Check for updates

An integrated evolutionary approach for modelling and optimization of wire electrical discharge machining

D Kondayya[1] and **A Gopala Krishna**[2*]
[1] Department of Mechanical Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, India
[2] Department of Mechanical Engineering, University College of Engineering, JNTUK, Kakinada, A. P., India

The manuscript was received on 9 January 2010 and was accepted after revision for publication on 15 June 2010.

DOI: 10.1243/09544054JEM1975

Abstract: This paper presents an application of an integrated evolutionary approach for modelling and optimization of a wire electrical discharge machining (WEDM) process. The proposed methodology consists of two parts. In the first part, a novel application of genetic programming (GP) is proposed. GP is an evolutionary modelling algorithm which uses principles similar to genetic algorithms to model highly non-linear and complex processes, resulting in accurate and reliable models. Two important aspects of machining performance of WEDM, namely metal removal rate and surface roughness, are modelled based on experimental data using GP in terms of four prominent input variables. The effect of machining parameters on the performance measures is also reported. In the second part, as the chosen machining performances are opposite in nature, the problem under consideration is formulated as a multi-objective optimization problem and solved using an efficient evolutionary optimization algorithm, non-dominated sorting genetic algorithm-II (NSGA-II). The outcome of Pareto optimal solutions is presented. The work presents a fully fledged evolutionary approach for optimization of the process.

Keywords: wire electrical discharge machining, modelling, optimization, NSGA-II

1 INTRODUCTION

Machining processes play a significant role in manufacturing industry where cost and quality are the main driving forces. Among the various manufacturing processes in existence today, wire electrical discharge machining (WEDM) is the most widely used [**1**] because of its capability to machine intricate shapes and profiles irrespective of hardness of the material. This process finds widespread applications in the mould-making tool and dies industries, as well as the aerospace, automobile, and electronics industries.

The principle of WEDM is shown in Fig. 1. In this process, a wire of small diameter is used as the electrode to cut a narrow kerf (width) in the workpiece. The workpiece is fed continuously and steadily past the wire in order to achieve the desired cutting path. Numerical control is used for precise control of the motions during cutting. As it cuts, the wire is continuously advanced between a supply reel and a take-up reel to present a fresh electrode of constant diameter to the work. This helps maintain a constant kerf during cutting. The dielectric fluid is continuously injected to flush away minute amounts of removed material.

One of the most investigated areas of machining is process modelling and optimization as this results in both reduced production cost and improved product quality. However, experimental optimization of any machining process is costly and time consuming owing to the complex, coupled, and non-linear nature of the input–output variables of machining processes [**2**]. Briefly, owing to the highly complicated interactions between process parameters, current analytical models and analyses cannot provide accurate process prediction for better quality control and higher throughput. Therefore, an efficient method is needed to determine the optimal machining parameters.

In the case of WEDM, the most significant performance indicators are metal removal rate (MRR) and surface finish. Metal removal rate is an indicator for

*Corresponding author: Department of Mechanical Engineering, University College of Engineering, JNTUK, Kakinada, A. P., India.
email: dr.a.gopalakrishna@gmail.com

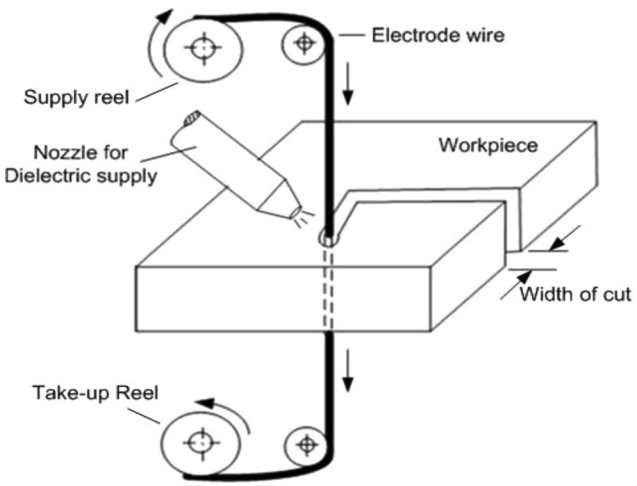


Fig. 1 Schematic diagram of WEDM

productivity while surface roughness (R_a) is a measure of work quality. In WEDM, because it is a complex stochastic process, it is very difficult to determine the optimal machining parameters for the best machining performance. Moreover, the performance measures, namely MRR and R_a, are conflicting in nature as it is desirable to have higher MRR with lower value of surface roughness.

The overall objective of this research is to apply a new process modelling and optimization methodology for the highly non-linear and complex manufacturing process of WEDM. With this aim, accurate prediction models to estimate MRR and R_a were developed from the experimental data using a potential evolutionary modelling algorithm called genetic programming (GP). Subsequently, the developed models were used for optimization of the process. As the chosen objectives, MRR and R_a, are conflicting in nature, the problem was formulated as a multi-objective optimization problem. Later, a popular evolutionary algorithm, non-dominated sorting genetic algorithm II (NSGA-II), was used to retrieve the multiple optimal sets of input variables.

2 LITERATURE SURVEY

Scott *et al.* [**3**] developed the empirical models to predict surface finish and material removal rate on machining D2 tool steel. They recognized that there was no single combination of levels of the different factors that could be optimal under all situations. To locate the optimal machining parameters, the non-dominated point approach was applied, using explicit enumeration of all possible combinations and the dynamic programming method. Tarng *et a1*. [**4**] used a feedforward neural network to determine the optimal cutting parameters for improving the cutting performance measures of machining speed and surface finish. Spedding and Wang [**5**] used response surface methodology and artificial neural networks (ANNs) for modelling of the WEDM process. They modelled the responses of cutting speed, surface roughness, and surface waviness for AISI 420. Huang *et al.* [**6**] investigated experimentally the effect of machining parameters on gap width, surface roughness, and the depth of white layer. Ramakrishnan and Karunamoorthy [**7**] developed a mathematical model with the response surface methodology in order to predict the performance characteristics of the WEDM process. Hewidy *et al.* [**8**] developed mathematical models correlating the various WEDM machining parameters with volumetric MRR, wear ratio, and surface roughness based on the response surface methodology. Sarkar *et al.* [**9**] used a feedforward back-propagation neural network to model the machining process. Manna and Bhattacharyya [**10**] studied seven different control parameters for optimization of the process using a Taguchi design methodology for the metal matrix composite of Al/SiC. They considered cutting speed, surface roughness, material removal rate, and spark gap as measures of the process performance. Han *et al.* [**11**] experimentally investigated the influence of machining parameters on surface roughness taking brass as the electrode and Cr12 alloy as the workpiece. Chiang [**12**] used the response surface methodology to investigate the influence of machining parameters on material removal rate, electrode wear ratio, and surface roughness of Al_2O_3+TiC mixed ceramic material.

3 CRITIQUE ON THE LITERATURE SURVEY

The literature survey reveals that particular efforts were devoted to determine the most accurate models for surface roughness and material removal rate. These empirical models were utilized as objective functions and were optimized to obtain the machining conditions for the required material removal rate and the surface roughness. Also the literature shows that the predominant modelling and optimization tools used to date have been Taguchi-based regression analysis, response surface methodology, and ANNs. However, the accuracy and possibility of determining the global optimum solution depend on the type of modelling technique used to express the objective function and constraints as functions of the decision variables [**13**]. Therefore, effective, efficient, and economic utilization of the WEDM process necessitates an accurate modelling and optimization procedure.

In most conventional deterministic approaches such as multiple regression and response surface methodology, a prediction model has to be determined in advance and a set of coefficients has to be found. The pre-specified size and shape of the model

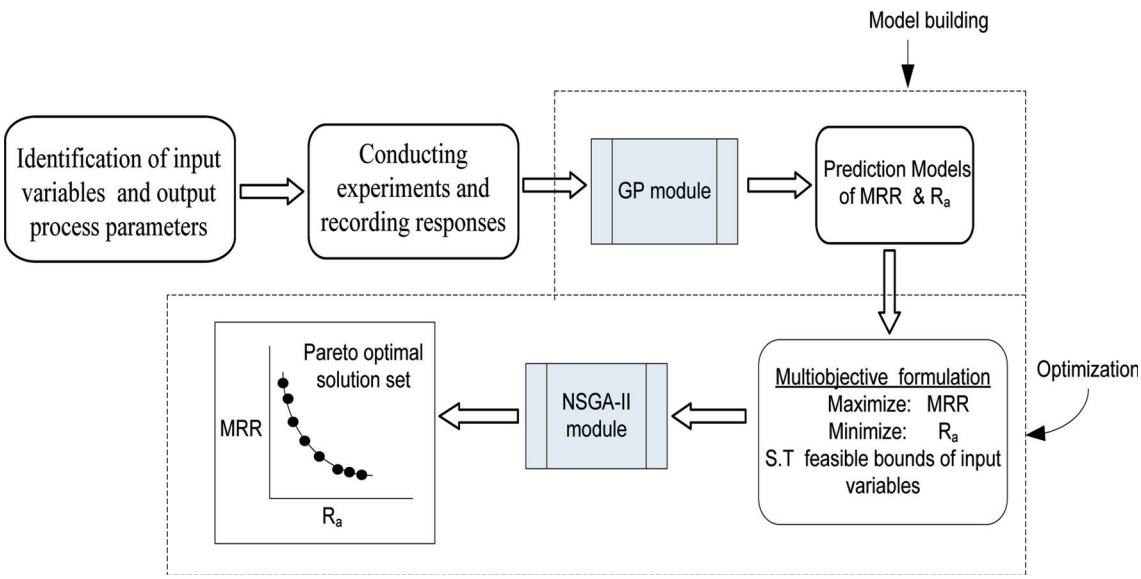


Fig. 2 Proposed methodology

mean that these are not adequate to capture a complex relation between the influencing variables and output parameters. Like the aforementioned approaches, although ANNs have also been used extensively in the literature for modelling, they have the drawback of not being able to quantify explicitly the relationships between inputs and outputs.

4 PROPOSED METHODOLOGY

In this paper, a novel approach is presented for modelling of material removal rate and surface roughness using GP. The distinctive aspect of GP as compared to traditional approaches is it does not make any presumption about the formulation to be made. Also the generated model helps directly to obtain an interpretation of the parameters affecting the process. More details of this methodology are discussed in section 5. The models developed by GP were subsequently used for optimization.

In the current study, the optimization problem of WEDM was explicitly formulated as a multi-objective optimization problem, as the determination of the optimal machining conditions involves a conflict between maximizing MRR and minimizing the surface roughness. It can be noted that the classical optimization methods (weighted sum methods, goal programming, min–max methods, etc.) are not efficient for handling multi-objective optimization problems because they do not find multiple solutions in a single run, and therefore it is necessary for them to be applied as many times as the number of desired Pareto-optimal solutions [**14**]. The above-mentioned difficulty of classical optimization methods is eliminated in evolutionary algorithms, as they can find the multiple solutions in a single run. As a result, a most commonly used evolutionary approach, the NSGA-II is proposed in this paper for multi-objective optimization of WEDM. Genetic algorithm (GA)-based multi-objective optimization methodologies have been widely used in the literature to find Pareto-optimal solutions [**15–18**]. In particular, NSGA-II has proven its effectiveness and efficiency in finding well-distributed and well-converged sets of near Pareto-optimal solutions [**19**]. The proposed methodology of integrating GP and NSGA-II is depicted in Fig. 2.

5 MODELLING USING GP

Evolutionary approaches attempt to find the best solution to a problem by mimicking the process of evolution in nature using Darwin's theory of 'survival of the fittest'. Individual potential solutions are selected based on their fitness from the initially generated random population. The potential solutions are then recombined to produce better solutions. The extreme popularity of these techniques is because of their success in searching highly complex non-linear spaces and their robustness in practical applications. Genetic programming is one such evolutionary method which can be used to generate models with interaction terms or higher order terms. Genetic programming is a relatively new approach when compared to other variations of evolutionary algorithms such as evolutionary strategies, GAs, and evolutionary programmes.

The main principles of GP and its related terminology were developed by Koza [**20**]. Since then this technique has found widespread applications in diversified fields such as industrial robotics [**21**], fault detection [**22**], prediction of shear strength of beams

[**23**], identification of fuzzy models [**24**], etc. However, it is noteworthy that the potentiality of this tool has not been much exploited in the field of manufacturing; very few applications in the field of manufacturing can be found in the literature [**25–27**]. Therefore, in the present work, the proposed method is used to model a machining process.

The main principles of GP are a logical extension of GAs since both are population-based methods and use evolutionary techniques to arrive at optimal solutions. This population-based approach avoids the possibility of the solution becoming trapped in a local optimum, which is a major problem with gradient-based methods.

In GP, a solution to a problem is represented as a computer program, which has a hierarchical composition (tree-like structure) of *terminals* and *functions* appropriate to a particular problem domain. In this algorithm, input variables are usually called *terminals* and the user specifies a number of *functions* that manipulate terminals. The primitive functions typically include: arithmetic operations (+, –, *, /), Boolean operations (AND, OR, NOT), logical operations (IF–THEN–ELSE), and non-linear functions (sin, cos, tan, exp, log). Compatibility between the functions and terminals must be ensured in order for them to pass information impeccably between each other [**28**]. Typical representation of an individual in GP is shown in Fig. 3, which corresponds to the expression of $x\left(15+y\right)/\left[y\left(x-z\right)\right]$. The set of *functions* in the representation are {+, –, *, /} whereas the set of *terminals* are $\{x, y, z\}$.

5.1 Generation of initial population

Genetic programming starts with generation of an initial population of the problem by random compositions of the functions and terminals. This population consists of a group of individuals, each comprising a single mathematical expression coded to conform to a tree structure. One of the most commonly used methods for ensuring this is the ramped half-and-half algorithm. This method was proposed by Koza [**20**] to produce a wide variety of trees of various sizes and shapes. Once the population is generated, each individual is evaluated by its fitness value, which is a measure of how well it solves the given problem. Only the fitter individuals are modified while the population size is kept constant.

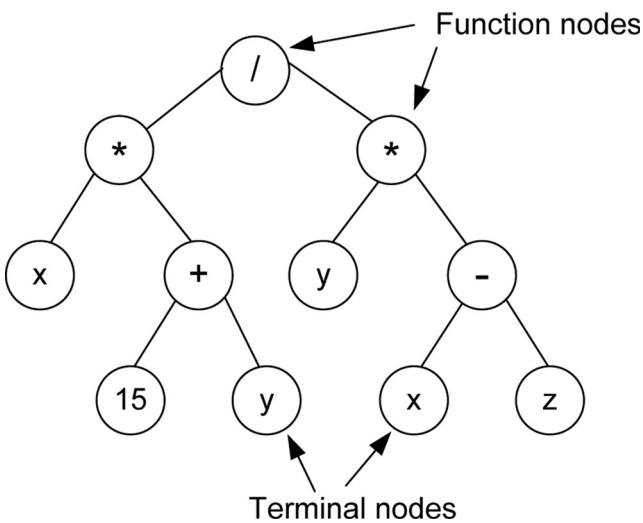


Fig. 3 Representation of GP tree

5.2 Genetic operators

The initial population is gradually improved through the genetic operators – *selection, crossover,* and *mutation*. Selection is the exploitation phase of search in which emphasis is given to the highly fit individuals. The selection operation ensures that good individuals remain in the population. There is a good number of selection operators in the literature for propagating the influence of the best fit individuals in the population, such as tournament selection, rank selection, and roulette wheel selection.

Exploration in GP is brought about by crossover and mutation operations. In crossover operation, two of the fittest individuals are randomly selected to be the parents and the selected parts of the parent schemata are swapped, hopefully to produce better ones. This process is illustrated in Fig. 4. It should be noted that highlighted parts of the parent trees in the figure exchange each other to produce two offspring. The expressions for the two parents and two offspring are also presented. To produce the good offspring so far obtained, not all are subjected to crossover. Mutation maintains genetic diversity and prevents the solution from becoming stuck on a local minimum. In mutation, one randomly selected node is replaced by another from the same set except itself. The process of mutation is illustrated in Fig. 5. Koza [**20**] recommended that crossover and reproduction should be the chief operators of GP while the exploitation of mutation should be very restricted.

5.3 Termination criterion

Implementation of the aforementioned three operators constitutes one generation and the procedure is repeated until a termination criterion is met. The termination criterion can be either a prescribed number of generations or sufficient quality of the solution. The number of runs required for a satisfactory solution depends on the complexity of the problem under consideration.

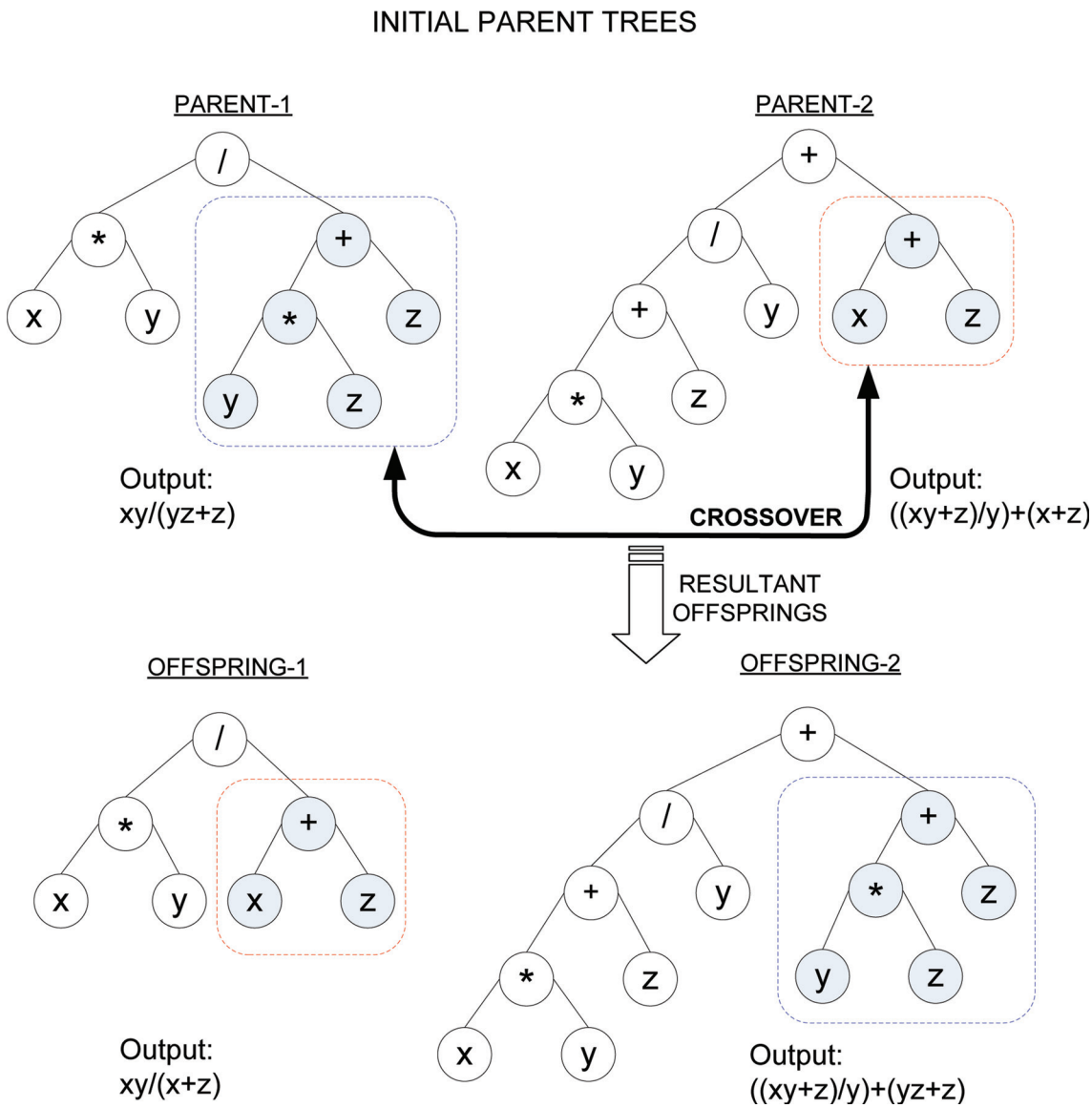


Fig. 4 Illustration of crossover operator

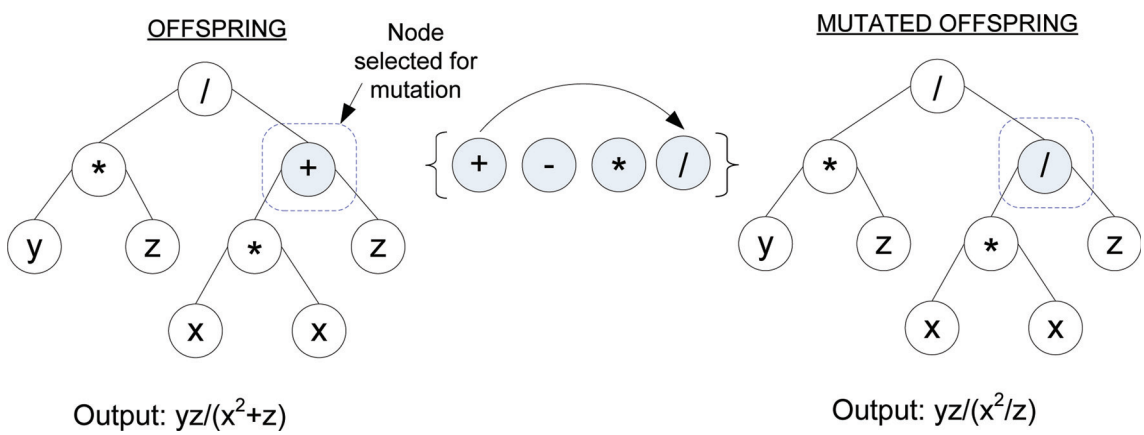


Fig. 5 Illustration of mutation operator

6 OPTIMIZATION USING NSGA-II

The application of evolutionary algorithms (EAs) for solving multi-objective optimization problems has been explored over a long period in the literature. Among these, the NSGA-II,, developed by Deb *et al.* [**19**], is a well-known and extensively used algorithm that has been applied successfully to several non-linear complex optimization problems. NSGA-II has proved to be more worthy over its predecessor non-dominated sorting genetic algorithm (NSGA), mainly in four aspects [**19**]:

(a) it has reduced the computational complexity;
(b) it uses an elitist approach, i.e. the best individuals are preserved in each generation;
(c) it uses a crowded comparison operator to maintain diversity of solutions;
(d) there is an absence of niched operators.

6.1 Non-dominated sorting

In this algorithm the population is initialized in a similar way to the single objective EAs. Once the population is initialized and the fitness function is evaluated, then the population is sorted based on non-domination into different fronts. The first front comprises the completely non-dominant individuals in the current population and the second front comprises only those dominated by the individuals in the first front and the third front, of only those dominated by the individuals in the first and second fronts, and so on. Each individual in each front is assigned rank value, based on the front in which it is placed. Individuals in the first front are given a fitness value of 1, those in the second a value of 2, and so on. The ranking scheme for a small number of solutions is illustrated in Fig. 6, where f_1 and f_2 are the two functions to be minimized.

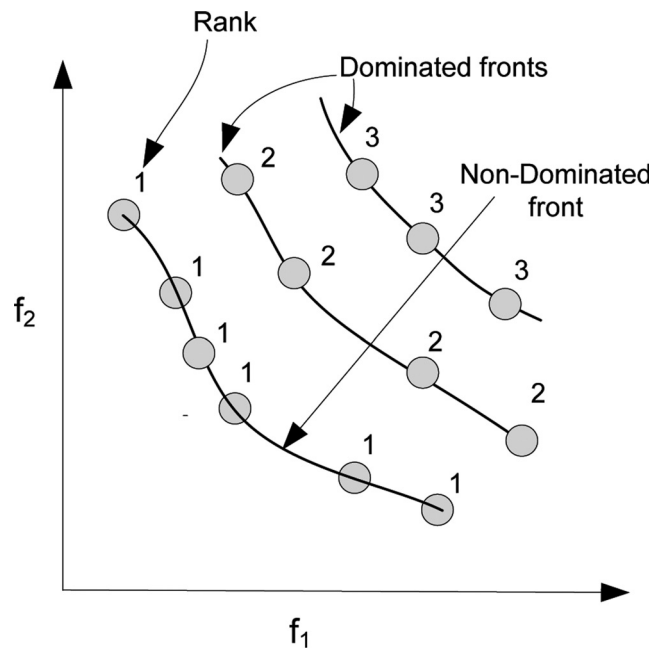


Fig. 6 Illustration of ranking procedure in NSGA-II

6.2 Crowding distance

In order to maintain diversity among solutions generated, a parameter called crowding distance [**29**] is calculated for each individual apart from fitness value. The crowding distance is a measure of how close an individual solution is to its neighbours. Large average crowding distance will result in better diversity in the population. Crowding distance gives an estimate of the density of solutions surrounding a solution i by taking the average distance of two solutions on either side of solution i along each of the objectives. As shown in Fig. 7, the crowding distance of the ith solution (d_i) represents the average side length of the cuboid.

The procedure for calculating the crowding distance in a front F involves the following steps.

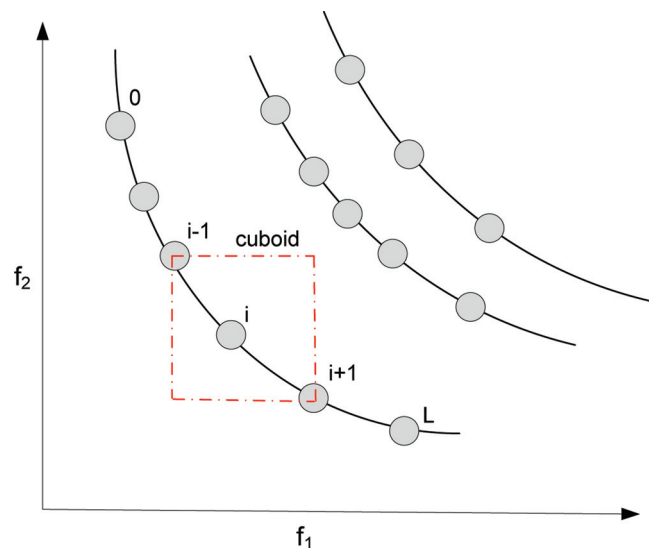


Fig. 7 Illustration of crowding distance calculation

1. The number of solutions in F is called L. For each solution i in the set, at first $d_i = 0$.
2. For each objective $m = 1, 2, \ldots, M$, the set is sorted in worst order of f_m. Hence the sorted indices vector $\boldsymbol{I}_m = \text{sort}(f_m)$ is calculated.
3. For $m = 1, 2, \ldots, M$ a large distance is assigned to the boundary solutions ($\boldsymbol{I}$ and L), or $d_{I1^m} = d_{I1^m} = \infty$ and for all other solutions $j = 2$ to $(L-1)$, the following equation is used to compute their corresponding values

$$d_{\boldsymbol{I}_j^m} = d_{\boldsymbol{I}_j^m} + \frac{f_m^{\mathbf{I}_{j+1}^m} - f_m^{\mathbf{I}_{j-1}^m}}{f_m^{\max} - f_m^{\min}} \tag{1}$$

where index $\boldsymbol{I}_j$ indicates the solution index for the jth member in the sorted list. For any objective, $\boldsymbol{I}_1$ and $\boldsymbol{I}_L$ denote the lowest and highest objective function values, respectively.

6.3 Selection

Parents for crossover are selected from the population using binary tournament selection as explained below.

Among two solutions i and j available for selection, the solution i wins the tournament over j if the following conditions are true:

(a) if solution i has better rank over j, i.e. $r_i < r_j$;
(b) if both solutions have the same rank, i.e. $r_i = r_j$, but i has larger crowding distance than j, i.e. $d_i > d_j$.

6.4 Procedural steps

The steps involved in NSGA-II based on the main framework of the algorithm [**19**] shown in Fig. 8 can be stated in the following steps.

1. Select control parameters: population size, maximum number of generation, crossover probability, and mutation probability (generation, $g = 0$)
2. Generate randomly a uniformly distributed parent population, P_g of size N based on the feasible ranges and constraints, if any.
3. Evaluate the fitness of individuals based on objective function and sort the population according to the non-domination.
4. Assign each solution a rank equal to its non-domination level as explained in section 6.1.
5. Calculate the crowding distance of each rank and sort the individuals that have the same rank in ascending order by the crowding distance as detailed in section 6.2.
6. Apply the binary tournament selection as discussed in section 6.3.
7. Use the simulated binary crossover operator and polynomial mutation [**30**] to create an offspring population Q_g of size N.
8. Combine the offspring and parent population $R(g) = (P_g \cup Q_g)$ to form extended population of size $2N$.
9. Sort the extended population based on non-domination to create different fronts.
10. Fill new population, P_{g+1} of size N with the individuals from the sorting fronts starting from the best.
11. Invoke the crowding-distance method to ensure diversity if a front can only partially fill the next generation. The crowding-distance method maintains diversity in the population and prevents convergence in one direction.
12. Update the number of generations, $g = g + 1$
13. Repeat the steps 3 to 12 until a stopping criterion is met.

7 EXPERIMENTAL DETAILS

The experiments were conducted on a five-axis computer numerically controlled (CNC) wire electrical discharge machine, model number CT 520A (Joemars Machinery and Electric Industrial Co. Ltd, Taiwan). WEDM involves several control variables such as pulse-on time, pulse-off time, pulse peak current, wire feed rate, servo reference voltage, wire tension, and dielectric flowrate. The dimensions of the workpiece and its material also influence the performance of the process. However, based on the literature survey and

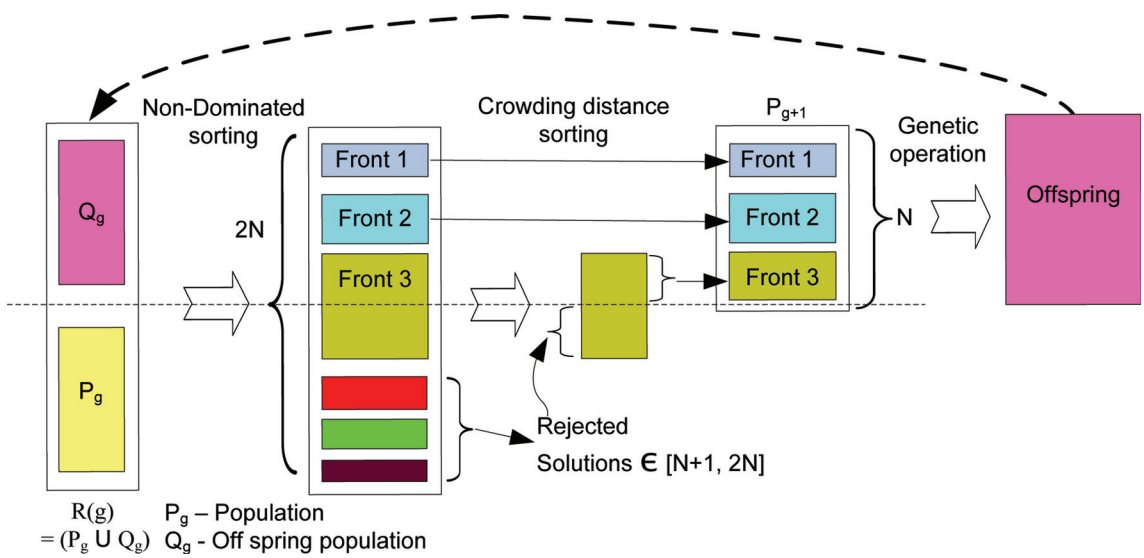


Fig. 8 Main framework of NSGA-II algorithm

Table 1 Machining conditions

(a)	Wire: uncoated brass wire of diameter 0.25 mm
(b)	Work piece material: AISI D3
(c)	Chemical composition: C – 2.00–2.35% ; Mn – 0.60% ; Si – 0.60% ; Cr –12.5% ; Ni – 0.30% ; W – 1.0% ; V – 1.0% ; Cu – 0.25% ; P – 0.03% ; S – 0.03%
(d)	Workpiece dimensions: 250 mm $L \times$ 12.5 mm $B \times$ 12.5 mm H
(e)	Length of cut: 12.5 mm
(f)	Angle of cut: vertical
(g)	Location of workpiece: at the centre of the table
(h)	Servo reference voltage: 35 V
(i)	Dielectric temperature: 25°C
(j)	Dielectric fluid: distilled water

Table 2 Feasible ranges of input variables

			Feasible range	
Input variable	Units	Notation	Lower limit	Upper limit
Pulse-on time	μs	x_1	5	13
Pulse-off time	μs	x_2	20	40
Wire tension	g	x_3	1	9
Wire feed rate	mm/min	x_4	3	7

the trial experiments, the variables, namely, pulse-on time (x_1), pulse-off time (x_2), wire tension (x_3), and wire feed (x_4) were considered as the decision (control) variables; and the MRR and the surface roughness R_a were considered as the output responses. MRR was calculated as the ratio of volume of material removed from workpiece to the machining time and R_a was measured perpendicular to the cutting direction using a Mitutoyo surface roughness tester at a 0.8 mm cut-off value and an average of six measurements were taken at six different locations to record the response value for R_a.

The machining conditions at which the experiments were conducted are listed in Table 1. The feasible ranges of process parameters are listed in Table 2. A good number of experiments were conducted within the feasible domains of individual variables so as to have sufficient data for effective modelling. The experimental data taken were grouped into two sets: the training dataset and the validation dataset. The training dataset was used to predict the expression that best suits the problem and the validation data were used to validate the developed model and to ensure the generalization capability of the predicted model for unseen cases. The training and validation datasets are shown in Tables 3 and 4 respectively.

8 IMPLEMENTATION OF GP

The unique aspect of GP compared to traditional regression-based methods is it does not assume any *a priori* functional form of the solution. Only the constituent elements of the desired function need to be defined to start the run. To decide the elements of functional sets, initially some trial runs were conducted with different combinations. It was found that the probability of successful solution was the greatest when only the basic arithmetic functions were used. Hence the arithmetic elements of addition (+), subtraction (−), multiplication (*), and division (\) were included in the functional set. The terminal set consists of all input variables of the WEDM process. However, a numerical constant has also to be added as a design variable because numerical constants are important in most models, as it will generally be a combination of input values and numeric constants that produces the required empirical model. One way to enable GP to evolve numeric constants is to include a numerical terminal in some reasonable range that represents a randomly generated real number. Well diversified model trees may result by these numerical constants. Therefore, the random floating-point numbers from the range (−20, 20), were added to the set of the terminals.

Preliminary experiments were performed to determine the best parameter settings for the GP algorithm. These preliminary test runs in the GP system were executed for the output parameters independently. Based on these experiments, the final parameters used to generate the models are given in Table 5. The parameter, depth of a tree, in Table 5 represents the length of the largest non-backtracking path from the root node to the end point. Generally, the evolutionary process of GP has a tendency to evolve uncontrollably large trees, resulting in much complicated model trees, which eventually leads to complex expressions. Hence to evolve parsimonious model trees it is necessary to limit the depth of the tree. In the present work, this is set to 6 to evolve simple expressions that are easy to interpret.

Evolutionary algorithms are generally robust to variations of control parameters [**31**] and some guidelines are provided in reference [**32**] for choosing the control parameters of standard GP. Tests with populations of different sizes of 250, 350, and 500 were performed. In all cases, the best results were achieved with the large populations. However, it was noted

Table 3 Training dataset

S. no.	x_1 (μs)	x_2 (μs)	x_3 (g)	x_4 (mm/min)	MRR (g/min)	R_a (μm)
1	5	25	1	3	0.181	1.70
2	5	25	1	7	0.168	1.61
3	5	25	5	3	0.186	1.75
4	5	25	5	5	0.234	1.73
5	5	25	9	5	0.182	1.67
6	13	25	1	3	0.657	3.45
7	13	25	1	5	0.633	3.13
8	5	25	9	7	0.157	1.67
9	5	25	5	7	0.189	1.61
10	5	30	1	5	0.154	1.58
11	5	30	1	7	0.129	1.48
12	5	30	5	3	0.162	1.72
13	5	30	5	7	0.147	1.48
14	5	30	9	3	0.154	1.52
15	5	30	9	5	0.143	1.49
16	5	30	1	3	0.170	1.58
17	5	35	1	5	0.105	1.47
18	13	35	9	5	0.543	3.07
19	5	35	1	7	0.110	1.35
20	5	35	5	3	0.122	1.39
21	5	35	5	7	0.157	1.45
22	5	35	9	3	0.151	1.49
23	5	35	9	5	0.120	1.47
24	5	35	9	7	0.137	1.55
25	5	40	1	5	0.158	1.35
26	5	40	1	7	0.140	1.22
27	5	40	5	3	0.119	1.36
28	5	40	9	5	0.112	1.24
29	5	40	1	3	0.129	1.47
30	13	40	9	5	0.480	2.94
31	13	40	1	3	0.450	3.19
32	9	25	1	5	0.375	2.43
33	9	25	1	7	0.350	2.31
34	9	25	5	3	0.417	2.55
35	9	25	5	5	0.393	2.53
36	9	25	5	7	0.368	2.40
37	9	25	9	3	0.434	2.34
38	9	25	9	5	0.410	2.42
39	9	25	9	7	0.391	2.50
40	9	30	1	5	0.312	2.30
41	9	30	1	7	0.287	2.18
42	9	30	5	3	0.354	2.22
43	9	30	5	7	0.305	2.28
44	9	30	9	3	0.372	2.22
45	9	30	9	5	0.347	2.29
46	9	30	1	3	0.337	2.42
47	9	35	1	5	0.249	2.17
48	9	35	1	7	0.225	2.10
49	9	35	5	3	0.291	2.19
50	9	35	5	7	0.242	2.15
51	9	35	9	3	0.309	2.29
52	9	35	9	5	0.285	2.17
53	9	35	9	7	0.271	2.06
54	9	40	1	5	0.219	2.05
55	9	40	1	7	0.192	1.92
56	9	40	5	3	0.229	2.16
57	9	40	5	7	0.197	2.02
58	9	40	9	3	0.246	2.16
59	9	40	9	5	0.222	2.04
60	9	40	1	3	0.261	2.08
61	13	25	1	7	0.608	3.21
62	13	25	5	3	0.675	3.35
63	13	25	5	7	0.626	3.30
64	13	25	9	3	0.693	3.44
65	13	25	9	5	0.668	3.32
66	13	25	9	7	0.644	3.36
67	13	30	1	5	0.570	3.27
68	13	30	1	7	0.546	2.98
69	13	30	5	3	0.612	3.22

continues

Table 3 Training dataset (continued)

S. no.	x_1 (μs)	x_2 (μs)	x_3 (g)	x_4 (mm/min)	MRR (g/min)	R_a (μm)
70	13	30	5	7	0.563	3.18
71	13	30	9	3	0.63	3.32
72	13	30	9	5	0.605	3.19
73	13	30	1	3	0.595	3.43
74	13	35	1	5	0.507	2.97
75	13	35	1	7	0.483	2.95
76	13	35	5	3	0.549	3.09
77	13	35	5	7	0.501	2.85
78	13	35	9	3	0.567	3.19
79	13	35	9	7	0.469	3.12
80	13	40	1	5	0.445	2.95
81	13	40	1	7	0.420	2.92
82	13	40	5	3	0.487	3.06
83	13	40	5	7	0.438	2.92
84	13	40	9	3	0.505	3.06
85	5	40	5	7	0.108	1.32
86	5	40	9	3	0.160	1.16

Table 4 Validation dataset

S. no	x_1	x_2	x_3	x_4	MRR (g/min)	R_a (μm)
1	5	25	1	5	0.198	1.68
2	5	30	5	5	0.161	1.53
3	5	30	9	7	0.154	1.51
4	5	35	1	3	0.156	1.60
5	5	35	5	5	0.146	1.57
6	5	40	5	5	0.128	1.44
7	5	40	9	7	0.120	1.42
8	9	25	1	3	0.439	2.55
9	9	30	5	5	0.33	2.40
10	9	30	9	7	0.323	2.37
11	9	35	1	3	0.309	2.30
12	9	35	5	5	0.290	2.12
13	9	40	5	5	0.254	2.01
14	9	40	9	7	0.224	1.94
15	9	40	9	5	0.222	1.94
16	13	25	5	5	0.651	3.45
17	13	30	5	5	0.588	3.31
18	13	30	9	7	0.581	3.23
19	13	35	1	3	0.513	3.31
20	13	35	5	5	0.510	3.21
21	13	40	5	5	0.430	3.04
22	13	40	9	7	0.410	3.00

Table 5 GP control parameters

Terminal set	$\{x_1, x_2, x_3, x_4\}$
Function set	$\{+, -, \times, \div\}$
Population size	500
Number of generations (maximum)	50
Number of independent runs	30
Crossover probability (%)	85
Mutation probability (%)	5
Reproduction probability (%)	10
Selection method	Tournament
Fitness measure	R^2
Maximum depth of tree	6

during the runs that there was an increase in computation times with increasing the population sizes. Thus, a reasonable size of 500 was considered. GP is probabilistic by nature, and therefore it is possible that it will land at differing approximate solutions every time. In order to exploit this intrinsic variation, multiple runs are performed.

After the investigation of various alternative models, the following expressions for MRR and R_a were found to have the largest fitness value.

$$\text{MRR} = \frac{1}{x_2}\left(x_1 + \frac{x_3}{x_2}\right) + \frac{x_1\,(x_1 - x_4)\,(1 + 0.0377 x_4 x_2^2)}{x_4 x_2^3} \tag{2}$$

$$R_a = \frac{x_4\left[x_1^2 + 2\,(x_1 - x_2) - x_3 + 144\right]}{72x_4 - x_1 + x_1 x_4} \tag{3}$$

While developing the above models, the correlation coefficient R^2 was taken as the fitness measure. GP was continued until a value as close as possible to

the maximum value of R^2 was obtained. R^2 was used as the fitness measure in order to check whether the fitted models actually describe the experimental data. Figures 9 and 10 show the convergence graphs of the models predicted by the proposed algorithm. The fitness measures for MRR and R_a have gradually improved with the number of generations and finally converged to 0.9793 and 0.9898 respectively. This shows that the models can explain the variation in the MRR and R_a up to the extents of 97.93 per cent and 98.98 per cent, respectively. On the basis of the high values of R^2 it can be said that the models are adequate in representing the process.

The normal probability plots of the residuals for the output responses are also shown in Figs 11 and 12. A check on these plots reveals that the residuals are located on a straight line, which means that the errors are distributed normally and the obtained models are fairly well fitted with the observed values.

A comparison of the predicted models and the experimental values for the validation datasets of MRR and R_a are shown in Figs 13 and 14 respectively. Very high values of R^2 for MRR and R_a of validation sets are obtained and found to be 0.9898 and 0.998, respectively; these indicate that the developed models satisfactorily represent the outputs.

8.1 Interpretation of the developed models

The developed equations reveal some unique aspects. For example, equation (2) obtained for MRR is short as well as accurate. Also, the explicit relation allows simple *ad hoc* interpretation of the problem at hand; for example, looking closely at the model, the x_1 variable (pulse-on time) is visible only in the numerator part as both a linear and a quadratic term. This indicates that pulse-on time has a pronounced effect on MRR and is directly related to it. On the other hand, the x_2 variable (pulse-off time) appears in the denominator more often. This gives a hint that the pulse-off time has an inverse effect on MRR. Similar conclusions can also be drawn from other machining parameters. The detailed main effects and interaction effects for both the outputs are discussed in the following sections. It should be noted that if a particular parameter does not influence the output, during the course of evolution, it gets eliminated.

8.2 Effect of process parameters on material removal rate

8.2.1 Direct effects

As seen in Fig. 15 (a), MRR increases with the increase in pulse-on time. Increase in pulse-on time increases

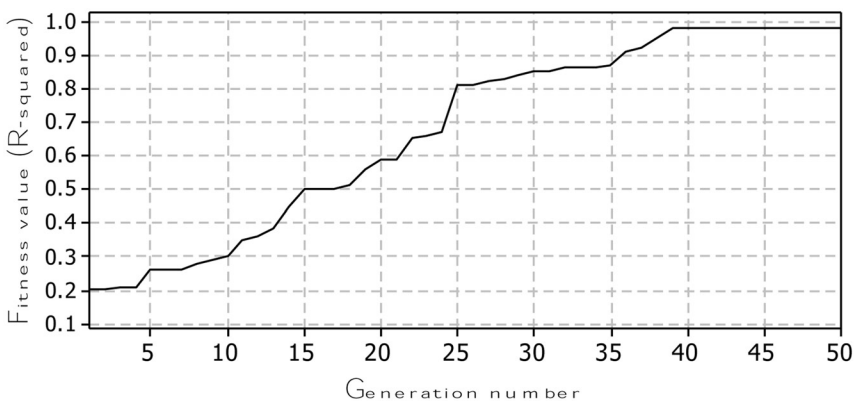


Fig. 9 Convergence plot of MRR model

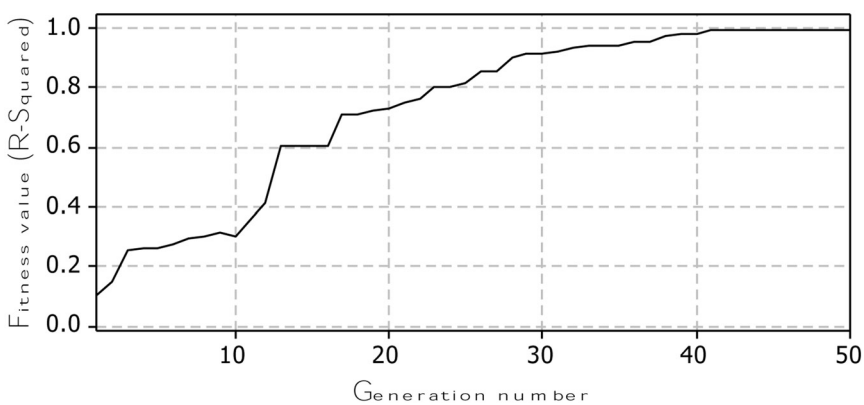


Fig. 10 Convergence plot of R_a model

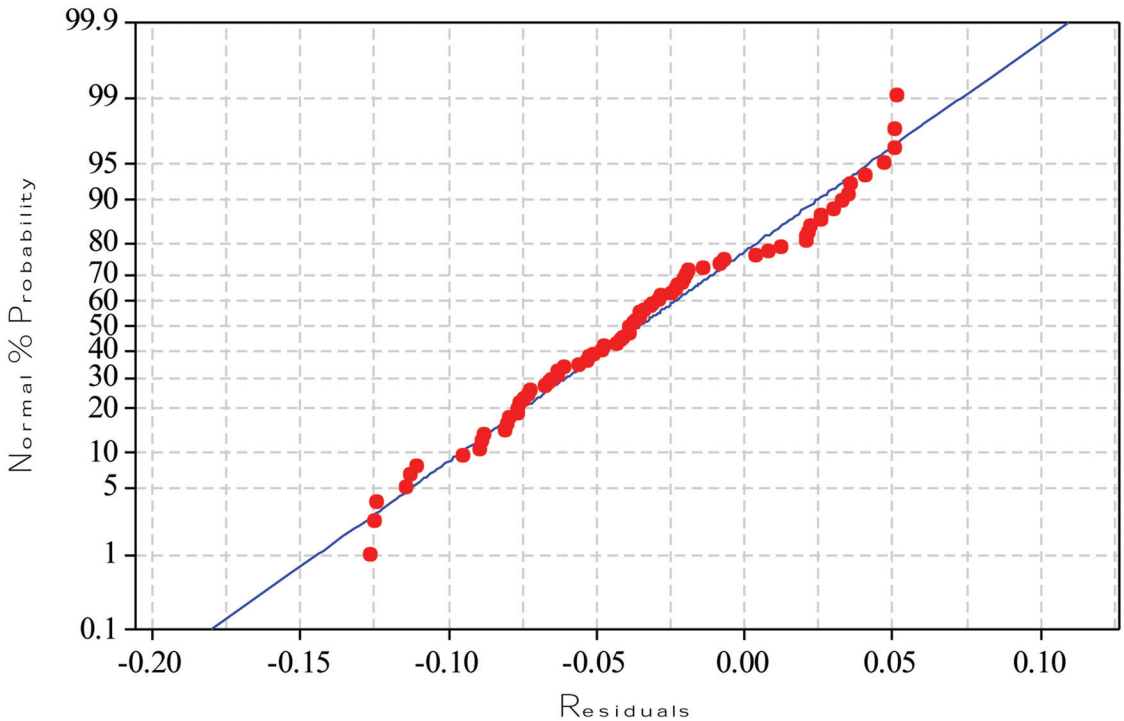


Fig. 11 Normal probability plot of residuals for MRR

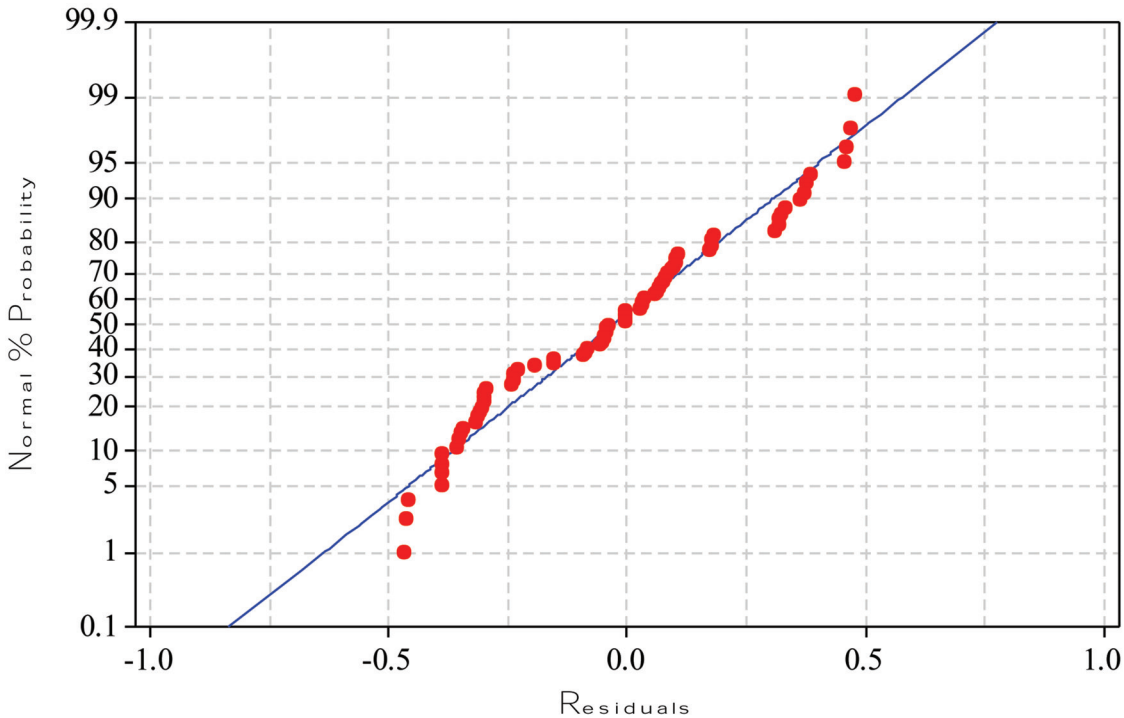


Fig. 12 Normal probability plot of residuals for R_a

the level of discharge energy, which subsequently causes a more powerful explosion and hence leads to more metal removal from the workpiece. Higher values of pulse-on time lead to a greater amount of metal removal.

On the contrary, Fig. 15(b) shows that MRR decreases with the increase in pulse-off time. This happens as the discharge current decreases and a smaller amount of material is removed. It is desirable to use lower values of pulse-off time to obtain higher

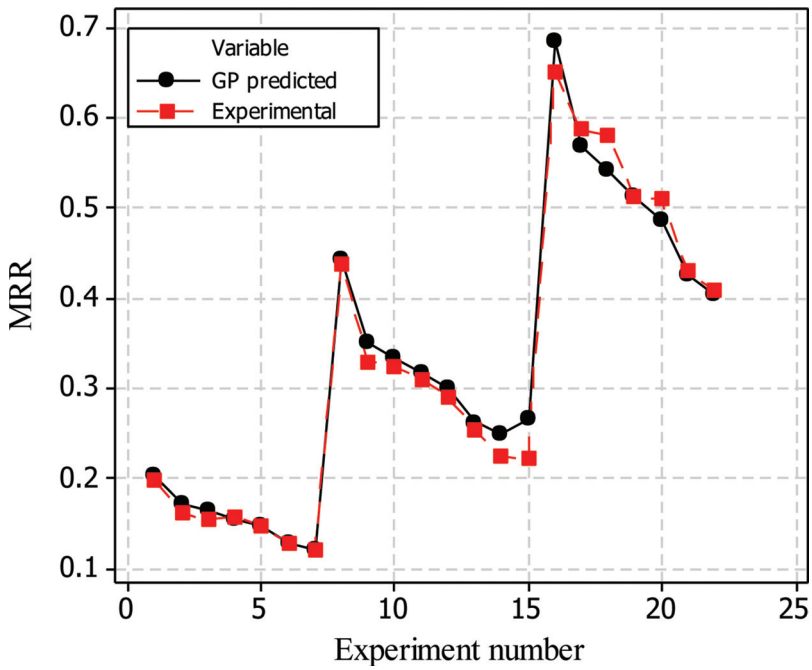


Fig. 13 Comparison of predicted values and experimental values of validation dataset for MRR

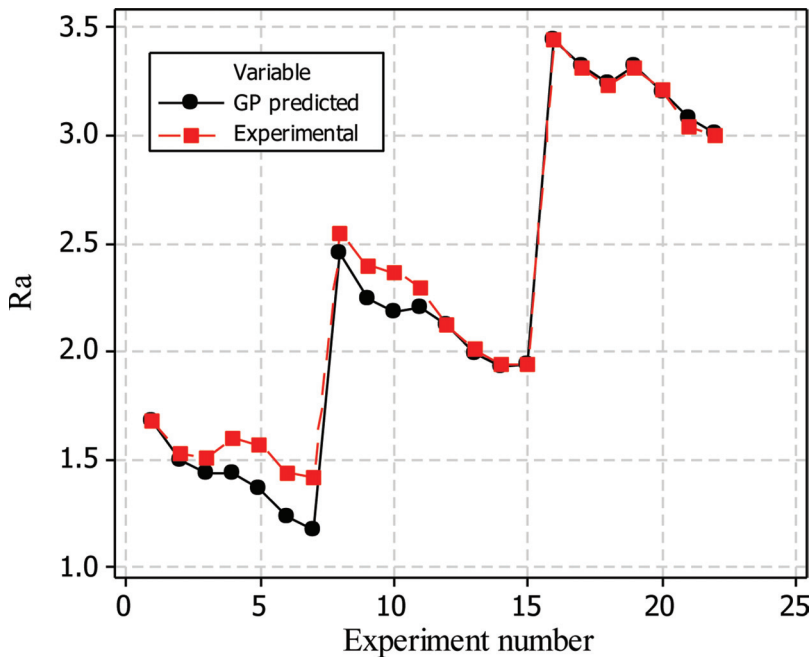


Fig. 14 Comparison of predicted values and experimental values of validation dataset for R_a

rates of metal removal. MRR is almost invariable with the change in wire tension, as seen in Fig. 15(c). A similar pattern is observed with change in wire feed, as depicted in Fig. 15(d); MRR is slightly decreased with increase in wire feed.

8.2.2 Interaction effects

Figure 16 shows the interaction plots for MRR. It can be observed from Fig. 16(a) that high values of pulse-on time together with low values of pulse-off time produce high MRR. This happens owing to the high levels of discharge energy, which causes a powerful explosion and results in deep craters on the surface of the workpiece. Deep craters indicate high rates of metal removal. The interaction plot of pulse-on time and wire tension on MRR is shown in Fig. 16(b). It can be observed from the figure that high MRR is achieved at both the high values of pulse-on time and wire tension. However, as can be observed, the

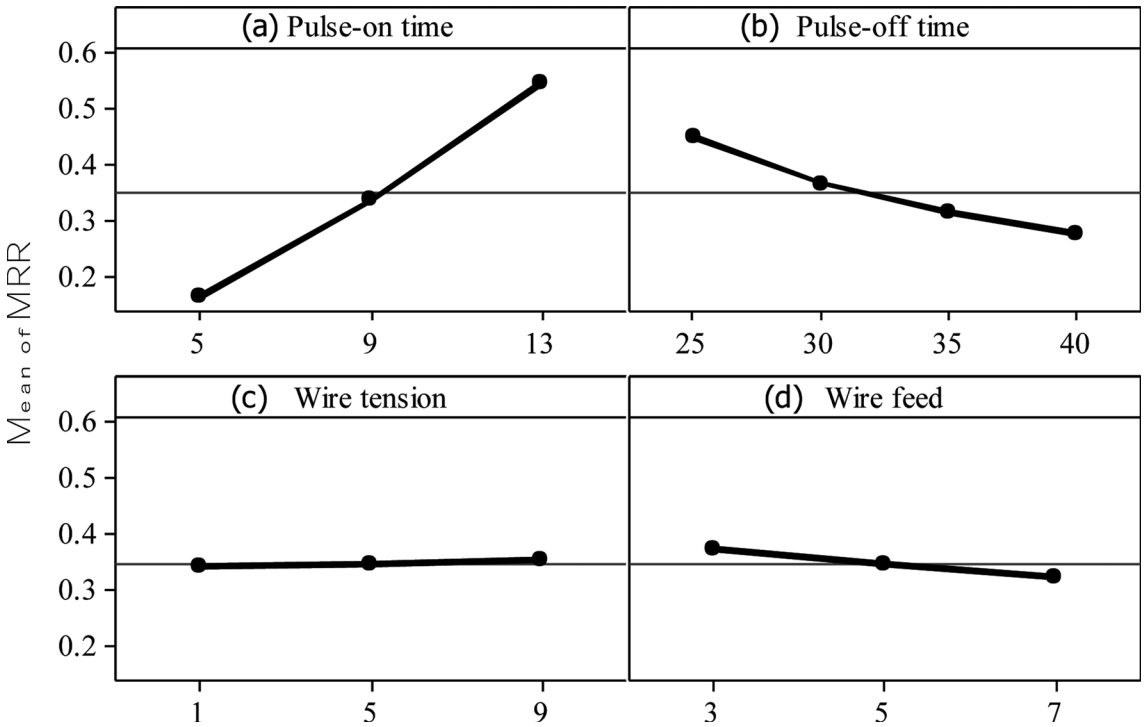


Fig. 15 Individual effects of input variables on MRR

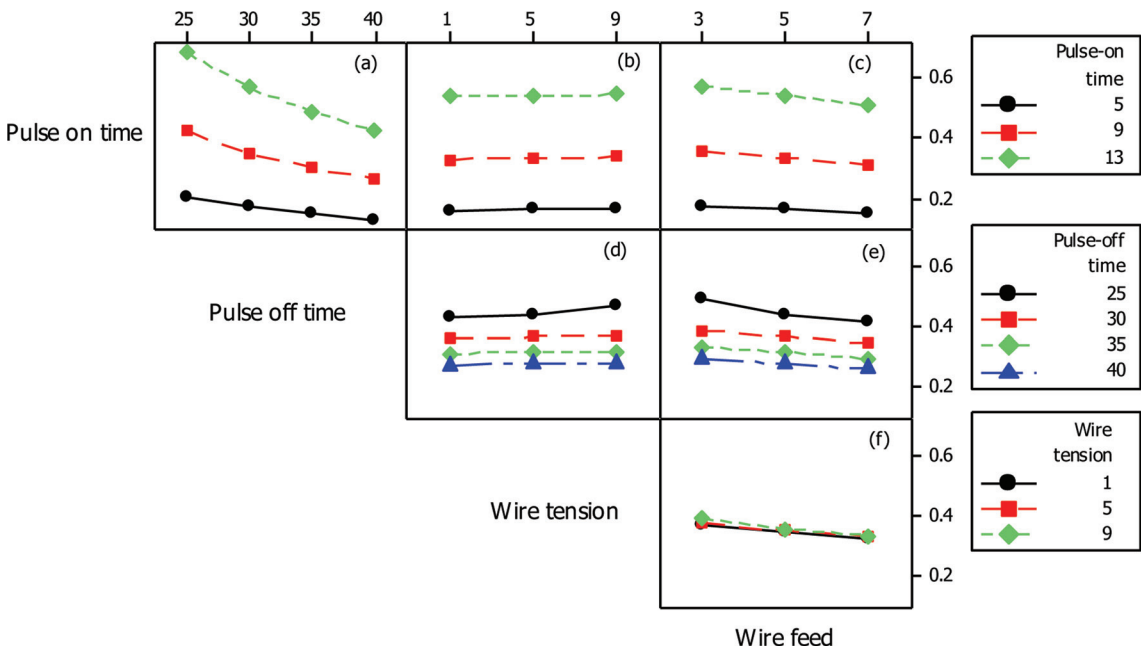


Fig. 16 Interaction effects of input variables on MRR

effect of wire tension is not predominant. The interaction plot of pulse-on time and wire feed is shown in Fig. 16(c). It can be observed that high values of pulse-on time together with low values of wire feed produce high MRR. Figure 16(d) shows the interaction plot of pulse-off time and wire tension on MRR. It is necessary to maintain low values of pulse-off time and high values of wire tension to obtain high MRR. On the contrary, MRR decreases with both the increase in pulse-off time and wire feed as shown in Fig. 16(e). The interaction plot of wire tension and wire feed, depicted in Fig. 16(f), shows that the effect of wire tension when compared to wire feed on MRR is almost insignificant. A greater value of MRR is achieved when lower values of wire feed are used.

8.3 Effect of process parameters on surface roughness

8.3.1 Direct effects

From Fig. 17(a), it is observed that R_a increases with the increase in pulse-on time. This indicates that the discharge energy becomes more intense with a higher value of pulse-on time. The higher discharge energy causes a more powerful explosion and deeper craters are formed on the machined surface. This results in a rougher surface. Hence to obtain good surface finish the pulse-on time should be as low as possible. On the other hand, low pulse-on time leads to longer machining time. The individual effect of pulse-off time on surface roughness can be observed from Fig. 17(b). The roughness decreases with the increase in pulse-off time, as discharge energy decreases and subsequently deep craters are not formed. Hence, this leads to good surface quality. However, high pulse-off time leads to low MRR. The effect of wire tension on surface roughness, as seen in Fig. 17(c), is that hardly any improvement in surface quality is achieved with any higher value of wire tension. A similar pattern is observed from Fig. 17(d) for wire feed on R_a.

8.3.2 Interaction effects

Figure 18 shows the interaction plots for surface roughness. As observed from Fig. 18(a), R_a decreases with low values of pulse-on time and high values of pulse-off time. The interaction plot of pulse-on time and wire feed, Fig. 18(b), indicates that R_a decreases with low values of pulse-on time and the effect of wire tension is observed to be not particularly significant. Figure 18(c) displays the interaction plot of pulse-on time and wire feed rate. To obtain good surface quality, low values of pulse-on time and high values of wire feed are to be used. From Fig. 18(d), it is noted that higher values of pulse-off time together with the lower values of wire tension produce better surface quality. On the other hand, both higher values of pulse-off time and wire feed rate produce better surface quality, as observed from Fig. 18(e). Figure 18(f), the interactive plot of wire tension and wire feed, shows that R_a is increased with low values of wire feed rate but the effect of wire tension can be observed to be insignificant.

Finally, it is evident from sections 8.2 and 8.3 that, among the chosen four control factors, only pulse-on time and pulse-off time have a profound effect on both the responses, whereas the effects of wire tension and wire feed are found to be insignificant.

9 FORMULATION OF MULTI-OBJECTIVE OPTIMIZATION PROBLEM

The two objective functions considered in this study are:

(a) maximization of material removal rate;
(b) minimization of surface roughness;

which are given by equations (2) and (3) respectively. The two objective functions are optimized subject to the feasible bounds of input variables. The optimization problem is defined as follows.

Maximize

$$\mathrm{MRR} = \frac{1}{x_2}\left(x_1+\frac{x_3}{x_2}\right)+\frac{x_1\,(x_1-x_4)\,\left(1+0.0377x_4x_2^2\right)}{x_4x_2^3}$$

Minimize

$$R_a=\frac{x_4\left[x_1^2+2\,(x_1-x_2)-x_3+144\right]}{72x_4-x_1+x_1x_4}$$

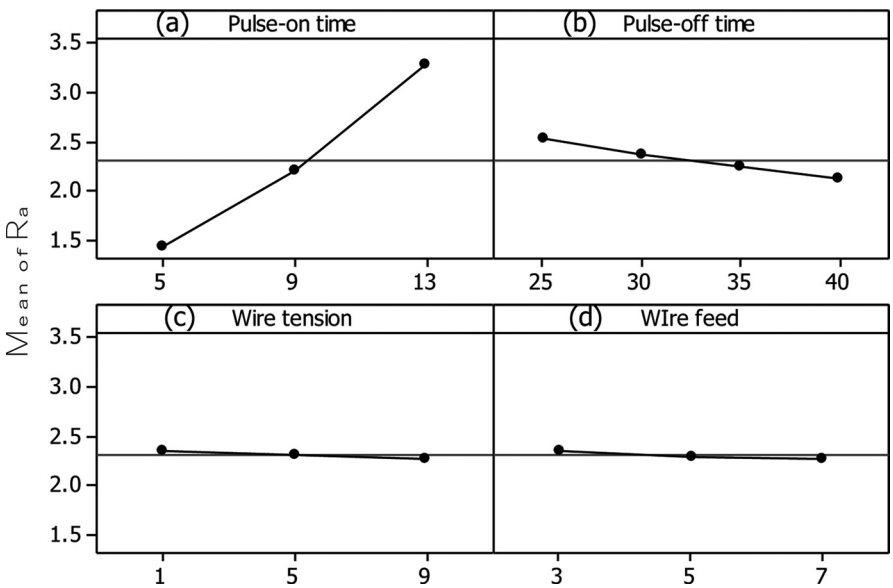


Fig. 17 Individual effects of input variables on R_a

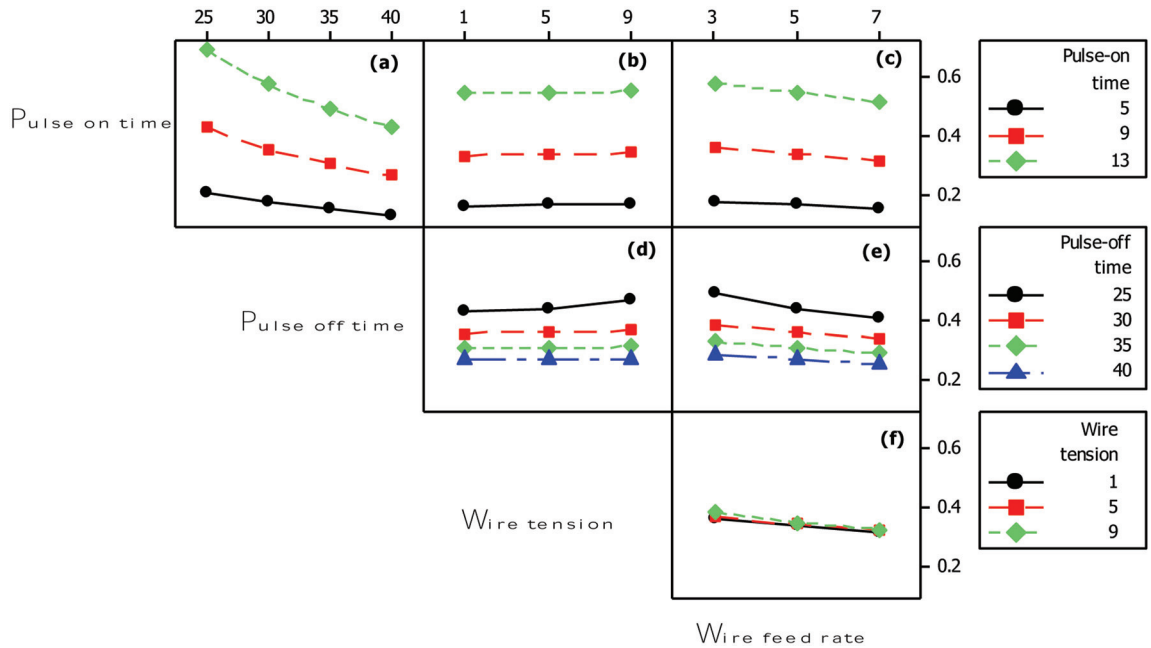


Fig. 18 Interaction effects of input variables on R_a

Subject to

$$5 \leqslant x_1 \leqslant 13$$
$$20 \leqslant x_2 \leqslant 30$$
$$1 \leqslant x_3 \leqslant 9$$
$$3 \leqslant x_4 \leqslant 7$$

Table 6 Control parameters

Population size (N)	50
Number of generations (n_{gen})	100
Crossover probability (p_c)	0.9
Mutation probability (p_m)	0.25
Distribution index for crossover operator (n_c)	20
Distribution index for mutation operator (n_m)	20

10 RESULTS AND DISCUSSION

The source code of the proposed optimization algorithm was run on a Pentium IV processor using Microsoft VC++ programming language on a Windows XP platform. The control parameters required for implementation of the algorithm are listed in Table 6. The algorithm found the Pareto optimal front of the conflicting objective functions with good diversity of the solutions, as shown in Fig. 19. The optimal input variables and their corresponding objective function values are presented in Table 7. By analysing the Pareto front, some decisions could be taken, depending upon specific requirements of the process. For instance, at point 1, a component can be machined with the best surface quality but with minimum MRR. On the other extreme of the front, i.e. at point 2, the highest material removal rate can be achieved but it will be of the worst surface quality. All the other points are in-between cases. As can be observed from the graph, no solution in the front is better than any other as they are non-dominated solutions. The choice of a solution has to be made purely based on production requirements. For example, if a manufacturing engineer chooses to machine a component with a surface quality of 1.27 μm, the set of input variables may be selected from the eighth row of Table 7. Accordingly, the MRR of 0.151 g/min would be achieved. To illustrate further the advantage of the proposed methodology, two more examples are explained as follows. From the experimental results of Table 3, the parameters listed in the 13th row lead to the R_a value of 1.48 μm and the MRR of 0.147 g/min. After optimization, it can be noted that the MRR is increased to 0.197 g/min for the same surface finish (Sl No. 18, Table 7), with a 34 per cent increase in MRR. In another instance, from Table 3, 38th row, the set of input variables leads to the MRR of 0.41 g/min and the R_a value of 2.42 μm. After optimization, the R_a value is reduced to 2.06 μm (Sl No. 36, Table 7) with almost the same value of MRR.

The scanning electron microscopy (SEM) photographs of the samples that correspond to the best values of MRR and R_a are shown in Figs 20(a) and (b). Those values of MRR and R_a correspond to the extreme positions (point-1 and point-2) of the Pareto optimal set shown in Fig. 19. As can be observed from the photographs, the variation of R_a is conspicuous with respect to the different optimal sets of input variables.

Table 7 Optimal solutions through NSGA-II

S. no.	x_1 (μs)	x_2 (μs)	x_3 (g)	x_4 (mm/min)	MRR (g/min)	R_a (μm)
1	10.438	20.137	8.96	3.062	0.688	2.84
2	12.96	20.006	8.989	3.257	0.912	3.57
3	5.306	22.658	8.976	3.015	0.272	1.70
4	6.548	20.041	8.941	3.005	0.394	1.98
5	9.744	20.149	8.955	3.11	0.629	2.66
6	5	38.283	8.997	3.746	0.143	1.23
7	6.039	36.748	8.99	3.03	0.19	1.45
8	5.068	37.497	8.969	3.173	0.151	1.27
9	5	25.735	8.968	3.024	0.223	1.57
10	9.744	20.148	8.97	3.095	0.63	2.66
11	6.2	38.713	8.985	3.721	0.181	1.42
12	11.821	20	8.993	3.004	0.814	3.23
13	6.262	20.128	8.994	3.753	0.363	1.91
14	7.822	20.137	8.96	3.062	0.482	2.22
15	5.532	36.986	8.99	3.026	0.17	1.36
16	9.887	20.006	9	3.22	0.643	2.7
17	10.765	20.093	8.996	3.272	0.712	2.92
18	6.711	39.716	8.918	3.179	0.197	1.48
19	9.114	20.041	8.869	4.059	0.565	2.49
20	12.992	20.004	8.929	3.151	0.918	3.58
21	8.06	20.727	8.869	3.379	0.479	2.25
22	5.172	25.815	8.921	3.385	0.227	1.59
23	5.01	26.54	8.918	3.198	0.214	1.55
24	6.077	39.036	8.949	3.198	0.179	1.39
25	5.172	21.054	8.934	4.164	0.275	1.71
26	10.805	20.145	9	3.01	0.72	2.94
27	8.152	20.037	8.758	3.175	0.507	2.29
28	11.433	20	8.919	3	0.78	3.12
29	5.494	39.301	8.959	3.2	0.158	1.29
30	13	20.005	9	3.039	0.922	3.59
31	10.248	20.005	9	3.002	0.678	2.8
32	6.487	39.729	8.967	3.032	0.19	1.45
33	6.27	39.811	8.98	3.386	0.18	1.40
34	5.001	39.658	8.982	4.341	0.135	1.20
35	9.109	20	8.928	3.003	0.585	2.51
36	6.959	20	8.984	3.011	0.413	2.06
37	5.072	20.016	8.997	4.104	0.285	1.73
38	9.594	20.033	9	3.004	0.623	2.63
39	9.856	20.011	8.927	3.15	0.642	2.69
40	6.715	39.517	8.926	3.626	0.196	1.49
41	8.152	20.026	8.758	3.152	0.507	2.29
42	5.727	26.478	8.966	3	0.252	1.67
43	10.727	20.051	8.941	3.036	0.716	2.92
44	12.969	20.004	9	3.271	0.913	3.57
45	5.698	38.391	9	3.004	0.17	1.35
46	5.015	21.604	8.967	3.286	0.267	1.68
47	7.055	20.043	8.78	3.015	0.429	2.07
48	13.157	20.001	8.994	3.006	0.938	3.64
49	5.523	20.019	9	4.469	0.309	1.79
50	12.775	20.001	9	3.048	0.901	3.52

11 CONCLUSIONS

WEDM is an important machining process in manufacturing industries to machine intricate shapes of electrically conductive materials irrespective of their hardness. However, the selection of the right combination of input variables in WEDM for optimal performance is not possible by the trial-and-error method. The present work proposes a novel approach for WEDM based on the integration of two evolutionary approaches, namely, GP and NSGA-II. GP is such powerful tool that it can extract any complex quantitative relationship between input variables and output parameters effectively, whereas NSGA-II is a widely used potential tool for multi-objective optimization.

In this work, the most important machining performances of WEDM, namely MRR and surface finish, are considered. Initially, GP was used to model the mathematical relations for the chosen machining performances. Then, the models developed by GP were tested for their suitability using statistical methods. The individual effects and the interaction effects of the input variables on the chosen output parameters were also studied. Later, the validated mathematical models of GP were used by NSGA-II to find the multiple sets of optimal solutions so as to enable a manufacturing engineer to choose a particular optimal set of input

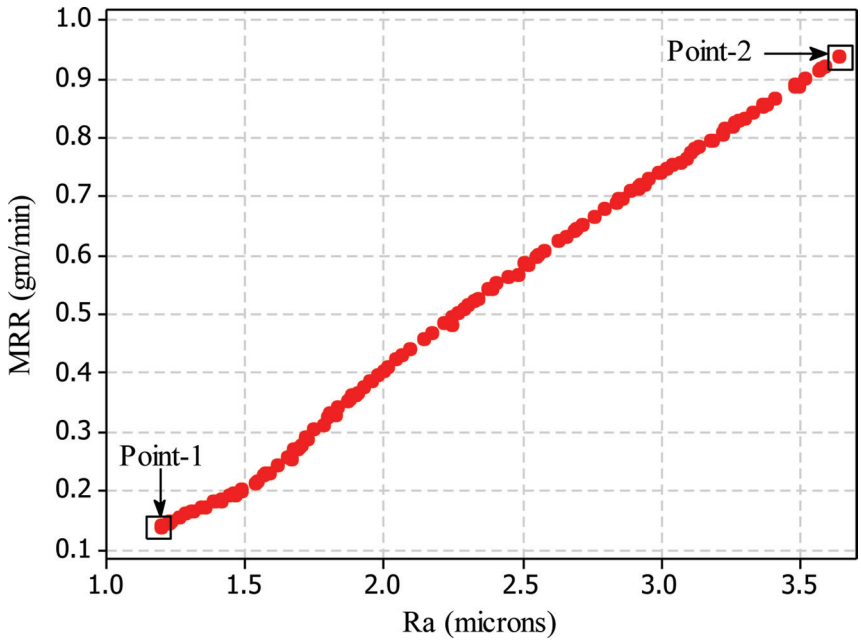
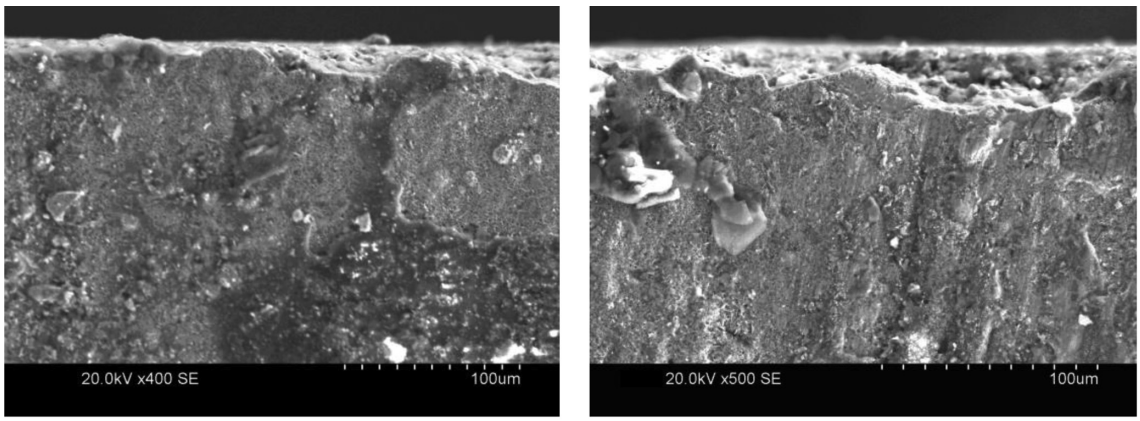


Fig. 19 Pareto optimal front



(a)
x_1= 5μs; x_2=39.65μs; x_3=8.98g
x_4=4.34mm/min; MRR = 0.135 gm/min
Ra = 1.20 μm

(b)
x_1= 13μs; x_2=20μs; x_3=9g
x_4=3mm/min; MRR = 0.922 gm/min
Ra = 3.49 μm

Fig. 20 SEM photographs at the optimal values of individual output responses

variables depending upon the specific requirements. The selection of optimum values is essential for the process automation and implementation of a computer integrated manufacturing system.

© Authors 2011

REFERENCES

1 **Ho, K. H., Newman, S. T., Rahimifard, S.,** and **Allen, R. D.** State of art in wire electrical discharge machining (WEDM). *Int. J. Mach. Tools Mf.*, 2004, **44**, 1247–1259.

2 **Warren Liao, T.** and **Chen, L. J.** Manufacturing process modeling and optimization based on multi-layer perceptron network. *Trans. ASME, J. Mfg Sci. Engng*, 1998, **120**, 109.

3 **Scott, D., Sreedhar, B.,** and **Rajurkar, K. P.** Analysis and optimization of parameter combinations in wire electrical discharge machining. *Int. J. Prod. Res.*, 1991, **29**(11), 2189–2207.

4 **Tarng, Y. S., Ma, S. C.,** and **Chung, L. K.** Determination of optimal cutting parameters in wire electrical

discharge machining. *Int. J. Mach. Tools Mf.*, 1995, **35**(12), 1693–1701.

5 **Spedding, T. A.** and **Wang, Z. Q.** Study on modeling of wire EDM process. *J. Mater. Processing Technol.*, 1997, **69**, 18–28..

6 **Huang, J. T., Liao, Y. S.,** and **Hsue, W. J.** Determination of finish cutting operation number and machining parameters setting in wire electrical discharge machining. *J. Mater. Processing Technol.*, 1999, **87**, 69–81.

7 **Ramakrishnan, R.** and **Karunamoorthy, L.** Surface roughness model for CNC wire electro discharge machining. *J. Mfg Technol. Today*, 2004, **3**(5), 8–11.

8 **Hewidy, M. S., El-Taweel, T. A.,** and **El-Safty, M. F.** Modeling the machining parameters of wire electrical discharge machining of Inconel 601 using RSM. *J. Mater. Processing Technol.*, 2005, **169**, 328–336.

9 **Sarkar, S., Mitra, S.,** and **Bhattacharyya, B.** Parametric optimisation of wire electrical discharge machining of γ titanium aluminide alloy through an artificial neural network model. *Int. J. Advd Mfg Technol.*, 2006, **27**, 501–508.

10 **Manna, A.** and **Bhattacharyya, B.** Taguchi and Gauss elimination method: A dual response approach for parametric optimization of CNC wire cut EDM of PRAl-SiCMMC. *Int. J. Advd Mfg Technol.*, 2006, **28**, 67–75.

11 **Han, F., Jiang, J.,** and **Yu**, D. Influence of machining parameters on surface roughness in finish cut of WEDM. *Int. J. Advd Mfg Technol.*, 2007, **34**, 538–546.

12 **Chiang, Ko-Ta** Modeling and analysis of the effects of machining parameters on the performance characteristics in the EDM process of Al_2O_3+TiC mixed ceramic. *Int. J. Advd Mfg Technol.*, 2008, **37**, 523–533.

13 **Jain, N. K., Jain, V. K.,** and **Jha, S.** Parametric optimization of advanced fine-finishing processes. *Int. J. Advd Mfg Technol.*, 2007, **34**, 1191–1213.

14 **Sardiñas, R. Q., Santana, M. R.,** and **Brindis, E. A.** Genetic algorithm based multiobjective optimization of cutting parameters in turning processes. *Engng Applic. Artif. Intell.*, 2006, **19**, 127–133. DOI:10.1016/j.engappai.2005.06.007.

15 **Fonseca, C. M.** and **Fleming, P. J.** Genetic algorithms for multiobjective optimization: formulation, discussion, and generalization. In Proceedings of the 5th International Conference on *Genetic algorithms*, Urbana-Champaign, IL, USA, 1 June 1993, pp. 416–423 (Morgan Kaufmann, San Mateo).

16 **Horn, J., Nafpliotis, N.,** and **Goldberg, D. E.** A niched Pareto genetic algorithm for multiobjective optimization. In Proceedings of the 1st IEEE Conference on *Evolutionary computation*, IEEE World Congress on *Computational intelligence*, Orlando, Florida, 27–29 June 1994, vol. 1 pp. 82–87.

17 **Zitzler, E.** and **Thiele, L.** Multiobjective optimization using evolutionary algorithms – a comparative case study. In Proceedings of the 5th International Conference on *Parallel problem solving from nature* (Eds A. E. Eiben, T. Bäck, M. Schoenauer, and H. P. Schwefel), Amsterdam, 27–30 September 1998, pp. 292–301 (Springer, Berlin).

18 **Srinivas, N.** and **Deb, K.** Multiobjective function optimization using nondominated sorting in genetic algorithms. *Evolutionary Comput.*, 1994, **2**(3), 221–248. DOI: 10.1162/evco.1994.2.3.221.

19 **Deb, K., Pratap, A., Agarwal, S.,** and **Meyarivan, T.** A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Comput.*, 2002, **6**(3), 187–192.

20 **Koza, J. R.** *Genetic programming: On the programming of computers by means of natural selection*, 1992 (MIT Press, Cambridge, MA).

21 **Dolinsky, J. U., Jenkinson, I. D.,** and **Colquhoun, G. J.** Application of genetic programming to the calibration of industrial robots. *Comput. Ind.*, 2007, **58**, 255–264.

22 **Zhang, L., Jack, L. B.,** and **Nandi, A. K.** Fault detection using genetic programming. *Mech. Syst. Signal Processing*, 2005, **19**, 271–289.

23 **Ashour, A. F., Alvarez, L. F.,** and **Toropov, V. V.** Empirical modelling of shear strength of RC deep beams by genetic programming. *Comput. Structs*, 2003, **81**, 331–338.

24 **Bastian, A.** Identifying fuzzy models utilizing genetic programming. *Fuzzy Sets and Syst.*, 2000, **113**, 333–350.

25 **Nastran, M.** and **Balic, J.** Prediction of metal wire behavior using genetic programming. *J. Mater. Processing Technol.*, 2002, **122**, 368–373.

26 **Brezocnik, M., Kovacic, M.,** and **Ficko, M.** Prediction of surface roughness with genetic programming. *J. Mater. Processing Technol.*, 2004, **157–158**, 28–36.

27 **Kovacic, M., Uratnik, P., Brezocnik, M.,** and **Turk, R.** Prediction of the bending capability of rolled metal sheet by genetic programming. *Mater. Mfg Processes*, 2007, **22**(5), 634–640.

28 **Sette, S.** and **Boullart, L.** Genetic programming: principles and applications. *Engng Applic. Artif. Intell.*, 2001, **14**, 727–736.

29 **Deb, K.** *Multi-objective optimization using evolutionary algorithms*, 1st edition, Wiley-Interscience Series in Systems and Optimization, 2001 (John Wiley, Chichester/New York).

30 **Deb, K.** and **Agrawal, R. B.** Simulated binary crossover for continuous search space. *Complex Syst.*, 1995, **9**(3), 115–148.

31 **Fonlupt, C.** Solving the ocean color problem using a genetic programming approach. *Appl. Soft Comput.*, 2001, **1**(2), 63–72.

32 **Zonker, D.** and **Punch, B.** *Lilgp user's manual*, 1996 (Michigan State University, East Lansing, MI).