Journal of Multidisciplinary Modeling and Optimization

Volume:3

Issue:2

Editor in Chief

Ahmet ŞAHİNER

E-ISSN: 2645-923X

http://dergipark.gov.tr/jmmo

TABLE OF CONTENTS

Journal of Multidisciplinary Modeling and Optimization 3(2) (2020), 47-56.

Structural and Spectral Analysis of Epicatechin Molecule by Density Functional Theory Method

Fatıma Demir[1], Mustafa Karakaya[2], Fatih UCUN[1,*]

[1]Department of Physics
Faculty of Science and Arts
Suleyman Demirel University, Isparta, Turkey
fd.fatma@hotmail.com; fatihucun@sdu.edu.tr

[2]Department of Energy Systems
Faculty of Engineering and Architecture
Sinop University, Sinop, Turkey
mkarakaya@sinop.edu.tr

Received: 17.10.2020, Accepted: 05.01.2021, Published: 25.03.2021

Abstract *—In this study, firstly the minimum energy structure of the title compound was determined by a result of the scanning of the potential energy surface at DFT(B3LYP)/6-31 G (d, p) from -180^o to -180^o at 20^o steps at a dihedral angle. Then, the ground state optimized structure and spectral results of the molecule were calculated by using DFT(B3LYP) method at 6-311++G(d,p) level of theory. Its optimized structure parameters (bond lengths, bond angles and torsion angles), vibrational frequencies and chemical shift values were listed and, compared with the corresponding experimental results.*

Keywords: Optimization, Epicatechin, Vibrational Frequency, Chemical Shift, DFT.
Mathematics Subject Classification: 90C26.

1 Introduction

To understand the properties of molecules, definitely, their experimental structural and spectral results should be taken in hand with together theoretical results The methods like ab initio Hatree-Fock (HF) and Density Functional Theory (DFT) are approach methods to understand the situation of systems with many particle, totally. They are withstand to calculate, numerically, ground state structure parameters (bond length and bond angle, i.e), energies, spectroscopic [Infrared (IR), Raman, Ultra-voile (UV), Nuclear Magnetic Resonance (NMR)] results and i.e. of molecules [1,2].

Tea is a popular beverage around the world as black tea and green tea. Green tea is also common in Asian countries, especially Japan, China, Korea and India [3]. Green tea contains a large amount of catechin (ECG; epigallocatechin gallate, EGC; epigallocatechin, EC; epicatechin). These tea catechins belong to the "flavonoid family". In recent years, due to their various pharmacological activities they have attracted serious attention as example, anti-mutagenic (mutation reducing agent) activities [4, 5], anti-carcinogen (substance that

Cite as: F. Demir, M. Karakaya and F.Ucun, Structural and spectral analysis of epicatechin molecule by density functional theory method, Journal of Multidisciplinary Modeling and Optimization 3(2) (2020), 47-56.

balances carcinogenic effects or inhibits cancer growth) effects [6], i.e. Widespread controversy in the mechanism of their biological and pharmacological effects is the antioxidant activities of this tea catechin (oxidation-inhibiting chemical or compound) [7,8]. These have shown that; EGCG, ECG, EGC and EC act as low-density in-hybrids to lipoprotein oxidation (biochemical compounds composed of both protein and lipids) [9]. Some catechins have been reported the destructive effects of free radicals; for example; DPPH radicals (1,1 diphenyl-2-pycrylhydrazyl)[10], superoxide anions [11], free lipid and hydroxyl radicals [12]. Some authors have studied the synergistic scavenging effects of these 4 catechins (EGCG, ECG, EGC and EC) in superoxide anions [13].

Here, in this study, after the determination of the optimized structure of the title molecule, its IR vibrational and NMR chemical shift analyzes were done. By comparing of the obtained theoretical results with the corresponding experimental results, the precise informations about the molecule were reached.

2 Theoretical Details

Gaussian 03 package and Gauss-View molecular visualization program were used in all the calculations [14]. Firstly the minimum energy structure of the compound was determined by a result of the scanning of the potential energy surface at DFT(B3LYP)/6-31 G (d, p) from -180^o to -180^o at 20^o steps at the dihedral angle C3-C2-C11-C16. After, the optimized structure parameters (bond lengths, bond angles and torsion angles) of the compound were calculated by the density functional theory (DFT) method at the B3LYP/6-311 ++ G (d, p) basis set level. The vibrational frequencies and chemical shifts were also calculated at the same level. The calculated vibrational frequencies were scaled with the scale factor of 0,9614 [16]. Approximate descriptions of vibration modes were made with VEDA 4 program [17]. The ^{1}H and ^{13}C NMR chemical shifts (in gas phase) were done by Gauge Including Atomic Orbitals **(**GIAO) method. In the chemical shift calculations tetramethylsilane (TMS) was used as a reference molecule, and the theoretical chemical shift ^{1}H and ^{13}C values were obtained by subtracting the GIAO isotropic magnetic shielding (IMS) values [18, 19].

3 Results and Discussion

The optimized molecular structure of (-)-epicatechin is given in Fig.1. In the figure is also given PLUTO drawing with the atomic numbering scheme [20]. The optimized energy value of the molecule is 1031.380486 a.u. The geometric parameters (bond length, bond angle and dihedral angles) according to the numbered atoms in Fig.1 are given in Table 1. To compare the experimental values are also given in the table [20]. The correlation values (R^2) between the theoretical and experimental values are written in the last line of the charts. From the R^2 values, we can say that the theoretical calculations are compatible with the experimental data, especially in the bond length.

The experimental and theoretical IR spectra of the compound are given in Fig .2. The experimental spectrum is taken from Ref [21]. Table 2 shows the experimental and corresponding theoretical vibrational frequencies. The approximate descriptions of the vibrational modes in the table are obtained with the Veda 4 program [17], which makes potential energy

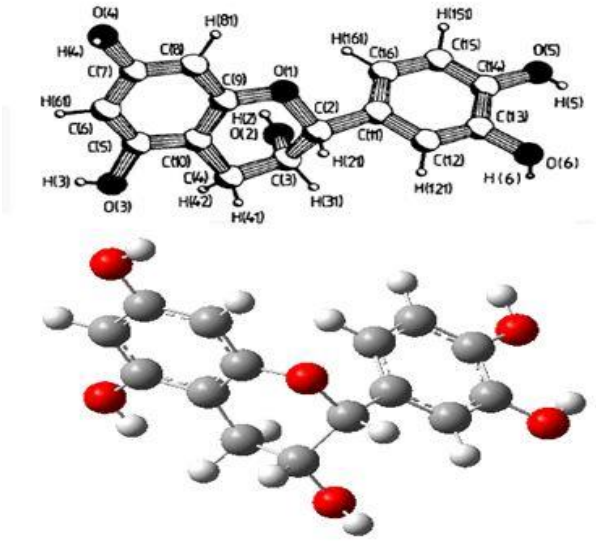

Figure 1: Molecular structure of (-)-epicatechin, a) PLUTO drawing with the atomic numbering scheme, and b) calculated at DFT(B3LYP)/6-311++ G(d,p) level.

Table 1: Calculated optimized geometrical parameters of (-)-epicatechin.

Bond length (A°)	Exp.[20]	Calculated	**Bond length (A°)**	Exp.[20]	Calculated
O(1)-C(2)	1.46	1.44	C(5)-C(10)	1.40	1.41
O(1)-C(9)	1.39	1.37	C(6)-C7)	1.36	1.39
O(2)-C(3)	1.44	1.42	C(7)-C8)	1.39	1.39
O(3)-C(5)	1.38	1.37	C(8)-C(9)	1.41	1.40
O(4)-C(7)	1.37	1.37	C(9)-C(10)	1.38	1.40
O(5)-C(14)	1.38	1.38	C(11)-C(16)	1.39	1.40
O(6)-C(13)	1.39	1.36	C(11)-C(12)	1.41	1.40
C(2)-C(11)	1.53	1.52	C(12)-C(13)	1.38	1.39
C(2)-C(3)	1.53	1.55	C(13)-C(14)	1.40	1.40
C(3)-C(4)	1.51	1.52	C(14)-C(15)	1.36	1.39
C(4)-C(10)	1.50	1.51	C(15)-C(16)	1.37	1.40
C(5)-C(6)	1.38	1.39			

R^2=0.91

Table 1: Continued

Bond angles (º)	Exp.[20]	Calculated	**Bond angle (º)**	Exp.[20]	Calculated
C(2)--O(1)--C(9	114.6	118.9	C(8)-C(9)-C(10)	122.5	121.9
C(3)-C(2)-C(11)	112.4	115.3	O(1)-C(9)-C(10)	122.7	122.7
O(1)-C(2)--C(3)	110.5	108.4	C(5)-C(10)-C(9)	116-5	117.1
O(1)-C(2)-C(11)	105.3	112.5	C(4)-C(10)-C(5)	121.I	122.4
O(2)-C(3)-C(2)	112.2	112.8	C(4)-C(10)-C(9)	122.4	120.4
O(2)-C(3)-C(4)	112.2	108.4	C(12)-C(11)-C(16)	120.2	118.9
C(2)-C(3)-C(4)	109.5	110.1	C(2)-C(I l)-C(16)	123.3	122.4
C(3)-C(4)-C(10)	109.7	109.6	C(2)-C(I l)-C(12)	116.4	118.8
C(6)-C(5)-C(10)	122.5	122.2	C(l l)-C(l 2)-C(13)	119.4	121.0
O(3)-C(5)-C(6)	122.1	116.4	O(6)-C(13)-C(12)	122.2	119.7
O(3)-C(5)-C(10)	115.3	121.4	C(12)-C(13)--C(14)	119.8	119.6
C(5)-C(6)-C(7)	118.8	118.8	O(6)-C(13)-C(14)	118.0)	120.7
C(6)-C(7)-C(8)	122.3	121.0	O(5)--C(14)-C(15)	119.0	124.7
O(4)-C(7)-C(8)	118.3	122.0	C(13)-C(14)-C(15)	119.3	120.0
O(4)-C(7)-C(6)	119.4	117.0	O(5)-C(14)-C(13)	121.7	115.3
C(7)-C(8)-C(9)	117.3	118.9	C(14)-C(15)-C(16)	123.0	120.2
O(1)-C(9)-C(8	114.7	115.3	C(I l)-C(16)-C(15	118.7	120.4

R^2=0.60

Table 2: Selected calculated and experimental vibrational frequencies (cm^{-1}) and, assignments. Experimental values are obtained from the IR spectrum of (-)- epicatechin [21].

Assinments	Exp.[21]	Calculated
νOH(100)	3603	3690
νOH(100)	3455	3689
νOH(100)	3409	3647
νCH(99)	3169	3091
νCH(98)	3151	3090
νCH(98)	2923	2929
νCH(98)	2868	2920
νCH(99)	2864	2889
νCH(90)	2727	2885
νCC(51)+δHCC(10)	1627	1604
νCC(56)	1608	1597
νCC(53)	1522	1573
δHCC(21)+δCCC(12)	1457	1468
νCC(15)+δHOC(15)+δHCH(11)	1444	1435
δHOC(18)+δHCO(17)+τH-COH(16)+τHCCC(11)	1378	1381
δHCO(18) +τHCCC(14)+νCC(12)	1349	1355
νCC(16)+νOC(12)	1313	1310
δHCC(14)+δHOC(13)	1296	1299
δHOC(22)	1289	1287
νOC(29)+δHCC(12)+νCC(11)	1261	1253
νOC(18)+δHCC(15)	1225	1232
δHOC(38)	1182	1187
δHCC(33)+δHOC(26)	1144	1148
δHCC(13)+ νCC(10)	1112	1115
νOC(30)+δHOC(42)+δHCC(12)	1096	1108
νOC(33)+δCCC(13)	1070	1068

Table 2: Continued

Assinments	Exp.[21]	Calculated
νCC(22)+νOC(19)	1045	1047
νOC(33)	1017	1028
νCC(30)+δHCC(15)+τHCCC(12)	979	973
τHCCC(24)	875	871
τHCCC(46)+τCCCC(14)	862	843
τHCCC(68)+ γOCCC(22)	833	806
δCCC(15)	808	792
τHCCC(52)	795	778
δOCC(15)+γOCCC(12)+δCOC(11)	627	636

R^2=0.996

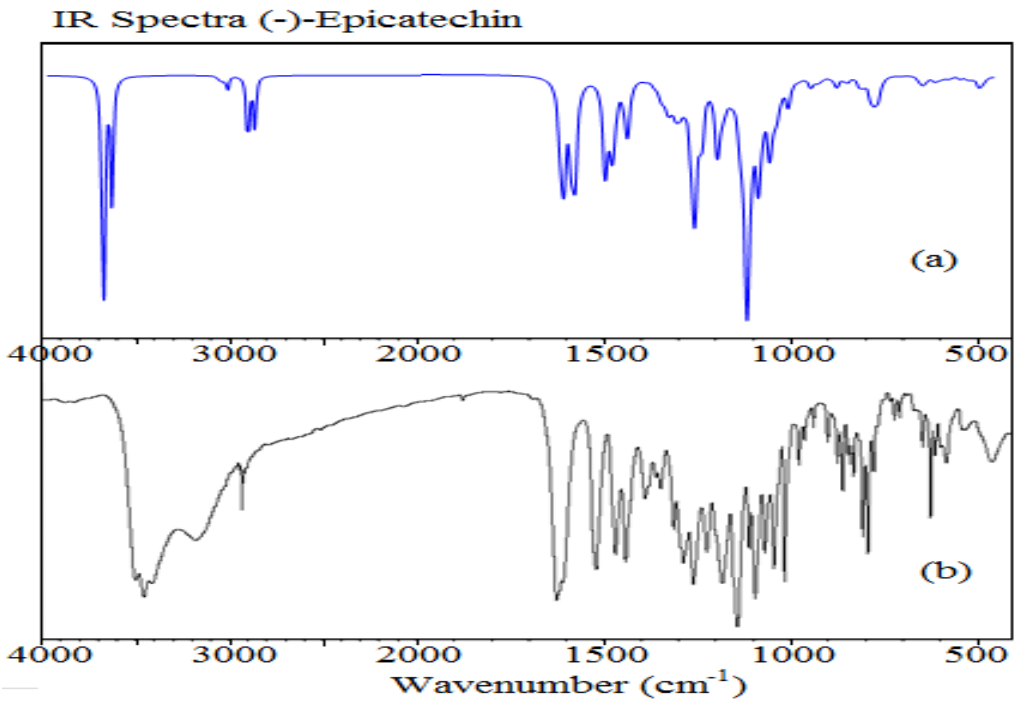

Figure 2: a) Theoretical and b) experimental IR spectrum of (-)-epicatechin

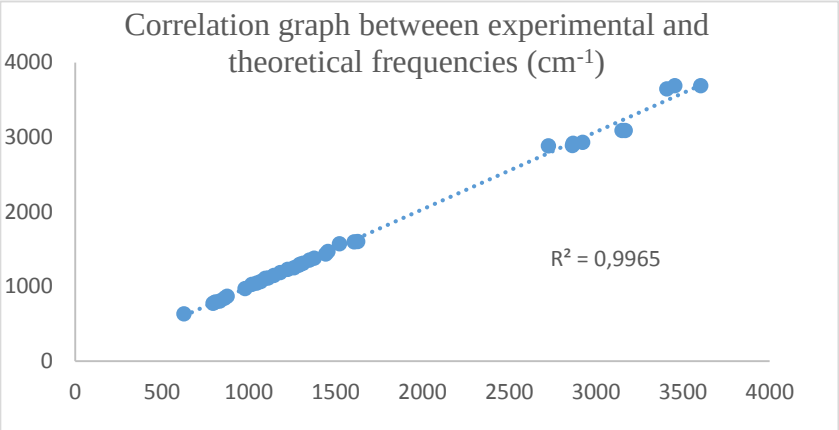

Figure 3: Correlation graph for frequencies.

distribution (PED) analysis. The correlation value between the theoretical and experimental frequencies is written at the bottom line of the table. The correlation graph can be seen in Fig.3. Here, from the R^2 value we can say that the experimental and theoretical frequencies are in a very good harmony.

As finally in Table 3 the calculated and experimental ^{1}H and ^{13}C NMR isotropic chemical shift values for (-)-epicatechin are given as corresponding to the atom numbering schema given in Fig.4. The experimental values are taken from Ref. [21]. Again, the R^2 values are written in the last line of the charts. They show a good agreement with the experimental data.

4 Conclusion

As conclusion, the ground state optimized structures of (-)- epicatechin molecule was determined, and its geometric parameters, vibration frequencies and chemical shift values were calculated and, compared with the experimental results. It has been found a good harmony between the experimental and theoretical values. The results determined here for the title molecule having various pharmacological activities will be useful in its future studies.

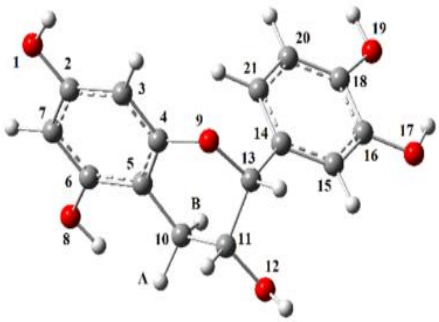

Figure 4: Atom numbering scheme for chemical shifts.

Table 3: Calculated and experimental ^{1}H experimental and ^{13}C NMR isotropic chemical shifts for (-)-epicatechin.

Atoms	Exp. (DMSO-d_6)	Calculated (Gas)
C2	156.41	165.81
C3	94.09	97.07
C4	155.68	162.74
C5	98.47	101.41
C6	156.16	163.88
C7	95.12	99.77
C10	28.07	25.46
C11	64.89	73.93
C13	78.01	85.25
C14	130.55	138.15
C15	114.85	121.80
C16	144.35	151.80
C18	144.42	148.88
C20	114.72	115.92
C21	117.91	122.74
R^2=0.995		
H3	5.65	5.84
H7	5.82	6.35
H10A	2.60	2.28
H10B	2.40	1.98
H11	3.93	4.37
H13	4.59	5.08
H15	6.82	6.99
H20	6.58	6.35
H21	6.59	6.45
R^2=0.964		

Acknowledgments

The authors declare that this work is completely private effort and it has not been supported by any governmental, financial, educational or otherwise establishment.

Conflict of Interest Declaration

The authors declare that there is no conflict of interest statement.

Ethics Committee Approval and Informed Consent

The authors declare that there is no ethics committee approval and/or informed consent statement.

References

[1] A. R. Leach, Molecular Modelling: Principles and Applications, Prentice Hall, Harlow, 88-92, 2001.

[2] J. B. Foresman and A. Frisch, Exploring Chemistry with Electronic Structure Methods, Gaussian, Inc., Pittsburgh, PA, 1996.

[3] H.N. Graham, Green tea composition, consumption, and polyphenol chemistry, Prev. Med. 21 1992, 334-350.

[4] H. Mukhtar, Z.Y. Wang, S.K. Katiyar and R. Agarwal, Tea components: antimutagenic and anti-carcinogenic e¡ects, Prev. Med. 21 1992, 351-360.

[5] Z.Y. Wang, S.J. Cheng, Z.C. Zhou, M. Athar, W.A. Khan, D.R. Bickers and H. Mukhtar, Anti-mutagenic activity of green tea polyphenols, Mutat. Res. 223 1989, 273-289.

[6] Z.G. Dong, W.Y. Ma, C.S. Huang and C.S. Yang, Inhibition of tumor promoter-induced activator protein 1 activation and cell transformation by tea polyphenols, (3)-epigallokateşin gallate, and thea£avins, Cancer Res. 57 1997, 4414-4419.

[7] C.S. Yang and Z.Y. Wang, Tea and cancer, J. Natl. Cancer Inst. 58 1993, 1038-1049.

[8] S. Yoshizawa, T. Horiuchi, H. Fujiki, T. Yoshida, T. Okuda and T. Sugimura, Antitumor promoting activity of (3)-epigallokateşin gallate, the main constituent of `tannin' in green tea, Photother. Res. 1 1987, 44-47.

[9] N. Salah, N.J. Miller, G. Paganga, L. Tijiburg, G.P. Bolwell and C. Rice-Evans, Polyphenolic £avonols as scavengers of aqueous phase radicals and as chain-breaking antioxidants, Arch. Biochem. Biophys. 322 1995, 339-346.

[10] T. Hatano, R. Edamatsu, M. Hiramatsu, A. Mori, Y. Fujita, T. Yasuhara and T. Okuda, Ejects of the interaction of tannins with co-existing substances. VI. Ejects of tannins and related polyphenols on superoxide anion radical, and on 1,1- diphenyl-2-picrylhydrazyl radical, Chem. Pharm. Bull. 37 1989, 2016-2021.

[11] G. Sichel, C. Corsaro, M. Scalia, A.J. DiBilio and R.P. Bonomo, In vitro scavenger activity of some £avonoids and melanins against O3W 2, Free Radic. Biol. Med. 11 1991, 1-8.

[12] Q. Guo, B.L. Zhao, M.F. Li, S.R. Shen and W.J. Xin, Studies on protective mechanisms of four components of green tea polyphenols against lipid peroxidation in synaptosomes, Biochem. Biophys. Acta 1304 1996, 210-222.

[13] S.R. Shen, X.Q. Yang, F.J. Ya, B.L. Zhao, W.J. Xin and H. Yukihiko, Coordinatingly synergic e¡ect of kateşins during their antioxidant, J. Tea. Sci. 13 1993, 141-146.

[14] M. J. Frisch, et al., GAUSSIAN 03, Revision C.02, Gaussian Inc., Pittsburgh, PA, 2003.

[15] A. Frish, A. B. Nielsen, A. J. Holder, Gauss View User Manual, Gaussian Inc. Pittsburg, PA, 2001.

[16] D. C. Young, Computational Chemistry: A Practical Guide for Applying Techniques to Real-World Problems, (Electronics), John Wiley & Sons, Inc., New York 2001, 92.

[17] M. H. Jamróz, Vibrational Energy Distribution Analysis VEDA 4, Warsaw, 2004.

[18] R. Ditchfield, Self-consistent perturbation theory of diamagnetism. 1. Gauge-invariant LCAO method for N.M.R. chemical shifts, Mol. Phys., 27 1974, 789-807.

[19] C. M. Rohlfing, L. C. Allen and R. Ditchfield, Proton and 13C chemical shifts: comparison between theory and experiment, Chem. Phys. 87 1984, 9-15.

[20] L. Spek and B. Kojıc-Prodı, Structure of (-)-Epicatechin: (2R,3R)-2-(3,4-Dihyroxyphenyl)-3,4-dihydro-2H- lbenzopyran-3,5,7-triol, C15H1406, Acta Cryst. C40 1984, 2068-2071.

[21] Spectral Database for Organic Compounds, SDBS. National Institute of Advanced Industrial Science and Technology (AIST), Japan. http://sdbs.db.aist.go.jp/sdbs/cgi-bin/direct_frame_top.cgi

Fatıma Demir, ORCID: https://orcid.org//0000-0003-0439-4820
Mustafa Karakaya, ORCID: https://orcid.org/0000-0001-6663-9008
Fatih Ucun, ORCID: https://orcid.org/0000-0001-7464-7788

Journal of Multidisciplinary Modeling and Optimization 3(2) (2020),57-69.

A New Conjugate Gradient Method for Learning Fuzzy Neural Networks

Hisham M. KHUDHUR[1] and Khalil K. ABBO[2]

[1]Mathematics Department, College of Computer Science and Mathematics,
University of Mosul, Mosul, Iraq
hisham892020@uomosul.edu.iq

[2]Department of Mathematics, College of Basic Education,
University of Telafer, Tall'Afar, Iraq
dr.khalil-khuder@uotelafer.edu.iq

Received: 02.01.2021, Accepted: 20.01.2021, Published: 25.03.2021

Abstract — *In this paper, we suggest a conjugate gradient method, which belongs to the optimization methods for learning a fuzzy neural network model that is based on Takagi Sugeno. A new algorithm based on the Polak–Ribière–Polak (PRP) method is introduced to overcome the slow convergence of Polak–Ribière–Polak (PRP) and Liu-Storey (LS) methods. The numerical results indicate the efficiency of the developed method for classifying data as shown in the Table (2) where the new method outperforms above mentioned methods in terms of average training time, average training accuracy, average test accuracy, average training MSE, and average test MSE.*

Keywords: Algorithm, Classification, Fuzzy neural networks, Techniques, Optimization.
Mathematics Subject Classification: 65K10, 90C26, 68T07.

1 Introduction

Fuzzy modeling is to create a large number of local input and output relationships. The purpose of this relation is to define a rule and to make clear a nonlinear manner instead of the classical modeling schemes which may use different equations. [1]. Therefore, by using the given input-output (I-O), a process identification data would become practically a different equivalent problem that concentrates on the description of a fuzzy model[2]. In general, the description of a fuzzy logic method or fuzzy neural (neuro-fuzzy) network method model covers chiefly two phases: construction description and parameter description [3].

Structure identification, In general, the determination of the construction of any fuzzy problem requires, in each law, the number of fuzzy regulations and the membership functions of the premise and consequent fuzzy sets. A variety of techniques is proposed for structure recognition. For the sake of extracting rules from the available input-output dataset to construct the initial rule base, one of these approaches is to use clustering

Cite as: *H. M. Khudhur and K. K. Abbo, A new Conjugate Gradient Method for learning Fuzzy Neural Networks, Journal of Multidisciplinary Modeling and Optimization 3(2) (2020), 57-69.*

algorithms. Typically, multiple clustering devices such as the K-means algorithm can be used to obtain the initial fuzzy rule base of a fuzzy logic system (FLS). [4] , (FCM) fuzzy c-means[5] [6] plus mountain clustering technique[7]. Furthermore, there are other methods for clustering, such as the so-called FPCM , PCM and PFCM in[4] [8] [9]. The fundamental view of the clustering method-based structure identification is to collect the specified samples and position them in various clusters linked by one cluster to a law. The number of laws is also equivalent to the number of clusters. The data must be obtained in advance through of clustering method-based structure recognition. Consequently, online structure recognition is not sufficient. In several experiments, however, scientists stress the use of fumigated neural networks to dynamically model the system.[10] [11]. It is also proposed that the Bayesian TSK fuzzy model in[12] [13], which can classify the number of fuzzy laws without returning to the knowledge of the previous expert. In this paper the researchers concentrate on the clustering of method-based structure recognition as a tool for resolving problems with static regression and classification. The "Gradient based Neuro-Fuzzy learning algorithm" is widely used to characterize the neuro-fuzzy system's feedback, similar to the Neural Network training feedback. [2] [3] [14] [15]. Inspired by the GNF for neuro-fuzzy structures, a GNF update, MGNF, is proposed in[16]. The error function type is revised by considering independent variables in the reciprocal widths of Gaussian membership to prevent singularity. Thus, The weight sequence update formulas are easily modified. This adjustment will help to evaluate the MGNF algorithm converging. In[16] The T-norm product, but the firing strength can be very low for the product, even for a moderate amount of inputs. While any atomic precedent clause can very well be fulfilled. One approach to this issue with other T criteria such as minimum standards [17] [18] [19]. Unfortunately, this is not differential; we want to use gradient-based procedures for T-norm differentiability. As a result, this paper uses a softer variant of the minimum, softmin, to calculate the value of the firing capacity. Softmin's purpose is distinguishable and can manage the Specimen with a wide number of features. [20] [21] [19]. The latter performs much better in terms of both efficiency and acceleration of convergence in general, compared to the common gradient descent technique with conjugate gradient (CG) techniques [22]. The first linear conjugate gradient (CG) technique is implemented in[23], The linear problems can be solved with positive definite coefficient matrices, which can be treated as an optimization algorithm. In addition to the above, the conjugate gradient (CG) method shown in [24] An effective way to solve large-scale nonlinear optimization problems has been found to be an effective tool. Hestenes-Stiefel and for (HS)[23] and Fletcher-Reeves (FR)[24], Another traditional Conjugate gradient (CG) technique Polak–Ribière–Polak (PRP) [25] The alternative direction of the descent is then suggested. Successfully, Conjugate gradient approaches can be extended to the training of neuro-fuzzy networks. [26] [27]. Eight methods of the conjugate gradient (CG) are described in[26] As they are used to equip the fuzzy logic system type-1 to solve the classification issue. The results of learned simulation in [26] Explain that the techniques of conjugate gradient (CG) converge more rapidly than the process of gradient descent (GD). Also, Compared to the ones generated by the optimized fuzzy logic system (FLS) using the gradient descent (GD) process, the classification results derived from conjugate gradient (CG) based fuzzy logic system (FLS) are the best. In [27], Recently, Ahmad et al. [21] developed a new numerical method for solutions of coupled burgers' equations. Also, Ahmad et al. [28] To obtain the numerical solutions of certain nonlinear PDEs such

as KdV, mKdV and combined KdV-mKdV equations, a new modification of the variational iteration algorithm-II was proposed.

The goal of this paper is to develop a new Polak–Ribière–Polak (PRP) based algorithm for learning a fuzzy-neural network model to obtain the lowest average training error.

This paper is organized as follows: In Section 2 inference method for Zero-order Takagi-Sugeno is introduced. In Section 3 we present new conjugate gradient (CG) techniques and show that our algorithm satisfies descent and global convergence conditions. Section 4 presents numerical experiments and comparisons.

2 Inference Method for Zero-Order Takagi-Sugeno (TS)

A fuzzy inference scheme that is used as an adaptive network is the neuro-fuzzy model. The neuro-fuzzy model adopted in this article is the zero-order Takagi-Sugeno inference method. Its topological structure can be seen in Fig.1. It is a four-layer network with m-input nodes $x = (x_1, x_2, \dots, x_m) \in \mathbb{R}^m$ and one output node y.

Let us first describe the inference method of the zero-order Takagi-Sugeno.

The basis of the fuzzy rule is defined as follows [29] [30] [31] [14] [32] [33].

Rule i: IF x_1 is A_{1i} and x_2 is A_{2i} and … and x_m is A_{mi} THEN y is y_i, (1)

where $i\ (i = 1,2,\dots,n)$ Matches with the ith fuzzy rule, n is the number of the fuzzy rules, yi is a real number, Ali is a fuzzy subset of xl, and $Ali(xl)$ It means the role of Gaussian membership of the fuzzy judgment ‘‘$xl\ is\ Ali$” defined by

$$A_{li} = \frac{\exp(-(x_l - a_{li})^2}{\sigma_{li}^2} \quad (2)$$

where ali is the center of $Ali(xl)$, and rli is the width of $Ali(xl)$.

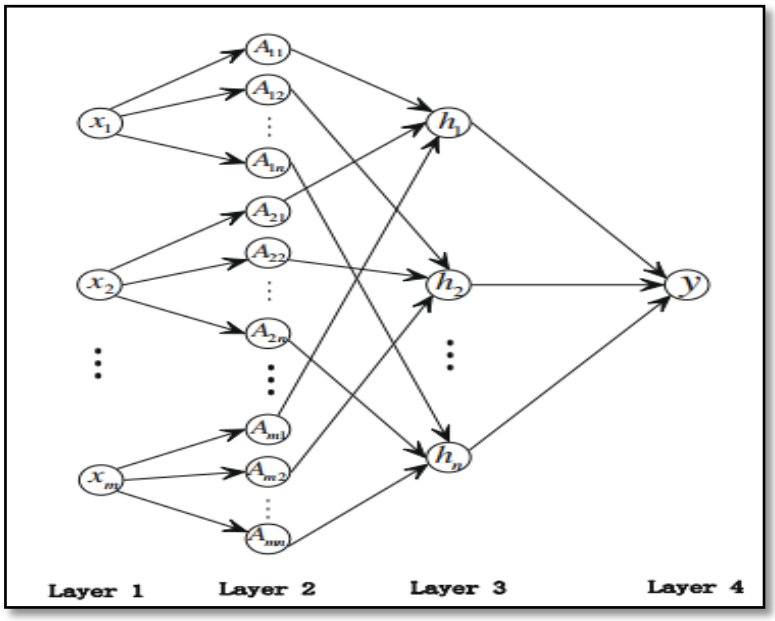

Figure 1: Topological structure of the zero-order takagi–sugeno inference system

For a stated observation $x = (x_1, x_2, \dots, x_m)$ the functions of the nodes in this model are as follows, according to the zero-order Takagi-Sugeno inference method:

Layer1: (input layer): In this layer, each neuron represents one input variable and the input variables are directly passed to the next layer.

Layer2: (membership layer): Each node in this layer represents the membership function of a linguistic variable and serves as a memory unit. Here, the Gaussian functions(2) are adopted as membership functions for the nodes. The weights connecting Layer1 and Layer2 can be interpreted as the Gaussian membership function's centers and widths, respectively.

Layer3: (rule layer): Nodes are referred to as rule nodes in this layer, and each of them denotes a term with a rule. For $i = 1,2,\dots,n$, Agreement on the ith Previous section is estimated by

$$h_i = h_i(x) = A_{1i}(x_1)A_{2i}(x_2) \dots A_{mi}(x_m) = \prod_{l=1}^{m} A_{ii}(x_l) \tag{3}$$

The connecting weights of layers 2 and 3 are set as constant 1.

Layer4: (output layer): This layer performs the summed-weight defuzzification process. The final product of this layer is y, which is a linear combination of the implications of Layer3:

$$y = \sum_{i=1}^{n} h_i y_i \tag{4}$$

The yi relation weights of the output layer are often referred to as conclusion parameters.

Remark 1. In original neuro-fuzzy models [29] [34] [32] [35], the final consequence *y* is calculated by using the gravity method as follows:

$$y = \frac{\sum_{i=1}^{n} h_i y_i}{\sum_{i=1}^{n} h_i} \tag{5}$$

A popular method is to achieve the fuzzy effect without measuring the center of gravity for ease of learning. Hence, the denominator in (5) is omitted [30] [31] [14] [33]. A further advantage of this operation is the rapid deployment of hardware. [36]. We therefore take the form of (4) in our discussions.

We then take the form of (4) in our debates.

The error function is defined as

$$E(\mathbf{W}) = \frac{1}{2}\sum_{j=1}^{J}(y^j - O^j)^2$$

where O^j is the desired output for the jth training pattern x^j, y^j is the corresponding fuzzy reasoning result, J is the number of training patterns.

The purpose of network learning is to find W^* such that $E(\mathbf{W}^*) = minE(\mathbf{W})$

To solve this optimization problem, the gradient descent approach is sometimes used [37] [38] [39].

3 New Conjugate Gradient (CG) Techniques

Development of new optimization algorithm Based on algorithm Polak–Ribière–Polak (PRP) for learning fuzzy neural networks in the field of data classification and comparison with other optimization algorithms

$$w_{k+1} = w_k + \alpha_k d_k, \quad k \geq 1, \tag{6}$$

where α_k is step-size obtained by a line search and d_k is the direction of search specified by

$$d_{k+1} = \begin{cases} -g_1, & k = 1 \\ -g_{k+1} + \beta_k d_k, & k \geq 1' \end{cases} \tag{7}$$

where β_k is a parameter. $\beta^{LS} = \frac{-g_{k+1}^T y_k}{g_k^T d_k}$, see [40] and $\beta^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}$, see [25] where $g_k = \nabla E(w_k)$, denotes the gradient of the function of error $E(w)$ in regard to w, k the number of iterations denotes the, and let $y_k = g_{k+1} - g_k$.

Now we suggest a new conjugate gradient algorithm for classifying data depend basically on Polak–Ribière–Polak (PRP) algorithm so we get a new formula:

$$-\theta g_{k+1} + \beta_k d_k = -\gamma g_{k+1} + \beta_k^{PRP} d_k$$

$$-\theta g_{k+1}^T g_{k+1} + \beta_k g_{k+1}^T d_k = -\gamma g_{k+1}^T g_{k+1} + \beta_k^{PRP} g_{k+1}^T d_k$$

$$\beta_k^{NEW} = \begin{cases} \frac{(\theta - \gamma) g_{k+1} g_{k+1}}{g_{k+1}^T d_k} + \beta_k^{PRP}, & if \quad g_{k+1}^T d_k \neq 0 \\ \beta_k^{PRP}, & if \quad g_{k+1}^T d_k = 0 \end{cases}$$

where $\theta < \gamma$ and $\theta, \gamma \in [0,1]$.

$$d_{k+1} = -g_{k+1} + (\frac{(\theta - \gamma) g_{k+1}^T g_{k+1}}{g_{k+1}^T d_k} + \frac{y_k^T g_{k+1}}{g_k^T g_k}) d_k \tag{8}$$

3.1 The Descent Property of a Conjugate Gradient (CG) Technique

Below we have to demonstrate the descending property for our proposed new conjugate gradient scheme, denoted by β_k^{NEW}. In the following part

Theorem 1. *The search direction d_{k+1} and β_k^{NEW} given in equation*

$d_{k+1} = -g_{k+1} + \beta_k^{NEW} d_k$

$$\beta_k^{NEW} = \begin{cases} \frac{(\theta - \gamma) g_{k+1} g_{k+1}}{g_{k+1}^T d_k} + \beta_k^{PRP}, & if \quad g_{k+1}^T d_k \neq 0 \\ \beta_k^{PRP}, & if \quad g_{k+1}^T d_k = 0 \end{cases}$$

$d_{k+1} = -g_{k+1} + (\frac{(\theta - \gamma) g_{k+1}^T g_{k+1}}{g_{k+1}^T d_k} + \frac{y_k^T g_{k+1}}{g_k^T g_k}) d_k$,

where $\theta < \gamma$ and $\theta, \gamma \in [0,1]$. It will hold for all $k \geq 1$.

Proof. The proof is by using inducement mathematical

1- If $k = 1$ then $g_1^T d_1 < 0$, $d_1 = -g_1 \rightarrow < 0$.

2- Let the relation $g_k^T d_k < 0$ for all k.

3- We are going to prove that the relationship is true when $k = k + 1$ by multiplying the equation (8) in g_{k+1} we obtain

$$g_{k+1}^T d_{k+1} = -g_{k+1}^T g_{k+1} + (\frac{(\theta - \gamma) g_{k+1}^T g_{k+1}}{g_{k+1}^T d_k} + \frac{y_k^T g_{k+1}}{g_k^T g_k}) g_{k+1}^T d_k$$

Let $\tau = \frac{(\theta - \gamma) g_{k+1}^T g_{k+1}}{g_{k+1}^T d_k}, v = \frac{y_k^T g_{k+1}}{g_k^T g_k}$

$g_{k+1}^T d_{k+1} = -g_{k+1}^T g_{k+1} + (\tau + v) g_{k+1}^T d_k$

Let $g_{k+1}^T d_{k+1} > 0$ and $\tau > v$ the $g_{k+1}^T d_{k+1} \geq 0$.

3.2. Global Convergence

We will display that conjugate gradient (CG) method with β_k^{NEW} convergences globally. For the convergence of the proposed new algorithm, we need a certain assumption.

Assumption 1. [41][42]

1- Assume E in the level set is bound below $S = \{w \in R^n : E(w) \leq E(w_0)\}$; In some Initial point.

2- E is continuously differentiable and its gradient is Lipchitz continuous, there exist $L > 0$ such that[43]:

$$\| g(x) - g(y) \| \leq Lx - y \| \forall x, y \in N \tag{9}$$

On the other hand, under Assumption(1), it is clear that there exist positive constants B such

$$\|w\| \leq B, \forall w \in S \tag{10}$$

$$\| \nabla E(w) \| \leq \overline{\gamma}, \forall x \in S \tag{11}$$

Lemma 1. *Assume that Assumption (1) and equation (10) hold. take into consideration any conjugate gradient method in from (6) and (7), where d_k is a decent direction and α_k is obtained by the S.W.L.S. If*

$$\sum_{k>1} \frac{1}{\| d_{k+1} \|^2} = \infty$$

then we have

$$\liminf_{k\to\infty} \| g_k \| = 0$$

more details can be found in [44][45][46].

Theorem 2. *Assume that Assumption (1) and equation (6) and the descent condition hold. Consider a conjugate gradient scheme in the form*

$$d_{k+1} = -g_{k+1} + \beta_k^{NEW} d_k,$$

where α_k is computed from strong Wolfe line search condition for more details see [47] [48] [49] [50] , If the objective function is uniformly on set S, then

$$lim_{n\to\infty}(inf \| g_k \|) = 0 .$$

Proof.

$$d_{k+1} = -g_{k+1} + \beta_k^{NEW1} d_k$$

$$\beta_k^{NEW} = \begin{cases} \frac{(\theta-\gamma) g_{k+1} g_{k+1}}{g_{k+1}^T d_k} + \beta_k^{PRP}, & if \quad g_{k+1}^T d_k \neq 0 \\ \beta_k^{PRP}, & if \quad g_{k+1}^T d_k = 0 \end{cases}$$

$$\| d_{k+1} \| = \| -g_{k+1} + (\frac{(\theta-\gamma) g_{k+1}^T g_{k+1}}{g_{k+1}^T d_k} + \frac{y_k^T g_{k+1}}{g_k^T g_k}) d_k \|$$

$$\| d_{k+1} \| \leq \| g_{k+1} \| + \| \frac{(\theta-\gamma) g_{k+1}^T g_{k+1}}{g_{k+1}^T d_k} + \frac{y_k^T g_{k+1}}{g_k^T g_k}) \| d_k \|$$

$$\| d_{k+1} \| \leq \| g_{k+1} \| + \| \frac{(\theta-\gamma) \| g_{k+1} \|^2}{\| d_k \| \| g_{k+1} \|} + \frac{\| y_k^T \| \| g_{k+1} \|}{\| g_k \|^2} \| \| d_k \|$$

$$\| d_{k+1} \| \leq (1+\| \frac{(\theta-\gamma) \| d_k \|}{\| d_k \|} + \frac{\| d_k \| \| y_k^T \|}{\| g_k \|^2}) \| g_{k+1} \|$$

$$\psi = \| \frac{(\theta-\gamma) \| d_k \|}{\| d_k \|} + \frac{\| d_k \| \| y_k^T \|}{\| g_k \|^2} \|$$

$$\| d_{k+1} \| \leq (1+\psi) \| g_{k+1} \|$$

$$\sum_{k\geq 1} \frac{1}{\| d_{k+1} \|^2} \geq (\frac{1}{(1+\psi)^2}) \frac{1}{\gamma^2} \Sigma 1 = \infty.$$

3 Numerical Examples

The conjugate gradient algorithm developed to teach the fuzzy neural networks described in Part Two is evaluated by comparing it with related algorithms such as LS and PRP to classify the data given by the following classification problems (Iris, Thyroid, Glass, Wine, Breast Cancer and Sonar) [51], The developed algorithm NEW showed high efficiency in data classification compared to LS and PRP algorithms as shown in the following table and graphs, The simulation was carried out using Matlab 2018b, running on a Windows 8 HP machine with an Intel Core i5 processor, 4 GB of RAM and 500 GB of hard disk drive.

Table 1: Problems in Real-World Classification [51]

	Classification dataset	*Data size*	*No. of training samples*	*No. of testing samples*
1	*Iris*	150	90	60
2	*Thyroid*	215	129	86
3	*Glass*	214	107	107
4	*Wine*	178	89	89
5	*Breast Cancer*	253	127	126
6	*Sonar*	208	104	104

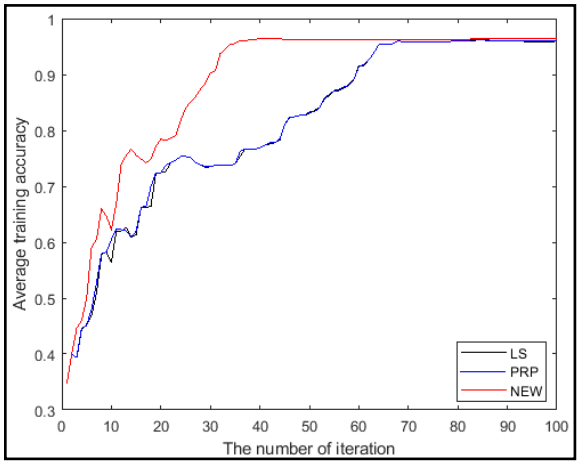

Figure 2: The average training accuracy for Iris

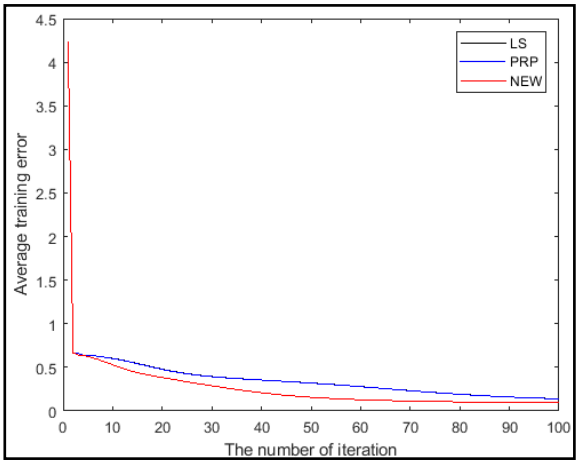

Figure 3: The average training error results for Iris

Table 2: Average Performance Comparison for Classification Problems for NEW

Datasets	*Algorithms*	*No. of training iteration*	*Average training time*	*Average training accuracy*	*Average test accuracy*	*Average training MSE*	*Average test MSE*
Iris	*LS*	*100*	***0.2883***	*0.9600*	*0.9367*	*0.1392*	*0.1541*
	PRP	*100*	*0.4328*	*0.9600*	*0.9367*	*0.1391*	*0.1541*
	NEW	*100*	*0.6028*	***0.9644***	***0.9500***	***0.0932***	***0.1110***
Thyroid	*LS*	*100*	*0.1702*	*0.6930*	*0.7070*	*0.3899*	*0.3874*
	PRP	*100*	*0.1703*	*0.8961*	*0.9000*	*0.1627*	*0.1632*
	NEW	*100*	***0.1686***	***0.9271***	***0.9163***	***0.1356***	***0.1370***
Glass	*LS*	*100*	*0.6123*	*0.3121*	*0.2617*	*0.7841*	*0.7944*
	PRP	*100*	***0.6059***	*0.3121*	*0.2673*	*0.7807*	*0.7990*
	NEW	*100*	*0.6565*	***0.5664***	***0.4636***	***0.5828***	***0.6414***
Wine	*LS*	*100*	*0.4195*	*0.4854*	*0.4427*	*0.6324*	*0.6568*
	PRP	*100*	***0.4147***	*0.9528*	*0.9258*	*0.1340*	*0.1827*
	NEW	*100*	*0.4230*	***0.9730***	***0.9348***	***0.1131***	***0.1597***
Breast Cancer	*LS*	*100*	*1.7727*	*0.4677*	*0.4619*	*0.9235*	*0.9347*
	PRP	*100*	***1.7482***	*0.4646*	*0.4619*	*0.9142*	*0.9257*
	NEW	*100*	*1.8412*	***0.6630***	***0.6349***	***0.6235***	***0.6727***
Sonar	*LS*	*100*	*2.1704*	*0.5442*	*0.5288*	*0.6071*	*0.6078*
	PRP	*100*	***2.1506***	*0.5115*	*0.5000*	*0.6063*	*0.6071*
	NEW	*100*	*2.1819*	***0.6558***	***0.5942***	***0.4296***	***0.4873***

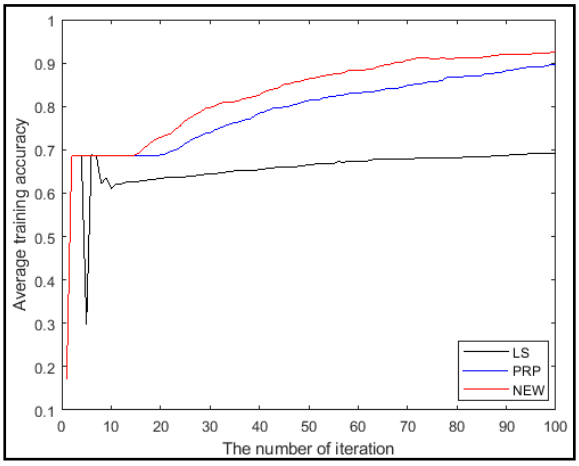

Figure 4: The average training accuracy for Thyroid

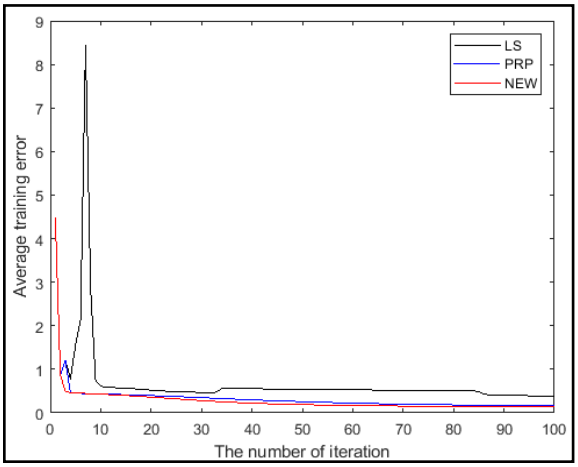

Figure 5: The average training error results for Thyroid

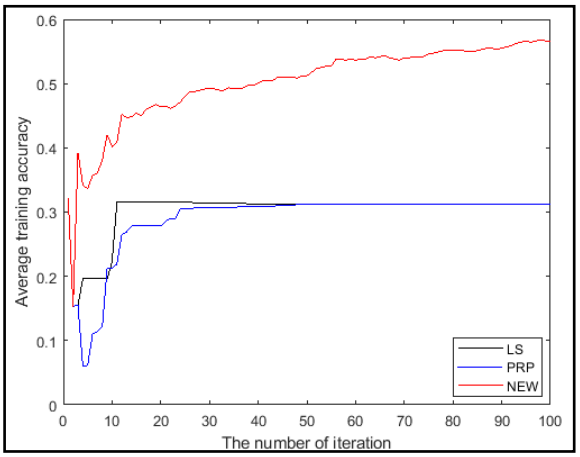

Figure 6: The average training accuracy for Glass

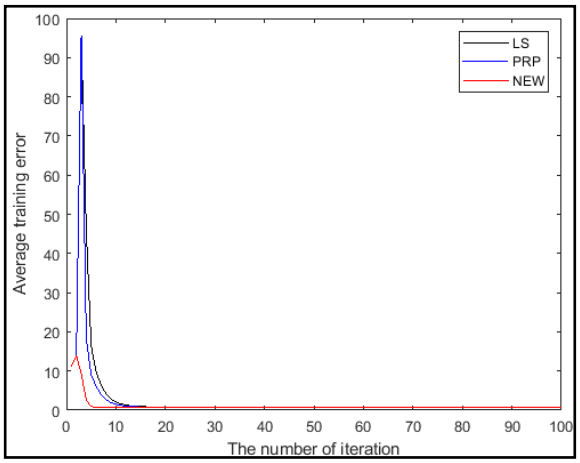

Figure 7: The average training error results for Glass

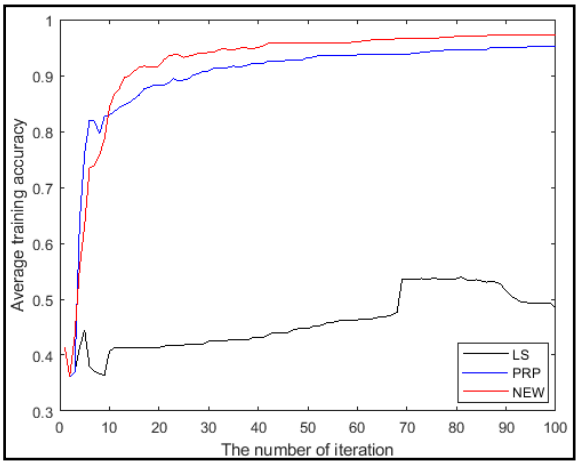

Figure 8: The average training accuracy for Wine

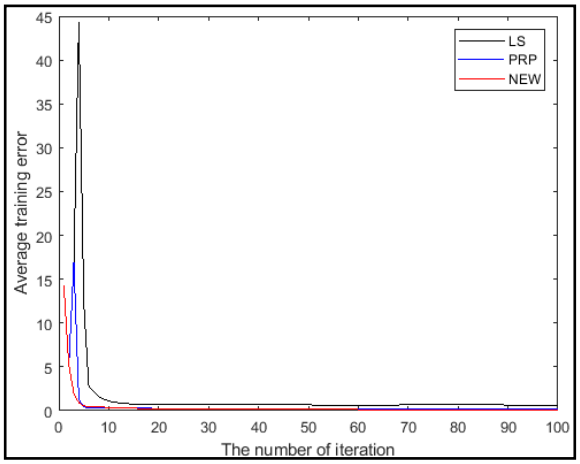

Figure 9: The average training error results for Wine

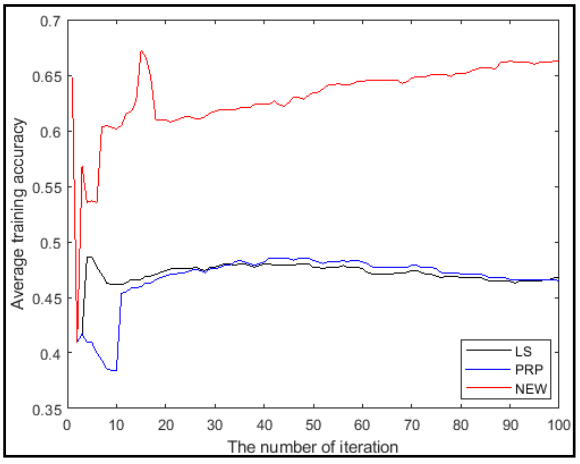

Figure 10: The average training accuracy for Breast Cancer

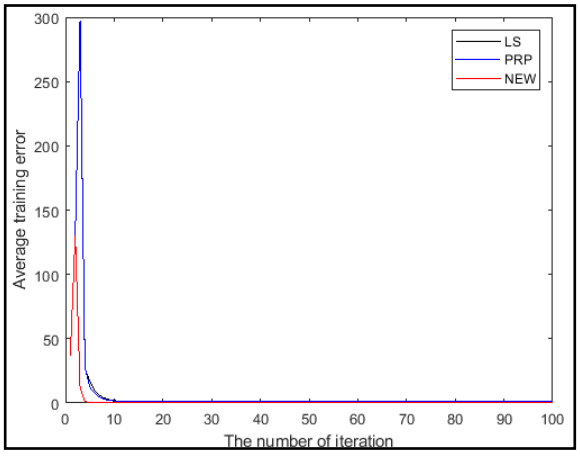

Figure 11: The average training error results for Breast Cancer

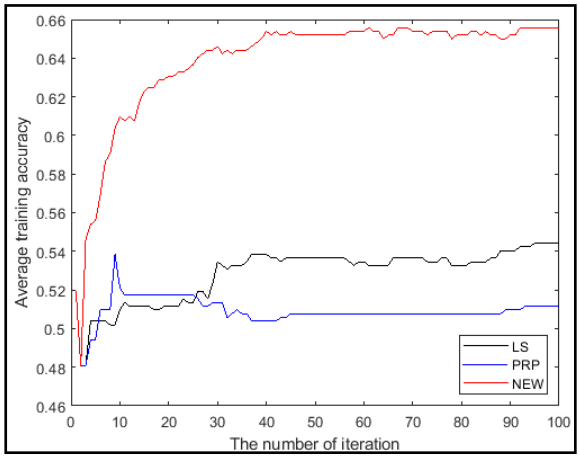

Figure 12: The average training accuracy for Sonar

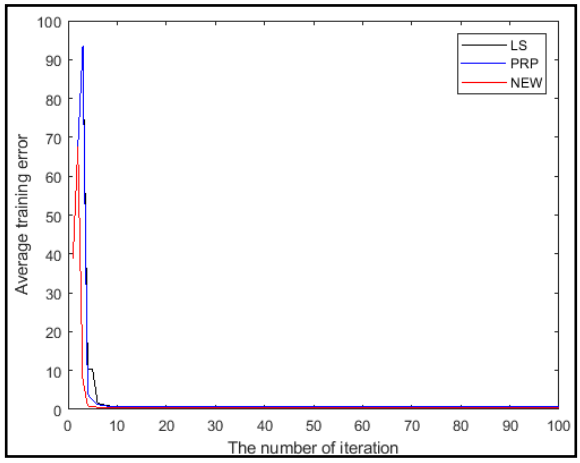

Figure 13: The average training error results for Sonar

4 Conclusion

Our Conjugate gradient technique is a good option to a gradient descent method for its faster convergence speed via looking for a conjugate descent path with adaptive learning coefficients. An updated conjugate gradient approach has been proposed in this paper to train the fuzzy neural network system of the 0-th order Takagi-Sugeno (TS). Numerical simulations shown that new algorithm has a better generalization efficiency than its current counterparts. Also, the simulations observed endorse the converging behavior of the suggested algorithm is very well. We also conclude that the proposed technique can solve the optimization functions and can be used in training artificial neural networks.

Acknowledgments

The authors are grateful to Dr. Tao Gao, Dr. Jian Wang at the College of information, Control Engineering and College of Science at the China University of Petroleum, China, for their precious support.

Conflict of Interest Declaration

The authors declare that there is no conflict of interest statement.

Ethics Committee Approval and Informed Consent

The authors declare that there is no ethics committee approval and/or informed consent statement.

References

[1] S. Chakraborty, A. Konar, A. Ralescu, and N. R. Pal, A fast algorithm to compute precise type-2 centroids for real-time control applications, *IEEE Trans. Cybern.*, 45(2) 2014, 340-353.

[2] M. Sugeno and G. T. Kang, Structure identification of fuzzy model, *Fuzzy Sets Syst.*, 28 (1), 1988, 15–33.

[3] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Syst. Man. Cybern.*, 1 1985, 116–132.

[4] J. C. Bezdek, J. Keller, R. Krisnapuram, and N. Pal, Fuzzy models and algorithms for pattern recognition and image processing, Springer Science & Business Media, vol. 4. 1999.

[5] J. C. Bezdek, Objective function clustering, in Pattern recognition with fuzzy objective function algorithms, Springer, 1981, 43–93.

[6] X. Gu and S. Wang, Bayesian Takagi–Sugeno–Kang fuzzy model and its joint learning of structure identification and parameter estimation, *IEEE Trans. Ind. Informatics*, 14(12) 2018, 5327–5337.

[7] R. R. Yager and D. P. Filev, Generation of fuzzy rules by mountain clustering, *J. Intell. Fuzzy Syst.*, 2(3) 1994, 209–219.

[8] R. Krishnapuram and J. M. Keller, A possibilistic approach to clustering, *IEEE Trans. fuzzy Syst.*, 1(2) 1993, 98–110.

[9] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, A possibilistic fuzzy c-means clustering algorithm, *IEEE Trans. Fuzzy Syst.*, 13(4) 2005, 517–530.

[10] C.-F. Juang and C.-T. Lin, An online self-constructing neural fuzzy inference network and its applications, *IEEE Trans. Fuzzy Syst.*, 6(1) 1998, 12–32.

[11] H. Shahparast, E. G. Mansoori, and M. Z. Jahromi, AFCGD: an adaptive fuzzy classifier based on gradient descent, *Soft Comput.*, 23(12) 2019, 4557–4571.

[12] X. Gu, F.-L. Chung, H. Ishibuchi, and S. Wang, Imbalanced TSK fuzzy classifier by cross-class Bayesian fuzzy clustering and imbalance learning, *IEEE Trans. Syst. Man, Cybern. Syst.*, 47(8) 2016, 2005–2020.

[13] X. Gu, F.-L. Chung, and S. Wang, Bayesian Takagi–Sugeno–Kang fuzzy classifier, *IEEE Trans. Fuzzy Syst.*, 25(6) 2016, 1655–1671.

[14] H. Ichihashi and I. B. Türksen, A neuro-fuzzy approach to data analysis of pairwise comparisons, *Int. J. Approx. Reason.*, 9(3) 1993, 227–248.

[15] J. M. Mendel, General type-2 fuzzy logic systems made simple: a tutorial, *IEEE Trans. Fuzzy Syst.*, 22(5) 2013, 1162–1182.

[16] W. Wu, L. Li, J. Yang, and Y. Liu, A modified gradient-based neuro-fuzzy learning algorithm and its convergence, *Inf. Sci. (Ny).*, 2010, doi: 10.1016/j.ins.2009.12.030.

[17] H. Ahmad, T. A. Khan, P. S. Stanimirović, Y. M. Chu, and I. Ahmad, Modified variational iteration algorithm-II: Convergence and applications to diffusion models, *Complexity*, 2020, doi: 10.1155/2020/8841718.

[18] A. Ghosh, N. R. Pal, and J. Das, A fuzzy rule based approach to cloud cover estimation, *Remote Sens. Environ.*, 100(4) 2006, 531–549.

[19] N. R. Pal and S. Saha, Simultaneous structure identification and fuzzy rule generation for Takagi–Sugeno models, *IEEE Trans. Syst. Man, Cybern. Part B*, 38(6) 2008, 1626–1638.

[20] H. Ahmad, A. Akgül, T. A. Khan, P. S. Stanimirović, and Y. M. Chu, New perspective on the conventional solutions of the nonlinear time-fractional partial differential equations, *Complexity*, 2020, doi: 10.1155/2020/8829017.

[21] H. Ahmad, T. A. Khan, and C. Cesarano, Numerical solutions of coupled burgers' equations, *Axioms*, 2019, doi: 10.3390/axioms8040119.

[22] J. Wang, W. Wu, and J. M. Zurada, Deterministic convergence of conjugate gradient method for feedforward neural networks, *Neurocomputing*, 74(14–15) 2011, 2368–2376.

[23] M. R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, 49(1). NBS Washington, DC, 1952.

[24] R. Fletcher and C. M. Reeves, Function minimization by conjugate gradients, Comput. J., 7(2) 1964, 149–154, doi: 10.1093/comjnl/7.2.149.

[25] E. Polak and G. Ribiere, Note sur la convergence de méthodes de directions conjuguées, ESAIM Math. Model. Numer. Anal. Mathématique Anal. Numérique, 3(1) 1969, 35–43.

[26] E. P. de Aguiar *et al.*, EANN 2014: a fuzzy logic system trained by conjugate gradient methods for fault classification in a switch machine, Neural Comput. Appl., 27(5) 2016, 1175–1189.

[27] T. Gao, J. Wang, B. Zhang, H. Zhang, P. Ren, and N. R. Pal, A polak-ribière-polyak conjugate gradient-based neuro-fuzzy network and its convergence, IEEE Access, 2018, doi: 10.1109/ACCESS.2018.2848117.

[28] H. Ahmad, A. R. Seadawy, and T. A. Khan, Study on numerical solution of dispersive water wave phenomena by using a reliable modification of variational iteration algorithm, Math. Comput. Simul., 2020, doi: 10.1016/j.matcom.2020.04.005.

[29] I. Del Campo, J. Echanobe, G. Bosque, and J. M. Tarela, Efficient hardware/software implementation of an adaptive neuro-fuzzy system, *IEEE Trans. Fuzzy Syst.*, 16(3) 2008, 761–778.

[30] K. T. Chaturvedi, M. Pandit, and L. Srivastava, Modified neo-fuzzy neuron-based approach for economic and environmental optimal power dispatch, *Appl. Soft Comput.*, 8(4) 2008, 1428–1438.

[31] H. Ichihashi, Iterative fuzzy modelling and a hierarchical network, 1991.

[32] C.-J. Lin and W.-H. Ho, An asymmetry-similarity-measure-based neural fuzzy inference system, *Fuzzy Sets Syst.*, 152(3) 2005, 535–551.

[33] M. Tang, K. jun Wang, and Y. Zhang, A research on chaotic recurrent fuzzy neural network and its convergence, in *2007 International Conference on Mechatronics and Automation*, 2007, 682–687.

[34] J.-S. Jang, ANFIS: adaptive-network-based fuzzy inference system, *IEEE Trans. Syst. Man. Cybern.*, 23(3) 1993, 665–685.

[35] X. G. Luo, D. Liu, and B. W. Wan, An adaptive fuzzy neural inferring network, *Fuzzy Syst. Math.*, 12(4) 1998, 26–33.

[36] C.-F. Juang and J.-S. Chen, Water bath temperature control by a recurrent fuzzy controller and its FPGA implementation, *IEEE Trans. Ind. Electron.*, 53(3) 2006, 941–949.

[37] A. Sahiner , N. Yilmaz and S. A. Ibrahem, Smoothing approximations to non-smooth functions, *J. Multidiscip. Model. Optim.*, 1(2) 2018, 69-74.

[38] A. S. Ahmed, Optimization Methods For Learning Artificial Neural Networks, University of Mosul, 2018.

[39] A. Sahiner and S. A. Ibrahem, A new global optimization technique by auxiliary function method in a directional search, *Optim. Lett.*, 2019, doi: 10.1007/s11590-018-1315-1.

[40] Y. Liu and C. Storey, Efficient generalized conjugate gradient algorithms, part 1: theory, *J. Optim. Theory Appl.*, 69(1) 1991, 129–137.

[41] K. K. Abbo and H. M. Khudhur, New A hybrid Hestenes-Stiefel and Dai-Yuan conjugate gradient algorithms for unconstrained optimization, *Tikrit J. Pure Sci.*, 21(1) 2015, 118–123.

[42] Y. A. Laylani, K. K. Abbo, and H. M. Khudhur, Training feed forward neural network with modified Fletcher-Reeves method, *J. Multidiscip. Model. Optim.*, 1(1) 2018, 14–22.

[43] K. K. Abbo, Y. A. Laylani, and H. M. Khudhur, Proposed new Scaled conjugate gradient algorithm for unconstrained optimization, *Int. J. Enhanc. Res. Sci. Technol. Eng.*, 5(7) 2016.

[44] Z. M. Abdullah, M. Hameed, M. K. Hisham, and M. A. Khaleel, Modified new conjugate gradient method for Unconstrained Optimization, *Tikrit J. Pure Sci.*, 24(5) 2019, 86–90.

[45] H. M. Khudhur, Numerical and analytical study of some descent algorithms to solve unconstrained Optimization problems, University of Mosul, 2015.

[46] K. K. Abbo, Y. A. Laylani, and H. M. Khudhur, A new spectral conjugate gradient algorithm for unconstrained optimization, *Int. J. Math. Comput. Appl. Res.*, 8 2018, 1–9.

[47] M. Al-Baali, Descent property and global convergence of the Fletcher—Reeves method with inexact line search, *IMA J. Numer. Anal.*, 5(1) 1985, 121–124.

[48] L. Zhang and W. Zhou, Two descent hybrid conjugate gradient methods for optimization, *J. Comput. Appl. Math.*, 216(1) 2008, 251–264.

[49] K. K. Abbo and H. M. Khudhur, New A hybrid conjugate gradient Fletcher-Reeves and Polak-Ribiere algorithm for unconstrained optimization, *Tikrit J. Pure Sci.*, 21(1) 2015,124–129.

[50] H. N. Jabbar, K. K. Abbo, and H. M. Khudhur, Four--Term Conjugate Gradient (CG) Method Based on Pure Conjugacy Condition for Unconstrained Optimization, *Kirkuk Univ. J. Sci. Stud.*, 13(2) 2018, 101–113.

[51] T. Gao, Z. Zhang, Q. Chang, X. Xie, P. Ren, and J. Wang, Conjugate gradient-based Takagi-Sugeno fuzzy neural network parameter identification and its convergence analysis, *Neurocomputing*, 2019, doi: 10.1016/j.neucom.2019.07.035.

Hisham M. Khudhur, ORCID: https://orcid.org/0000-0001-7572-9283
Khalil K. Abbo, ORCID: https://orcid.org/0000-0001-5858-625X

Journal of Multidisciplinary Modeling and Optimization 3(2) (2020),70-79.

Bezier Curve Based Smoothing Penalty Function for Constrained Optimization

Ahmet SAHINER[1,a], Nurullah YILMAZ[1,b], Gulden KAPUSUZ[1,c] and Gamze OZKARDAS[1,d]

[1]Deparment of Mathematics,
Suleyman Demirel University, Isparta, TURKEY
[a] ahmetsahiner@sdu.edu.tr
[b] nurullahyilmaz@sdu.edu.tr
[c] guldenkapusuz92@gmail.com
[d] gamzeozkrds@gmail.com

Received: 01.02.2021, Accepted: 07.03.2021, Published: 25.03.2021

Abstract — *In this study, we consider nonlinear inequality constrained optimization problems. We introduce l_1 exact penalty function approach with a new smoothing function based on Bezier curve. Then, we propose a new algorithm by using the differentiation based methods to solve for solving l_1 exact penalty functions. Finally, we apply our algorithm to test problems to demonstrate the effectiveness of the algorithm.*

Keywords: l_1 penalty function, Smoothing, Non-smooth optimization.
Mathematics Subject Classification: 90C30, 65D10, 90C26.

1 Introduction

In this study, we deal with the constrained optimization problem as follows

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) \\ &s.t.\ \ g_i(x) \leq 0, \quad i = 1, 2, ..., m. \end{aligned} \tag{1}$$

where $f, g_i : \mathbb{R}^n \to \mathbb{R}$, $i \in I = \{1, 2, ..., m\}$ are continuously differentiable functions. The set of feasible solution is defined as $G_0 := \{x \in \mathbb{R}^n | g_i(x) \leq 0, i = 1, 2, ...m\})$ and it is assumed that G_0 is not empty.

The penalty function is used in order to transform a constrained problem to an unconstrained one. The following problem is one of the well-known penalty form of problem 1:

$$\min_{x\in\mathbb{R}^n} F_2(x, \rho) = f(x) + \rho \sum_{i=1}^{m} \left(g_i^+(x)\right)^2, \tag{2}$$

Cite as: *A. Sahiner, N. Yilmaz, G. Kapusuz and G. Ozkardas, Bezier curve based smoothing penalty function for constrained optimization, Journal of Multidisciplinary Modeling and Optimization 3(2) (2020),70-79.*

where $\rho > 0$ is a penalty parameter and $g_i^+(x) = \max\{0, g_i(x)\}$, $i \in I$. Clearly, $F_2(x, \rho)$ is continuously differentiable exact penalty function. According to Zangwill [1], an exact penalty function has been defined by

$$\min_{x \in \mathbb{R}^n} F_1(x, \rho) = f(x) + \rho \sum_{i=1}^{m} g_i^+(x). \tag{3}$$

The obvious difficulty in minimization of F_1 is the non-differentiability of F_1 which originates from the presence of "$\max$" operator (when the power of $\max$ is equal 1). The exact l_1 penalty function has been studied by many interesting studies [2, 3]. The penalty approach is used many areas such as academic problems: image processing problems [4], min-max problems [5], PDE constrained control optimization problems [6] and also many engineering problems [7].

One of the most popular way of solving these kind of non-smooth problems is smoothing techniques. The idea of behind the smoothing techniques is based on the approximation to the non-smooth objective function by smooth functions. The degree of approximation is controlled by parameters. The first studies are on smoothing techniques [8, 9, 10, 11, 12]. In order to improve the smoothing approaches, different types of valuable techniques and algorithms are developed [13, 14, 16, 15, 17]. Smoothing techniques are widely used for solving exact penalty functions. The first study is given in [18] and many new studies has been arisen with different smoothing techniques [19, 20, 21, 22, 24, 25, 26]. The smoothing exact penalty functions has been an active research area in recent years [27, 28, 29]

In this paper, we first present a new smoothing function based on Bezier curve. Then, we apply smoothing approach with exact penalty functions and construct the smoothing l_1 exact penalty functions. Finally, we develop a new algorithm by using the differentiation based methods and the implementation of our algorithm to test problems is demonstrated.

2 Preliminaries

Throughout the paper, x^k is denoted as local minimizer and x^* is denoted as the global minimizer. $\mathbb{R}_+$ denote the non-negative real numbers and $\|\cdot\|$ denote the Euclidean norm.

The smoothing function of non-smooth functions is defined by the following definition:

Definition 1. *[30] Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous function and $\varepsilon > 0$. The function $\tilde{f} : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is called a smoothing function of $f(x)$, if $\tilde{f}(\cdot, \varepsilon)$ is continuously differentiable in $\mathbb{R}^n$ for any fixed ε, and for any $x \in \mathbb{R}^n$,*

$$\lim_{z \to x, \varepsilon \to 0} \tilde{f}(z, \varepsilon) = f(x).$$

The Bezier curve is successfully used for smoothing of the $\min$ operator in [31] to obtain filled function for global optimization. We plan to construct a new smoothing function for penalty problem by the help of Bezier curve. A Bezier curve is defined as follows:

Definition 2. *[32] A Bezier curve of degree n is a parametric curve with control points $P_0, P_1, ..., P_n$, and it is expressed in terms of Bernstein polynomials given by*

$$B_i^n(t) = \begin{pmatrix} n \\ i \end{pmatrix} (1-t)^{n-i} t^i$$

where the binomial coefficients are

$$\begin{pmatrix} n \\ i \end{pmatrix} = \begin{cases} \frac{n!}{i!(n-i)!} & if \quad 0 \leq i \leq n \\ 0 & else \end{cases}$$

Therefore, a Bezier curve of degree n *is explicitly defined by*

$$\beta(t) = \sum_{i=0}^{n} B_i^n(t) P_i, \quad t \in [0,1].$$

In general, finding exact solution is quite hard task for the complicated constrained optimization problems. Therefore, the approximate solution is useful for these types of problems.The $\varepsilon-$feasible solution for inequality constrained optimization problems is defined as follows:

Definition 3. *[20] Assume* $\varepsilon > 0$*, a point* x_ε *is called* $\varepsilon-$*feasible solution of problem* (1)*, if*

$$g_i(x) \leq \varepsilon, \quad i = 1, 2, \ldots, m.$$

3 A New Smoothing Approach Based on Bezier Curve for Exact Penalty Functions

Let us define the $h : \mathbb{R} \to \mathbb{R}$ such that $h(t) = \max\{t, 0\}$. It easy to see that, the function $h(t)$ is re-written as

$$h(t) = t\chi_A(t), \tag{4}$$

where $A = \{t \in \mathbb{R} : t > 0\}$ and $\chi_A : \mathbb{R} \to \mathbb{R}$ is indicator function of a set A defined as

$$\chi_A(t) = \begin{cases} 1 & , t \in A, \\ 0 & , t \notin A. \end{cases}$$

Considering the Eqn. (4), if anyone smooth out the function $\chi_A(t)$, then smoothing function of $h(t)$ is obtained. Therefore, we plan to construct a new smoothing function by the help of Bezier curves. The smoothing function is obtained as follows:

$$\tilde{h}(t, \varepsilon) = t\tilde{\chi_A}(t, \varepsilon),$$

where $\tilde{\chi}_A(x, \varepsilon)$ is the smoothing function of indicator function $\chi_A(t)$ and

$$\tilde{\chi}_A(t, \varepsilon) = \begin{cases} 0 & , t \leq -\varepsilon/2 \\ \dfrac{(t + 0.5\varepsilon)^2}{\varepsilon^3} (3\varepsilon - 2(t + 0.5\varepsilon)) & , -\varepsilon/2 \leq t < \varepsilon/2 \\ 1 & , t \geq \varepsilon/2 \end{cases}$$

It can easily verify that the function $\tilde{h}(t, \varepsilon)$ is continuously differentiable on $\mathbb{R}$.

Lemma 1. *Assume that* $\varepsilon > 0$ *then*

$$0 \leq h(t) - \tilde{h}(t, \varepsilon) \leq \frac{\varepsilon}{4} \tag{5}$$

for any $t \in \mathbb{R}$.

Proof. Since $\chi_A(t) = \tilde{\chi}_A(t,\varepsilon)$ when $t \notin [-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}]$, it is enough to show that the inequality (5) holds for any $t \in [-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}]$. For $\varepsilon > 0$ we have

$$\begin{aligned} 0 \leq h(t) - \tilde{h}(t,\varepsilon) &= t\chi_A(t) - t\tilde{\chi_A}(t,\varepsilon) \\ &\leq \frac{\varepsilon}{4}. \end{aligned}$$

It completes the proof. □

It can be concluded from the Lemma 1 that $\tilde{h}(t,\varepsilon) \to h(t)$ as $\varepsilon \to 0$.

By the help of the smoothing and penalty formulation we can construct the following problem

$$\min_{x \in \mathbb{R}^n} \tilde{F}_1(x,\varepsilon,\rho) \tag{6}$$

instead of the problem given in (3). Here the function $\tilde{F}_1(x,\rho,\varepsilon)$ is defined as

$$\tilde{F}_1(x,\varepsilon,\rho) := f(x) + \rho \sum_{i=1}^{m} \tilde{h}(g_i(x),\varepsilon).$$

Now, we are ready to give the following theoretical results.

Theorem 3.1. *Let $x \in \mathbb{R}^n$ and $\varepsilon > 0$ then,*

$$0 \leq F_1(x,\rho) - \tilde{F}_1(x,\varepsilon,\rho) \leq \frac{m}{4}\rho\varepsilon. \tag{7}$$

Proof. From Lemma 1 we obtain

$$\begin{aligned} F_1(x,\rho) - \tilde{F}_1(x,\varepsilon,\rho) &= \rho \sum_{i=1}^{m} h(g_i(x)) - \rho \sum_{i=1}^{m} \tilde{h}(g_i(x),\varepsilon) \\ &= \rho \sum_{i=1}^{m} \Big(h(g_i(x)) - \tilde{h}(g_i(x),\varepsilon) \Big) \\ &\leq \frac{m}{4}\rho\varepsilon. \end{aligned}$$

□

Theorem 3.2. *Suppose that $\{\varepsilon_j\} \to 0$ and x^j is a solution of* (6) *for any $\rho > 0$. Assume that $\overline{x}$ is an accumulation point of $\{x^j\}$. Then $\overline{x}$ is an optimal solution for* (3).

Proof. The proof is obtained from the Theorem 3.1. □

Theorem 3.3. *Let x^* be an optimal solution for the problem* (3) *and $\overline{x}$ be an optimal solution for the problem* (6). *Then we have the following:*

$$0 \leq F_1(x^*,\rho) - \tilde{F}_1(\overline{x},\varepsilon,\rho) \leq \frac{m\rho\varepsilon}{4}. \tag{8}$$

Proof. From the Theorem 3.1 we have the following:

$$\begin{aligned} F_1(x^*,\rho) - \tilde{F}_1(x^*,\rho,\varepsilon) &\leq F_1(x^*,\rho) - \tilde{F}_1(\bar{x},\varepsilon,\rho) \\ &\leq F_1(\bar{x},\rho) - \tilde{F}_1(\bar{x},\varepsilon,\rho) \\ &\leq \frac{m\rho\varepsilon}{4}. \end{aligned}$$

□

Theorem 3.4. *Let x^* be an optimal solution for* (3)*, $\overline{x}$ be an optimal solution for* (6) *and let x^* be a feasible solution for (P) and $\overline{x}$ be an $\varepsilon-$ feasible solution for (P), then we have*

$$0 \leq f(x^*) - f(\bar{x}) \leq \frac{m\rho\varepsilon}{2}. \tag{9}$$

Proof. From the Theorem 3.3, we have

$$\begin{aligned} F_1(x^*,\rho) - \tilde{F}_1(\bar{x},\varepsilon,\rho) &= f(x^*) + \rho\sum_{i=1}^{m} h(g_i(x^*)) - \left(f(\bar{x}) + \rho\sum_{i=1}^{m} \tilde{h}\left(g_i(\bar{x}),\varepsilon\right)\right) \\ &\leq \frac{m\rho\varepsilon}{4} \end{aligned}$$

and since $\sum_{i=1}^{m} h(g_i(x^*)) = 0$, we obtain

$$\rho\sum_{i=1}^{m} \tilde{h}\left(g_i(\bar{x}),\varepsilon\right) \leq f(\bar{x}) - f(x^*) \leq \rho\sum_{i=1}^{m} \tilde{h}\left(g_i(\bar{x}),\varepsilon\right) + \frac{m\rho\varepsilon}{4}.$$

Since $\bar{x}$ is $\varepsilon-$feasible then we have

$$\rho\sum_{i=1}^{m} \tilde{h}\left(g_i(\bar{x}),\varepsilon\right) \leq \frac{m\rho\varepsilon}{4}.$$

Therefore, we obtain

$$0 \leq f(x^*) - f(\bar{x}) \leq \frac{m\rho\varepsilon}{2}.$$

□

4 Algorithm and Numerical Examples

In this section, we first propose an algorithm to solve (6) as follows:

Penalty Function Algorithm (PFA)

Step 1 Choose the initial point x^0. Determine $\varepsilon_0 > 0$, $\rho_0 > 0$, $0 < \delta < 1$, and $M > 1$, let $k = 0$ and go to Step 2.

Step 2 Use x^k as an initial point to solve (6). Let x^{k+1} be the solution.

Step 3 If x^{k+1} is $\varepsilon-$feasible for (1), then stop and x^{k+1} is the optimal solution. If not, determine $\rho_{k+1} = M\rho_k$, $\varepsilon_{k+1} = \delta\varepsilon_k$ and $k = k+1$, then go to Step 2.

In order to guaranteed that the algorithm is worked straightly, we have to prove the following theorem.

Theorem 4.1. *Assume that the set*

$$\underset{x\in\mathbb{R}^n}{argmin}\ \tilde{F}_1(x,\varepsilon,\rho) \tag{10}$$

is not empty for $\rho \in [\rho_0, \infty)$ *and* $\varepsilon \in (0, \varepsilon_0]$. *Further assume that* x^k *is generated by* PFA *when* $\delta M < 1$. *If* $\{x^k\}$ *has a limit point, then the limit point of* x^k *is the solution for the problem* (1).

Proof. Assume $\overline{x}$ is a limit point of $\{x^k\}$. Then there exists set $K \subset \mathbb{N}$, such that $x^k \to \overline{x}$ for $k \in K$. We have to show that $\overline{x}$ is the optimal solution for (1). Thus, it is sufficient to show (i) $\overline{x} \in G_0$ and (ii) $f(\overline{x}) \leq \inf_{x\in G_0} f(x)$.

i. Let us consider the contrary that $\overline{x} \notin G_0$, i.e. for sufficiently large $k \in K$, there exist $\tau_0 > 0$ and $i_0 \in \{1, 2, \ldots, m\}$ such that

$$g_{i_0}(x^k) \geq \tau_0 > 0.$$

Since x^j is the global minimum according $k-$th values of the parameters ρ_k, ε_k, for any $x \in G_0$ we have

$$\begin{aligned} F_1(x^k,\varepsilon_k,\rho_k) &= f(x^k) + \rho_k(\tau_0 + \frac{\varepsilon_k}{2}) + \frac{(m-1)}{2}\rho_k\varepsilon_k \\ &= f(x^k) + \rho_k\tau_0 + \frac{m}{2}\rho_k\varepsilon_k \\ &\leq f(x) + \frac{m}{2}\rho_k\varepsilon_k. \end{aligned}$$

If $k \to \infty$ then, $\rho \to \infty$, $\rho_k\varepsilon_k \to 0$ and $\rho_k\tau_0 \to \infty$. Thus, $f(x)$ takes infinite values on G_0 and it contradicts with the boundedness of f on G_0.

ii. By considering the Step 2 in *PFA* and for any $x \in G_0$,

$$\tilde{F}_1(x^k,\varepsilon_k,\rho_k) \leq \tilde{F}_1(x,\varepsilon_k,\rho_k) = f(x) + \frac{1}{4}m\rho_k\varepsilon_k$$

When $k \to \infty$, we have $f(\overline{x}) \leq f(x)$.

□

Now we are ready to apply *PFA* to numerical examples. The *PFA* is programmed in Matlab R2016A. For these tables we use some symbols in order to abbreviate the expressions. The symbols are described as follows:

$Iter$	:	The total number of iterations.
Obj	:	The value of solution minimum point x^*.
$C.val$	:	The maximum value of error value for constraints.
$Time$	:	The total time in seconds.

We consider the 4 different test problems which are given in details [26].

Problem 1. *Let us consider the Example in [19]*

$$\begin{aligned} \min f(x) &= x_1^2 + x_2^2 - \cos(17x_1) - \cos(17x_2) + 3 \\ \textit{s.t.} \quad & g_1(x) = (x_1 - 2)^2 + x_2^2 - 1.6^2 \leq 0, \\ & g_2(x) = x_1^2 + (x_2 - 3)^2 - 2.7^2 \leq 0, \\ & 0 \leq x_1 \leq 2, \;\; 0 \leq x_2 \leq 2. \end{aligned}$$

The global minimum is obtained at a point $x^* = (0.7254, 0.3993)$ *with the corresponding value* 1.8376.

Problem 2. *Let us consider the example in [21],*

$$\begin{aligned} \min f(x) &= 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3 \\ \textit{s.t.} \quad & x_1^2 + x_2^2 + x_3^2 - 25 = 0, \\ & (x_1 - 5)^2 + x_2^2 + x_3^2 - 25 = 0 \\ & (x_1 - 5)^2 + (x_2 - 5)^2 + (x_3 - 5)^2 - 25 \leq 0 \end{aligned}$$

The global minimum is obtained at a points $x^* = (2.5000, 4.2196, 0.9721)$ *and the value of the point is* 944.2157.

Problem 3. *The Rosen-Suzuki problem in [19]*

$$\begin{aligned} \min f(x) &= x_1^2 + x_2^2 + 2x^3 + x_4^2 - 5x_1 - 21x_3 + 7x_4 \\ \textit{s.t.} \quad & 2x_1^2 + x_2^2 + x_3^2 + 2x_1 + x_2 + x_4 - 5 \leq 0, \\ & x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \leq 0, \\ & x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \leq 0. \end{aligned}$$

In the paper [19], the obtained global value is obtained as -44.23040.

Problem 4. *Let us consider the Example in [21, 18]*

$$\begin{aligned} \min f(x) &= 10x_2 + 2x_3 + x_4 + 3x_3 + 4x_6 \\ \textit{s.t.} \quad & x_1 + x_2 - 10 = 0, \\ & -x_1 + x_3 + x_4 + x_5 = 0, \\ & -x_2 - x_3 + x_5 + x_6 = 0, \\ & 10x_1 - 2x_3 + 3x_4 - 2x_5 - 16 \leq 0, \\ & x_1 + 4x_3 + x_5 - 10 \leq 0, \\ & 0 \leq x_1 \leq 12, \; 0 \leq x_2 \leq 18, \\ & 0 \leq x_3 \leq 5, \; 0 \leq x_4 \leq 12, \\ & 0 \leq x_5 \leq 1, 0 \leq x_6 \leq 16, \end{aligned}$$

In the paper [21], the obtained global minimum value is obtained as 117.000004.

The *PFA* is applied to test problems and the detailed result is presented in Table 1. In Table 1, the total number of function iterations, the value of the objective function at the optimal point, the maximum error values of constraints at the optimal point and the total spending time obtained from our algorithm and competing algorithms have been reported. The numerical results show that our algorithm is present better results among the all algorithms.

Table 1: The numerical results

	PFA				Algorithm I				Algorithm II			
Problem No	Iter	Obj	C.val	Time	Iter	Obj	C.val	Time	Iter	Obj	C.val	Time
1	3	1.8376	−0.0000	0.446089	3	1.8376	−0.0000	0.458735	3	1.8376	−0.0000	0.482673
2	2	944.2156	0.0000	0.345145	4	944.2157	0.0000	0.486354	3	944.2157	0.0000	0.448798
3	3	−44.2338	−0.0000	0.444549	3	−44.2338	−0.0000	0.519692	4	−44.2322	−0.0000	0.552898
4	4	117.0100	0.0000	0.474952	3	117.0182	0.0000	0.795644	3	117.0071	0.0000	0.884352

5 Conclusion

In this study, we propose new smoothing technique based on Bezier Curve for l_1 exact penalty function. We design a new algorithm to solve smoothing penalty expression of the problem (1). We perform some numerical experiments on test problems and obtain satisfactorily results.

Our new smoothing technique needs to tune just one parameter. Thus, it is easy to set the best parameter value in the process of the algorithm. It can be conclude that our approach provide good approximations to this kind of penalty functions. The algorithm is user friendly and effective. It has fast convergence properties in comparing with the other penalty algorithms. Moreover, the numerical results consolidate the efficiency of the algorithm.

Acknowledgments

The authors declare that this work is completely private effort and it has not been supported by any governmental, financial, educational or otherwise establishment.

Conflict of Interest Declaration

The authors declare that there is no conflict of interest statement.

Ethics Committee Approval and Informed Consent

The authors declare that there is no ethics committee approval and/or informed consent statement.

References

[1] W. I. Zangwill, Nonlinear programing via penalty functions, Manage Sci., 13 1967, 344–358.

[2] M. V. Dolgopolik, Smooth exact penalty functions: a general approach, Optim. Lett. 10(3) 2016, 635–648.

[3] M. V. Dolgopolik, Smooth exact penalty functions II: a reduction to standard exact penalty functions, Optim. Lett., 10(7) 2016, 1541–1560.

[4] J.Liu, R. Ma, X. Zeng, W. Liu, M. Wang, H. Chen, An efficient non-convex total variation approach for image deblurring and denoising, Appl. Math. Comput., 397 2021, 125977.

[5] C. Ma, X. Li, K. F. Cedric Yiu, L.-S Zhang, New exact penalty function for solving constrained finite min-max problems, Appl. Math. Mech., 33(2) 2012, 253–270 .

[6] A. Jayswal, An exact l_1 penalty function method for multi-dimensional first-order PDE constrained control optimization problem, Eur. J. Control, 52 2020, 34-41.

[7] R. Manikantan, S. Chakraborty, T. K. Uchida, C. P. Vyasarayani, Parameter Identification in Nonlinear Mechanical Systems with Noisy Partial State Measurement Using PID-Controller Penalty Functions Mathematics 8 2020, 1084.

[8] D. Bertsekas, Nondifferentiable optimization via approximation, Mathematical Programming Study, 3 1975, 1–25.

[9] I. Zang, A smooting out technique for min-max optimization, Math. Programm., 19 1980, 61–77.

[10] A. Ben-Tal, M. Teboule, Smoothing technique for nondifferentiable optimization problems, Lecture notes in mathematics, 1405, Springer-Verlag, Heidelberg, 1989, 1-11.

[11] C. Chen, O.L. Mangasarian, A Class of Smoothing Functions for Nonlinear and Mixed Complementarity Problem, Comput. Optim. Appl., 5 1996, 97–138.

[12] A. M. Bagirov, A. Al Nuamiat, N. Sultanova Hyperbolic smoothing functions for nonsmooth minimization, Optimization, 62 (6), 2013, 759–782.

[13] A. E. Xavier, The hyperbolic smoothing clustering method, Pattern Recognition, 43 2010, 731–737.

[14] A. E. Xavier,A. A. F. D. Oliveira, Optimal covering of plane domains by circles via hyperbolic smoothing, J. Glob. Optim. **31** (2005) 493-504.

[15] C. Grossmann, Smoothing techniques for exact penalty function methods, Contemporary Mathematics, In book Panaroma of Mathematics: Pure and Applied, 658 249–265.

[16] N. Yilmaz, A. Sahiner, New smoothing approximations to piecewise smooth functions and applications, Numer. Funct. Anal. Optim., 40(5) 2019, 523–534.

[17] N. Yilmaz, A. Sahiner, On a new smoothing technique for non-smooth, non-convex optimization, Numer. Algebra Control Optim., 10 (3) 2020, 317–330.

[18] M.C. Pinar, S. Zenios, On smoothing exact penalty functions for convex constrained optimization, SIAM J. Optim., 4 1994, 468-511.

[19] S. J. Lian, Smoothing approximation to l_1 exact penalty for inequality constrained optimization, Appl. Math. Comput., 219 2012, 3113–3121.

[20] B. Liu, On smoothing exact penalty function for nonlinear constrained optimization problem, J. Appl. Math. Comput., 30 2009, 259–270.

[21] X. Xu, Z. Meng, J. Sun, R. Shen A penalty function method based on smoothing lower order penalty function, J. Comput. Appl. Math., 235 2011, 4047-4058.

[22] Z. Meng, C. Dang, M. Jiang, R. Shen A smoothing objective penalty function algorithm for inequality constrained optimization problems, Numer. Funct. Anal. Optim., 32 2011, 806-820.

[23] Z. Y. Wu, H. W. J. Lee, F. S. Bai, L. S. Zhang, Quadratic smoothing approximation to l_1 exact penalty function in global optimization, J. Ind. Manage. Optim., 53 2005, 533–547.

[24] Z.Y.Wu, F.S.Bai, X.Q.Yang, L.S.Zhang, An exact lower orderpenalty function and its smoothing in nonlinear programming, Optimization, 53 2004, 51-68.

[25] F.S.Bai, Z.Y.Wu, D.L. Zhu, Lower order calmness and exact penalty fucntion, Optim. Methods Softw., 21 2006, 515–525.

[26] A. Sahiner, G. Kapusuz, N. Yilmaz, A new smoothing approach to exact penalty functions for inequality constrained optimization problems, Numer. Algebra Control Optim., 6 (2) 2016, 161–173.

[27] X. Xu, C. Dang, F. T. S. Chan and Yongli Wang, On smoothing l_1 exact penalty function for constrained optimization problems Numer. Funct. Anal. Optim., 40 (1) 2019, 1–18

[28] J. Min, Z. Meng, G. Zhou, R. Shen, On the smoothing of the norm objective penalty function for two-cardinality sparse constrained optimization problems, Neurocomputing, 2020 Doi: 10.1016/j.neucom.2019.09.119.

[29] Qian Liu, Yuqing Xu, Yang Zhou, A class of exact penalty functions and penalty algorithms for nonsmooth constrained optimization problems, J. Glob. Optim., 76 2020, 745–768.

[30] X. Chen, Smoothing Methods for nonsmooth, nonconvex minimzation, Math. Programm. Serie B, 134 2012, 71–99.

[31] A. Sahiner, N. Yilmaz and G. Kapusuz, A new global optimization method and applications, Carpathian Math. J., 33(3) 2017, 373-380.

[32] G.E. Farin, Curves and Surfaces for CADG: A Partical Guide, Morgan Kaufmann, San Fransico, 2002.

Ahmet Sahiner, ORCID: https://orcid.org/0000-0002-4945-2476
Nurullah YILMAZ, ORCID: https://orcid.org/0000-0001-6429-7518
Gulden Kapusuz, ORCID: https://orcid.org/0000-0002-6316-1501
Gamze Ozkardas, ORCID: https://orcid.org/0000-0003-4262-0120