Journal of Artificial Intelligence and Data Science (JAIDA)
e-ISSN: 2791-8335
https://dergipark.org.tr/pub/jaida

Vol. 2, No. 1, pp. 43-52, 2022
Research Article

Optimization of Wire Electrical Discharge Machining (WEDM) Process Parameters Using Neuro-Regression Analysis for Fabrication of Precision Electrodes with Complex Shapes

Meliha BAŞTÜRK

Volt Elektrik Motor San. ve Tic. A.Ş., Electrical-Electronics Engineer, Kemalpasa, Izmir, Turkey

Abstract
The wire electrical discharge machining (WEDM) process is extremely important in the fabrication of complex electrodes with delicate structures. Identifying optimal operating combinations is a challenge in industry due to the large number of process variables. To overcome this difficulty, neuro-regression analysis was used and optimization was carried out. In this study, the six-parameter WEDM process was first modeled in twelve different regression models using a neuro-regression analysis. These parameters are discharge current, pulse duration, pulse frequency, wire speed, wire tension and dielectric flow rate. In addition, multiple regression model types including linear, quadratic, trigonometric and rational forms were tested in twelve different regression models. Then, optimization study was carried out with four different algorithms to obtain minimum kerf and surface roughness. These algorithms are Nelder-Mead Algorithm, Differential Evolution Algorithm, Simulated Annealing Algorithm and Random Search Algorithm. The study shows that WEDM process parameters can be adjusted to achieve better surface finish and cutting width at the same time. The process is optimized by minimizing kerf and surface roughness. Optimization results are 0.17044 mm for kerf and 3.60393 µm for surface roughness. It is seen that the processing model is suitable and the optimization technique meets the practical requirements.
Keywords: Kerf; neuro-regression analysis; optimization; surface roughness; WEDM.

1. Introduction

WEDM technology is based on the principle of electro-erosion. During this process, metals corrode each other. With the control of the electro-erosion process, it may be possible to manufacture conductive materials of complex shapes. It is a very popular and widely used technology for cutting and manufacturing plastic molds, which is also used in the production of gear wheels and machine parts. WEDM is preferred by researchers/industry to save time, energy and money with the spread of mechanization and the development of technology [1].

Many investigations have been performed in the field of WEDM, which include process modeling studies. Mouralova et al. [2] investigated it is important that the kerf width is as small as possible to ensure precision machining. For this reason, the slit was investigated in metallographic sections using light and electron microscopy. Surface roughness is important for the finishing cut of WEDM. Han et al. [3] experimentally investigated the effects of WEDM parameters including pulse time, discharge current, continuous pulse time, pulse interval time, polarity effect, material and dielectric or surface roughness in finishing. Saha and Mondal [4] examined the influence of WEDM parameters such as discharge pulse time, discharge stop time, servo voltage. In addition to experimental studies, combination of Taguchi's robust design concept and principal component analysis were applied to optimize those process parameters. Kumar et al. [5] analyzed the analytical hierarchy process and genetic algorithm studies to determine the best WEDM conditions for hybrid composites. In that research, three response parameters such as metal removal rate, surface roughness and spark gap were considered for process optimization. Shihab [6] used the Box-Behnken design to maximize metal removal rate while achieving low kerf and surface roughness in optimization. He realized the optimum process parameters. A new approach to determinate cutting parameters in WEDM, integrated artificial neural network (ANN), and wolf pack algorithm based on the strategy of the leader (LWPA) proposed by Ming et al. [7]. He presented an ANN-LWPA integration system with multiple fitness functions to solve the modeling problems. Two smart prediction tools, namely general regression neural network (GRNN) and multiple regression analysis (MRA) models, were developed in the research of Majumder and Maity [8]. In that study they predicted and compared some key machinability properties such as average notch width, average surface roughness, and material removal rate. Analysis of variance was performed by Phate et al. [9] to understand the effect of WEDM process parameters on overall WEDM efficiency. WEDM response properties such as surface roughness (Ra), material removal rate and kerf width were considered in that study. The ANN was employed for enhancing the performance of the process. Manjaiah et al. [10] used multi-objective optimization by using Taguchi-based utility approach to optimize Ra.

Corresponding author e-mail address: meliha.basturk98@gmail.com

Received: 12/05/2022; Accepted: 27/06/2022

In this study it is aimed to optimize kerf (cutting width) and surface roughness. Taguchi design and neuro-regression method were used during optimization process. Neuro-regression analysis were performed methodically, including linear, quadratic, trigonometric, logarithmic, and their rational forms. $R^2_{training}$, $R^2_{testing}$ and $R^2_{validation}$ values were checked for the validity of the models. Stochastic optimization methods were used to maximize or minimize the objective functions to optimum values. Finally, different direct search methods (modified differential evaluation, Nelder-Mead [11], random search and simulated annealing algorithms) are methodically performed.

2. Materials and Methods

2.1 Modeling

In the modeling phase, a method combining the use of neuro-regression analysis and ANN was used. In order to apply this method, two different data separation was used. In the first one, all data were divided into three parts. 80% of the data was used for training, 15% for testing, and 5% for validation. In the second one, the same data set were split into two parts as 80% for training and 20% for testing. Moreover 5% validation data was selected from the second part of the data whose percentage was determined as 20% before. Both splitting methods mentioned above were applied and the most suitable method for modeling was chosen. The second is more likely to pass stages. If we explain these stages briefly, the training stage aims to minimize the error between the experimental and predicted values by changing the given regression models and coefficients. The test stage is performed to obtain the prediction results by minimizing the effects of regression model mismatches. It is crucial to check the boundary values with the predicted ones to show whether the model is realistic or not. After obtaining suitable models for training and testing, the maximum and minimum values of the given models were calculated for each design variable. Equations of twelve different regression models are given in Table 1. Multiple regression models and types were used including linear, quadratic, trigonometric, logarithmic and rational forms [12].

Table 1. *Regression models name with nomenclature – formula.*

Model Name	Nomenclature	Formula
Multiple Linear	L	a[1] + a[2] x1 + a[3] x2 + a[4] x3 + a[5] x4 + a[6] x5 + a[7] x6
Multiple Linear Rational	LR	(a[1] + a[2] x1 + a[3] x2 + a[4] x3 + a[5] x4 + a[6] x5 + a[7] x6)/(b[1] + b[2] x1 + b[3] x2 + b[4] x3 + b[5] x4 + b[6] x5 + b[7] x6)
Second-Order Multiple Nonlinear	SON	a[1] + a[2] x1 + a[3] x2 + a[4] x3 + a[5] x4 + a[6] x5 + a[7] x6 + a[8] x1^2 + a[9] x2^2 + a[10] x3^2 + a[11] x4^2 + a[12] x5^2 + a[13] x6^2+ 2 a[14] x1 x2 + 2 a[15] x1 x3 + 2 a[16] x1 x4 + 2 a[17] x1 x5 + 2 a[18] x1 x6 + 2 a[19] x2 x3 + 2 a[20] x2 x4 + 2 a[21] x2 x5
Second-Order Multiple Nonlinear Rational	SONR	(a[1] + a[2] x1 + a[3] x2 + a[4] x3 + a[5] x4 + a[6] x5 + a[7] x6 + a[8] x1^2 + a[9] x2^2 + a[10] x3^2 + a[11] x4^2 + a[12] x5^2 + a[13] x6^2 + 2 a[14] x1 x2 + 2 a[15] x1 x3 + 2 a[16] x1 x4 + 2 a[17] x1 x5 + 2a[18] x1 x6 + 2 a[19] x2 x3 + 2 a[20] x2 x4 + 2 a[21] x2 x5)/(b[1] + b[2] x1 + b[3] x2 + b[4] x3 + b[5] x4 + b[6] x5 + b[7] x6 + b[8] x1^2 + b[9] x2^2 + b[10] x3^2 + b[11] x4^2 + b[12] x5^2 + b[13] x6^2 + 2 b[14] x1 x2 + 2 b[15] x1 x3 + 2 b[16] x1 x4 + 2 b[17] x1 x5 + 2 b[18] x1 x6 + 2 b[19] x2 x3 + 2 b[20] x2 x4 + 2 b[21] x2 x5)
First-Order Trigonometric Multiple Nonlinear	FOTN	a[1] + a[2] Sin[x1] + a[3] Sin[x2] + a[4] Sin[x3] + a[5] Sin[x4] + a[6] Sin[x5] + a[7] Sin[x6] + a[8] Cos[x1] + a[9] Cos[x2] + a[10] Cos[x3] + a[11] Cos[x4] + a[12] Cos[x5] + a[13] Cos[x6]
First-Order Trigonometric Multiple Nonlinear Rational	FOTNR	a[1] + a[2] Sin[x1] + a[3] Sin[x2] + a[4] Sin[x3] + a[5] Sin[x4] + a[6] Sin[x5] + a[7] Sin[x6] + a[8] Cos[x1] + a[9] Cos[x2] + a[10] Cos[x3] + a[11] Cos[x4] + a[12] Cos[x5] + a[13] Cos[x6])/(b[1] + b[2] Sin[x1] + b[3] Sin[x2] + b[4] Sin[x3] + b[5] Sin[x4] + b[6] Sin[x5] + b[7] Sin[x6] + b[8] Cos[x1] + b[9] Cos[x2] + b[10] Cos[x3] + b[11] Cos[x4] + b[12] Cos[x5] + b[13] Cos[x6])
Second-Order Trigonometric Multiple Nonlinear	SOTN	a[1] + a[2] Sin[x1] + a[3] Sin[x3]^2 + a[4] Sin[x3]^2 + a[5] Sin[x4]^2 + a[6] Sin[x5]^2 + a[7] Sin[x6]^2 + a[8] Cos[x1] + a[9] Cos[x2] + a[10] Cos[x3] + a[11] Cos[x4] + a[12] Cos[x5] + a[13] Cos[x6] + a[14] Sin[x1] + a[15] Sin[x2] + a[16] Sin[x3] + a[17] Sin[x4] + a[18] Sin[x5] + a[19] Sin[x6] + a[20] Cos[x1]^2 + a[21] Cos[x2]^2 + a[22] Cos[x3]^2 + a[23] Cos[x4]^2 + a[24] Cos[x5]^2 + a[25] Cos[x6]^2
Second-Order Trigonometric Multiple Nonlinear Rational	SOTNR	a[1] + a[2] Sin[x1] + a[3] Sin[x3]^2 + a[4] Sin[x3]^2 + a[5] Sin[x4]^2 + a[6] Sin[x5]^2 + a[7] Sin[x6]^2 + a[8] Cos[x1] + a[9] Cos[x2] + a[10] Cos[x3] + a[11] Cos[x4] + a[12] Cos[x5] + a[13] Cos[x6] + a[14] Sin[x1] + a[15] Sin[x2] + a[16] Sin[x3] + a[17] Sin[x4] + a[18] Sin[x5] + a[19] Sin[x6] + a[20] Cos[x1]^2 + a[21] Cos[x2]^2 + a[22] Cos[x3]^2 + a[23] Cos[x4]^2 + a[24] Cos[x5]^2 + a[25] Cos[x6]^2)/(b[1] + b[2] Sin[x1] + b[3] Sin[x3]^2 + b[4] Sin[x3]^2 + b[5]

		Sin[x4]^2 + b[6] Sin[x5]^2 + b[7] Sin[x6]^2 + b[8] Cos[x1] + b[9] Cos[x2] + b[10] Cos[x3] + b[11] Cos[x4] + b[12] Cos[x5] + b[13] Cos[x6] + b[14] Sin[x1] + b[15] Sin[x2] + b[16] Sin[x3] + b[17] Sin[x4] + b[18] Sin[x5] + b[19] Sin[x6] + b[20] Cos[x1]^2 + b[21] Cos[x2]^2 +b[22] Cos[x3]^2 + b[23] Cos[x4]^2 + b[24] Cos[x5]^2 + b[25] Cos[x6]^2)
First-Order Logarithmic Multiple Nonlinear	FOLN	a[1] + a[2] Log[x1] + a[3] Log[x2] + a[4] Log[x3] + a[5]Log[x4] + a[6] Log[x5] + a[7] Log[x6]
First-Order Logarithmic Multiple Nonlinear Rational	FOLNR	(a[1] + a[2] Log[x1] + a[3] Log[x2] + a[4] Log[x3] + a[5] Log[x4] + a[6] Log[x5] + a[7] Log[x6])/(b[1] + b[2] Log[x1] + b[3] Log[x2] + b[4] Log[x3] + b[5] Log[x4] + b[6] Log[x5] + b[7] Log[x6])
Second-Order Logarithmic Multiple Nonlinear	SOLN	a[1] + a[2] Log[x1] + a[3] Log[x2] + a[4] Log[x3] + a[5] Log[x4] + a[6] Log[x5] + a[7] Log[x6] + a[8] Log[x1 x2] + a[9] Log[x1 x3] + a[10] Log[x1 x4] + a[11] Log[x1 x5] + a[12] Log[x1 x6] + a[13] Log[x2 x3] + a[14] Log[x2 x4] + a[15] Log[x2 x5] + a[16] Log[x2 x6] + a[17] Log[x3 x4] + a[18] Log[x3 x5] + a[19] Log[x3 x5] + a[20] Log[x3 x6] + a[21] Log[x4 x5] + a[22] Log[x4 x6] + a[23] Log[x5 x6] + a[24] Log[x1]^2 + a[25] Log[x2]^2 + a[26] Log[x3]^2 + a[27] Log[x4]^2 + a[28] Log[x5]^2 + a[29] Log[x6]^2
Second-Order Logarithmic Multiple Nonlinear	SOLNR	a[1] + a[2] Log[x1] + a[3] Log[x2] + a[4] Log[x3] + a[5] Log[x4] + a[6] Log[x5] + a[7] Log[x6] + a[8] Log[x1 x2] + a[9] Log[x1 x3] + a[10] Log[x1 x4] + a[11] Log[x1 x5] + a[12] Log[x1 x6] + a[13] Log[x2 x3] + a[14] Log[x2 x4] + a[15] Log[x2 x5] + a[16] Log[x2 x6] + a[17] Log[x3 x4] + a[18] Log[x3 x5] + a[19] Log[x3 x5] + a[20] Log[x3 x6] + a[21] Log[x4 x5] + a[22] Log[x4 x6] + a[23] Log[x5 x6] + a[24] Log[x1]^2 + a[25] Log[x2]^2 + a[26] Log[x3]^2 + a[27] Log[x4]^2 + a[28]Log[x5]^2 + a[29] Log[x6]^2)/(b[1] + b[2] Log[x1] + b[3] Log[x2] + b[4] Log[x3] + b[5] Log[x4] + b[6] Log[x5] + b[7] Log[x6] + b[8] Log[x1 x2] + b[9] Log[x1 x3] + b[10] Log[x1 x4] + b[11] Log[x1 x5] + b[12] Log[x1 x6] + b[13] Log[x2 x3] + b[14] Log[x2 x4] + b[15] Log[x2 x5] + b[16] Log[x2 x6] + b[17] Log[x3 x4] + b[18] Log[x3 x5] + b[19] Log[x3 x5] + b[20] Log[x3 x6] + b[21] Log[x4 x5] + b[22] Log[x4 x6] + b[23] Log[x5 x6] + b[24] Log[x1]^2 + b[25] Log[x2]^2 + b[26] Log[x3]^2 + b[27] Log[x4]^2 + b[28] Log[x5]^2 + b[29] Log[x6]^2)*)

2.2 Optimization

Optimization means achieving results and fixing problems after making the best use of available resources. Optimization techniques can be classified as conventional and non-conventional. Conventional optimization techniques are only used for continuous and differentiable functions, such as limited variation and Lagrange multipliers. In the present study, Nelder-Mead Algorithm, Differential Evolution Algorithm, Simulated Annealing Algorithm and Random Search Algorithm were chosen to solve the optimization scenarios [13-14].

2.2.1 Nelder Mead Algorithm

Nelder Mead is a simplex method for finding the local minimum point(s) of a function containing several variables. The Nelder Mead method creates a simplex triangle for two variables and compares the function values of this triangle at the vertices. Thus, the optimum point is found. It is an easy and practical method applied in engineering problems [13].

2.2.2 Differential Evolution Algorithm

This technique is based on genetic algorithm. It is an intuitive optimization technique and has a random nature. The productive parameters of this algorithm are population size, crossover and scaling factor. This indicates that you are dealing with a population of solutions rather than repeating them. Among other approaches, differential evolution algorithm is a simple yet powerful technique [14].

2.2.3 Simulated Annealing Algorithm

In this technique, the global optimum value of the function ıs determined. The purpose of the algorithm is to achieve a global optimization and generally used in the computing applications [15].

2.2.4 Random Search Algorithm

Random Search Algorithm can be expressed as a kind of local random search in which each iteration depends on the candidate solution of the previous iteration. It takes samples from the entire search space. In this technique random search methods are available such as pure random search or uniform general random search. For example, it is known that random search is used for hyper parameter optimization in ANN [16].

2.3 Problem Definition

The wire is arranged as follows to ensure optimum working conditions in WEDM.

- The levels for the various control factors are given in Table 2.
- Electrical discharge process input parameters were modeled with Taguchi design and regression analysis was performed. In Table 3, the parameters related with the reference [17] are given.
- Design variables, where A: Discharge Current, B: Pulse Duration, C: Pulse Frequency, D: Wire Speed, E: Wire Tension, F: Dielectric Flow Rate. These variables have three different levels and given in Table 2. That table is provided from the reference source [17].
- $R^2_{training}$, $R^2_{testing}$ and $R^2_{validation}$ must be greater than 0.90,0.85 and 0.85 respectively . When these conditions are satisfied, it can be concluded that the model is appropriate.
- The validity of the obtained models was checked, and then the suitable models were solved by four different optimization algorithm methods.

2.4 Optimization Scenario

In this optimization problem, the objective function contained kerf (cutting gap) and surface roughness. All design variables are assumed to be real numbers and the search space is continuous. For this case, 16 Amp < Discharge Current (A) < 32 Amp, 3.20 µs < Pulse Duration (B) <12.80µs, 40KHz < Pulse Frequency (C) < 60 KHz, 7.60 m/min < Wire Speed (D) < 9.20 m/min, 1000 g <Wire Tension (E) < 1200 g, 1.20 Bars < Dielectric Flow Rate (F) < 1.40 Bars. The main purpose is to operate the WEDM at an optimal level. Therefore, kerf and surface roughness parameters should be minimized. Mathematically, the boundaries of the objective function can also be obtained with this approach.

Table 2. *Levels for various control factors [17]*

Control Factor	I	II	III
A. Discharge Current (Amp)	16.00	24.00	32.00
B. Pulse Duration (µs)	3.20	6.40	12.80
C. Pulse Frequency (KHz)	40	50	60
D. Wire Speed (m/min)	7.60	8.60	9.20
E. Wire Tension (g)	1000	1100	1200
F. Dielectric Flow Rate (Bars)	1.20	1.30	1.40

Table 3. *Taguchi Experimental Design, input parameters and experimental results [17]*

Expt. No.	A	B	F	C	D	E	R_a(µm)	Kerf(mm)
1	1	1	1	1	1	1	3.68	0.236
2	1	1	2	2	2	2	3.61	0.190
3	1	1	3	3	2	3	3.53	0.161
4	1	2	1	2	2	2	3.82	0.286
5	1	2	2	3	3	3	3.77	0.224
6	1	2	3	1	1	1	3.70	0.217
7	1	3	1	3	3	3	3.86	0.308
8	1	3	2	1	1	1	3.83	0.248
9	1	3	3	2	2	2	3.77	0.204
10	2	1	1	1	2	3	3.64	0.211
11	2	1	2	2	3	1	3.63	0.184
12	2	1	3	3	1	2	3.67	0.256
13	2	2	1	2	3	1	3.89	0.332
14	2	2	2	3	1	2	3.87	0.306
15	2	2	3	1	2	3	3.90	0.372
16	2	3	1	3	1	2	3.86	0.246
17	2	3	2	1	2	3	3.83	0.218
18	2	3	3	2	3	1	3.86	0.278
19	3	1	1	1	3	2	3.73	0.234
20	3	1	2	2	1	3	3.75	0.294
21	3	1	3	3	2	1	3.73	0.254
22	3	2	1	2	1	3	3.80	0.225

23	3	2	2	3	2	1	3.84	0.285
24	3	2	3	1	3	2	3.83	0.253
25	3	3	1	3	2	1	3.99	0.263
26	3	3	2	1	3	2	3.89	0.262
27	3	3	3	2	1	3	3.89	0.259

3. Results and Discussion

It was desired to minimize the surface roughness and kerf values in the wire electrical discharge machine. Various regression models were tested in the literature using $R^2_{training}$, $R^2_{testing}$, $and\ R^2_{validation}$. Multiple regression model types including linear, quadratic, trigonometric, and rational forms were tested. Taguchi design and regression analysis were used to test output-input models in the reference work. In this study, twelve regression models with six parameters were tested. The models are given in Table 1 in detail. In addition, the outputs for the models are given in Table 4 and Table 6. Optimization results and suggested designs are given in Table 5 and Table 7.

Table 5 and Table 7 show that four different optimization algorithms are used. These algorithms are Nelder-Mead Algorithm, Differential Evolution Algorithm, Simulated Annealing Algorithm and Random Search Algorithm. Mathematically, the limits of the objective function can also be obtained with this approach. This approach gives evidence of correct modeling and limitations in optimization after applying twelve different regression models. The results for all algorithms seem close to each other. However, these differences should be analyzed well in sensitive interpretations. The scope of the study is to see the effect of the results of these optimization algorithms on the optimization. Based on this, it can be said that the optimization algorithms in this study give similar values for different algorithms in the results. However, in experimental studies with different data, algorithms can give different results, so it is easier to interpret the effect of the algorithm on optimization. However, it is difficult to explain the difference between optimization algorithms by looking at the results in this study.

The kerf value is minimized for optimum operating conditions. Kerf value is 0.17044 mm. SOLN (Second Order Logarithmic Multiple Nonlinear) as a model. The simulated annealing algorithm was used as a method. Optimal conditions are Discharge Current :32 Amp, Pulse Duration: 3.2 µs, Dielectric Flow Rate: 1.26657 Bars, Pulse Frequency: 60 KHz, Wire Speed:9.2 m/min, Wire Tension: 1200 g.

Table 4. *The results of the regression models and limitation are for kerf.*

Models	$R^2_{training}$	$R^2_{testing}$	$R^2_{validation}$	**Maximum**	**Minimum**
Y=0.345525 + 0.00227546 x1 + 0.00170305 x2 +0.0106114 x3 - 0.00111863 x4 - 0.00218462 x5 - 0.0000948113 x6	0.96762	-1.55346	-1.04871	0.298836	0.199127
Y= (-156.984 + 2.91334 x1 + 4.70894 x2 - 155.444 x3 + 1.88426 x4 - 1.54588 x5 + 0.172973 x6)/(-814.977 + 11.7874 x1 + 21.5156 x2 - 679.797 x3 + 7.04886 x4 + 5.43501 x5 + 0.83685 x6)	0.99741	-278.943	-1.0293	178469.	-1.16373*10^{12}
Y=3.02691 + 0.0852935 x1 - 0.000278251 x1^2 + 0.0220959 x2 - 0.000478143 x1 x2 - 0.00253284 x2^2 - 3.72551 x3 + 0.0299919 x1 x3 - 0.0269791 x2 x3 + 1.2682 x3^2 + 0.0349188 x4 - 0.000104581 x1 x4 + 0.000716035 x2 x4 - 0.000415591 x4^2 + 0.0622959 x5 - 0.00214178 x1 x5 + 0.00365543 x2 x5 - 0.00495281 x5^2 - 0.00414701 x6 - 0.0000749325 x1 x6 +2.64107*10^-6 x6^2	0.987461	-4.64949	-0.0116655	0.512067	-0.0131406
Y=(1.00142 + 1.02142 x1 - 0.363206 x1^2 + 1.01119 x2 + 2.51615 x1 x2 - 2.03674 x2^2 + 0.996731 x3 + 2.02719 x1 x3 + 1.87877 x2 x3 + 0.989473 x3^2 + 1.00466 x4 + 8.66948 x1 x4 + 5.29075 x2 x4 - 2.41314 x4^2 + 1.0293 x5 + 3.7472 x1 x5 + 3.58219 x2 x5 + 1.39927 x5^2 + 1.65865 x6 + 5.41413	0.969526	-0.310494	-0.880369	0.269572	0.179999

x1 x6 - 0.0653058 x6^2)/(0.999656 + 0.998974 x1 + 1.55153 x1^2 + 1.00405 x2 + 2.47433 x1 x2 + 1.82439 x2^2 + 1.0005 x3 + 1.99323 x1 x3 + 2.05869 x2 x3 + 1.0018 x3^2 + 0.991958 x4 + 0.534298 x1 x4 + 2.55489 x2 x4 + 1.13551 x4^2 + 0.993309 x5 + 1.74626 x1 x5 + 1.8483 x2 x5 + 0.910865 x5^2 + 0.842715 x6 + 21.4831 x1 x6 - 0.239944 x6^2)					
Y=2.6147 + 0.0248117 Cos[x1] + 0.0767145 Cos[x2] - 0.695018 Cos[x3] + 0.00804462 Cos[x4] + 0.00206104 Cos[x5] - 0.0623113 Cos[x6] - 0.0158866 Sin[x1] - 0.485028 Sin[x2] - 2.23478 Sin[x3] + 0.0179596 Sin[x4] + 0.0467603 Sin[x5] + 0.00575167 Sin[x6]	0.983099	0.922505	-4.75779	0.943884	-0.320025
Y=2.6147 + 0.0248117 Cos[x1] + 0.0767145 Cos[x2] - 0.695018 Cos[x3] + 0.00804462 Cos[x4] + 0.00206104 Cos[x5] - 0.0623113 Cos[x6] - 0.0158866 Sin[x1] - 0.485028 Sin[x2] - 2.23478 Sin[x3] + 0.0179596 Sin[x4] + 0.0467603 Sin[x5] + 0.00575167 Sin[x6]	0.992522	-5.13909	-8.59265	$3.7212*10^7$	$-5.94437*10^6$
Y=0.0328006 + 0.035645 Cos[x1] + 0.0325077 Cos[x1]^2 + 0.0757116 Cos[x2] + 0.0400001 Cos[x2]^2 - 0.741854 Cos[x3] + 1.31834 Cos[x3]^2 + 0.00877994 Cos[x4] + 0.0394002 Cos[x4]^2 - 0.000300729 Cos[x5] + 0.0403801 Cos[x5]^2 + 0.0301413 Cos[x6] + 0.0290802 Cos[x6]^2 - 0.0304481 Sin[x1] - 0.471283 Sin[x2] + 0.027312 Sin[x3] + 0.0506393 Sin[x3]^2 + 0.00332714 Sin[x4] + 0.0721788 Sin[x4]^2 + 0.0377993 Sin[x5] + 0.0511693 Sin[x5]^2 - 0.000961744 Sin[x6] + 0.097492 Sin[x6]^2	0.983099	0.92225	-4.75779	0.838918	-0.433362
Y=(-0.143231 + 1.02503 Cos[x1] + 0.319877 Cos[x1]^2 + 1.33965 Cos[x2] + 1.64897 Cos[x2]^2 - 0.686355 Cos[x3] + 0.626178 Cos[x3]^2 + 0.337058 Cos[x4] + 0.442023 Cos[x4]^2 + 0.667737 Cos[x5] + 0.198311 Cos[x5]^2 + 0.172114 Cos[x6] + 0.375179 Cos[x6]^2 + 2.71112 Sin[x1] - 4.19607 Sin[x2] + 0.0334958 Sin[x3] + 0.461181 Sin[x3]^2 + 0.187331 Sin[x4] + 0.414746 Sin[x4]^2 + 0.783651 Sin[x5] + 0.658458 Sin[x5]^2 + 0.190198 Sin[x6] + 0.48159 Sin[x6]^2)/(1.50951 + 3.28362 Cos[x1] + 2.51938 Cos[x1]^2 + 2.60966 Cos[x2] + 1.00835 Cos[x2]^2 + 2.13008 Cos[x3] + 1.45514 Cos[x3]^2 + 1.24312 Cos[x4] + 1.49287 Cos[x4]^2 + 2.84951 Cos[x5] + 0.669053 Cos[x5]^2 + 1.64504 Cos[x6] + 1.67608 Cos[x6]^2 + 10.5281 Sin[x1] + 2.57822 Sin[x2] + 1.27688 Sin[x3] + 2.10872 Sin[x3]^2 + 0.812532 Sin[x4] + 1.01664 Sin[x4]^2 + 1.47664 Sin[x5] + 1.84045 Sin[x5]^2 + 1.11108 Sin[x6] + 0.83343 Sin[x6]^2)	0.99487	-2.43018	-4.29465	$7.71308*10^6$	$-2.57775*10^7$

2.6147 + 0.0248117 Cos[x1] + 0.0767145 Cos[x2] - 0.695018 Cos[x3] + 0.00804462 Cos[x4] + 0.00206104 Cos[x5] - 0.0623113 Cos[x6] - 0.0158866 Sin[x1] - 0.485028 Sin[x2] - 2.23478 Sin[x3] + 0.0179596 Sin[x4] + 0.0467603 Sin[x5] + 0.00575167 Sin[x6]	0.969421	-1.71268	-0.755777	0.304101	0.191605
(-410.542 + 505.756 Log[x1] + 86.3088 Log[x2] + 137.106 Log[x3] + 168.48 Log[x4] - 1260.57 Log[x5] + 104.764 Log[x6])/(-100.648 + 2029.13 Log[x1] + 468.664 Log[x2] - 344.646 Log[x3] + 358.247 Log[x4] - 4832.5 Log[x5] + 299.616 Log[x6])	0.996957	-17.7241	-2966.27	12704.9	-307377.
30.344 + 0.765363 Log[x1] - 0.221966 Log[x1]^2 + 0.509759 Log[x2] - 0.130349 Log[x2]^2 + 0.378032 Log[x1 x2] - 3.01852 Log[x3] + 2.01673 Log[x3]^2 + 0.639298 Log[x1 x3] + 0.355116 Log[x2 x3] + 0.616684 Log[x4] - 0.164074 Log[x4]^2 + 0.342368 Log[x1 x4] + 0.339301 Log[x2 x4] + 0.531829 Log[x3 x4] + 2.07915 Log[x5] - 1.36013 Log[x5]^2 + 0.612108 Log[x1 x5] + 0.711383 Log[x2 x5] + 3.25295 Log[x3 x5] + 0.518864 Log[x4 x5] - 2.90909 Log[x6] + 0.805314 Log[x6]^2 - 1.30757 Log[x1 x6] - 1.802 Log[x2 x6] - 2.71383 Log[x3 x6] - 1.12419 Log[x4 x6] - 1.59307 Log[x5 x6]	0.983099	0.92225	-4.75779	0.363967	0.139315
(0.392182 + 2.6288 Log[x1] + 3.23866 Log[x1]^2 + 7.93825 Log[x2] - 11.1651 Log[x2]^2 + 9.56705 Log[x1 x2] + 2.15288 Log[x3] + 1.93372 Log[x3]^2 + 3.78168 Log[x1 x3] + 9.09114 Log[x2 x3] - 0.31849 Log[x4] - 0.943493 Log[x4]^2 + 1.31031 Log[x1 x4] + 6.61976 Log[x2 x4] + 0.834391 Log[x3 x4] - 0.808902 Log[x5] - 4.41187 Log[x5]^2 + 0.819894 Log[x1 x5] + 6.12935 Log[x2 x5] + 0.687958 Log[x3 x5] - 2.12739 Log[x4 x5] - 1.3626 Log[x6] - 1.6747 Log[x6]^2 + 0.266192 Log[x1 x6] + 5.57565 Log[x2 x6] - 0.209723 Log[x3 x6] - 2.68109 Log[x4 x6] - 3.17151 Log[x5 x6])/(0.981338 + 3.25945 Log[x1] + 19.6328 Log[x1]^2 - 0.569116 Log[x2] + 4.65743 Log[x2]^2 + 1.69033 Log[x1 x2] + 0.303267 Log[x3] + 0.565446 Log[x3]^2 + 2.56272 Log[x1 x3] - 1.26585 Log[x2 x3] + 0.769713 Log[x4] - 0.298893 Log[x4]^2 + 3.02916 Log[x1 x4] - 0.799403 Log[x2 x4] + 0.0729798 Log[x3 x4] + 0.432774 Log[x5] - 1.1629 Log[x5]^2 + 2.69222 Log[x1 x5] - 1.13634 Log[x2 x5] - 0.527919 Log[x3 x5] + 0.202487 Log[x4 x5] + 0.604482 Log[x6] - 3.80114 Log[x6]^2 + 2.86393 Log[x1 x6] - 0.964634 Log[x2 x6] - 0.0922513 Log[x3 x6] + 0.374195 Log[x4 x6] + 0.0372559 Log[x5 x6])	0.984678	0.842674	-11.7328	0.404415	0.0481186

Table 5. *Results of optimization problems for the four models selected for kerf minimization.*

Objective Functions	Constraints	Optimization Algorithm	Kerf(mm) Minimization	Suggested Design
FOTN	16 < x1 < 32, 3.2 < x2 <12.8, 1.2 < x3 < 1.4 ,40 < x4 < 60, 7.6 < x5 < 9.2, 1000 < x6 < 1200	MDE MSA MRS MNM	-0.320025 -0.320025 -0.320025 -0.320025	x1 -> 21.4216, x2 -> 8.01085, x3 -> 1.26928, x4 -> 48.2736, x5 -> 9.2, x6 -> 1200. x1 -> 27.7048, x2 -> 8.01085, x3 -> 1.26928, x4 -> 54.5567, x5 -> 9.2, x6 -> 1181.15 x1 -> 27.7048, x2 -> 8.01085, x3 -> 1.26928, x4 -> 41.9904, x5 -> 9.2, x6 -> 1086.9 x1 -> 27.7048, x2 -> 8.01085, x3 -> 1.26928, x4 -> 54.5567, x5 -> 9.2, x6 -> 1105.75
SOTN	16 < x1 < 32, 3.2 < x2 < 12.8,1.2 < x3 < 1.4 ,40 < x4 < 60, 7.6 < x5 < 9.2,1000 < x6 < 1200	MDE MSA MRS MNM	-0.433362 -0.433362 -0.433362 -0.415825	x1 -> 27.0769, x2 -> 7.99065, x3 -> 1.27036, x4 -> 59.735, x5 -> 9.2, x6 -> 1077.56 x1 -> 27.0769, x2 -> 7.99065, x3 -> 1.27036, x4 -> 47.1687, x5 -> 9.2, x6 -> 1165.53 x1 -> 27.0769, x2 -> 7.99065, x3 -> 1.27036, x4 -> 59.735, x5 -> 9.2, x6 -> 1014.73 x1 -> 27.0769, x2 -> 7.99065, x3 -> 1.27036, x4 -> 50.2068, x5 -> 9.2, x6 -> 1064.99
SOLN	16 < x1 < 32, 3.2 < x2 < 12.8, 1.2 < x3 < 1.4 ,40 < x4 < 60, 7.6 < x5 < 9.2, 1000 < x6 < 1200	MDE MSA MRS MNM	0.139315 0.17044 0.149927 0.139315	x1 -> 16., x2 -> 3.2, x3 -> 1.26657, x4 -> 60., x5 -> 9.2, x6 -> 1200. x1 -> 32., x2 -> 3.2, x3 -> 1.26657, x4 -> 60., x5 -> 9.2, x6 -> 1200. x1 -> 16., x2 -> 12.8, x3 -> 1.26657, x4 -> 60., x5 -> 9.2, x6 -> \1200. x1 -> 16., x2 -> 3.2, x3 -> 1.26657, x4 -> 60., x5 -> 9.2, x6 -> 1200.
SOLNR	16 < x1 < 32, 3.2 < x2 < 12.8,1.2 < x3 < 1.4 ,40 < x4 < 60, 7.6 < x5 < 9.2, 1000 < x6 < 1200	MDE MSA MRS MNM	0.0481184 0.0481207 0.0481185 0.0481186	x1 -> 16., x2 -> 3.2, x3 -> 1.2, x4 -> 60., x5 -> 9.2, x6 -> 1200. x1 -> 16., x2 -> 3.2, x3 -> 1.2, x4 -> 59.9994, x5 -> 9.2, x6 -> 1200. x1 -> 16., x2 -> 3.2, x3 -> 1.2, x4 -> 60., x5 -> 9.2, x6 -> 1200. x1 -> 16., x2 -> 3.2, x3 -> 1.2, x4 -> 60., x5 -> 9.2, x6 -> 1200.

Table 6. *The results of the regression models and limitation are for surface roughness.*

Model	Maximum	Minimum	$R^2_{training}$	$R^2_{testing}$	$R^2_{validation}$
4.06381 + 0.104384 Log[x1] + 0.156206 Log[x2] - 0.218486 Log[x3] - 0.0115373 Log[x4] + 0.0443204 Log[x5] - 0.126955 Log[x6]	0.999779	0.865904	0.89335	3.9628	3.60393

Table 7. *Results of optimization problems for the one model selected for surface roughness minimization.*

Objective Functions	**Constraints**	**Optimization Algorithm**	**Surface Rouhness (μm) Minimization**	**Suggested Design**
FOLN	16 < x1 < 32, 3.2 < x2 <12.8, 1.2 < x3 < 1.4 ,40 < x4 < 60, 7.6 < x5 < 9.2, 1000 < x6 < 1200	MDE	3.60393	x1 -> 16., x2 -> 3.2, x3 -> 1.4, x4 -> 60., x5 -> 7.6, x6 -> 1200.
		MSA	3.60394	x1 -> 16.0001, x2 -> 3.20001, x3 -> 1.4, x4 -> 59.9918, x5 -> 7.60005, x6 -> 1199.99
		MRS	3.60393	x1 -> 16., x2 -> 3.2, x3 -> 1.4, x4 -> 59.9994, x5 -> 7.60001, x6 -> 1200.
		MNM	3.60393	x1 -> 16., x2 -> 3.2, x3 -> 1.4, x4 -> 59.9999, x5 -> 7.6, x6 -> 1200.

When the tables are examined, the suitability of the regression models is very important in optimizing. Because, in fact, every experiment has a working principle suitable for maximizing or minimizing in accordance with the working dynamics.

The surface roughness value is minimized for optimum operating conditions. Surface roughness value is 3.60393 **μ**m. FOLN (First Order Logarithmic Multiple Nonlinear) as a model The differential evolution was used as a method. Optimal conditions are Discharge Current :16 Amp, Pulse Duration: 3.2 μs, Dielectric Flow Rate: 1.4 Bars, Pulse Frequency: 60 KHz, Wire Speed: 7.6 m/min, Wire Tension: 1200 g.

For surface roughness and kerf, the results fit the input ranges. Provided the training, testing and validation phase for surface roughness. Provided the training and testing phase for Kerf. Looking at the tables in general, it can be seen that the study was successful (Table 4, Table 5, Table 6 and Table 7).

4. Conclusions

In this study, important processing parameters were tried to be determined separately for performance measurements such as surface roughness and kerf in the WEDM process. It has been seen that factors such as discharge current, pulse frequency and pulse duration and their interactions play an important role in rough cutting for kerf and surface roughness minimization.

Taguchi's experimental design method and multiple regression models are used to obtain the optimum parameter combination for minimization of kerf and surface roughness. The data separated as 80%, 15% and 5% in the modeling were randomly selected. This increased the reliability of the experiment. Thus, there was no accumulation in a certain number range. The most important part of the work is to check the accuracy of the limits of the proposed input parameters in the minimization results. This gives a great idea of the accuracy of the model. After using twelve different regression models, using four different optimization algorithms (Nelder-Mead Algorithm, Differential Evolution Algorithm, Simulated Annealing Algorithm and Random Search Algorithm) increased the accuracy of the optimization results.

This work can be extended in the future for different processing parameters and outputs. In addition, the work can be further improved with hybrid regression modeling techniques.

Declaration of Interest

The authors declare that there is no conflict of interest.

References

[1] Khatri, Bharat C., and Rathod, Pravin P. (2017). "Investigations on the performance of concentric flow dry wire electric discharge machining (WEDM) for thin sheets of titanium alloy." Int J Adv Manuf Technol Springer-Verlag London , 92,1945–1954.

[2] Mouralova, K., Kovar, J., Klakurkova, L.,and Prokes, T. (2018). "Effect of Width of Kerf on Machining Accuracy and Subsurface Layer After WEDM." JMEPEG ASM International, 27(4), 1908-1916

[3] Han, F., Jiang, J., Yu, D. (2006). "Influence of machining parameters on surface roughness in finish cut of WEDM." Int J Adv Manuf Technol Springer-Verlag London Limited, 34, 538–546.

[4] Saha, A., and Mondal, S. C. (2016). "Experimental investigation and modelling of WEDM process for machining nano-structured hardfacing material. " J Braz. Soc. Mech. Sci. Eng. The Brazilian Society of Mechanical Sciences and Engineering 2, 39, 3439–3455.

[5] Kumar, A., Grover N., Manna, A., Kumar, R., Chohan J. S., Singh, S., Singh, S., and Prunchu, C. L. (2021). "Multi-Objective Optimization of WEDM of Aluminum Hybrid Composites Using AHP and Genetic Algorithm." Arabian Journal for Science and Engineering Crown 2.

[6] Shihab, K. (2018). "Optimization of WEDM Process Parameters for Machining of Friction-Stir-Welded 5754 Aluminum Alloy Using Box–Behnken Design of RSM." Arabian Journal for Science and Engineering King Fahd University of Petroleum & Minerals , 43, 5017-5027.

[7] Ming, W., Hou, J., Zhang, Z., Huang, H., Xu, Z., Zhang, G., and Huang, Y. (2015). "Integrated ANN-LWPA for cutting parameter optimization in WEDM." Int J Adv Manuf Technol Springer-Verlag London , 84, 1277–1294.

[8] Majumder, H., and Maity, K. P. (2018) "Predictive Analysis on Responses in WEDM of Titanium Grade 6 Using General Regression Neural Network (GRNN) and Multiple Regression Analysis (MRA)." Silicon Springer Science+Business Media B.V. 2018, 10,1763–1776.

[9] Phate, M. R., Toney, S. B., and Phate, V. R. (2020). "Multi-parametric Optimization of WEDM Using Artificial Neural Network (ANN)-Based PCA for Al/SiCp MMC." J. Inst. Eng. India Ser. C The Institution of Engineers (India), 102(1), 169–181.

[10] Manjaiah, M., Laubscher, R. F., Kumar, A., and Basavarajappa, S. (2016). "Parametric optimization of MRR andsurface roughness in wire electro dischargemachining (WEDM) of D2 steel using Taguchi-based utility approach." International Journal of Mechanical and Materials Engineering Springer-Verlag London Limited, 11(7), 237-248.

[11] Polatoğlu, I., Aydın, L., Nevruz, B. Ç.,and Öze, S. (2020). "A Novel Approach for the Optimal Design of a Biosensor." Analytical Letters Taylor & Francis Group, LLC.

[12] Akcair, M., Savran, M., Aydın, L., Ayakdas, O., Ozturk, S.,and Kucukdogan, N. (2019) "Optimum design of anti-buckling behaviour of graphite/epoxy laminated composites by differential evolution and simulated annealing method. " Research on Engineering Structures and Materials, 5, 175–88.

[13] Zhang, Y., Price, C. J., Coope, I. D., Byatt, D. (2002). "A Convergent Variant of the Nelder–Mead Algorithm." JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS Plenum Publishing Corporation, 113(1), 5-19.

[14] Majumder, A., Das, A., Das, P. Kr. (2016). "A standard deviation based firefly algorithm for multi-objective optimization of WEDM process during machining of Indian RAFM steel." Neural Comput & Applic The Natural Computing Applications, 29, 665–677.

[15] Somashekhar, K. P., Mathew, J., & Ramachandran, N. (2012). "A feasibility approach by simulated annealing on optimization of micro-wire electric discharge machining parameters." Int J Adv Manuf Technol Springer-Verlag London, 61, 1209–1213.

[16] Kaelo, P., and Alİ, M. M. (2006). "Some Variants of the Controlled Random Search Algorithm for Global Optimization." JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS Springer Science+Business Media, Inc., 130(2), 253-264.

[17] Mahapatra, S. S., and Patnaik, A. (2007). "Optimization of wire electrical discharge machining (WEDM) process parameters using Taguchi method." The International Journal of Advanced Manufacturing Technology, 34, 911-925