Ain Shams Engineering Journal (2015) 6, 315–331

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com

MECHANICAL ENGINEERING

Parametric optimization of ultrasonic machining process using gravitational search and fireworks algorithms

Debkalpa Goswami, Shankar Chakraborty *

Department of Production Engineering, Jadavpur University, Kolkata 700 032, West Bengal, India

Received 30 April 2014; revised 10 September 2014; accepted 16 October 2014
Available online 24 November 2014

KEYWORDS

Ultrasonic machining process;
Optimization;
Gravitational search algorithm;
Fireworks algorithm;
Response

Abstract Ultrasonic machining (USM) is a mechanical material removal process used to erode holes and cavities in hard or brittle workpieces by using shaped tools, high-frequency mechanical motion and an abrasive slurry. Unlike other non-traditional machining processes, such as laser beam and electrical discharge machining, USM process does not thermally damage the workpiece or introduce significant levels of residual stress, which is important for survival of materials in service. For having enhanced machining performance and better machined job characteristics, it is often required to determine the optimal control parameter settings of an USM process. The earlier mathematical approaches for parametric optimization of USM processes have mostly yielded near optimal or sub-optimal solutions. In this paper, two almost unexplored non-conventional optimization techniques, i.e. gravitational search algorithm (GSA) and fireworks algorithm (FWA) are applied for parametric optimization of USM processes. The optimization performance of these two algorithms is compared with that of other popular population-based algorithms, and the effects of their algorithm parameters on the derived optimal solutions and computational speed are also investigated. It is observed that FWA provides the best optimal results for the considered USM processes.

1. Introduction

Due to ever increasing use of advanced materials, such as carbides, ceramics and nimonics in aerospace, nuclear, automobile industries because of their high strength-to-weight ratio, hardness and heat resistant properties, it becomes essential to develop non-traditional machining processes that can efficiently machine those materials into intricate shapes along with improved dimensional features. Ultrasonic machining (USM) is such a non-traditional machining process

* Corresponding author. Tel./fax: +91 33 2414 6153.
E-mail address: s_chakraborty00@yahoo.co.in (S. Chakraborty).
Peer review under responsibility of Ain Shams University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.asej.2014.10.009
2090-4479

for precision machining of hard and brittle materials, having many unique characteristics. This process is non-thermal, non-chemical, non-electrical and creates no change in the metallurgical, chemical or physical properties of the workpiece material. This process is characterized by low material removal rate (MRR) and almost no surface damage to the work material machined. It can be used for machining both electrically conductive and non-conductive materials preferably with low ductility and high hardness into complex shapes with good accuracy and reasonable surface finish. The process is particularly suitable to machine holes with a curved axis of any shape on the workpiece material.

In USM process, low-frequency electrical energy is first converted to a high-frequency electrical signal, which is then fed to a transducer. The transducer transforms the high-frequency electrical energy into mechanical vibrations, which are then transmitted through an energy-focusing device (horn/tool assembly). This causes the tool to vibrate along its longitudinal axis at high frequency (usually $\geqslant$20 kHz). For efficient material removal, the tool and tool holder are so designed considering their mass and shape that resonance can be achieved within the frequency range capability of the machine. A controlled static load is applied to the tool and abrasive slurry (composing of a mixture of abrasive materials, such as silicon carbide, boron carbide, alumina, etc. suspended in oil or water) is pumped around the cutting zone. The vibration of the tool causes the abrasive particles, held in slurry between the tool and the workpiece, to impact the workpiece surface causing material removal by micro-chipping. The schematic diagram of a typical USM setup is shown in Fig. 1. An excellent overview on the mechanism of USM process is available in [1–3].

As the USM process is characterized by low MRR, it is therefore extremely important to adopt proper steps so as to improve its rate of metal removal without affecting the surface finish of the workpiece. This can only be achieved through optimal selection of various machining parameters influencing MRR and surface roughness (SR) in USM process. A comprehensive qualitative and quantitative study on the material removal mechanism and subsequent development of relevant analytical models for MRR and SR is therefore necessary to achieve the optimal machining performance of USM process. Several attempts have already been made to investigate the influence of different process parameters on the two most important performance measures of USM process, i.e. MRR and SR.

2. Literature review

Singh and Khamba [4] deduced the relationship between MRR and other controllable machining parameters, i.e. power rating, tool type, slurry concentration, slurry type, slurry temperature and slurry size by using Taguchi technique for an USM process. Dvivedi and Kumar [5] studied the effects of workpiece material, grit size, slurry concentration, power rating and tool material on SR of an USM process. Taguchi method was applied to obtain the optimal parametric setting for that process. Jain et al. [6] optimized an USM process using genetic algorithm (GA), giving details of formulation of the optimization model, solution methodology used and optimization result. Singh and Khamba [7] selected tool material, power rating, slurry type, slurry temperature, slurry concentration and slurry grit size as the input parameters, and SR as the single response for an USM process. The outcome of a Taguchi method-based model was adopted for developing a mathematical formulation of SR using Buckingham's π-theorem. Jadoun et al. [8] applied Taguchi method for identifying the optimal settings for workpiece material, tool material, grit size of the abrasive, power rating and slurry concentration of an USM process. The effects of those process parameters on oversize, out-of-roundness and conicity were also studied.

Kumar and Khamba [9] determined the optimal combination of various input factors, such as type of abrasive slurry, their size and concentration, nature of tool material and power rating of an USM process applying Taguchi's multi-objective optimization technique. Kumar and Khamba [10] determined

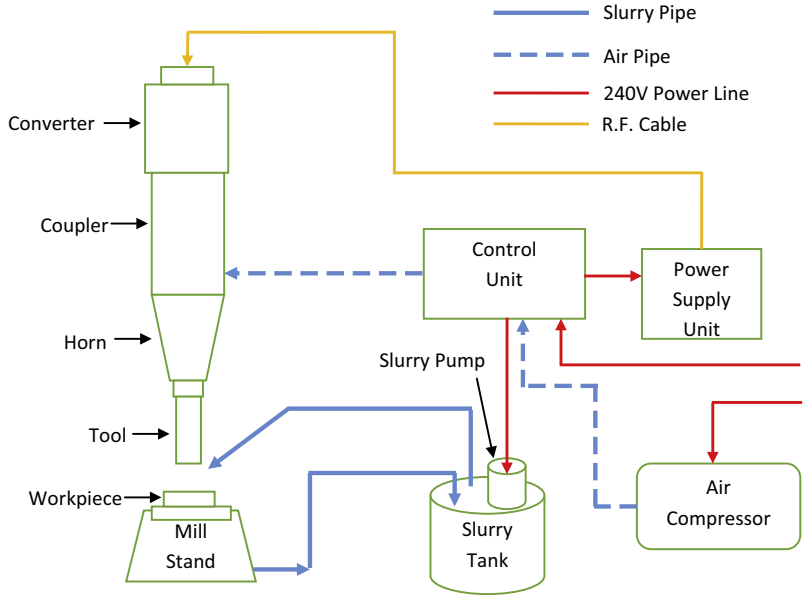

Figure 1 Schematic diagram of a typical USM setup.

the optimal settings of parameters for an USM process using Taguchi method and developed a micro-model for prediction of MRR in USM process using dimensional analysis. Rao et al. [11] applied simulated annealing (SA) technique for optimization of an USM process and observed that it had a better performance as compared to GA. Rao et al. [12] considered amplitude of ultrasonic vibration, frequency of ultrasonic vibration, mean diameter of abrasive particles, volumetric concentration of abrasive particles and static feed force of an USM process as the control parameters, and maximized the value of MRR subjected to a given SR constraint. The optimization of USM process was also carried out using artificial bee colony (ABC), harmony search (HS) and particle swarm optimization (PSO) algorithms, and the results were compared with that obtained using GA. Gauri et al. [13] optimized the correlated multiple responses of two USM processes using weighted principal component, principal component analysis (PCA)-based technique for order preference by similarity to ideal solution (TOPSIS) and PCA-based gray relational analysis (GRA) methods. Rao and Kalyankar [14] applied teaching-learning-based optimization (TLBO) algorithm for optimization of an USM process. The optimization performance of TLBO algorithm was compared with that of GA, SA, ABC, PSO, HS and shuffled frog leaping (SFL) algorithms, and it was observed that TLBO algorithm showed the best optimization performance. Lalchhuanvela et al. [15] considered abrasive grit size, slurry concentration, power rating, tool feed rate and slurry flow rate as the predominant USM process parameters, and determined the optimal combination of those process parameters for maximum MRR and minimum SR using a response surface methodology (RSM)-based multi-objective optimization technique. In a follow-up paper, using RSM, Lalchhuanvela et al. [16] studied the effects of the above-mentioned process parameters on profile accuracy of machined hexagonal holes. Das et al. [17] also applied RSM to develop regression models for MRR and SR in USM of zirconia bio-ceramics. Those models were then optimized using GA technique. Chakravorty et al. [18] compared the performance of weighted signal-to-noise (WSN) ratio method, GRA method, multi-response signal-to-noise (MRSN) ratio method and utility theory (UT) approach for multi-response optimization of USM processes. It was shown that WSN ratio and UT methods would provide better overall optimization results.

The past researchers have also adopted various evolutionary algorithms and hybrid techniques for solving diverse machining and manufacturing related optimization problems. Yıldız [19] developed a hybrid optimization approach based on immune algorithm and hill climbing local search algorithm for solving design and manufacturing optimization problems. Its results were also compared with those of GA, feasible direction method and handbook recommendation. Sayadi et al. [20] applied a discrete firefly metaheuristic to minimize the makespan in permutation flow shop scheduling problems. The permutation flow shop problem was formulated as a mixed integer programming problem and the adopted method was observed to outperform the existing ant colony optimization (ACO)-based solutions. Yıldız [21] applied an optimization approach based on ABC algorithm for optimal selection of cutting parameters in multi-pass turning operation and compared its performance with that of other evolutionary-based optimization techniques. Mukherjee et al. [22] also applied ABC algorithm to optimize two Nd:YAG laser beam machining processes of practical importance. A comparison of results with GA, PSO and ACO using two sample paired t-test demonstrated the superiority of ABC algorithm over the others. Yıldız [23] developed a novel hybrid optimization algorithm called hybrid robust differential evolution for minimizing production cost associated with multi-pass turning operation and applied it to two case studies to illustrate its effectiveness and robustness. Its performance was also validated against other evolutionary algorithms. Goswami and Chakraborty [24] employed differential search algorithm (DSA) to select the optimal process parameters for electrochemical micromachining processes. Two unique characteristics of DSA were identified to make it a successful search tool for solving multi-modal functions. Firstly, DSA may simultaneously use more than one individual; and secondly, it has no inclination to go toward the so-called *best* possible solution of the problem. Branke et al. [25] applied the state-of-the-art covariance matrix adaptation evolution strategy to find out the best dispatching rules in a complex job shop scheduling problem. The robustness of the evolved dispatching rules against variations in the underlying job shop scenario was also analyzed. Yıldız [26] applied cuckoo search (CS) algorithm for solving a milling optimization problem, and demonstrated its superiority as an effective and robust approach over the other popular evolutionary algorithms. The CS algorithm was also implemented by Goswami and Chakraborty [27] to predict trends of multi-dependent responses in laser transmission welding processes. Subsequently, a parametric optimization was also performed to establish that CS algorithm had a fast convergence rate and an exceptionally low variability.

Although the past researchers have attempted to solve single- and multi-response optimization problems for various machining operations, especially USM processes, employing different non-conventional optimization techniques, in most of the cases, it is observed that sub-optimal or near optimal results have been arrived at. In this paper, a maiden venture is taken to optimize the machining parameters of USM processes using two almost new non-conventional optimization techniques, i.e. gravitational search algorithm and fireworks algorithm which have immense potential to deal with complex multi-dimensional optimization problems.

3. Gravitational search algorithm

The gravitational search algorithm (GSA) is a newly developed stochastic optimization technique based on the law of gravity and mass interactions [28]. The mindset of engineers is such that they observe and learn from various natural phenomena. In this algorithm, agents are considered as objects and their performance is measured by their masses. A collection of masses, acting as search agents, interacts with each other based on the Newtonian laws of gravitation and motion. Evidently, this approach is completely different from other well-known population-based optimization methods inspired by swarm behaviors.

All of the objects attract each other by the gravity force, while this force causes a global movement of all objects toward the objects with larger masses. This concept is illustrated in Fig. 2. The large masses correspond to good solutions of the problem. In other words, each mass represents a solution, and the algorithm is navigated by properly adjusting the

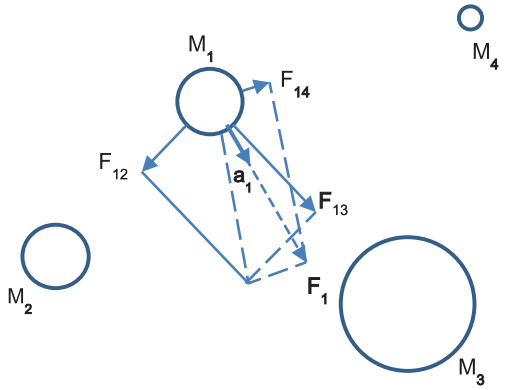

Figure 2 Gravity force causes a global movement of all objects toward the largest mass.

gravitational and inertial masses. With lapse of time, it is expected that the masses will be attracted by the largest mass, which represents the optimal solution in the search space [29].

Every agent in GSA is specified by four parameters, i.e. position of the mass in dth dimension, inertia mass, active gravitational mass and passive gravitational mass. The positions of the mass of an agent at specified dimensions represent a solution of the problem and the inertial mass of an agent reflects its resistance to make its movement slow. Both the gravitational mass and the inertial mass, which control the velocity of an agent in specified dimension, are computed by fitness evaluation of the problem. The positions of the agents in specified dimensions (solutions) are updated with every iteration and the best fitness along with its corresponding agent is recorded. The termination condition of the algorithm is defined by a fixed amount of iterations, reaching which the algorithm automatically stops. After termination of the algorithm, the recorded best fitness at final iteration becomes the global fitness for a particular problem and the position of the mass at a specified dimension of the corresponding agent becomes the global solution of that problem.

The algorithm can be summarized as follows [28,30–32]:

Step 1: Initialize the agents.

Initialize the positions of N agents randomly within the given search interval as follows:

$$X_i = \left(x_i^1, \ldots, x_i^d, \ldots, x_i^n\right) \quad \text{for} \quad i = 1, 2, \ldots, N$$

where x_i^d represents the position of ith agent in dth dimension and n is the space dimension.

Step 2: Evaluate the fitness values and compute the best fitness for each agent.

Perform the fitness evaluation for all agents at each iteration, and also compute the *best* and *worst* fitness at each iteration as defined below:

$$best(t) = \begin{cases} \min\limits_{j\in\{1,\ldots,N\}} fit_j(t), & \text{for minimization problems} \\ \max\limits_{j\in\{1,\ldots,N\}} fit_j(t), & \text{for maximization problems} \end{cases} \tag{1}$$

$$worst(t) = \begin{cases} \max\limits_{j\in\{1,\ldots,N\}} fit_j(t), & \text{for minimization problems} \\ \min\limits_{j\in\{1,\ldots,N\}} fit_j(t), & \text{for maximization problems} \end{cases} \tag{2}$$

where $fit_j(t)$ represents the fitness of jth agent at iteration t, and $best(t)$ and $worst(t)$ denote the best and the worst fitness at generation t.

Step 3: Compute the gravitational constant, G.

$$G(t) = G_0 e^{-\alpha t/T} \tag{3}$$

where G_0 represents the initial value of G, α is a constant and T is the maximum number of iterations.

Step 4: Calculate the mass of the agents.

$$M_{ai} = M_{pi} = M_{ii} = M_i, \quad i = 1, 2, \ldots, N \tag{4}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{5}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{6}$$

where M_{ai}, M_{pi} and M_{ii} are respectively the active, passive and inertial gravitational masses of ith agent.

Step 5: Calculate accelerations of the agents.

Compute the acceleration of ith agent at iteration t as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{7}$$

where $F_i^d(t)$ is the total force acting on ith agent, calculated as follows:

$$F_i^d(t) = \sum_{j\in Kbest, j\neq i} rand_j F_{ij}^d(t) \tag{8}$$

Kbest is the set of first K agents with the best fitness value and biggest mass. *Kbest* is computed in such a manner that it decreases linearly with time. $F_{ij}^d(t)$ is the force acting on agent i from agent j at dth dimension and tth iteration, and is computed as follows:

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right) \tag{9}$$

where $R_{ij}(t)$ is the Euclidian distance between the two agents i and j at iteration t, and ε is a small positive constant (used to avoid division by zero).

Step 6: Update velocity and position of the agents.

Compute the velocity and position of the agents at next iteration $(t + 1)$ using the following relations:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{10}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{11}$$

Step 7: Repeat step (2) to step (6) until the iteration process reaches its set maximum limit.

Return the best fitness computed at final iteration as the global fitness, and positions of the corresponding agents at specified dimensions as the global solution of the problem.

For a simpler and more qualitative perspective of the above procedure, a flowchart for GSA is presented in Fig. 3.

4. Fireworks algorithm

In recent years, algorithms inspired from swarm intelligence have become quite popular as a powerful global optimization tool for solving multi-dimensional problems. These algorithms evolve from mathematical modeling of biological or social phenomena, or other laws of nature. Inspired by observing explosion of fireworks in the night sky, a novel swarm intelligence algorithm called fireworks algorithm (FWA) has recently been proposed for global optimization of complex functions [33,34].

When a firework is set off, a shower of sparks fills the local space around the firework. The explosion process of a firework can be viewed as a search in the local space around a specific point where the firework is set off through the sparks generated in the explosion. When asked to find a point x_j satisfying $f(x_j) = y$, 'fireworks' in potential space can be continually set off until one 'spark' targets or is fairly near the point x_j. Simulating the process of setting off fireworks, a basic framework of FWA is depicted in Fig. 4.

The effectiveness of FWA lies in the good design of the explosion phenomenon and a proper method for selection of locations. By carefully observing fireworks displays, two distinct behaviors of fireworks can be observed. In a good explosion, numerous sparks are generated and these sparks centralize the explosion center. In a bad firework, however, few sparks are generated which are scattered in space. These two behaviors are illustrated in Fig. 5.

Keeping this in mind, FWA is designed for the general optimization problems as follows:

Minimize $f(\boldsymbol{x}) \in \mathbf{R}$, $\boldsymbol{x}_{\min} \leqslant \boldsymbol{x} \leqslant \boldsymbol{x}_{\max}$, where $\boldsymbol{x} = x_1, x_2, \ldots, x_d$ denotes a location in the potential space, $f(\boldsymbol{x})$ is an objective function, and $\boldsymbol{x}_{\min}$ and $\boldsymbol{x}_{\max}$ denote the bounds of the potential space.

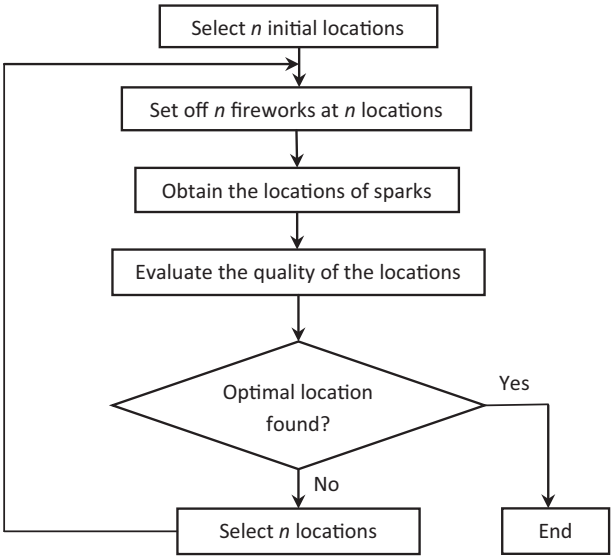

Figure 4 Flowchart for fireworks algorithm.

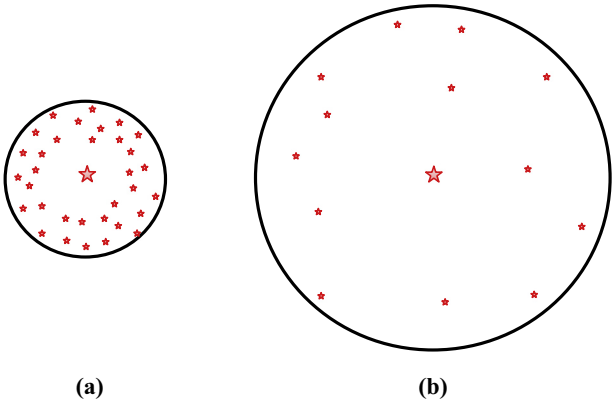

Figure 5 Two types of fireworks explosion: (a) Good explosion and (b) bad explosion.

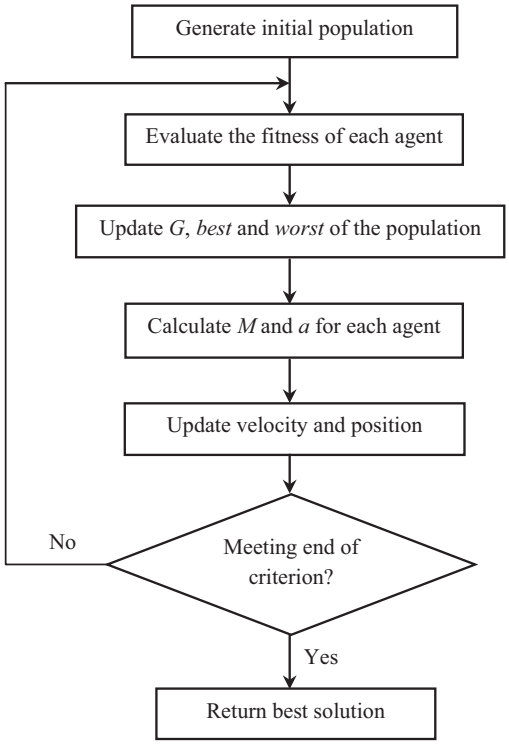

Figure 3 Flowchart for gravitational search algorithm.

The framework for FWA is given as below:

1. Randomly select n locations for fireworks
2. **while** stop criteria = false **do**
3. Set off n fireworks respectively at n locations
4. **for** each firework $\boldsymbol{x_i}$ **do**
5. Calculate the number of sparks that the firework yields, $\hat{s}_i$ according to Snippet 1
6. Obtain locations of $\hat{s}_i$ sparks of the firework $\boldsymbol{x_i}$ using pseudo-code 1
7. **end for**
8. **for** k = 1: $\hat{m}$ **do**
9. Randomly select a firework $\boldsymbol{x_j}$
10. Generate a specific spark for the firework using pseudo-code 2
11. **end for**
12. Select the best location and keep it for the next explosion generation

13. Randomly select $(n-1)$ locations from the two types of sparks and the current fireworks according to the probability given in Snippet 2.
14. **end while**

Snippet 1

$$\hat{s}_i = \begin{cases} round(a \cdot m) & \text{if } s_i < am \\ round(b \cdot m) & \text{if } s_i > bm, a < b < 1 \\ round(s_i) & \text{otherwise} \end{cases}$$

where $s_i = m \cdot \frac{y_{\max} - f(x_i) + \xi}{\sum_{i=1}^{n}(y_{\max} - f(x_i)) + \xi}$,

m is a parameter controlling the total number of sparks generated by n fireworks,

$y_{\max} = \max(f(x_i))$, ξ is a small positive constant used to avoid division by zero, and a and b are two constant parameters.

Snippet 2

Selection probability of a location x_i is defined as follows: $p(\boldsymbol{x}_i) = \frac{R(\boldsymbol{x}_i)}{\sum_{j \in K} R(\boldsymbol{x}_j)}$,

where $R(\boldsymbol{x}_i) = \sum_{j \in K} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$, and K is the set of all current locations of both fireworks and sparks.

Pseudo-code 1: Obtain the location of a spark.

1. Initialize the location of the spark: $\tilde{\boldsymbol{x}}_j = \mathbf{x}_i$
2. $z = round(d.rand(0,1))$
3. Randomly select z dimensions of $\tilde{\boldsymbol{x}}_j$
4. Calculate the displacement: $h = A_i.\ rand(-1,1)$, where A_i is the amplitude of explosion of each firework
5. **for** each dimension $\tilde{x}_k^j \in \{\text{pre} - \text{selected}\, z \text{ dimensions of } \tilde{\boldsymbol{x}}_j\}$ **do**
6. $\tilde{x}_k^j = \tilde{x}_k^j + h$
7. **if** $\tilde{x}_k^j < x_k^{\min}$ or $\tilde{x}_k^j > x_k^{\max}$ **then**
8. map $\tilde{x}_k^j$ to the potential space: $\tilde{x}_k^j = x_k^{\min} + \|\tilde{x}_k^j\| \% (x_k^{\max} - x_k^{\min})$
9. **end if**
10. **end for**

Pseudo-code 2: Obtain the location of a specific spark.

11. Initialize the location of the spark: $\hat{\boldsymbol{x}}_j = \boldsymbol{x}_i$
12. $z = round(d.rand(0,1))$;
13. Randomly select z dimensions of $\hat{\mathbf{x}}_j$
14. Calculate the coefficient of the Gaussian explosion: g = $Gaussian(1,1)$
15. **for** each dimension $\hat{x}_k^j \in \{\text{pre} - \text{selected}\, z \text{ dimensions of } \hat{\mathbf{x}}_j\}$ **do**
16. $\hat{x}_k^j = \hat{x}_k^j \cdot g$
17. **if** $\hat{x}_k^j < x_k^{\min}$ or $\hat{x}_k^j > x_k^{\max}$ **then**
18. map $\hat{x}_k^j$ to the potential space: $\hat{x}_k^j = x_k^{\min} + |\hat{x}_k^j| \% (x_k^{\max} - x_k^{\min})$
19. **end if**
20. **end for**

5. Optimization of USM processes

In order to validate the applicability and effectiveness of GSA and FWA algorithms, the machining performances of two USM processes are optimized here. The first example is taken from Das et al. [17], whereas, the second example is cited from Jain et al. [6]. For arriving at a suitable compromise between computation speed (CPU time) and solution accuracy, in this paper, the following parameter settings are used (unless specifically otherwise mentioned) for the two adopted algorithms.

For GSA: $N = 20$, $G_0 = 100$, maximum number of iterations = 500 and $\alpha = 5$.
For FWA: $n = 5$, $m = 75$, maximum number of iterations = 150, $a = 0.01$ and $b = 0.40$.

A detailed justification for this choice of parameter settings is provided later in Section 6.

5.1. Example 1

Das et al. [17] performed USM operation on zirconia bio-ceramic materials using a Sonic-Mill, 1000 W ultrasonic machine (having a frequency of vibration of 20 kHz). A flat plate of 58.5 mm × 58.5 mm × 5.1 mm of zirconia bio-ceramic was used as the workpiece. Boron carbide powder of different grain sizes mixed with water at room temperature was used as the abrasive slurry. Tubular stainless steel (S304) tools, 20 mm long with 8.2 mm hole diameter, of hexagonal shape were fabricated and used for the machining operation. Das et al. [17] considered abrasive grit size, slurry concentration, power rating and tool feed rate as the predominant control parameters. The values of those USM process parameters along with their corresponding levels are provided in Table 1. The effects of those USM process parameters on MRR and SR (in terms of Ra value) were studied by carrying out a set of planned experiments using central composite design with 31 experimental runs. Based on the observed experimental data of Das et al. [17], two second order regression equations for MRR and SR are developed using RSM technique. The corresponding response surface plots for MRR and SR with respect to the USM process parameters are respectively shown in Figs. 6 and 7.

$$\begin{aligned} Y_{\text{MRR}} = {} & 0.136843 + 0.0170333x_1 - 4.91667\text{E}-04x_2 \\ & + 0.000458333x_3 - 1.41667\text{E}-04x_4 \\ & + 0.00471429x_1^2 - 7.32143\text{E}-05x_2^2 - 8.10714\text{E} \\ & - 04x_3^2 + 0.000126786x_4^2 + 0.000150000x_1x_2 \\ & - 1.37500\text{E}-04x_1x_3 - 0.00210000x_1x_4 \\ & + 0.000437500x_2x_3 - 4.50000\text{E}-04x_2x_4 \\ & - 1.87500\text{E}-04x_3x_4 \end{aligned} \quad (12)$$

Table 1 USM process parameters with their levels.

Process parameter	Level				
	−2	−1	0	1	2
Grit size (μm) (x_1)	16	24	34	44	63
Slurry concentration (g/l) (x_2)	30	35	40	45	50
Power rating (W) (x_3)	300	350	400	450	500
Feed rate (mm/min) (x_4)	0.84	0.96	1.08	1.20	1.32

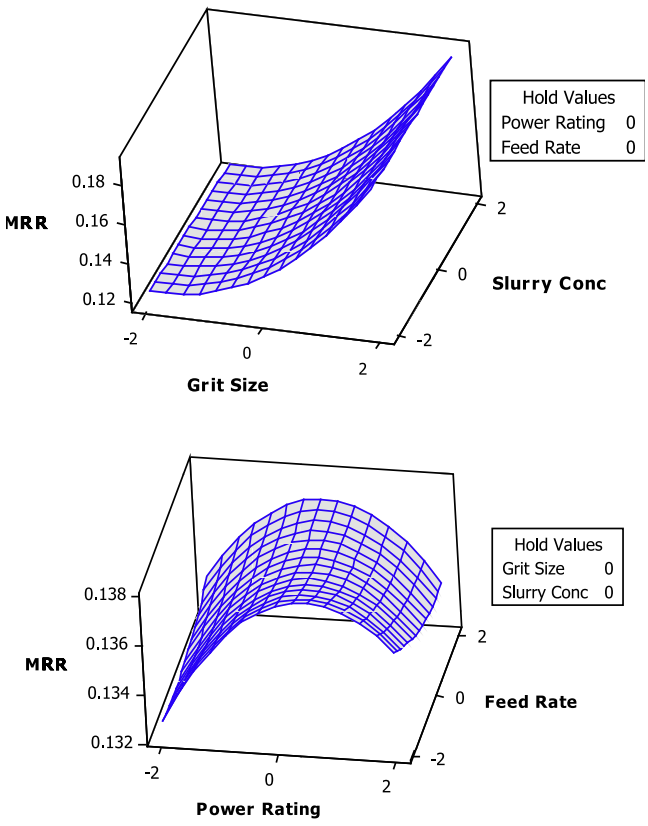

Figure 6 Response surfaces for MRR (coded parameter values).

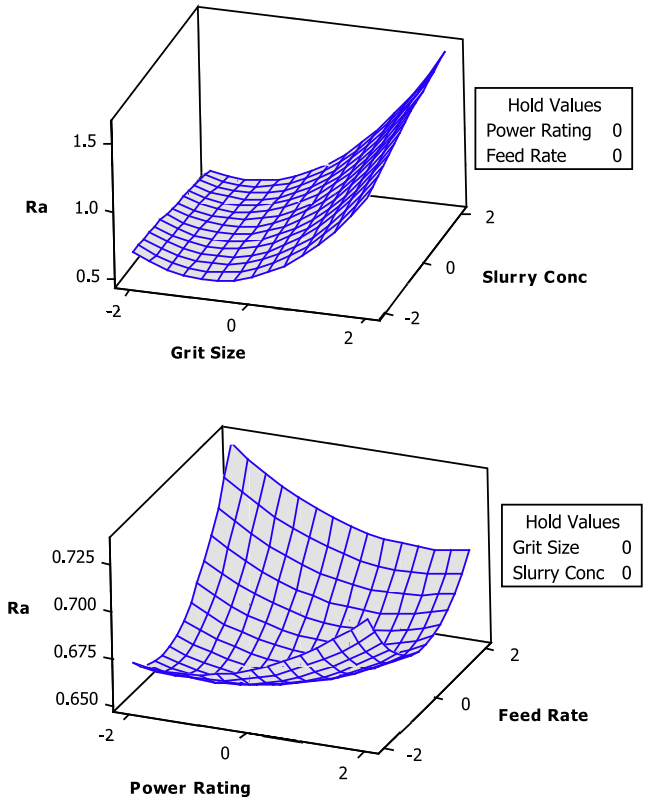

Figure 7 Response surfaces for SR (coded parameter values).

$$
\begin{aligned}
Y_{SR} = {} & 0.652857 + 0.210000x_1 + 0.0250000x_2 \\
& + 0.000833333x_3 + 0.00583333x_4 + 0.0976190x_1^2 \\
& - 0.00488095x_2^2 + 0.00386905x_3^2 + 0.00886905x_4^2 \\
& + 0.0300000x_1x_2 - 0.00250000x_1x_3 \\
& - 0.00250000x_1x_4 - 0.00250000x_2x_3 \\
& + 0.0100000x_2x_4 - 0.00500000x_3x_4 \qquad (13)
\end{aligned}
$$

Employing Derringer's desirability function approach for multi-response optimization, the optimal parametric settings of grit diameter = 45.91 μm, slurry concentration = 30 g/l, power rating = 384.85 W and tool feed rate = 1.03 mm/min were obtained [17]. The maximum value of gravimetric MRR was observed as 0.1483 g/min and the minimum SR value was 0.69 μm. Das et al. [17] did not consider single response optimization of the two responses (MRR and SR).

Based on the respective pseudo-codes, the computer codes for both GSA and FWA algorithms are developed in MATLAB 7.10.0 (R2010a) in an Intel® Core™ i5-2450 M CPU @ 2.50 GHz, 4.00 GB RAM operating platform. At first, the two second order RSM-based equations are separately optimized using both the algorithms and the corresponding single response optimization results are shown in Table 2. The constraints for these two optimization problems are set as $16 \leqslant x_1 \leqslant 63$ (μm), $30 \leqslant x_2 \leqslant 50$ (g/l), $300 \leqslant x_3 \leqslant 500$ (W) and $0.84 \leqslant x_4 \leqslant 1.32$ (mm/min). From Table 2, it is observed that using GSA algorithm, the maximum value of MRR is achieved as 0.1904 g/min and the minimum value of SR is obtained as 0.4826 μm. On the other hand, the single response optimization of this USM process using FWA technique provides a maximum MRR of 0.2007 g/min and a minimum SR of 0.4678 μm. It is found that while applying both GSA and FWA techniques, the value of MRR is increased and the value of SR is trimmed down as compared to those obtained by the desirability function approach. A comparative study revealing the average CPU times taken by GSA, FWA and other popular population-based optimization algorithms while solving this single response optimization problem shows that except for FWA algorithm, the remaining algorithms are quite similar to each other with respect to their average CPU times: for GSA, CPU time = 1.25 s; for ABC, CPU time = 2.93 s; for ACO, CPU time = 3.35 s; for PSO, CPU time = 4.53 s; and for GA, CPU time = 3.05 s. For FWA algorithm, the average CPU time is slightly higher as 7.79 s. In Table 2, the optimal values of two responses (MRR and SR) and settings of different USM process parameters as achieved by GA, ACO, PSO and ABC algorithms are also provided. Fig. 8 shows the convergence diagram for all the considered optimization algorithms with respect to SR response. It proves the faster convergence of GSA and FWA algorithms toward the minimum SR value against the other optimization algorithms. In order to compare the relative optimization performance of GSA and FWA algorithms, two sample paired t-tests are also performed in Table 3 for both the responses to study the existence of significant differences between these two algorithms. Table 3 reveals that these two algorithms are statistically different at 5% significance level with respect to their optimization performance. It is also observed that the optimization performance of FWA is relatively more consistent than that of GSA.

Table 2 Results of single response optimization.

Optimization method	Response	Optimal value	Parameter			
			x_1	x_2	x_3	x_4
GSA	MRR (g/min)	0.1904	60.4666	49.9385	440.5590	0.8815
	SR (μm)	0.4826	22.4998	48.9586	388.1507	0.9190
FWA	MRR (g/min)	0.2007	63.0000	50.0000	444.1997	0.8400
	SR (μm)	0.4678	22.7672	50.0000	339.3737	0.8401
GA	MRR (g/min)	0.1535	52.7098	44.8183	300.0000	1.2483
	SR (μm)	0.6977	44.7634	30.0008	369.5321	0.9307
ACO	MRR (g/min)	0.1730	54.6579	50.0000	455.1514	0.8790
	SR (μm)	0.5253	32.8093	30.0000	403.1287	1.1735
PSO	MRR (g/min)	0.1698	53.2673	50.0000	496.9831	0.8400
	SR (μm)	0.5647	22.6487	33.7365	319.6699	1.0291
ABC	MRR (g/min)	0.1795	57.8587	32.6227	438.4538	0.8818
	SR (μm)	0.5125	22.5449	45.6502	341.6648	0.9681

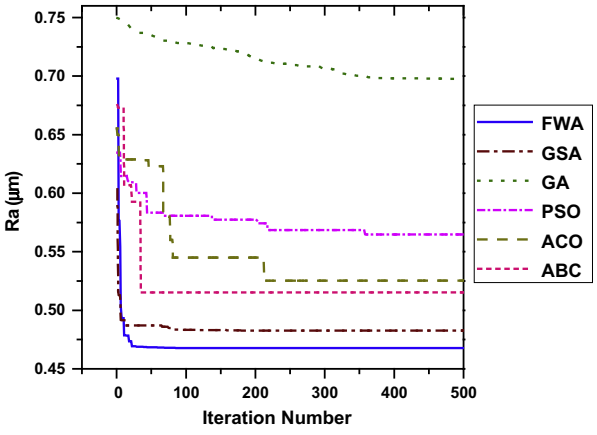

Figure 8 Convergence diagram with respect to SR response.

Now, for multi-response optimization of the above-considered USM process, the following objective function is developed.

$$\text{Min}(Z) = \frac{w_1 Y_{\text{SR}}}{\text{SR}_{\text{min}}} - \frac{w_2 Y_{\text{MRR}}}{\text{MRR}_{\text{max}}} \tag{14}$$

where w_1 and w_2 are the weights (relative importance) assigned to SR and MRR respectively, and SR_{min} is the minimum value of SR, and MRR_{max} is the maximum value of MRR. These minimum and maximum values of the responses are obtained from the single response optimization results. The weight values allotted to the responses are so chosen that $w_1 + w_2 = 1$. The choice for these weights entirely depends on the preference of the process engineers or they can be determined using analytic hierarchy process. Table 4 shows the results of multi-response optimization while employing both GSA and FWA techniques for three different situations, i.e. case 1: $w_1 = w_2 = 0.5$; case 2: $w_1 = 0.1$, $w_2 = 0.9$; and case 3: $w_1 = 0.9$, $w_2 = 0.1$. For all these three cases, FWA provides better solutions as compared to GSA, and it is also revealed that the most acceptable and compromised optimization results are obtained when equal importance is allotted to both the responses (for case 1).

The scatter plots in Figs. 9 and 10 respectively exhibit the variations of MRR and SR with respect to four USM process parameters, based on the results as derived using FWA technique. The dots in these scatter plots represent the locations in the search space that are navigated by the algorithm in one simulation run. Together with the developed response surfaces of Figs. 6 and 7, these figures can be used to understand the trends in both responses within the selected range of process parameters.

For USM operation of zirconia bio-ceramics, the following conclusions regarding the parametric influences on MRR and SR can be drawn from Figs. 6, 7, 9 and 10.

(a) Grit size is the most influential factor for both MRR and SR. As grit size increases, causing an increase in the average diameter of abrasive particles, more material can be chipped away from the workpiece in the same time, thus leading to an increase in MRR. Larger abrasive particles also signify that the surface finish will gradually deteriorate, thus causing an increase in SR with increasing grit size.

Table 3 Two sample paired *t*-tests between GSA and FWA algorithms.

Optimization method	Response	Sample size	Optimal value	Mean	Standard deviation	Standard error
GSA	MRR (g/min)	20	0.1904	0.1753	0.0123	0.00275
	SR (μm)	20	0.4826	0.5987	0.0634	0.0142
FWA	MRR (g/min)	20	0.2007	0.1946	0.0083	0.00186
	SR (μm)	20	0.4678	0.5186	0.0292	0.0065

MRR: Estimate for average difference = 0.01934, 95% CI for mean difference = (0.01146, 0.02721), *t*-test of mean difference = 0 (vs not = 0), *t*-value = 5.14, *p*-value = 0.000.
SR: Estimate for average difference = 0.0801, 95% CI for mean difference = (0.0414, 0.1189), *t*-test of mean difference = 0 (vs not = 0), *t*-value = 4.33, *p*-value = 0.000.

Table 4 Multi-response optimization results using GSA and FWA.

Condition	Method	Response	Value	Z	Parameter			
					x_1	x_2	x_3	x_4
Case 1: $w_1 = w_2 = 0.5$	GSA	MRR	0.1079	0.2353	16.0000	50.0000	311.5241	0.8400
		SR	0.5006					
	FWA	MRR	0.1182	0.2101	24.1460	50.0000	389.0573	0.8971
		SR	0.4722					
Case 2: $w_1 = 0.1$, $w_2 = 0.9$	GSA	MRR	0.1755	−0.5069	61.3941	47.2714	314.0399	1.2248
		SR	1.5569					
	FWA	MRR	0.1972	−0.5972	63.0000	30.0000	379.0712	0.8400
		SR	1.3438					
Case 3: $w_1 = 0.9$, $w_2 = 0.1$	GSA	MRR	0.1072	0.8556	17.4949	50.0000	300.0000	0.8400
		SR	0.4890					
	FWA	MRR	0.1144	0.8432	22.9322	50.0000	347.9497	0.8490
		SR	0.4679					

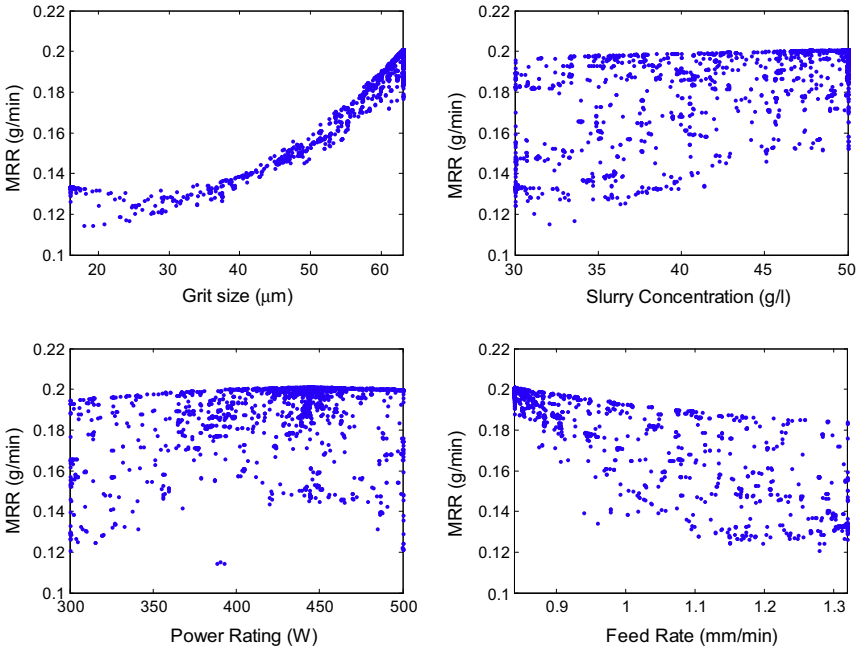

Figure 9 Effects of USM process parameters on MRR.

(b) Both the response surfaces and scatter plots indicate that MRR and SR are least sensitive to slurry concentration. A slight increase in MRR is although observed with increase in slurry concentration. At lower values of MRR (as governed by the other process parameters), this increasing effect is more pronounced. Although negligible for all practical purposes, SR varies non-linearly with slurry concentration. Maximum SR is observed at values near the central level (~40 g/l) of slurry concentration while SR values are lower at both the extreme levels.

(c) MRR varies non-linearly with power rating. Highest values of MRR are obtained when power rating is maintained around its central level (~400 W). At either end of the investigated power rating range, MRR values tend to decrease. The SR response shows quite an interesting behavior for changing values of power rating. The overall trend in Fig. 10 indicates a steady increase in SR with increasing power ratings. However, when grit size and slurry concentration are held at their central levels, it is observed from Fig. 7 that for low values of feed rate, SR increases with increasing power rating, while for high values of feed rate, SR is found to decrease with increasing power rating.

(d) A gradual increase in tool feed rate leads to decreasing values for MRR. For SR response, the effect of feed rate is rather more complex. For low power ratings, surface finish deteriorates (i.e. SR increases) as tool feed rate is increased, while for high power ratings, the best surface finish (i.e. minimum SR) can be achieved at intermediate values of feed rate. This behavior indicates that there exists a high degree of interaction between these two process parameters, i.e. power rating and feed rate.

The most challenging aspect in optimizing any machining process is that MRR and SR are the two conflicting objectives.

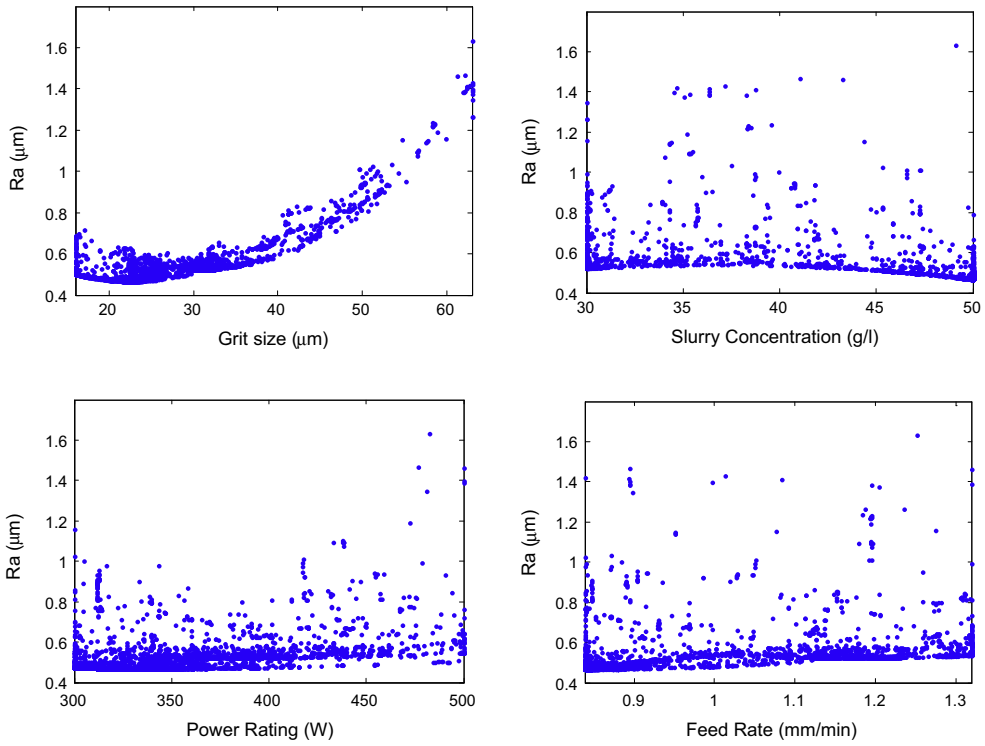

Figure 10 Effects of USM process parameters on SR.

It can be reconfirmed in Fig. 11 which clearly shows that when higher values of MRR are desired, the resulting surface quality must worsen, leading to higher SR values. Thus, during multi-response optimization of a machining process, there must be some trade-off between these two conflicting objectives.

5.2. Example 2

Jain et al. [6] considered a constrained optimization problem for USM process having five machining parameters as amplitude of vibration (A_v) (mm), frequency of vibration (f_v) (Hz or cycles/s), mean diameter of abrasive grains (d_m) (mm), volumetric concentration of abrasive particles in slurry (C_{av}) and static feed force (F_s) (N). A mathematical model for maximizing the volumetric MRR (mm^3/s) was also developed as given in Eq. (15).

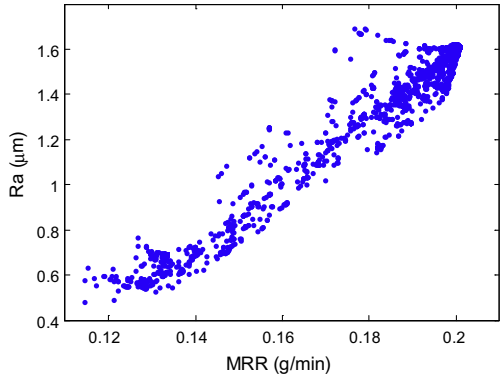

Figure 11 Overall variation of SR with respect to MRR.

$$\mathrm{MRR} = \frac{4.963 A_t^{0.25} K_{\mathrm{usm}}^{0.75} F_s^{0.75} A_v^{0.75} C_{\mathrm{av}}^{0.25} d_m f_v}{[\sigma_{\mathrm{fw}}(1+\lambda)]^{0.75}} \tag{15}$$

Here A_t is the cross-sectional area of cutting tool (mm^2), K_{usm} is a constant of proportionality (mm^{-1}) relating mean diameter of abrasive grains and diameter of projections on an abrasive grain, σ_{fw} is the flow strength of the workpiece material (MPa or N/mm^2), λ is the indentation ratio, and F_s is the sphericity factor of the abrasive particles. The developed model for MRR is subjected to a constraint for SR, as given by Eq. (16).SR constraint:

$$1 - \frac{1154.7}{[A_t \sigma_{\mathrm{fw}}(1+\lambda)]^{0.5} (R_a)_{\max}} \left[\frac{F_s A_v d_m}{C_{\mathrm{av}}}\right]^{0.5} \geq 0.0 \tag{16}$$

The bounds used by Jain et al. [6] for the considered process variables are given as below:

$$0.005 \leqslant A_v \leqslant 0.1 \text{ (mm)}; 10,000 \leqslant f_v \leqslant 40,000 \text{ (Hz)}; 0.007 \leqslant d_m \leqslant 0.15 \text{ (mm)}; 0.05 \leqslant C_{\mathrm{av}} \leqslant 0.5; 4.5 \leqslant F_s \leqslant 45 \text{ (N)}$$

Using GA, Jain et al. [6] solved this constrained optimization problem to achieve the maximum MRR value of 3.553 mm^3/s with a constrained value of SR (in terms of Ra) as 0.0214 µm. Subsequently, Rao et al. [12] applied ABC, HS and PSO algorithms to solve the same model, and then adopted SA algorithm for parametric optimization of the same USM process [11]. In a recent work, Rao and Kalyankar [14] applied TLBO algorithm for solving the same constrained optimization problem and obtained a maximum value of MRR as 4.004 mm^3/s with a constrained R_a value of 0.0003 µm. The optimal settings for the considered process parameters were obtained as amplitude of vibration = 0.0611 mm, frequency of vibration = 40,000 Hz, mean

diameter of abrasive grains = 0.15 mm, volumetric concentration of abrasive particles in slurry = 0.5, and static feed force = 4.5 N. The execution time for TLBO algorithm was observed around 0.35 s.

Now, using GSA and FWA techniques, this constrained optimization problem for maximizing MRR is solved, and the corresponding results are shown in Table 5, along with those obtained from the other popular optimization techniques. Here, the values for all the constants involved in the optimization model are kept same as those considered by Jain et al. [6]. For dealing with the complicated inequality constraint, the death penalty method is employed here. From the results of Table 5, it is observed that for GSA, the maximum value of MRR is 3.9600 mm^3/s which is better than that obtained by GA (3.553 mm^3/s) [6], SA (3.660 mm^3/s) [11], ABC (3.941 mm^3/s) [12], HS (3.870 mm^3/s) [12], PSO (3.950 mm^3/s) [12], ACO (3.8781 mm^3/s) and SFL (3.894 mm^3/s) [14] algorithms, but is worse than the MRR value as resulted from applying TBLO algorithm (4.004 mm^3/s) [14]. It indicates that GSA is not so efficient for constrained optimization problems, as it is mainly developed for unconstrained problems. However, it is an excellent algorithm for optimization problems with respect to very fast convergence and low computational time.

On the other hand, using FWA technique, the maximum MRR value is derived as 4.0061 mm^3/s which is better than that obtained by Rao and Kalyankar [14] employing TLBO algorithm. The convergence diagram for the considered algorithms with respect to MRR is exhibited in Fig. 12.

Table 6 compares the performance of GSA and FWA algorithms while solving this single response optimization problem for the considered USM process. The average CPU times for these two algorithms are observed as 3.96 and 6.97 s respectively. From Table 6, it is quite clear that FWA technique outperforms GSA algorithm with respect to consistency of the optimal solutions although it takes almost twice of the average CPU time as compared to that of GSA algorithm. The result of paired *t*-test, as given in Table 6, highlights that the optimization performance of FWA algorithm is statistically different (at 5% significance level) from that of GSA algorithm.

Fig. 13 displays the effects of amplitude of vibration, frequency of vibration, mean diameter of abrasive grains, volumetric concentration of abrasive particles in slurry and static feed force on MRR for the considered USM process. Since FWA provides the best optimal results, it is used to generate the data for the scatter plots. Maximum MRR can be achieved at lower values of amplitude of vibration and static feed force.

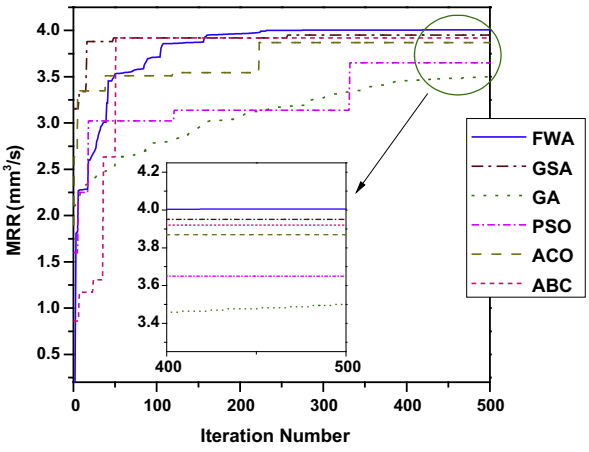

Figure 12 Convergence diagram with respect to MRR.

On the other hand, MRR goes on increasing with the increased values of frequency of vibration, mean diameter of abrasive grains and volumetric concentration of abrasive particles in slurry. Hence, the desired value of maximum MRR can be achieved at higher values frequency of vibration, mean diameter of abrasive grains and volumetric concentration of abrasive particles in slurry, and lower values of amplitude of vibration and static feed force. These phenomena can be confirmed from the optimization results as already given in Table 5.

6. Effects of algorithm parameters on optimization performance

It is quite obvious that the optimal solutions as derived using GSA and FWA algorithms will be affected by the varying values of their algorithm specific parameters. To investigate their effects on the optimal solution quality and average CPU time for both the algorithms, sample experiments are performed on the MRR objective function (which is a maximization problem) of the USM process considered in Example 1. The results obtained are shown in Tables 7-15 and graphically exhibited in Figs. 14-22. In these figures, the vertical error bars denote the standard deviation values obtained during the simulation runs. Generalized trends in performance with respect to different algorithm parameters are expected to be similar if other objective functions are deployed. Results are of course subject to statistical variability. If a minimization objective function is used (say SR) for this purpose, then the increasing MRR trends in these figures will become decreasing SR trends. The

Table 5 Parametric optimization of USM process using different methods.

Optimization method	MRR_{max}	Constraint	A_v	f_v	d_m	C_{av}	F_s
GSA	3.9600	8.0410E−4	0.0176	39,876	0.1441	0.5	16.2511
FWA	4.0061	7.1248E−5	0.0257	39956.28	0.1500	0.5	10.7071
ABC [12]	3.941	0.0124	0.0167	40,000	0.15	0.5	16.4
ACO	3.8781	0.0199	0.0591	40,000	0.15	0.5	4.5
PSO [12]	3.95	0.0095	0.06	40,000	0.15	0.5	4.5
GA [6]	3.553	0.0214	0.0263	39333.9	0.1336	0.479	10.8
SA [11]	3.660	0.0185	0.077	40,000	0.114	0.5	4.53
SFL [14]	3.894	0.0079	0.02271	40,000	0.14	0.5	12.78
HS [12]	3.870	0.0244	0.0582	40,000	0.15	0.5	4.5
TLBO [14]	4.004	0.0003	0.0611	40,000	0.15	0.5	4.5

Table 6 Result of two sample paired *t*-test between GSA and FWA algorithms.

Optimization method	Response	*N*	Optimal value	Mean	Standard deviation	Standard error
GSA	MRR (mm^3/s)	20	3.9600	3.9395	0.0118	0.0026
FWA	MRR (mm^3/s)	20	4.0061	4.0045	0.0015	0.0003

Estimate for average difference = 0.0650, 95% CI for mean difference = (0.0594, 0.0705), *t*-test of mean difference = 0 (vs not = 0), *t*-value = 24.49, *p*-value = 0.000.

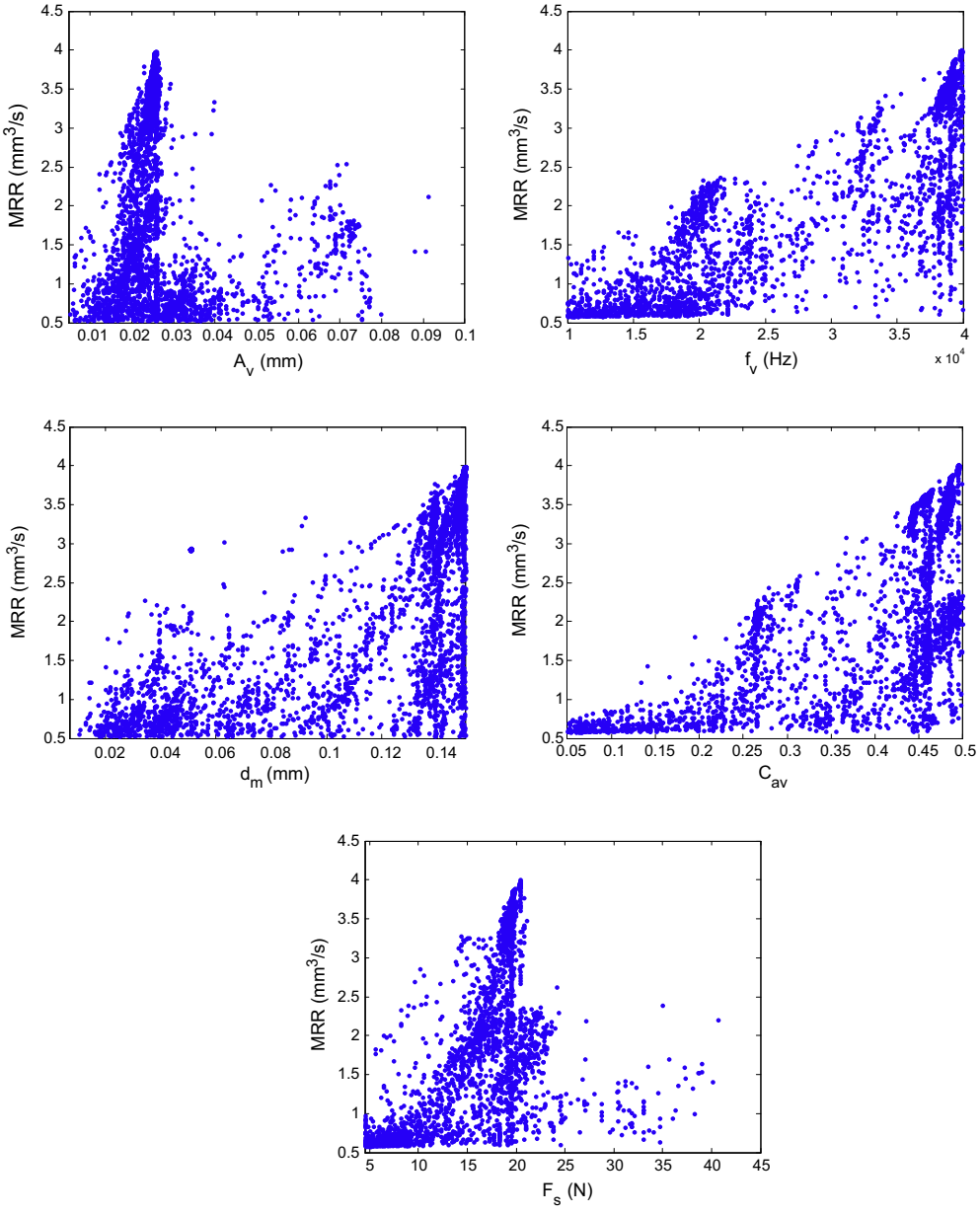

Figure 13 Effects of different USM process parameters on MRR.

CPU time trends will however remain almost the same. The effects of algorithm parameters for these two algorithms were not addressed by the past researchers [28,33]. They simply suggested a combination of those parameters without any scientific justification.

6.1. GSA algorithm

From the pseudo-code of GSA, as given in Section 3, it is clear that its optimization performance will be influenced by population size (*N*), maximum number of iterations (*T*), initial value

Table 7 Effects of population size on MRR and CPU time at varying values of α.[a]

N	$\alpha = 5$				$\alpha = 20$			
	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.
10	0.1807	0.00425	0.7878	0.00719	0.17612	0.00548	0.7722	0.00239
20	0.1824	0.00406	1.2556	0.00518	0.17516	0.00668	1.2328	0.00804
30	0.1868	0.00282	1.9066	0.00537	0.17888	0.00707	1.9234	0.02858
40	0.1865	0.00129	2.8136	0.01873	0.17410	0.00373	2.8068	0.00691
50	0.1873	6.02E−04	3.9350	0.02215	0.17922	0.00576	3.9394	0.00658

[a] Hold values: $G_0 = 100$ and maximum number of iterations = 500.

Table 8 Effects of number of iterations on MRR and CPU time at different values of N.[a]

Max. iterations	$N = 10$				$N = 20$			
	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.
200	0.17132	0.00803	0.5876	0.00288	0.17204	0.00676	0.7690	0.00596
400	0.17546	0.00873	0.7082	0.00445	0.18408	0.00334	1.0812	0.00476
600	0.17656	0.00273	0.8314	0.00577	0.18552	0.00325	1.3920	0.01086

[a] Hold values: $G_0 = 25$ and $\alpha = 5$.

Table 9 Effects of G_0 value on MRR and CPU time.[a]

G_0	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.
25	0.17965	0.00617	1.2425	0.00453
50	0.17206	0.00857	1.5394	0.27711
75	0.16764	0.01222	1.2400	0.00822
100	0.17516	0.00668	1.2328	0.00804

[a] Hold values: $N = 20$, maximum number of iterations = 500 and $\alpha = 20$.

Table 10 Effects of the value of α on MRR and CPU time.[a]

α	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.
5	0.18958	0.00105	1.2338	0.00311
10	0.18310	0.00428	1.2374	0.00744
15	0.17818	0.01022	1.2452	0.01223
20	0.17965	0.00617	1.2425	0.00453
25	0.16590	0.01162	1.2352	0.00715

[a] Hold values: $N = 20$, maximum number of iterations = 500 and $G_0 = 25$.

Table 11 Effects of number of fireworks on MRR and CPU time.[a]

n	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.
5	0.18676	0.00193	4.4338	0.02284
10	0.18544	0.00148	6.3664	0.05430
15	0.18778	7.95E−04	8.9032	0.07527

[a] Hold values: Maximum number of iterations = 150, $a = 0.04$, $b = 0.8$, $m = 50$.

Table 12 Impacts of total number of generated sparks on MRR and CPU time.[a]

m	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.
25	0.18676	0.00521	1.5874	0.08099
50	0.18460	0.00193	4.4338	0.02284
75	0.19370	0.00491	8.8194	0.00811
100	0.20001	0.00317	14.6522	0.10938

[a] Hold values: Maximum number of iterations = 150, $a = 0.04$, $b = 0.8$, $n = 5$.

Table 13 Effects of number of iterations on MRR and CPU time.[a]

Max. iterations	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.
50	0.16244	0.00520	0.5204	0.00688
100	0.17882	0.00246	1.0398	0.00904
150	0.18676	0.00521	1.5874	0.08099
300	0.19266	0.00667	3.1102	0.03408
500	0.18802	8.81E−04	5.1388	0.0154
700	0.18778	8.58E−04	7.1686	0.02948

[a] Hold values: $a = 0.04$, $b = 0.8$, $n = 5$, $m = 25$.

Table 14 Effects of value of a on MRR and CPU time.[a]

a	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.
0.01	0.17508	0.00555	0.9730	0.00632
0.04	0.17882	0.00246	1.0398	0.00904
0.10	0.17574	0.00574	1.2410	0.01198
0.50	0.18536	0.00667	4.2762	0.03343

[a] Hold values: Maximum number of iterations = 100, $b = 0.8$, $n = 5$, $m = 25$.

Table 15 Effects of value of b on MRR and CPU time.[a]

b	Mean MRR_{max} (g/min)	St. dev.	Mean CPU time (s)	St. dev.
0.1	0.17226	0.00572	0.40704	0.00692
0.4	0.19174	0.00887	0.97758	0.01900
0.8	0.17882	0.00246	1.03980	0.00904
1.5	0.18362	0.00795	1.03300	0.00652
2.5	0.19588	0.00690	1.05874	0.00727

[a] Hold values: Maximum number of iterations = 100, a = 0.04, n = 5, m = 25.

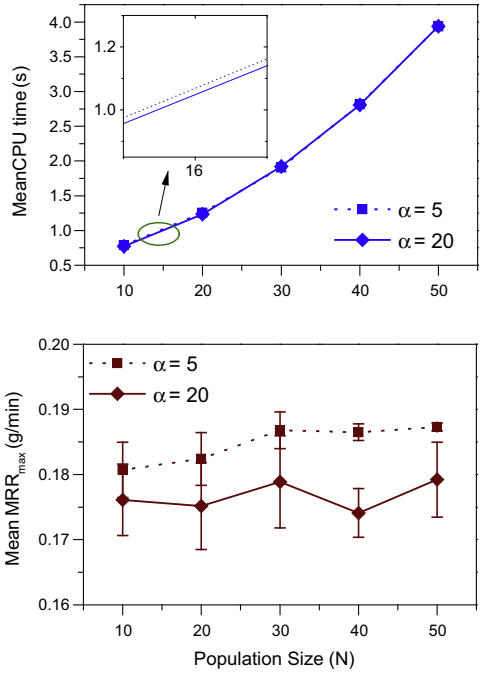

Figure 14 Effects of population size on MRR and CPU time at varying values of α.

of G (G_0) and value of constant α. Table 7 and Fig. 14 first show the effects of population size on mean achieved MRR_{max} and average CPU time for varying values of α (i.e. α = 5 and α = 20). It is observed that with increasing values of population size, mean CPU time also goes on increasing almost linearly for both the values of α. On the other hand, with the increase in population size, higher mean MRR_{max} values are achieved for a lower value of α at 5. But, for both the values of α, population size has almost no effect on the value of mean MRR_{max}. In Table 8 and Fig. 15, the influences of number of iterations on mean MRR_{max} and mean CPU time at changing values of population size are investigated. With the increment in number of iterations, both mean CPU time and mean MRR_{max} increase almost linearly, and the increase in their values is higher for larger value of population size (i.e. N = 20). The effects of the initial value of G (G_0) on mean CPU time and mean MRR_{max} are demonstrated in Table 9 and Fig. 16. It is clear that the initial value of G (G_0) has almost no effect on mean CPU time and mean MRR_{max}. Lastly, the influences of the value of α on MRR and CPU time are studied in Table 10 and Fig. 17. From there, it can be visualized that the value of mean CPU time remains almost unaffected with the changing values of α. On the other hand, the MRR_{max} goes

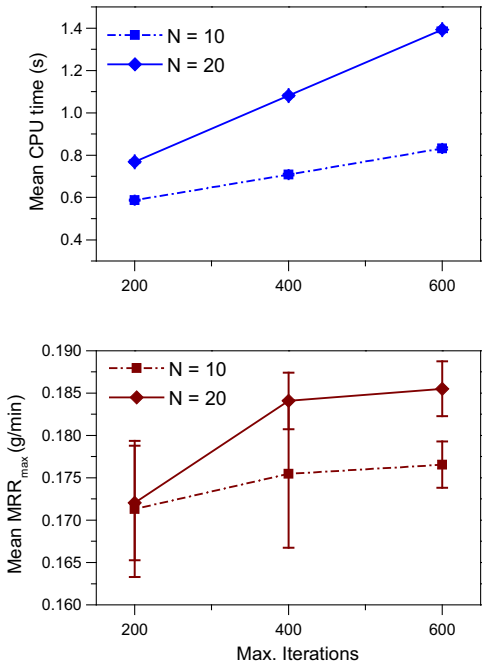

Figure 15 Effects of number of iterations on MRR and CPU time at changing values of N.

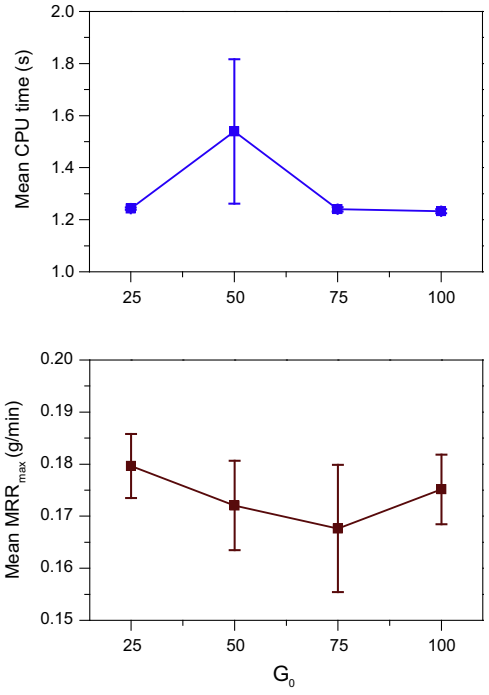

Figure 16 Effects of G_0 value on MRR and CPU time.

on decreasing with the increasing values of α. Quite interestingly, this observation can also be re-confirmed from a different perspective through Fig. 14.

6.2. FWA algorithm

The optimization performance and computational speed of FWA algorithm will also be controlled by various parameters

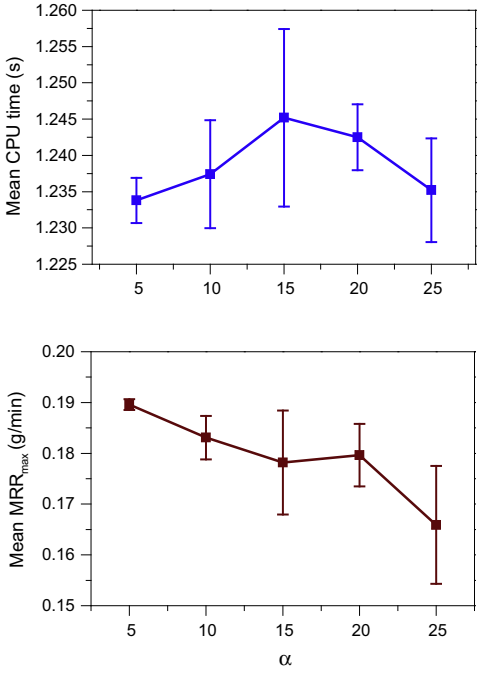

Figure 17 Effects of α value on MRR and CPU time.

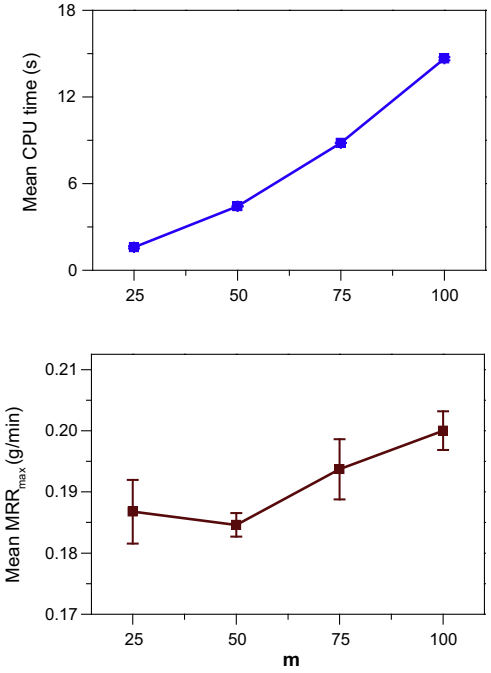

Figure 19 Effects of total number of generated sparks by *n* fireworks on MRR and CPU time.

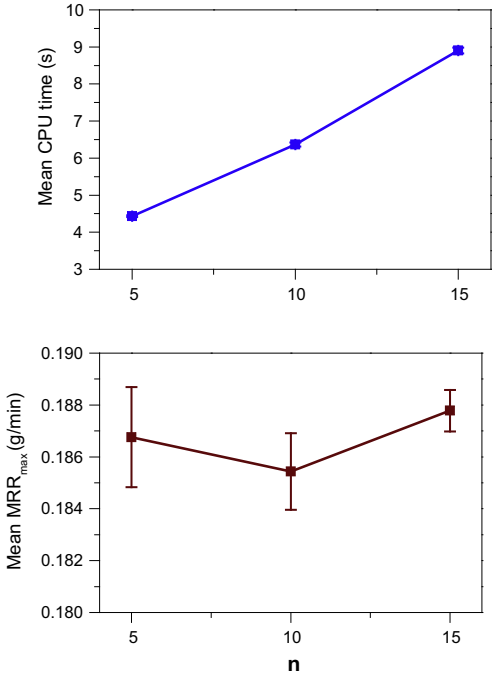

Figure 18 Effects of number of fireworks on MRR and CPU time.

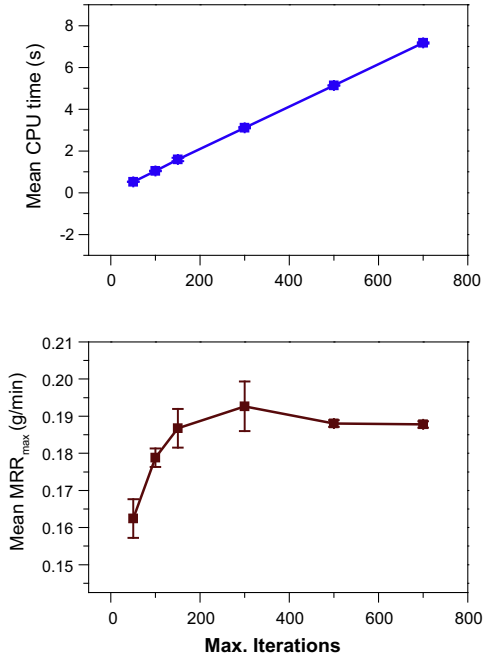

Figure 20 Effects of number of iterations on MRR and CPU time.

such as: number of fireworks (*n*), total number of sparks generated by *n* fireworks (*m*), maximum number of iterations, and values of two constants, *a* and *b*. The effects of number of initial locations/number of fireworks on mean CPU time and mean MRR_{max} are studied in Table 11 and Fig. 18. It can be shown that mean CPU time goes on increasing linearly with the increase in the value of number of fireworks. But with the increasing values of number of fireworks, the mean MRR_{max} remains almost unaltered. Table 12 and Fig. 19 exhibit the impacts of the total number of sparks generated by *n* fireworks on mean CPU time and mean MRR_{max}. It is clear from these analyses that both mean CPU time and mean MRR_{max} increase almost linearly with the increment in the value of total number of generated sparks. Table 13 and

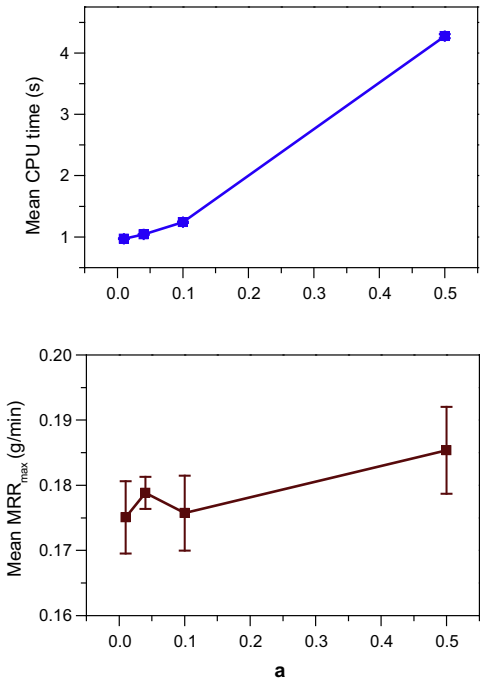

Figure 21 Effects of value of a on MRR and CPU time.

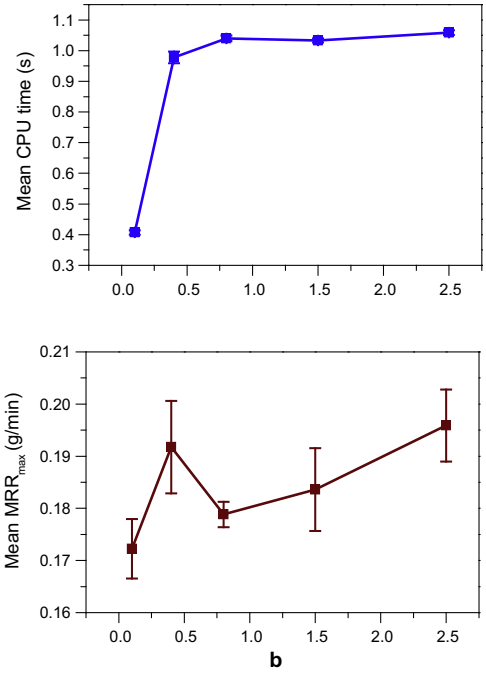

Figure 22 Effects of value of b on MRR and CPU time.

Fig. 20 demonstrate the effects of number of iterations of FWA algorithm on its performance with respect to mean CPU time and mean MRR_{max}. These analyses exhibit that, quite obviously, the mean CPU time goes on increasing linearly with the increase in the number of iterations for FWA algorithm. In case of maximum achieved MRR, it first increases with the increasing values of number of iterations and then at higher number of iterations, its value remains undisturbed. Lastly, the effects of two constants, a and b (see algorithm Snippet 1, Section 4) on mean CPU time and mean MRR_{max} are depicted in Tables 14 and 15 and Figs. 21 and 22. In both the cases, the mean CPU time increases with the increase in the values of a and b. On the other hand, there are marginal increments in achieved maximum MRR, when the values of both the constants are increased within the given boundaries.

7. Conclusions

In this paper, two almost unexplored non-conventional optimization techniques, i.e. GSA and FWA algorithms are applied for optimizing the responses of USM process. It is observed that the optimization performance of FWA technique is better than that of GSA and other popularly adopted population-based algorithms, although on an average, it takes slightly more computational time. It is also found that the optimization performances of these two algorithms are statistically different at 5% significance level. The study of investigating the effects of different algorithm-specific parameters on the achieved optimal solution and CPU time of these two algorithms will help in exploiting the best compromise performance from the considered algorithms. The derived optimal parametric combinations for USM process will guide the process engineers in achieving better machining performance, exploiting the full potential of those processes. The process engineers can now select the best combination of USM process parameters, not solely depending on the manufacturer's data or handbook data. The FWA technique can be applied as a global optimization tool for multi-response optimization of other non-traditional machining processes too.

References

[1] Thoe TB, Aspinwall DK, Wise MLH. Review on ultrasonic machining. Int J Mach Tools Manuf 1998;38(4):239–55.
[2] Singh R, Khamba JS. Ultrasonic machining of titanium and its alloys: a review. J Mater Process Technol 2006;173(2):125–35.
[3] Singh R, Khamba JS. Investigation for ultrasonic machining of titanium and its alloys. J Mater Process Technol 2007;183(2–3):363–7.
[4] Singh R, Khamba JS. Taguchi technique for modeling material removal rate in ultrasonic machining of titanium. Mater Sci Eng A 2007;460–461:365–9.
[5] Dvivedi A, Kumar AP. Surface quality evaluation in ultrasonic drilling through the Taguchi technique. Int J Adv Manuf Technol 2007;34(1–2):131–40.
[6] Jain NK, Jain VK, Deb K. Optimization of process parameters of mechanical type advanced machining processes using genetic algorithms. Int J Mach Tools Manuf 2007;47(6):900–19.
[7] Singh R, Khamba JS. Mathematical modelling of surface roughness in ultrasonic machining of titanium using Buckingham-Π approach: a review. Int J Abras Technol 2009;2(1):3–24.
[8] Jadoun RS, Kumar P, Mishra BK. Taguchi's optimization of process parameters for production accuracy in ultrasonic drilling of engineering ceramics. Prod Eng Res Develop 2009;3(3):243–53.
[9] Kumar V, Khamba JS. Parametric optimization of ultrasonic machining of Co-based super alloy using the Taguchi multi-objective approach. Prod Eng, Res Develop 2009;3(4–5):417–25.
[10] Kumar J, Khamba JS. Modeling the material removal rate in ultrasonic machining of titanium using dimensional analysis. Int J Adv Manuf Technol 2010;48(1–4):103–19.

[11] Rao RV, Pawar PJ, Davim JP. Optimisation of process parameters of mechanical type advanced machining processes using a simulated annealing algorithm. Int J Mater Prod Technol 2010;37(1/2):83–101.
[12] Rao RV, Pawar PJ, Davim JP. Parameter optimization of ultrasonic machining process using nontraditional optimization algorithms. Mater Manuf Process 2010;25(10):1120–30.
[13] Gauri SK, Chakravorty R, Chakraborty S. Optimization of correlated multiple responses of ultrasonic machining (USM) process. Int J Adv Manuf Technol 2011;53(9–12):1115–27.
[14] Rao RV, Kalyankar VD. Parameter optimization of modern machining processes using teaching-learning-based optimization algorithm. Eng Appl Artif Intell 2013;26(1):524–31.
[15] Lalchhuanvela H, Doloi B, Bhattacharyya B. Enabling and understanding ultrasonic machining of engineering ceramics using parametric analysis. Mater Manuf Process 2012;27(4):443–8.
[16] Lalchhuanvela H, Doloi B, Bhattacharyya B. Analysis on profile accuracy for ultrasonic machining of alumina ceramics. Int J Adv Manuf Technol 2013;67(5–8):1683–91.
[17] Das S, Doloi B, Bhattacharyya B. Optimisation of ultrasonic machining of zirconia bio-ceramics using genetic algorithm. Int J Manuf Technol Manage 2013;27(4/5/6):186–97.
[18] Chakravorty R, Gauri SK, Chakraborty S. Optimization of multiple responses of ultrasonic machining (USM) process: a comparative study. Int J Indust Eng Comput 2013;4(2):165–72.
[19] Yıldız AR. A novel hybrid immune algorithm for global optimization in design and manufacturing. Robot Computer-Integrated Manuf 2009;25(2):261–70.
[20] Sayadi MK, Ramezanian R, Ghaffari-Nasab N. A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. Int J Indust Eng Comput 2010;1(1):1–10.
[21] Yıldız AR. Optimization of cutting parameters in multi-pass turning using artificial bee colony-based approach. Inf Sci 2013;220:399–407.
[22] Mukherjee R, Goswami D, Chakraborty S. Parametric optimization of Nd:YAG laser beam machining process using artificial bee colony algorithm. J Indust Eng 2013. Article ID 570250.
[23] Yıldız AR. Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations. Appl Soft Comput 2013;13(3):1433–9.
[24] Goswami D, Chakraborty S. Differential search algorithm-based parametric optimization of electrochemical micromachining processes. Int J Indust Eng Comput 2014;5(1):41–54.
[25] Branke J, Hildebrandt T, Scholz-Reiter B. Hyper-heuristic evolution of dispatching rules: a comparison of rule representations. Evolut Comput 2014. http://dx.doi.org/10.1162/EVCO_a_00131.
[26] Yıldız AR. Cuckoo search algorithm for the selection of optimal machining parameters in milling operations. Int J Adv Manuf Technol 2013;64(1–4):55–61.
[27] Goswami D, Chakraborty S. Optimal process parameter selection in laser transmission welding by cuckoo search algorithm. In: Proc. of int. conf. on advanced engineering optimization through intelligent techniques, India; 2013. p. 40–4.
[28] Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009;179:2232–48.
[29] Khajehzadeh M, Eslami M. Gravitational search algorithm for optimization of retaining structures. Indian J Sci Technol 2012;5(1):1821–7.
[30] Chatterjee A, Mahanti GK, Pathak N. Comparative performance of gravitational search algorithm and modified particle swarm optimization algorithm for synthesis of thinned scanned concentric ring array antenna. Progr Electromag Res B 2010;25:331–48.
[31] Sahu RK, Panda S, Pradhan S. Optimal gravitational search algorithm for automatic generation control of interconnected power systems. Ain Shams Eng J 2014;5(3):721–33.
[32] Bahrololoum A, Nezamabadi-pour H, Bahrololoum H, Saeed M. A prototype classifier based on gravitational search algorithm. Appl Soft Comput 2012;12(2):819–25.
[33] Tan Y, Zhu Y. Fireworks algorithm for optimization. In: Tan Y, Shi Y, Tan KC, editors. Advances in swarm intelligence. Berlin: Springer-Verlag; 2010. p. 355–64.
[34] Pei Y, Zheng S, Tan Y, Takagi H. An empirical study on influence of approximation approaches on enhancing fireworks algorithm. In: Proc. of IEEE international conference on systems, man, and cybernetics, Korea; 2012. p. 1322–7.

Shankar Chakraborty is an Associate Professor in Production Engineering Department of Jadavpur University. He had been graduated in 1986 from University of Calcutta and had obtained his post-graduate degree from Jadavpur University in 1989. He had been awarded with Ph.D. (Engg.) degree from Jadavpur University in 1994. His research interests are in the areas of applications of different multi-criteria decision-making methods in manufacturing environment, control chart pattern recognition, and development of MIS and ERP systems for diverse engineering applications. He has guided several M.E. and Ph.D. (Engg.) theses, and published numerous papers in international journals. He is also a regular reviewer of several journals of international repute.

Debkalpa Goswami is currently a senior undergraduate student at the Department of Production Engineering, Jadavpur University, India. He is the recipient of the prestigious Erasmus Mundus Fellowship of the European Commission, and had been a guest researcher at the University of Bremen, Germany from Sept 2013 to June 2014. Prior to this, he interned at the Materials Characterization Division, Central Glass and Ceramic Research Institute, India in Summer 2013. After the completion of his bachelor's degree (c. Spring 2015), he is motivated in pursuing graduate studies, and subsequently a career in research. His diverse research interests include: complex optimization and decision making; heat and mass transfer; materials charactertization; and materials processing. He has already authored a few peer-reviewed articles in international journals.