Availableonline at: http://publikasi.mercubuana.ac.id/index.php/ijiem

IJIEM (Indonesian Journal of Industrial Engineering & Management)

ISSN (Print) : 2614-7327 ISSN (Online) : 2745-9063

Optimization and Selection of Boring Process Parameters for IS 2062 E250 Steel Plates Using Hybrid Taguchi-Pareto Box Behnken-Genetic Algorithm Method

Yakubu Umar Abdullahi [1], Sunday Ayoola Oke[1*]
[1] Department of Mechanical Engineering, University of Lagos, Lagos, 101017, Nigeria

ARTICLE INFORMATION

Article history:

Received: 4 May 2022
Revised: 24 May 2022
Accepted: 29 May 2022

Category: Research paper

Keywords:
Genetic algorithm
Boring operation
Optimization
Selection
Steel plates

A B S T R A C T

The integrated Taguchi-Pareto-Box-Behnken design (TP-BBD) method has been recognized as an effective method for boring operation optimization. Yet it has further optimization opportunities even with less information availability. In this study, the genetic algorithm (GA) was integrated with the TP-BBD method to form a novel TP-BBD-GA method to effectively deal with the paucity of boring data and generate multiple optimal solutions. Numerical simulation coupled with experimental data analysis was conducted to ascertain the effectiveness of the proposed method using literature data. To combine the procedure of the constituent methods, the authors analysed the literature data with the Taguchi-Pareto method. Then the output was used as inputs to the Box Behnken design method. Afterwards, linear programs with objective functions and constraints were formulated and introduced into the genetic algorithm structure and then solved using the python language. The results revealed that the proposed method exhibits good performance for boring operations as it predicts the best parameter i.e. speed, feed rate, depth of cut and noise radius values for optimal surface roughness values. This article offers a unique contribution to the boring literature since it examines an additional optimization procedure to the existing one. The study analyzes the optimization behaviour of the IS 2062 E250 steel plates in the boring process and gives an easy procedure for process engineers on improving the boring operations.

***Corresponding Author**
Sunday Ayoola Oke
E-mail: sa_oke@yahoo.com

1. INTRODUCTION

The prevailing practice of operators and process engineers in the boring industry is to seek optimization of resources, including those related to the implementation of parameters (Reddy et al., 2015; Nugroho et al., 2016; Izelu et al., 2021). Managers in the machining industry often weigh the likely operational cost of boring operations against the same cost when the parameters of the boring operation are

optimized (Atia et al., 2017). However, with the contemporary machining practice whereby dual/multiple optimization methods replace the recommendations of a single optimization method, a re-orientation of the operators and process engineer would radically transform the boring operation's performance. Unfortunately, a possible method of the Taguchi-Pareto Box Behnken approach, which may be deemed fit for optimizing the process machine parameters of the IS 2062 E250 steel plates cannot survive with the paucity of data (Abdullahi and Oke, 2022). It is also unable to provide multiple solution points through which several feasible alternatives could be chosen (Abdullahi and Oke, 2022). Further, in the present era of scarcity of operational resources where assurance about waste avoidance is required by the managers of machining processes from operators and process engineers. Besides, the methods to prioritise parameters according to the importance of prudent resource distribution are essential (Atia et al., 2017). From these arguments, it is obvious that an optimization method to re-assure an already optimized structure for further optimization is required. In this context, the genetic algorithm with useful attributes of operating successfully in situations involving scarcity of data and with multiple solution points is introduced (Dennison et al., 2012). In a combined form, the genetic algorithm is introduced to an existing method of Taguchi-Pareto Box Behnken to form the Taguchi-Pareto Box Behnken-genetic algorithm method.

The purpose of this research is to introduce and tests a new method called the Taguchi-Box Behnken design-genetic algorithm (TP-BBD-GA) that optimizes and selects the parameters of a boring process involving the use of IS2062 E250 steel plates. The developed approach has the competence of optimizing the inputs, which are the speed, depth of cut, feed and nose radius while the output is the surface roughness of the IS 2062 E250 steel plates. The experimental data obtained from Patel and Deshpande (2014) validated the method. The contribution of Patel and Desphande (2014) launched an initiative in performance enhancement in boring for the IS 2062 E250 steel plates using the Taguchi method. But Abiola and Oke (2021) probably discovered the negative influence of the inability to distinguish which of the Taguchi's parameters is more important than the other. Thus they argued that the use of three methods can avoid this effect. Consequently, they deployed the entropy, decision tree and the VIKOR approach to address this issue and illustrated the effectiveness of their method with the IS 2062 E250 steel plates data produced by Patel and Desphande (2014).

Yet some other authors such as Kilickap et al. (2011), Abiola and Oke (2022) exposed the unique opportunity of synergy in combining methods to solve the boring problem. In particular, Abiola and Oke (2022) addressed the surface roughness minimization issue using the IS 2062 E250 steel plates. Thus, they contributed a fuzzy analytic hierarchy process on one part and the combination of two other methods on the other part. These methods are the Markov chain together with any of the weighted sum-product, weighted product method and WASPAS method. Still in the endeavor of enhancing the performance of surface roughness of the IS 2062 E250 steel plates, Abdullahi and Oke (2022) established two different methods each containing the merger of Box Behnken design with either the Taguchi-Pareto or the Taguchi-ABC method. The authors expanded previous submissions in the literature using the data by Patel and Desphande (2014) to justify the two methods while boring the IS 2062 E250 steel plates. Despite the proposal by Abdullahi and Oke (2022), it is felt that an enhanced optimization may still be achieved using a robust optimizer called the genetic algorithm. Therefore, using their results and Patel and Deshpande's (2014) experimental data, the scope of the present study is to introduce the genetic algorithm as a potential enhancement tool, to further enhance the Taguchi-Pareto Box Behnken design and the Taguchi-ABC Box Behnken design using the IS 2062 E250 steel plate with the surface roughness improvement in focus.

By using Abdullahi and Oke's (2022) methods, the present study identifies what optimum solutions are obtainable when the genetic algorithm data is generated from the objective function formulated from the Box Behnken Design equation obtainable from Abdullahi and Oke (2022). The results were validated using

the regression equation as the objective function and running the genetic algorithm-based method using the coded programme from the Python language. Furthermore, based on the opinions from previous studies described in a state of the art review that showcases the differences between the previous and current studies, the novelty of the present research is stated as follows. This research explores the optimization behavior of the IS 2062 E250 steel plates under boring conditions and analyses this optimization attribute through the incorporation of the genetic algorithm into a known Taguchi-Pareto Box Behnken design method. The chief novelty of this article is the development of an efficient method for the boring operation of the IS 2062 E250 steel plates. Next, a multiphase optimization approach using the genetic algorithm integration, capable of operating successfully where the paucity of data exists is created to quickly achieve optimal parameters and outputs for the boring operation planning. Besides, the superiority and effectiveness of the developed approach are demonstrated using the IS 2062 E250 steel plates machined in a boring process on a CNC machine.

Nevertheless, for the benefits of this method, it could be stated that the multistage optimization approach incorporating initiation and mutation characteristics exploits the utmost benefits of soft computing aspects where the complicated and multiple computer statements are transformed quickly into feasible solutions for practical implementations (Reddy and Rao, 2005; Sardinas et al., 2006). Besides, the other benefits of this new method include the availability of multiple optimization points for choices of the most robust option. Also, a display of interaction behavior among variables is an additional benefit of the method. Lastly, the prioritization of parameters during the optimization process is possible. Thus, by implementing this developed framework, one can shed new light on the optimization characteristics of the IS 2062 E250 steel plates during the boring operation.

2. LITERATURE REVIEW

Optimization of process parameters in a boring operation is very vital in achieving a high-quality product of optimal surface roughness. In the present study, an original methodology is proposed to predict optimal parameters that could result in achieving high-quality surface roughness in the boring operation of E250 B0 steel material on a CNC machine tool. The proposed methodology is an integration of the Taguchi method, Pareto principle, Box Behnken design approach and an evolutionary genetic algorithm optimization approach. In this section, opinions from previous studies are described in a state of the art to clearly show the difference between previous studies and the present study. The literature review contains the contents of references related to materials, research methods, machines used, outputs considered by authors and the most common input parameters discussed by researchers.

2.1 General

In several studies, Dave et al. (2016), Ahmad et al. (2005) and Zoelan et al. (2013) used aluminium and other alloys as test materials. Khundrakpam et al. (2018), Palanisamy et al. (2007) and Patel and Deshpande (2014) employed steel and their alloys as the test materials. Furthermore, metallic and their composites were deployed by Kumar et al. (2012) and Zain et al. (2010) as their work materials. From these reviews, extremely scanty research was conducted with the IS 2062 E250 steel plates as it occurred only in a study. In that study, Patel and Deshpande (2014) the materials were used to establish the optimal parameters for minimum roughness in the boring activity. However, the IS 2062 E250 material is advantageous in its many features. Currently, the IS 2062 E250 steel grade is preferred to other materials because of its several features such as dimensional accuracy, thermal stability, corrosion resistance and finishing. Moreover, its practical use includes gear housing, metal plates, industrial pipes, bolts and nuts. However, with an expected annual growth rate in volume and revenue forecasts and the relative paucity of reports on it in the literature, it was taken as the choice material for the present study.

In the literature, different research methods were deployed in conjunction with a genetic algorithm, which is the principal addition of the present study to the literature. For instance, methods like response surface methodology, analysis of variance, self-optimizing adaptive

penalty, Taguchi method and Pareto sorting were integrated with the genetic algorithm optimization process to obtain optimal machining parameters for the best value of various output parameters (Kilickap et al., 2011). Furthermore, other authors used genetic algorithms alone to obtain the optimal machining parameters (Saffar et al., 2009; Marimuthu et al., 2015; Palanisamy et al., 2007). By thinking about the possibility of minimizing the experimental design cost, the Taguchi method is useful.

Also, considering the prioritization of the parameters, where the possibility of increasing productivity and the opportunity to connect production errors, the Taguchi-Pareto method is essential. Besides, by contemplating the probable interfaces between parameters, the Box Behnken design method is useful. Thus, by combining Taguchi methods, Abdullahi and Oke (2022) revealed how new and interesting results may be produced. However, with a continuous intensive demand for improvements by stakeholders in manufacturing, it is useful to extend Abdullahi and Oke's line of thinking for scholars to connect the strength of these various methods for improved operational performance. In this article, the authors achieve this by amalgamating the genetic algorithm with the methods presented by Abdullahi and Oke (2022). This task is aimed at motivating researchers and process engineers in thinking about multi-phase optimization since the present article further optimizes the results of previous authors.

Notwithstanding, the literature concerning machine usage considers the CNC as the main machine tool for verification experiments in different machining operations. These include turning, end milling, face milling, drilling and boring in verifying the optimal parameters from various analytical studies. In the following, the sole or combination of machining operations was demonstrated by the authors: boring and milling (Cubonova et al., 2019) and turning (Ganesan et al., 2011). Equally interesting are the varieties of machine tools used by authors in the literature. The CNC turning machine Moriseiki NV 2500 (Nugroho et al., 2016), CNC Batliboi Sprint (Patel and Deshpande, 2014), universal milling machine and the electric discharge machine are the varieties employed by authors in machining. Although several versions of machines were used by authors, the CNC Batliboi Sprint used by Patel and Deshpande (2014) is the choice in this work.

Besides, an important feature of the literature is an analysis of the outputs considered by authors. Common responses include surface roughness, drilled diameter, tool path, cutting force, machine power, tool life, operation time, material removal rate, milling force, vibration amplitude, production time, tool deflection, rake angle and burr heights. Among the responses, surface roughness as a single output was mostly used throughout the literature (Khundrakpan et al., 2018; Zain et al., 2010; Tien et al., 2020). Burr height was considered by Mahesh et al. (2015). Nonetheless, multiple outputs were considered by Ganesan et al. (2011), Sangwan and Kant (2017) and Dhavamani and Alwarsamy (2012). But surface roughness is a surface texture indicator calculated from deviations along the normal vector of the object's real surface against its ultimate mode. This output is essential to establish how an actual entity interrelates with its environment. Thus, the surface roughness measure may seem more important in certain cases than other output types. As in the present case of evaluating the IS 2962 E250 steel plate on a CNC machine, the surface roughness measure is adapted to be of interest to the present researchers.

Apart, based on the machining process and machine tools employed by various researchers, the most common input parameters used for CNC machining are spindle speed, feed rate, depth of cut and nose radius (Kumar et al., 2012). The pulse on-time current and pulse off time are used in electrical machining (Dave et al., 2016).

Notwithstanding, by considering the task of enlarging holes developed using the casting process or drilling on the lathe machine, the boring operation has relevance to the present industrial economy and the current article. Besides, the essential characteristics and benefits include the following. Firstly, regarding efficiency, boring operations are

efficient and ease the whole production process. Consequently, a growing number of industries are augmenting boring machines with their existing facilities. Secondly, boring has a long lifespan. Third, during its operation, noise pollution and odour are avoided. Therefore, using the increasingly demanded IS 2062 E250 steel plates; Patel and Deshpande (2014) showed the possibility of producing interesting outcomes from experiments. Since the IS 2062 E250 steel plates are growing in commercial appeal, in this article, the authors re-examine the data based on the IS 2062 E250 steel plates. This is aimed at broadening researchers' understanding of the important attributes and optimization possibilities with the material using the experimental data from Patel and Deshpande (2014).

Furthermore, as the literature search commenced, the researchers' attention was drawn to the article by Patel and Deshpande (2014), which tackled the performance analysis of the boring process while machining the IS 2062 E250 steel plates. But it became evident that an important gap exists in the article regarding the difficulty of the Taguchi method in specifying which of the parameters is better than the rest. Thus, the present authors monitored the attempts by previous authors to bridge this gap by Abiola and Oke (2021) and subsequently in another article by the same authors. However, more striking is the approach by Abdullahi and Oke (2022) as it diverges from the two mentioned earlier articles in improving on the Taguchi method as opposed to the earlier contributors that provided solutions to the problem through the amalgamation of Markov chains and WSM, WPM/WASPAS (Abiola and Oke, 2022) and the joint consideration of three methods, namely the entropy, VIKOR and decision tries (Abiola and Oke, 2021). The unique approach conducted by Abdullahi and Oke (2022) amalgamated the Taguchi method on one side with the Box Behnken design and Taguchi-Pareto and Box Behnken design on the other side. The two parts of the contributions by these authors contain two optimization methods each, namely the Taguchi and the Box Behnken on one side and the other two optimization methods, namely Taguchi-Pareto and Box Behnken design on the other side. However, the other method containing the Taguchi-Pareto is capable of prioritizing the parameters.

Besides, while progress is made in research to produce these methods by Abdullahi and Oke (2022), there is a continuous pressure mounting up to build more efficiency into existing friction stir welding processes. Although the method proposed by Abdullahi and Oke (2020) was optimized, further optimization was attempted by introducing the genetic algorithms into the proposal by Abdullahi and Oke (2020). Consequently, the major difference between the current research and the previous one is the introduction and demonstration of the genetic algorithms as a tool capable of further optimizing the integrated Taguchi-Box Behnken design and Taguchi-Pareto Box Behnken design method proposed by Abdullahi and Oke (2020). Towards the introduction of the genetic algorithms, an objective function was formed based on the box Behnken design outputs to be used in the codes written on genetic algorithm problems using python programming.

It should be noted that for the two articles of the present and the previous, the same IS 2062 E250 steel plates are used as the basis while the operation is the boring machining process. Furthermore, it is required to emphasize the coding of the problem using python programming. Previously, hand computations have been used by all the past researchers in the present study but the current article takes a unique turn from this practice to introduce programming codes in python language that will simplify the use of the procedures of the proposed method referred to as the integrated Taguchi-Pareto box Behnken design genetic algorithm.

2.2 Research gap

The proceeding section on the view of literature in the extant literature within the domain of boring operation has provided a comprehensive understanding of the perspectives of the literature. The review helped examine to what extent authors have impacted the literature on the use of optimization methods and also revealed how the issues of prioritization of factors while optimizing have not been tackled in most studies. The review also showed the

degree to which studies with sparse experimental data could be treated, which are completely ignored in most studies. Moreover, the search heuristic approach regarding the concurrent optimization, prioritization of factors and interaction analysis between factors are not found in the existing literature. This may reveal substantial importance in the boring operation's productivity, efficient control and economy. Also, the literature associated with approaches to evaluating boring operation optimization is less. To the authors' surprise, only a study on the Taguchi-Pareto-Box Behnken Design, which appears to be pioneering (Abdullahi and Oke, 2022) appears to be recently documented in this regard. Hence, it is understandable that optimization in concurrence with prioritization, interactions of factors and search algorithm in the context of probabilistic transition rules of genetic algorithm, which offers possible outcomes with related probabilities have not been identified. This is particularly true for the boring operation where the IS 2062 E250 plates are studied. The body of knowledge on boring operations has failed to examine the appropriate method necessary to provide results for boring operations where less processing information is available, the probabilistic nature of the process is considered and where an opportunity for a large set of solution space is provided. This could hamper the boring workshop operations despite the availability of skilled manpower and boring resources to achieve the system's objectives. However, it was also understood from the engineering literature that researchers deploy the combined genetic algorithm method with the Taguchi method, Pareto analysis and Box Behnken design to enhance performance. However, this has not been deployed to the boring operation's domain. No recommendations were offered to utilize the appropriate tools for hybridized optimization and concurrent prioritization and interactive analysis. In the current literature, it was understandable that no paper has shed light on the complete effort to improve performance using the Taguchi-Pareto-Box Behnken-genetic algorithm.

3. METHOD

3.1 Terminologies used in the boring operation and optimization

In this article, a new method for the optimization of boring activities is presented. Nonetheless, some terminologies were used, which relate to both the boring operation of the IS 2062 E250 steel plates on the CNC machine and the methods used. The meanings of these terms are explained in the present section.

Term	Description
Mutation	an operator that generates offspring chromosomes from a chromosome to maintain randomness in the process and to prevent premature convergence.
Mutation probability	the probability that indicates the amount of mutation that may take place in a set of chromosomes. The mutation probability is usually chosen to be low in optimization processes to prevent over-randomness in the process. For instance, if the mutation probability is chosen as 0.2, this means that mutation of 20% of a set of the chromosome is likely to take place.
Crossover	an operator that generates offspring from a population of parent chromosomes. It is implemented by swapping the chromosomes of two randomly selected parent chromosomes in the population of chromosomes at the cutting or swapping point. It is randomly selected too to give to two offspring.
Cut off point	is a randomly selected point at which crossover or interchanging of chromosomes in a population takes place in a crossover operation.
Crossover probability	the probability that determines if a crossover would take place or not, is usually fixed high, to increase the chances of its occurrence. It comes to play by randomly selecting a number between 0 and 1, if the selected random number is less than the chosen cross over probability, cross over takes place else it does not occur.
Bits	the encoded real parameters or factors into the binary parameter of 0s(zeros) and 1s(ones). They represent the chromosomes.
Parents	are best performing chromosomes in a population from which new solutions (offspring) are generated.

Offsprings	are newly generated solutions in a population, produced through the mutation and crossover operations.
Iteration	the number of times an algorithm is performed to reach the desired objective is also termed generation in the population-based optimization process.
Convergence	the description of how the values of a process tend to behave in the same way over a fixed number of iterations.
Convergence rate	the rate at which convergence occurs is how fast a process tends to converge in a given maximum number of iterations. A fast convergence rate converges earliest in iteration and vice versa.
Signal to noise ratio	the ratio of the information-carrying signal compared to the undesirable interferences (noise).it is of three distinct types; which are lower the better, nominal the best and greater the better.
Optimized parameter	the global best solution vector that results in the convergence of process optimization.
Surface roughness	a measure of how much deviation of a surface texture from its usual form. Where deviation is large then the surface is considered rough and vice versa.
Input parameter	a set of data that is introduced into a function, they can be referred to as independent variables in a process.
Output parameter	the result of the activity of a function to which input parameters were deployed.

3.2 Procedure for the implementation of the proposed method

In Table 1, a detailed description of the procedure and working of the TP-BBD-GA method is clearly explained. Here, the objective function from the concept of linear programming is generated and the search space, which is the constraints of each parameter is established. This is followed by initializing all algorithmic parameters of the genetic algorithm such as the crossover probability, mutation probability, the population size, the number of bits and the maximum number of iterations. The following is the procedure for the implementation of the proposed method:

Table 1. Procedures for the TP-BBD-GA approach

Step 1a	**Scenario 1**: Generate an objective function from the optimized parameter using the Pareto-Box Behnken design approach part of the work of Abdullahi and Oke (2022), using the concept of linear programming. For instance, assuming the optimized parameter from the Box Behnken design approach were 50 rpm of spindle speed, 0.002 for feed rate, 0.01 for depth of cut, 0.0 for nose radius. Using the linear programming concept we have the objective function generated as Equation (1), $F(x) = 50S + 0.002F + 0.01\ DC + 0.0\ NR$ (1) **Scenario 2**: Use the regression equation in the Pareto-Box Behnken design approach part of the work of Abdullahi and Oke (2022) as the objective function
Step 1b	Generate constraints for the objective function from the bounds of each parameter i.e. the upper and lower bounds. Take for instance the experimental value is in three-level of the four parameters under consideration. Say the speed parameter for the three-level of the experiment are 15rpm for level one, 35rpm for level 2 and 60rpm for level 3. Therefore, the constraint for the speed parameter would be generated as $15 \leq S \leq 60$. The same process is followed to generate constraints for the feed rate, depth of cut and nose radius parameters. Assume the constraint for the feed rate parameter is $0.001 \leq F \leq 0.005$ and that of the depth of cut is $0.005 \leq DC \leq 0.03$, while that of nose radius is also assumed to be $0.0 \leq NR \leq 0.006$.
Step 2a	Set or initialize the population size of chromosome to be considered. Take for example the population size to be 4, i.e. there are 4 different chromosomes in the population, which are represented by a binary number.
Step 2b	Set a constant number of bits for each factor or parameter under consideration for all chromosomes in the population, for instance, take the number of bits to be 4.
Step 2c	Set the crossover probability and the mutation probability i.e. the probability that determines if crossover and mutation would take place in the chromosome population respectively. Say for instance the crossover probability and the mutation probability are set as 0.5 and 0.2 respectively.
Step 2d	Set the number of generations or iterations to be considered in the algorithm. For instance, 5 iterations or generations are to be considered.
Step 3a	Randomly select four bits of binary numbers throughout the population. For each of the factors in consideration, note that the compositions of each chromosome in the population are the factors under consideration i.e. speed, feed rate, depth of cut and nose radius. Therefore, each factor in the chromosome must have 4 binary numbers throughout the population. For example, a chromosome can be represented as thus [1101 0011 1010 1110]. That is the first group of a four-bit binary number, which represents the

	speed parameter. The second group of a four-bit binary number represents the feed rate parameter while the third group represents the depth of cut parameter. Lastly, the fourth group of four binary numbers represents the nose radius parameter. Likewise, the rest of the chromosome in the population is presented in the same way.
Step 3b	Decode all the binary numbers in the population from base 2 to base 10 real numbers. Therefore, there will be four real numbers each in all the chromosomes in the population. For instance, decoding a chromosome in the population may result as thus: [12 7 15 1] where 12 represents the decoded binary number of the speed parameter, 7 represents the decoded binary number for the feed rate parameter, while 15 represents the decoded binary number for the depth of cut parameter, lastly, 1 represents the decoded binary number for the nose radius parameter, respectively. The same should apply to the rest of the chromosomes in the population.
Step 3c	Find the actual number of each decoded real number in the population using Equation (2) $x = x_{\min} + \frac{x_{\max} - x_{\min}}{2^n - 1} \times (Dv)$ (2) where x_{min} is the lower bound for the parameter in view, x_{max} represents the upper bound for the parameter studied, n is the number of bits while *Dv* is the decoded value. Recall that from the example in step 3b, the first decoded value for the speed parameter is 12. Thus, in computing the actual value, substitute all appropriate values into Equation (2) for the speed parameter, i.e. $x = 15 + \frac{60-15}{2^4 - 1} \times (12) = 51$ This represents the actual value of the speed parameter in the above chromosome instance. Use the same process to compute actual values for the remaining decoded values in the chromosome and throughout the population.
Step 4	Compute the fitness value of the population by substituting it into the objective function. For instance take a chromosome in the population to be [51,0.004,0.01,0.02], where 51,0.004,0.01, and 0.02 represents the actual value obtained from the decoded value of the speed, feed rate, depth of cut and nose radius, respectively, in a population. Now, these values are substituted into the objective function generated in step 1a, yielding F(x) =50 x 51+ 0.02 x 0.004 +0.01 x 0.01 +0.0 x 0.02, which gives 2550.0202 as the fitness value of this [51, 0.004, 0.01, 0.02] chromosome in the population. By so doing, all fitness values of chromosomes in the population are computed.
Step 5a	Conduct tournament competition between two randomly selected chromosomes in the population four times (i.e. the population size) with different contending mates each time. Use their fitness value to contend to populate the mating pool with four winners, and eliminate the weakest chromosome in the population. For example, take the fitness values of the chromosomes in the population as 2550.0202 P_1, 2003.0102 P_2, 2367.9001 P_3, 2451.1103 P_4. Then randomly select two pairs of chromosomes as P_2 and P_4, P_1 and P_2, P_1 and P_4, P_2 and P_3. The next activity is to compare the fitness value of each pair and keep our objective of minimization in mind i.e. the lower fitness value is more desirable to our course of finding optimal parameters which would lead to the lowest surface roughness. Therefore, the winners are introduced into the mating pool as thus; mating pool [2003.0102P_2, 2367.9001P_3, 2451.1103P_4, 2003.0102P_2]. Note that P_1 being the weakest in the population is eliminated while P_2 being the strongest in the population appears twice in the mating pool.
Step 5b	The binary number corresponding to the fitness value of chromosomes in the matting pool is considered as the parent population i.e. the fitness value in the mating pool is traced back to the various binary numbers that gave rise to them. For instance, the binary number corresponding to 2003.0102P_2 could be traced back to [1101 1111 0000 1010]. This is done for all the fitness values in the mating pool to give rise to four sets of binary chromosomes.
Step 6	Generate a random number between 0 and 1. If this number is less than the crossover probability then crossover would take place in the population of chromosomes. But if otherwise, the crossover would not occur in the population. Recall that our chosen crossover probability is 0.5 and taking the generated random number as 0.3, therefore, in this case, the crossover would take place. In applying crossover, select two pairs of parent chromosomes randomly then randomly choose a crossover point (i.e. when considering a single point cross over) from which the crossover would take place. Then all binary numbers after the crossover point are swapped between the two pairs of parent chromosomes. For example, consider this pair of chromosomes; [1101 1111 0000 1010] [1101 1101 0110 1010] Taking the crossover point to be at the 5th bit in both chromosomes, i.e. [1101 1111 0000 1010] [1101 1101 0110 1010]

	Then exchanging all the bits after the crossover point in the two-parent chromosomes, gives [11011|101 0110 1010] [11011|111 0000 1010] This same crossover procedure is applied to the second pair of parent chromosomes in the population. After the cross over, the resulting chromosomes are referred to as the offspring, and together, the offspring population. For example, the offspring population could be [11011101 0110 1010] [11011111 0000 1010] [01011100 0111 1010] [10011100 0011 1010]
Step 7	Conduct mutation of the offspring population based on the mutation probability chosen i.e. 0.2. The mutation is conducted by generating random numbers between 0 and 1 based on the number of bits considered in a chromosome. Each random number generated is attached to every bit in the chromosome. For instance, take the first chromosome in the population of the offspring and attach 16 randomly generated numbers between 0 and 1 to each bit in the chromosome as thus: [1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0] [0.1 | 0.4 | 0.6 | 0.2 | 0.7 | 0.5 | 0.1 | 0.9 | 0.8 | 0.1 | 0.3 | 0.2 | 0.1 | 0.9 | 0.2 | 0.5] Considering the mutation probability of 0.2, the mutation process is such that if the random number attached to a particular bit is less than the mutation probability that bit is changed, i.e. if it were 0, it is changed to 1 and if it were 1 it would be changed to 0. Applying mutation on the chromosome above, we have a mutated version of the chromosome as thus [0 1 0 1 1 1 1 1 0 0 1 0 0 0 1 0] which represents a stronger offspring chromosome. Therefore applying the mutation process on the whole chromosome is the population to yield a stronger offspring population.
Step 8	Repeat step 3b to step 4 on the mutated offspring population above. Recall that each chromosome is a group of four bits each representing four factors under consideration, i.e. a chromosome in the population must have 16 bits of binary numbers. Executing step 3b to step 4 on the offspring population would result in having four fitness values of the offspring population computed. For instance assuming [52, 0.002, 0.02, 0.004] actual chromosome values resulted in a fitness value of 2600.000204, [40, 0.003, 0.01, 0.002] actual chromosome values also result in a fitness value of 2000.000106, while [46, 0.001, 0.04, 0.001] actual chromosome values resulted in a fitness value of 2300.000402 and lastly [30, 0.005, 0.03, 0.006] actual chromosome values resulted in a fitness value of 1500.00031. With our objective of minimization in mind and comparing the fitness values, the lowest is 1500.0031 which represents the best fitness value in the population, its corresponding actual chromosome values are [30, 0.005, 0.03, 0.006] which represents the optimal solution in the first generation or iteration to be considered, where 30, 0.005, 0.03, 0.006 are the optimal speed, feed rate, depth of cut and nose radius parameter respectively.
Step 9a	Use the μ +ʎ strategy to combine the initial population with their respective fitness values with the offspring population and their respective fitness values Here, the initial population and its corresponding fitness values are placed above while the population of offspring and its corresponding fitness values are placed below (Note that the initial population and offspring population should be both in binary chromosome form).
Step 9b	Compare the fitness values of the combined population and select the four smallest fitness values since our objective is minimization, to generate a new population i.e. the binary chromosomes that correspond to the four selected fitness values would now represent a new population used for the next iteration or generation. This signifies the end of the first iteration.
Step 10	Repeat step 3b to step 9b on the new population each time till the set number of generations or iterations is reached.
Step 11	The genetic algorithm optimization approach explained above is then coded using the python programming language.
Step 12	Report results from the two scenarios outputted from the python coded genetic algorithm optimization approach

Furthermore, the flowchart of the research method is shown in Fig. 1.

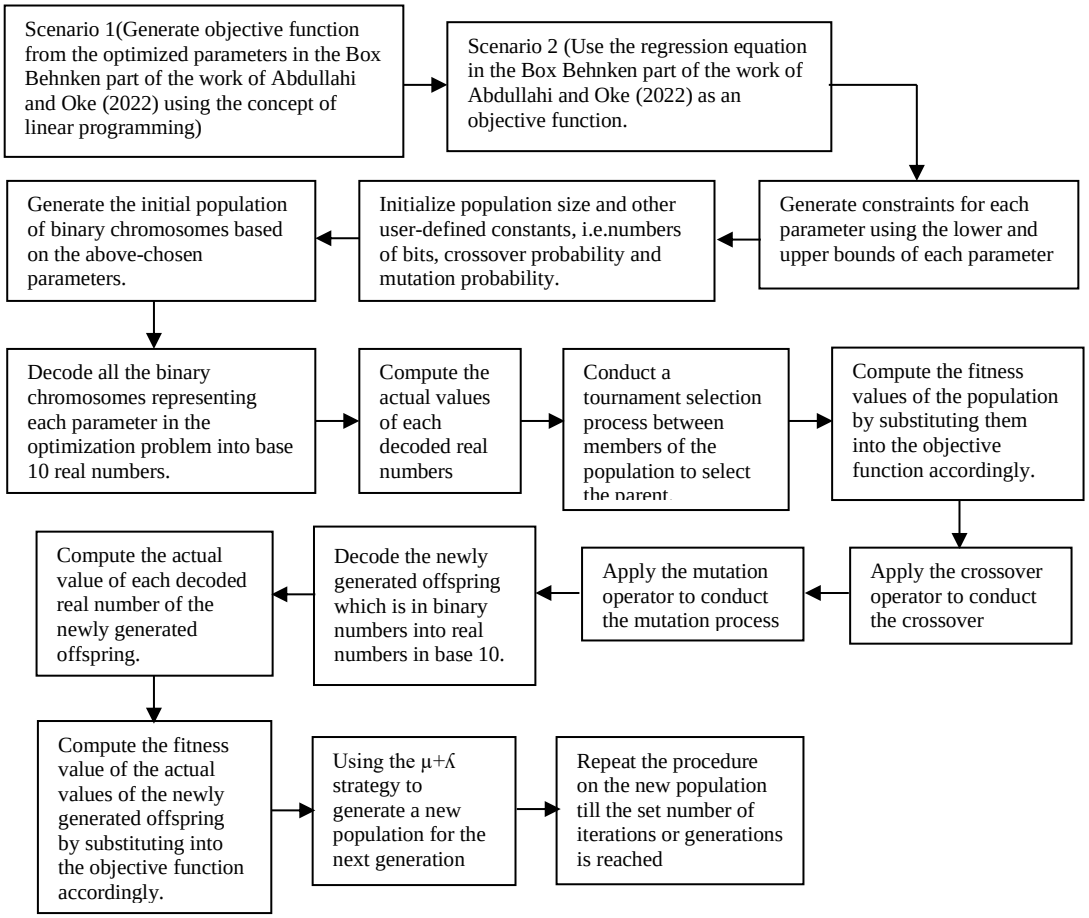

Fig. 1. Flowchart of the research method

4. RESULTS AND DISCUSSION

The Taguchi-Pareto-Box Behnken approach in a previous study is extended to include genetic algorithm optimization to further optimize the parameters in the boring Process of IS 2062 E2500 Plate on a CNC machine. This is done by creating an objective function from the optimized parameters of the Box Behnken design approach from Minitab software. This is followed by generating the constraints for the objective function from the bands of the parameters. With the objective function and its constraints, one can proceed to the genetic algorithm procedures, generation of the objective function and its constraints. The output obtained from the Box Behnken approach in the Minitab software is speed, feed rate, depth of cut and nose radius. This represents the optimized parameters at the end of the Box Behnken approach, from which the objective function is generated; they serve as the coefficient in the objective function while their various abbreviated names stand as the variable names of the function.

The objective function generated is stated in Equation (3):

$$F(x) = 1090.9091S + 0.06F + 1.250DC + 0.6061NR \quad (3)$$

where:
S is the speed, F is the feed rate, DC is the depth of cut and NR is the nose radius

The constraints serve as the bounds of the various parameter for example the speed parameter ranges between 800 to 1400, so the constraint of the speed parameter is: $800 \leq S \leq 1400$, the Sam applies to feed, depth of cut, nose radius, their constraints are given in Equation (4) to (6), respectively

$$0.06 \leq F \leq 0.12 \quad (4)$$
$$1 \leq DC \leq 1.5 \quad (5)$$
$$0 \leq NR \leq 1.2 \quad (6)$$

4.1 Genetic algorithm optimization procedures

The genetic algorithm is an optimization method which aims to mimic the natural selection of good traits or genes and discarding of bad traits or genes in nature or the human body.

1. The procedure starts with the initialization of the population size to be considered and other user-defined constants like the numbers of bits, probability of crossover or crossover probability, mutation probability, and iteration number:

n, number of bits = 5
Np, Population size = 4
Pc, Crossover probability = 0.8
Pm, Mutation probability = 0.3

2. For genetic algorithm illustration purposes, randomly select 17 bits of binary numbers, for the chosen population size, for each parameter, so each parameter would have four sets of 5bit binary numbers as follows:

S	F	DC	NR
10101	11001	11111	10110
01100	01101	00111	00100
11100	10001	11101	01011
01110	00001	10101	11110

3. Decode all the binary numbers in each parameter or population into base 10 i.e. real number so we have four different real numbers in each parameter as below.

$P_0 =$

21	25	31	22
12	13	7	4
28	17	29	11
14	1	2 1	30

4. The actual value of each decoded number is determined using Equation (2):

$$x = x_{\min} + \frac{x_{\max} - x_{\min}}{2^n - 1} \times (Dv) \qquad (2)$$

where x_{min} is the lower bound for the parameter in view, x_{max} represents the upper bound for the parameter studied, n is the number of bits while *Dv* is the decoded value.

The actual value were computed as thus,

PA =

1206.452	0.108387	1.5	0.851612903
1031.258	0.06	1.112903226	0.15483871
1341.935	0.06	1.467741935	0.425806452
1070.968	0.06	1.338709677	1.161290323

5. The fitness value is computed for each population as thus. For example

F(*x*) = 1090.9091 (1206.452) + 0.06 (0.108387) + 1.250 (1.5) + 0.6061 (0.851612903) = 1316131.863

This represents the fitness value for the first population.

6. Tournament is then conducted between two randomly selected populations, four times, with different mates each time using their various fitness values to contend. To populate the matting pool with 4 four winners and do away with the weakest population as thus.

The fitness value for each population is:

1316131.863 P_1
1126101.134 P_2
1463931.119 P_3
1168331.118 P_4

P_2 = 1126101.134
P_4 = 1168331.118

P_1 = 1316131.863
P_3 = 1463931.199

P_1 = 1316131.863
P_4 = 1168331.118

P_2 = 1126101.134
P_3 = 1463931.199

Mating pool

P_2 = 1126101.134
P_1 = 1316131.863
P_4 = 1168331.118
P_2 = 1126101.134

The first tournament is between P_2 and P_4 the population with the lowest. The fitness value is the winner, which means making it to the mating pool. Comparing P_2 and P_4, P_4 is the winner of the tournament, so it made it to the mating pool. This process is repeated for all pairs and the winners are introduced to the mating pool. Note that P_3 as the weakest in the population did not make it to the pool.

7. The binary numbers corresponding to the fitness values of the populations in the mating pool are then considered as the parent population, as thus:

Parent population

10101	11001	11111	10110
01100	01101	00111	00100
01110	00001	10101	11110
01100	01101	00111	00100

8. Applying one point crossover method to pairs of the population based on the crossover probability 0.8 to form the population of offspring. For example,

Crossover Point

10101	11001	11111	10110
01100	01101	00111	00100
01110	00001	10101	11110
01100	01101	00111	00100

Crossover Point

Beyond the crossover point, the bits are swapped as thus, to create the offspring

offspring population

10100	01101	00111	00100
01101	11001	11111	10110
01110	01101	00111	00100
01100	00001	10101	11110

9. Conducting mutation of the offspring based on the mutation probability 0.3 as thus:
Select randomly 20 different numbers between 0 and 1, each of which is attached to the 20 bits in a population.

	1.010001101			00111			00100		
bits	0.3	0.5	0.1	0.2	0.6	0.5	0.9	0.7	0.3	0.3
	0.1	0.0	0.9	0.0	0.6	0.2	0.1	0.7	0.9	0.9

Since the chosen mutation probability is 0.3, all random numbers less than 0.3 corresponding to each bit is changed or altered i.e. If it were 1, it is changed to 0 and If it were 0, is changed to 1, the process is called the mutation process.
This results in the offspring:
1001001101 11101 11100 represents the offspring after mutation.
The process is repeated for the rest of the offspring population and it result in the following:

Offspring after mutation

10010	01101	11101	11100
01101	11000	10110	10101
01111	01110	00110	00110
01101	00001	10001	11110

10. The offspring population is then decoded as thus:

18	13	29	28
13	24	22	21
15	14	6	6
13	1	17	30

$= 0_0$

11. Actual value of decoded value is computed and results in the following:

1148.387	0.0852	1.4677	1.0839
1051.613	0.1065	1.3548	0.8129
1090.321	0.0871	1.0968	0.2323
1051.613	0.0619	1.2742	1.1613

12. Computing the fitness value of the actual offspring's decoded value by substituting into the objective function as thus:

The first offspring population for example is computed as:
F(x) = 1090.9091 (1148.387) + 0.06 (0.0852) + 1.250 (1.4677) + 0.6061 (1.0839) = 1252788.325

This represents the fitness value of the first offspring population. The fitness values are then computed for the rest of the offspring population in the same way thus yielding

1252788.325
1147216.384
1189442.618
1147216.492

The minimum fitness is 1147216.384 and its corresponding optimal solution in the first iteration is:

(1051.613, 0.0852, 1.4677, 1.0839)

13. Using the μ +ʎ strategy, we combine the initial population and the offspring population as follows:

10101	11001	11111	10110		1316131.863
01100	01101	00111	00100	P_0	1126101.134
11100	10001	11101	01001		1463931.119
01110	00001	10101	11110		1168331.118
10010	01101	11101	11100		1252788.325
01101	11000	10110	10101	O	1147216.384
01111	01110	00110	00110		1189442.618
01101	00001	10001	11110		1.147216.492

14. Next is to pick the first four biggest fitness values of the combined population. These are
1126101.134
1147216.384
1147216.492
1168331.118
The bit population corresponding to these fitness values becomes the new population or generation i.e.

01100	01101	00111	00100
01101	11000	10110	10101
01101	00001	10001	11110
01110	00001	10101	11110

The above procedure is repeated on the new population till the set number of iterations is reached. The genetic algorithm procedure is then coded with a python programming language for accuracy and ease of computing large iterations numbers. Some user-defined variables used in the python coded genetic algorithm program are maximum iteration = 50,

Bits = 100, Population size = 200, Crossover rate = 0.8 and Mutation rate = 0.3

Table 2 shows the result of the genetic algorithm from the python code.

Table 2. Genetic algorithm data when the objective function is generated using optimized Box Behnken design parameters

Iteration	Optimal Solutions
1	[806.3112017890312, 0.06853017828708838, 1.1244688186350014, 0.6577763076899129]
2	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.6577763076899129]
3	[806.3112017890312, 0.06852974312774945, 1.1794134895833177, 0.6577763076899129]
4	[806.3112017890312, 0.07034109137302588, 1.1244689743550462, 0.6577763076899129]
5	[803.9786965106678, 0.09315552528565782, 1.0679278206718763, 0.3876489115813109]
6	[802.7837100344519, 0.09917560815126998, 1.340941670972866, 0.29745054051285336]
7	[806.3112017890312, 0.07034109137302588, 1.124468818636553, 0.8837651129839498]
8	[806.3112017890312, 0.06852974313010786, 1.179413489583316, 0.6577763076899129]
9	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.6577763078472952]
10	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.6577763076899129]
	The Best chromosome fitness value at end of the 10th iteration is 875765.917
11	[808.6661965106678, 0.09315552528565782, 1.0679278206718763, 0.3876828729306819]
12	[806.3112017890312, 0.07034109137302588, 1.124468818636593, 0.6577763076899129]
13	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.6577763076899129]
14	[806.3112017890312, 0.06852974312774944, 1.179095283480303, 0.8837597029022295]
15	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.88375970290201]
16	[806.3112017890312, 0.06852974312774944, 1.1794134884643719, 0.8837597029022295]
17	[806.3112017890312, 0.0685370673462385, 1.1794134895833177, 0.6577763076899129]
18	[806.3112017890312, 0.06852974312774944, 1.1794134895832609, 0.6577763076899129]
19	[806.3112017890312, 0.06852974312792506, 1.1244665901692552, 0.6577763076899129]
20	[806.3112017890312, 0.06852974312774944, 1.124468818636553, 0.6577763076899129]
	The Best chromosome fitness value at end of the 20th iteration is 875765.917
21	[806.3112017890312, 0.06852974312774944, 1.124468818636553, 0.6577763076899129]
22	[808.6661965106678, 0.09315552528565782, 1.0679278206718763, 0.3876828729306819]
23	[806.3112017890312, 0.0685297430986757l, 1.124468818629277, 0.8839481826899128]
24	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.8837597027448493]
25	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.6577763076899129]
26	[806.3112017890312, 0.06852974312774944, 1.179413489583204, 0.6577763076899278]
27	[806.3112017890312, 0.06852974312774944, 1.1794134828764364, 0.8837597029022295]
28	[806.3112017890312, 0.06853200934177589, 1.124468818636553, 0.8837597027625311]
29	[806.3112017890312, 0.07034109137302588, 1.124468818636553, 0.6577763076899129]
30	[806.5402442995863, 0.06852974312774944, 1.1794134895833177, 0.6577763076899129]
	The Best chromosome fitness value at end of the 30th iteration is 875765.917
31	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.6577763076899129]
32	[806.3112017890312, 0.06852974312774944, 1.124468818636553, 0.8839481826899128]
33	[806.3112017890312, 0.06852974312774944, 1.179413489628525, 0.6577763076899129]
34	[806.3112017890312, 0.06852974312774943, 1.124468818636593, 0.6577763076899129]
35	[806.3112017890312, 0.07034109137301398, 1.1794134895833177, 0.6577763076899129]
36	[806.3112017890312, 0.0703410911785822, 1.1794134895833177, 0.6577763076899129]
37	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.6577763076899129]
38	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.6577763076899129]
39	[806.3112017890312, 0.06852974312774945, 1.124468818636553, 0.6577763076899129]
40	[806.3112017890312, 0.06852974312774944, 1.1794133703740282, 0.6577763076899129]
	The Best chromosome fitness value at end of the 40th iteration is 875765.917
41	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.6577763076899129]
42	[806.3112017890312, 0.06852974312776136, 1.124468818636553, 0.6577763076899129]
43	[806.3112017890312, 0.07034109137302588, 1.1794134895833177, 0.6577763076899129]
44	[806.3112017890312, 0.06852974312774944, 1.1794134895833177, 0.6577763076899129]
45	[806.3113624636488, 0.07034109137302588, 1.124468818636553, 0.8837597029022295]
46	[806.3112017890312, 0.06846609137302588, 1.1794134895833177, 0.6577763076899129]
47	[806.3112017890312, 0.06852974312774944, 1.1794134893956945, 0.6577763076899129]
48	[802.7837100344519, 0.09917560815126998, 1.3409416709781485,0.06981382176285335]
49	[806.3112017890307, 0.07034109137415993, 1.1794134895833177, 0.6577763076899129]
50	[806.3112017890307, 0.07034109137302588, 1.124468818636553, 0.6577763076899129]
	The Best chromosome fitness value at end of the 50th iteration is 875765.917
Optimal solution	**[802.7837100344519, 0.09917560815126998, 1.3409416709781485, 0.06981382176285335].** This is interpreted as a speed of 802.78, feed of 0.09, depth of cut of 1.34 and nose radius of 0.07

Note: Iteration = 50, Bits = 100, Population size = 200, Crossover rate = 0.8 and Mutation rate = 0.3

In addressing the issue of the behavior of the results obtained in this article and comparing it with the literature, it is interesting to discuss how the genetic algorithm behaved in this case application. Points of interest include the fast convergence attribute and the high computational speed of the procedure. For the convergence rate, the researchers are interested in knowing the iteration at which the results stop changing. To the knowledge of the researchers, convergence may occur when the input parameter stops changing. This may result in the outputs not changing also. The convergence rate was quite fast such that it happens at the first iteration. This is because the population of 200 and the number of bit of 100 were used in the study. Furthermore, it is known that working with some evolutionary methods takes time to compute the optimal solutions. However, the experience obtained in running the data shows high computational speed on a computer of processor speed 2.3 Hz, 4 gigabytes RAM, system size of 64-bit operating system and the Windows Edition of Window 10 home while the hard disk is 500 gigabytes. From Table 2, the best fitness value is 875765.917 and the corresponding best solution is 802.7837 for speed, 0.0992 for feed rate, 1.3409 for depth of cut and 0.0698 for nose radius. Fig. 2 shows the behaviour of the programme after specific iterations.

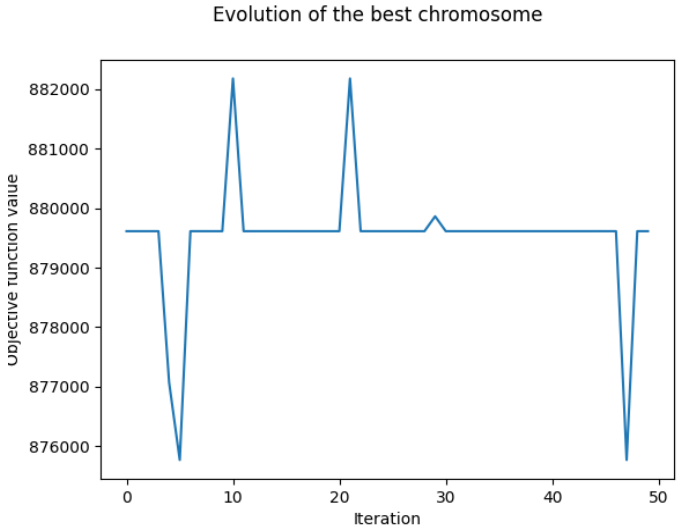

Fig. 2. Plot when the objective function is generated by optimized BBD parameters

To conclude, the optimized parameters from the box Behnken part of the present work which were 1090.9091 for speed, 0.06 for feed rate, 1.250 for depth of cut and 0.6061 for nose radius, were all further optimized by generating an objective function from the optimized values, which is then used for further analysis using genetic algorithm approach, which resulted in the results: Speed is 802.7837, Feed yields 0.0992, Depth of cut gives 1.3409 and Nose radius is obtained as 0.0698. The results show complete disagreement in that the optimal solution is both approaches are not the same but are in a similar range of numbers.

4.2 Regression equation (objective function) used for GA

The regression equation generated by the Box Behnken design approach part of the current study from the Minitab software is given by Equation (7):

SNR = -69.3+0.0233 x speed – 3 x feed + 2.5 x depth of cut +10.92 x Nose radius – 0.00010 x speed x speed + 35 x feed X feed -1.01 x depth of cut x depth of cut -7.72 x Nose radius x Nose radius -0.0000 x speed x feed +0.0000 x speed x Depth of Cut – 0.000092 x Nose radius -0.0 x feed x depth of cut -9.2 x Feed x Nose radius + 0.0 x Depth of cut x Nose radius (7)

This regression equation was used as the objective function in the python coded genetic algorithm optimization, where the bounds of each parameter are the constraints of the regression equation i.e.

$800 \leq S \leq 1400$ (8)

$0.06 \leq F \leq 0.12$ (9)

$0 \leq NR \leq 1.2$ (10)

$1 \leq DC \leq 1.5$ (11)

The following are outputs when the regression equation from the Box Behnken design approach part of the current study is used as the objective function in the python coded genetic algorithm optimization approach when population size is 200, several bits is 100, maximum iteration is 50, crossover rate is 0.8 and mutation rate is 0.3.

The number of iterations was plotted against the best or maximum objective function values at the end of each iteration (Fig. 3).

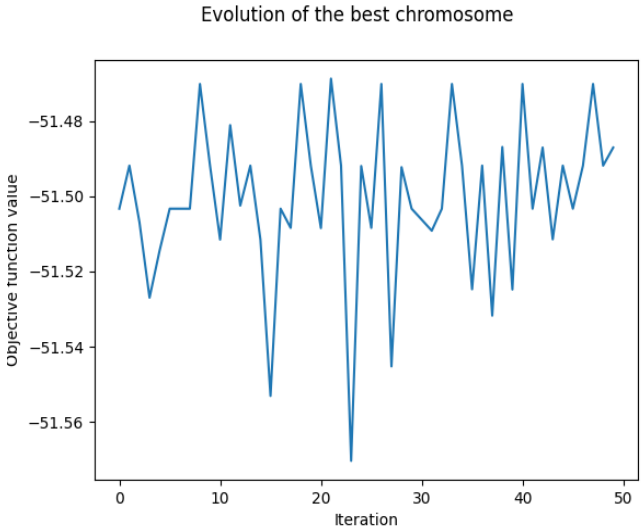

Fig. 3. Genetic algorithm when regression equation is used as the objective function

To conclude, the regression equation from the box Behnken design approach part of the current studies is used as the objective function using the bounds of each parameter as constraints for the regression equation in the python coded genetic algorithm for the present work which resulted in the following (Table 3):

The speed parameter was optimized to 1128.3867 the feed rate was optimized to 0.0610, the depth of cut was optimized to 1.4241, and lastly the nose radius was optimized to 0.6024.

Table 3. Genetic algorithm data when regression equation was used as the objective function

Iteration	Optimal solutions
1	[1145.9297576684646, 0.06218635396655763, 1.1887081067005236, 0.5120491779223741]
2	[1100.8221312134901, 0.07051009815884461, 1.2083608569421385, 0.5684125940929167]
3	[1181.1657581946051, 0.06613258204223611, 1.36946483436625, 0.5512075358320482]
4	[1112.405227958149, 0.07159251014246057, 1.4523538407402263, 0.5931165856034069]
5	[1181.1657581945904, 0.06613258204223611, 1.36946483436625, 0.541702682261272]
6	[1145.9297576684646, 0.06218635396655763, 1.1887081067005236, 0.5120491779223741]
7	[1145.9297576684646, 0.06218635396655763, 1.1887081067005236, 0.5120491779223741]
8	[1145.9297576684646, 0.06218635396655763, 1.1887081065841083, 0.5120491779223741]
9	[1122.3945527044639, 0.06102202171021221, 1.42405716024528, 0.6024019611953872]
10	[1100.8221312134901, 0.07051009815884461, 1.2083608569421385, 0.5684125940912343]
	The Best chromosome fitness value at end of the 10th iteration is -51.470
11	[1161.580206933474, 0.07999764433908133, 1.1688630882697963, 0.5812429982349516]
12	[1119.5721312134945, 0.07051009815884461, 1.2083608569421385, 0.5684125940929167]

Table 3 (Cont'd). Genetic algorithm data when regression equation was used as the objective function

Iteration	Optimal solutions
13	[1145.9297576684646, 0.06218635396655763, 1.197922379499772, 0.5120491779223741]
14	[1100.8221312134901, 0.07051009815884461, 1.2083608569421385, 0.5684125940928485]
15	[1161.580206933474, 0.07999647530502633, 1.1688630882697963, 0.5812422400569364]
16	[1145.9297576684646, 0.06218635396655763, 1.0104223795004328, 0.5120491779223741]
17	[1145.9297576684646, 0.06218635396655763, 1.1887081067005236, 0.5120491779223741]
18	[1068.4996651426716, 0.06661027882842234, 1.2279069718374507, 0.5775469429932851]
19	[1122.3946435053535, 0.06102202171021221, 1.4240571602395922, 0.6024019611953872]
20	[1100.8221312134901, 0.07051009815884461, 1.2083608569421385, 0.5684125940929167]
	The Best chromosome fitness value at end of the 20th iteration is -51.470
21	[1194.5721312134901, 0.07051009815884461, 1.2083608569421385, 0.5684125940929396]
22	[1128.3866791693076, 0.06102202171021221, 1.4240571602395922, 0.6024019611953872]
23	[1100.8221312134901, 0.07051009815884461, 1.2083608569421385, 0.5684125940929167]
24	[1112.4052286625283, 0.09637394452288814, 1.3981088456230388, 0.5931165856034069]
25	[1100.8221312134901, 0.0705407437519141, 1.2083608569421385, 0.5684125940929167]
26	[1068.4996651426716, 0.06661027882842234, 1.2279069718374507, 0.5775469429932851]
27	[1122.3946435053535, 0.06102202171021221, 1.4240571602395922, 0.6024019611953872]
28	[1161.580206933474, 0.07999647530502633, 1.4327276882018096, 0.5811697560474516]
29	[1100.2738143704933, 0.07051009815884461, 1.2083608569421385, 0.5684125940929167]
30	[1145.9297576684646, 0.06218635396655763, 1.1887081067003178, 0.5120491779223741]
	The Best chromosome fitness value at end of the 30th iteration is -51.469
31	[1159.0812897309836, 0.06218635396655763, 1.1887081067005236, 0.5120491779223741]
32	[1195.1956893704933, 0.07051009815884461, 1.2083608569421385, 0.5684125940929167]
33	[1145.9297576684646, 0.06218635396655763, 1.1887081067005236, 0.5120491779223741]
34	[1122.3946435053535, 0.06102202171021221, 1.4240571602395922, 0.6024019611953872]
35	[1100.8221312134901, 0.07051009815884449, 1.2083608569421385, 0.5684125940929167]
36	[1181.1657581946042, 0.0716014784278874, 1.1354051337193183, 0.5417026822612713]
37	[1100.8221312134901, 0.07051009815884461, 1.2083608569421385, 0.5684125940929167]
38	[1181.1657581946042, 0.0716014784278874, 1.36946483436625, 0.5417026822612713]
39	[1122.5273041693076, 0.06099669918381786, 1.0107548184199513, 0.6024019611953872]
40	[1181.1657581946042, 0.0716014784278874, 1.13508983436625, 0.5417043620039232]
	The Best chromosome fitness value at end of the 40th iteration is -51.469
41	[1122.3946435053454, 0.06102202171021221, 1.4240571602395922, 0.6024019611953872]
42	[1145.9297576684646, 0.06218635396655763, 1.1887081067005236, 0.5120491779223741]
43	[1122.3946435053535, 0.06102201923691845, 1.0107548184199513, 0.6024019611953872]
44	[1145.9297576684646, 0.06218635396655763, 1.135422379499772, 0.5120491779223741]
45	[1100.8221312134988, 0.07051009815884461, 1.2083608569421385, 0.5684125940929167]
46	[1145.9297576684646, 0.06218636243227984, 1.1887081067005236, 0.5120491779223741]
47	[1100.8221312134901, 0.07051009815884461, 1.2083608569421385, 0.5684125940929167]
48	[1122.3946435040493, 0.06102202171021221, 1.4240571602395922, 0.6024019611953872]
49	[1100.8221312134901, 0.07051009815884461, 1.2083608569421385, 0.5684125940929167]
50	[1122.3946435053535, 0.061022021695610044, 1.0107548184199513, 0.6024019611953872]
	The Best chromosome fitness value at end of the 50th iteration is -51.46878753096503
Optimal solution	**[1128.3866791693076, 0.06102202171021221, 1.4240571602395922, 0.6024019611953872]**

Note: Iteration = 50, Bits = 100, Population size = 200, Crossover rate = 0.8 and Mutation rate = 0.3

This result is in agreement in that the maximum objective function value i.e. -51.468 is in agreement with the maximized signal to noise ratio in the optimization plot of the Box Behnken design approach part of the current work, which was -51.952 but the optimal solution is both approaches are in the similar or same range of numbers. Except for the speed parameter which was 1090.9091 in the box Behnken design approach and which is now 1128.3867 in the current approach. Therefore, at the end of the 50th iteration, which is the maximum iteration chosen in the python coded programmed genetic algorithm, the best or minimum objective function value is -51.468 and the optimal solution of the genetic algorithm optimization is (1128.3867, 0.0610, 1.4241, 0.06024) i.e. Speed yields 1128.3867, Feed rate gives 0.610, Depth of cut is obtained as 1.4241 while Nose radius gives 0.0624.

4.3 Summarised results for the TP-BBD-GA method

The result of the proposed methodology presented in the current paper, suggest the optimal parameter to achieve high quality surface roughness when the objective function is generated using the optimized parameter from the Pareto-BBD part of the work of Abdullahi and Oke (2022) using the concept of

linear programming is 802.7937 rpm for speed, 0.0992 for feed rate, 1.3409 for depth of cut and 0.0698 for nose radius, while when the regression equation generated in the work of Abdullahi and Oke (2022) is used as the objective function, the optimal parameters for high quality surface roughness is; 1128.3867 rpm for speed, 0.0610 for feed rate, 1.4241 for depth of cut and 0.6024 for nose radius, it was observed that the result, when the objective function is generated using linear programming concept is used in the python coded genetic algorithm optimization is compared with the result obtained by Abdullahi and Oke (2022) in the Pareto-Box Behnken design part of their work both result are in partial agreement in that feed rate and depth of cut parameter values were in agreement but vary slight in value while the speed and nose radius were not in agreement as their values differ quite widely, and that the optimal signal to noise ratio when the objective function is generated by concept of linear programming, computed as -54.85dB. This was obtained by introducing the optimal parameter from the first scenario into the regression equation obtained by Abdullahi and Oke (2022). This is to facilitate a comparison of the results. The obtained value of -54.85dB is less than what was reported in the Pareto-BBD part of Abdullahi and Oke (2022). This shows that the optimal parameters in this regard would not give a better surface roughness quality as those from the work of Abdullahi and Oke (2022).

Meanwhile, the result, when the regression equation from the work of Abdullahi and Oke (2022) is used as the objective function in the python coded genetic algorithm optimization is compared to that obtained by Abdullahi and Oke (2022) in the Pareto-Box Behnken design part of their work, it was observed that all parameters were in agreement and that the optimal signal to noise ratio from the current study which is -51.468 is greater than that from the Pareto-Box Behnken design part of the work of Abdullahi and Oke (2022) which shows that the parameters when the regression equation is used as the objective function in the present study using genetic algorithm are better for obtaining high-quality surface roughness in the boring operation of EN250 steel material. The best result from the three scenarios is when the regression equation is used as the objective function in the python coded genetic algorithm optimization approach; which could be applied to parts that require very high-quality surface roughness in their design.

Besides, the concluding part of the data shows the optimum solution and the corresponding output for the regression equation. It was observed from it that the optimum output as understood from the analysis of the signal to noise ratios on the data while using the regression equation as the objective function in the genetic algorithm is -51.46dB. But by comparing it to that of the Box Behnken design part of Abdullahi and Oke (2022), -51.95dB was reported. This reveals that the regression equation-based objective function genetic algorithm method is superior to Abdullahi and Oke's (2022) reported results. Furthermore, as the linear programming concept is used to generate the objective function, a signal to noise ratio was not obtained. However, after the optimization process, to compare the result, with that of the previous one, the researcher can pick the optimum parameter and put it into the regression equation to reveal some understanding of the magnitude of the signal to noise ratio. It was discovered that the optimal signal to noise ratio after being run with the genetic algorithm module was -54.84dB. Besides, the authors argue that the results can be improved by querying the parameters. It should be noted that when ANOVA analysis is being done the following information is revealed. When the model is linear none of the parameters were significant to the model. However, when it is squared, the square model is significant to the objective. So, when this additional information was discovered to make the present study robust, the results were close to that of Abdullahi and Oke (2022), better than that previously obtained. But it does not exceed that of Abdullahi and Oke (2022). Thus instead of having about -54dB, the best result yield roughly -52dB. Furthermore, there is a need for more clarification on the signal to noise ratios, which arise from the regression equation. The regression equation generated by Abdullahi and Oke (2022) was based on the output of the signal to noise ratios. It is the same regression equation that is being used in the present study as the second scenario part.

5. CONCLUSIONS

In this article, a new method called the Taguchi-Box Behnken design-genetic algorithm method was established to optimize and select the parameters of a boring process involving the use of IS 2062 E250 steel plates. The method optimized the input parameters of speed, depth of cut, feed and nose radius while the output is the surface roughness of the IS 2062 E250 steel plates. After using the experimental data obtained from Patel and Desphande (2014) for the new method proposed, it was concluded as follows: The T-BBD-GA method proved its efficiency to optimize the boring process parameters. The present article optimizes the surface roughness, an output of the boring process using the CNC lathe machine through moderating the influential parameters in boring operation, such as the depth of cut, feed, speed and nose radius by applying the T-BBD-GA method for the IS 2062 E250 steel plates.

This article is an extension of a previously published work where the earlier authors had established the possibility of enhancing the surface roughness of the IS 2062 E250 steel plates subjected to the boring operation. In the work, the Taguchi method and Taguchi-Pareto method were independently integrated with the Box Behnken design. These two amalgamated methods proved feasible and a demonstrated analysis based on the experimental data contained in Patel and Deshpande (2014) was showcased in Abdullahi and Oke (2022). However, the current machining environment continues to be stiffer and business survival becomes increasingly difficult. But with the continuous improvement philosophy that currently engages the industry, further optimization is required from the already optimized results of the boring situation regarding the IS 2062 E250 steel plates. Therefore, it varies with the literature, the present study improved on the work of Abdullahi and Oke (2022) with the following contributions:

1. Improving initially optimized parameters by incorporating an evolutionary approach. Hence, the optimized parameters were first optimized by the Taguchi-Box Behnken design method and additionally, the genetic algorithm was added in the present article.
2. Introducing a new approach to the surface roughness in a boring operation. Besides, the developed, method has not been previously found in the literature. Therefore, the new mechanism of a multi-step optimization approach is rare to find in the literature and has been proved effective.

Future research may consider the replacement of the genetic algorithm with other evolutionary methods such as the firefly, teaching-learning based optimization and grey wolf optimization procedures. The results may be compared with those of the present study.

REFERENCES

Abdullahi Y.U. and Oke S.A. (2022). Optimizing the boring parameters on CNC machine using IS 2062 E250 steel plates: Taguchi-Pareto-Box Behnken design and Taguchi-ABC-Box Behnken design perspectives, *Engineering Access*, *8*(2), 219-241.

Ahmad, N., Tanaka, T., & Saito, Y. (2005). Optimization of cutting parameters for end milling operation by soap based genetic algorithm. *Power , 318(2)*, 5.

Atia M., Khalil J., Mokhtar M. (2017). A cost estimation model for machining operations: An ANN parametric approach, *Journal of Al-Azhar University Engineering Sector*, *12*(44), 878-885. https://doi.org/10.21608/auej.2017.19195

Čuboňová, N., Dodok, T., & Ságová, Z. (2019). Optimisation of the machining process using genetic algorithm. *Scientific Journal of Silesian University of Technology. Series Transport. 104*, 15-25. http://dx.doi.org/10.20858/sjsutst.2019.104.2

Dave, S., Vora, J. J., Thakkar, N., Singh, A., Srivastava, S., Gadhvi, B., Patel, V.V., Kumar, A. (2016). Optimization of EDM drilling parameters for aluminium 2024 alloy using response surface methodology and genetic algorithm. *Key Engineering Materials*, *706*, pp. 3–8. http://dx.doi.org/10.4028/www.scientific.net/KEM.706.3

Dennison M.S., Abdul Khadar S.D., Karuppusami G. (2012). Optimization of

machining parameters for face milling operation in a vertical CNC milling machine using genetic algorithm. *IRACST-Engineering Science and Technology: An International Journal*, *2*(4), pp. 544-548

Dhavamani, C., & Alwarsamy, T. (2012). Optimization of machining parameters for aluminium and silicon carbide composite using genetic algorithm. *Procedia Engineering*, *38*, 1994–2004. https://doi.org/10.1016/j.proeng.2012.06.241

Ganesan, H., Mohankumar, G., Ganesan, K., & Ramesh Kumar, K. (2011). Optimization of machining parameters in turning process using genetic algorithm and particle swarm optimization with experimental verification. *International Journal of Engineering Science and Technology* , *3*(2), 1091–1102.

Izelu, C.O., Essien, E.I., Okwu, M.O., Garba, D.K. & Agunobi-Ozoekwe, C.N. (2021). Lathe boring operation on ASTM A304 steel parameter optimization using response surface methodology, Australian Journal of Mechanical Engineering, *19*(5), 544-558. https://doi.org/10.1080/14484846.2019.1662534

Khundrakpam, N. S., Brar, G. S., & Deepak, D. (2018). Genetic algorithm approach for optimizing surface roughness of Near dry EDM. *IOP Conference Series: Materials Science and Engineering*, *376*(1), p. 012130. http://dx.doi.org/10.1088/1757-899X/376/1/012130

Kilickap, E., & Huseyinoglu, M. (2010). Selection of optimum drilling parameters on burr height using response surface methodology and genetic algorithm in drilling of AISI 304 stainless steel. *Materials and Manufacturing Processes*, *25*(10), 1068–1076. https://doi.org/10.1080/10426911003720854

Kilickap, E., Huseyinoglu, M., & Yardimeden, A. (2011). Optimization of drilling parameters on surface roughness in drilling of AISI 1045 using response surface methodology and genetic algorithm. *The International Journal of Advanced Manufacturing Technology* , *52*(1), 79–88. http://dx.doi.org/10.1007/s00170-010-2710-7

Kumar, S., Meenu, Satsangi, P.S. (2012). A genetic algorithmic approach for optimization of surface roughness prediction model in turning using UD-GFRP composite. *19*(6), 386-396.

Mahesh, G., Muthu, S., & Devadasan, S. R. (2015). Prediction of surface roughness of end milling operation using genetic algorithm. *The International Journal of Advanced Manufacturing Technology* , *77*(1-4), 369–381. https://doi.org/10.1007/s00170-014-6425-z

Marimuthu, P., Kumar, K., Raja, S., & Karthikeyan, D. S. (2015). Analyse and optimise machining parameters setting for CNC turning of inconel X-750 using genetic algorithm. *10*(55), 3978-3981.

Nugroho, W., Baba, N.B. and Saptari A. (2016). Optimization on surface roughness of boring process by varying damper position, *ARPN Journal of Engineering and Applied Sciences*, *11*(20), 11911-11918

Palanisamy, P., Rajendran, I., & Shanmugasundaram, S. (2007). Optimization of machining parameters using genetic algorithm and experimental validation for end-milling operations. *The International Journal of Advanced Manufacturing Technology* , *32*(7), 644–655. https://doi.org/10.1007/s00170-005-0384-3

Rao, V. D., Raju, K. M., Subbarayan, N. V., & Mahesh, P. (2018). Multi objective optimization of surface roughness and material removal rate in end milling using genetic algorithm. *AUT Journal of Mechanical Engineering*, *2*(1), 117-123, https://doi.org/10.22060/ajme.2018.12581.5373

Reddy, B.S.K., Nagaraju, S.K., Salmanm M.D. (2015), A study on optimisation of resources for multiple projects by using primavera, *Journal of Engineering Science and Technology*, *10(*2), 235–248

Reddy N.S.K. & Rao, P.V. (2005). A genetic algorithmic approach for optimization of surface roughness prediction model in dry milling. *Machine Science and*

Technology, *9*, 63–84. https://doi.org/10.1081/MST-200051263

Saffar, R. J., Razfar, M. R., Salimi, A. H., Khani, M. M. (2009). Optimization of machining parameters to minimize tool deflection in the end milling operation using Genetic Algorithm. *World Applied Sciences Journal* , *6*(1), 64–69.

Sangwan, K. S., & Kant, G. (2017). Optimization of machining parameters for improving energy efficiency using integrated response surface methodology and genetic algorithm approach. *Procedia CIRP* , *61*, 517–522, https://doi.org/10.1016/j.procir.2016.11.162

Sardinas, R. Q., Santana, M. R., & Brindis, E. A. (2006). Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. *Engineering Applications of Artificial Intelligence* , *19*(2), 127–133, https://doi.org/10.1016/j.engappai.2005.06.007

Tien, D.H., Nguyen, N.-T., Trung, D.D., Nguyen, V.C.C., Nguyen, Q., Luat, N.V., Huu, P.N. (2020). Optimization of cutting parameters and cutter helix angle for minimum surface roughness in flat-end milling of Al6061. *Optimization*, *62*(4), pp. 1321-1331.

Zain, A. M., Haron, H., & Sharif, S. (2010). Application of GA to optimize cutting conditions for minimizing surface roughness in end milling machining process. *Expert Systems with Applications*, *37*, 4650–4659. https://doi.org/10.1016/j.eswa.2009.12.043

Zeelan, B. N., Kumar, S. G., & Mosisa, E. (2013). Determining the Effect of Cutting Parameters on Surface Roughness Using Genetic Algorithm. *Science, Technology and Arts Research Journal*, *2*, 98–101. https://doi.org/10.4314/star.v2i4.17