Scientia Iranica E (2017) 24(2), 868–875

SCIENTIA IRANICA

Sharif University of Technology

Scientia Iranica

Transactions E: Industrial Engineering

www.scientiairanica.com

Research Note

A new optimization algorithm for parameter optimization of nano-finishing processes

R. Venkata Rao[a,*], D.P. Rai[a] and J. Balic[b]

a. *Department of Mechanical Engineering, S.V. National Institute of Technology, Surat, Gujarat 395007, India.*
b. *Faculty of Mechanical Engineering, University of Maribor, Slovenia.*

Received 4 September 2015; received in revised form 10 December 2015; accepted 7 March 2016

KEYWORDS
Nano-finishing processes;
Parameters optimization;
Jaya algorithm.

Abstract. Material removal rate and surface roughness are the most important performance measures in nano-finishing processes, and these are largely influenced by the process parameters. The optimum combination of process parameters for nano-finishing processes is determined in this paper using a recently proposed optimization algorithm, named as Jaya algorithm. The results show the better performance of the Jaya algorithm over the other approaches attempted by the previous researchers such as genetic algorithm and desirability function approach for the same nano-finishing processes. The results obtained by the Jaya algorithm are useful for the real production systems.

1. Introduction

Manufacturing industries are experiencing a profound need to manufacture products using materials with extraordinary properties, stringent design requirements, complex geometries, miniature features, improved quality and control, reduced loss of power due to friction, and increased longevity of the product by reducing the wear in sliding components. These requirements are taxing the engineers to manufacture parts with micro and nano-level surface finish. It is evident that the performance of any machining process is greatly influenced by its process parameters. Thus, researchers have recognized the need to investigate the effect of process parameters of the finishing processes on performance measures such as surface roughness, material removal rate, cutting forces, etc. and to keep abreast of the environmental footprint and sustainability of the process.

It is observed from the literature review [1-13] that researchers had proposed theoretical and empirical models for predicting the performance of nano-finishing processes. However, the combination of process parameters recommended by the previous researchers for the best performance of different nano-finishing processes was based on either experimental observations or statistical analysis of the experimental data. Moreover, predictive models developed by previous researchers are nonlinear and complex in nature. Therefore, there is a need to apply advanced optimization algorithms to solve the predictive models in order to obtain the optimal process parameter settings for nano-finishing processes.

Many population-based advanced optimization algorithms have been developed by researchers in the past two decades. Researchers have widely applied these heuristic algorithms to solve complex engineering optimization problems of continouous and discrete nature. However, these algorithms require common control parameters, like population size, number of generations, etc., for their working. Besides, the common control parameters, different algorithms require their own algorithm-specific parameters. For example, Genetic Algorithm (GA) uses mutation rate and crossover rate; Differential Evolution (DE) uses scaling factor

*. *Corresponding author.*
E-mail address: ravipudirao@gmail.com (R. Venkata Rao)

and cross-over rate; and Particle Swarm Optimization (PSO) algorithm uses inertia weight, social cognitive parameters, maximum velocity, etc. Improper tuning of algorithm-specific parameters either increases the computational effort or yields local optimal solution. In addition to the tuning of algorithm-specific parameters, the common control parameters need to be tuned which further enhances the effort.

Rao et al. [14] introduced the Teaching-Learning-Based Optimization (TLBO) algorithm which does not require any algorithm-specific parameters. The TLBO algorithm has gained wide acceptance among the optimization researchers [15,16]. Keeping in view of the success of the TLBO algorithm, another algorithm-specific parameterless algorithm has been proposed very recently by Rao [17]. However, the proposed new algorithm has only one phase, and it is comparatively simpler to apply. The algorithm is named as "Jaya algorithm", and it has proved its effectiveness in solving a number of constrained and unconstrained benchmark functions [17]. The Jaya algorithm is described in the following section.

2. The Jaya algorithm

Let $f(x)$ be the objective function to be minimized (or maximized). At any iteration, i, assume that there are 'm' number of design variables, and 'n' number of candidate solutions (i.e., population size, $k = 1, 2, ..., n$). Let the *best* candidate *best* obtain the best value of $f(x)$ (i.e., $f(x)_{best}$) in the entire candidate solutions, and let the *worst* candidate *worst* obtain the worst value of $f(x)$ (i.e., $f(x)_{worst}$) in the entire candidate solutions. If $X_{j,k,i}$ is the value of jth variable for kth candidate during ith iteration, then this value is modified as per the following equation:

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\left(X_{j,best,i} - |X_{j,k,i}|\right) - r_{2,j,i}\left(X_{j,worst,i} - |X_{j,k,i}|\right), \quad (1)$$

where $X_{j,best,i}$ is the value of the variable j for the *best* candidate, and $X_{j,worst,i}$ is the value of the variable j for the *worst* candidate. $X'_{j,k,i}$ is the updated value of $X_{j,k,i}$; $r_{1,j,i}$ and $r_{2,j,i}$ are the two random numbers for the jth variable during ith iteration in the range [0,1]. The term "$r_{1,j,i}\ (X_{j,best,i} - |X_{j,k,i}|)$" indicates the tendency of the solution to move closer to the best solution; the term "$-r_{2,j,i}\ (X_{j,worst,i} - |X_{j,k,i}|)$" indicates the tendency of the solution to avoid the worst solution. $X'_{j,k,i}$ is accepted if it gives better function value. All the accepted function values at the end of iteration are maintained, and these values become the input to the next iteration. The random numbers, r_1 and r_2, ensure good exploration of the search space. The absolute value of the candidate solution ($|X_{j,k,i}|$) considered in Eq. (1) further enhances the exploration ability of the algorithm. Figure 1 shows the flowchart of the Jaya algorithm. More details of Jaya algorithm are available at https://sites.google.com/site/jayaalgorithm/.

Now, in order to distinguish the working and to highlight the merits of Jaya algorithm as compared

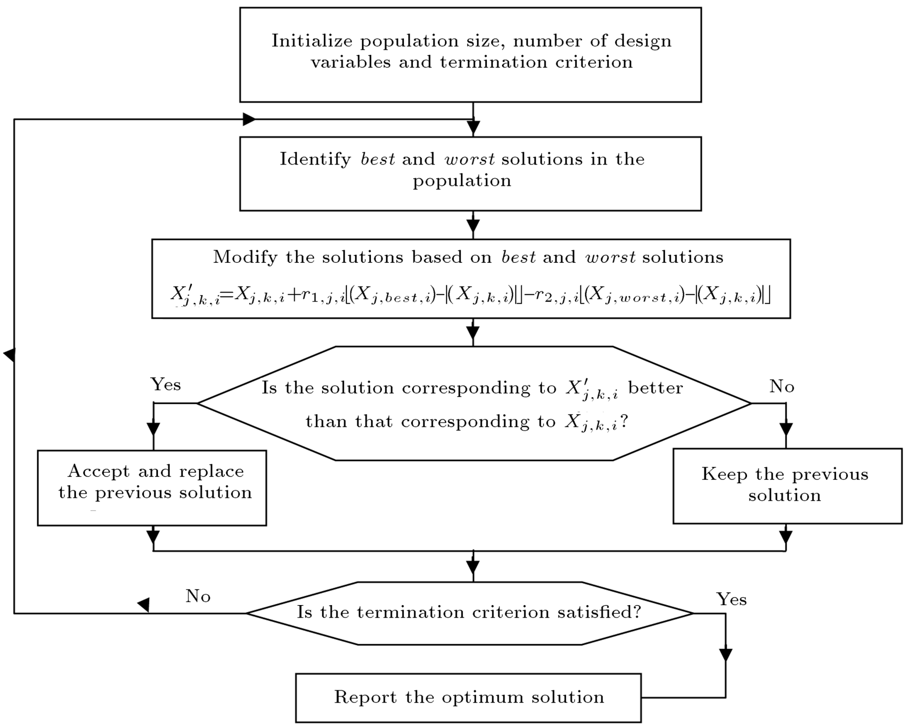


Figure 1. Flow chart for the Jaya algorithm.

to other well-known optimization algorithms, such as DE, ABC, and PSO, a brief discussion is provided as follows.

In DE algorithm, a solution is modified in two phases, i.e. the mutation and the crossover phases. However, the Jaya algorithm involves only one phase which makes it simpler to apply compared to DE. The working of DE is governed by two important parameters, i.e. scaling factor and cross-over rate. However, the Jaya algorithm requires no such algorithm-specific parameters. In the mutation phase of DE, a vector difference of randomly chosen vectors (solutions) is added to a third vector in order to generate a new vector (solution). However, in the case of Jaya algorithm, difference between the absolute values of a solution and the *best* and the *worst* solutions is found and scaled using a random number in the range [0,1], and the obtained value is added to the old value of the solution in order to generate the new solution.

ABC algorithm is inspired by the foraging behavioral patterns of honeybees. The ABC algorithm involves three phases, i.e. employed bee phase, onlooker bee, and abandoned food source phases. However, the Jaya algorithm requires only one phase, making it much simpler to apply than the ABC algorithm. In the employed bee phase and the onlooker bee phase, a candidate solution is updated by adding the scaled difference between the candidate solution and its neighbor to the initial value of the candidate solution. A random number in the range [-1,1] is used as the scaling factor. However, in the case of Jaya algorithm, difference between a candidate solution and the *best* and the *worst* solutions is found and scaled using a random number in the range [0,1], and the obtained value is added to the old value of the solution in order to generate the new solution. Furthermore, the *best* and the *worst* may not be necessarily the neighbors of the candidate solution to be updated.

The PSO algorithm simulates the social behavior of organisms by using the physical movements of the individuals in the swarm. The velocity of a particle (candidate solution) in the swarm is updated based on the personal best (*pbest*) solution of a particle and the global best (*gbest*) solution of the whole swarm, i.e. the best solution found so far in all the iterations. In addition to *pbest* and *gbest*, the velocity updating depends upon tuning of algorithm-specific parameters such as inertia weight and learning factors c_1 and c_2. The updated velocity is then added to the initial position of the particle in the swarm in order to obtain the new position of the particle in the swarm. Therefore, updating a solution requires execution of two separate equations in PSO. On the other hand, the Jaya algorithm does not require tuning of any algorithm-specific parameters for its working, and the solution is updated using a single equation based on the best solution (*best*) and the worst solution (*worst*) found in the current iteration. The significant difference between the working of PSO and Jaya algorithm is that the PSO algorithm does not consider the effect of the worst solution while updating a solution. Also, it is worthy noting that the best in the case of Jaya algorithm is not the best solution found so far in all the iterations, rather it is the best solution found only in the current iteration.

Therefore, it can be concluded from the above discussion that the Jaya algorithm is a new optimization algorithm. It is a simple, free from tuning of algorithm specific parameters and is a powerful algorithm for solving the engineering optimization problems.

3. Examples

3.1. Optimization of process parameters of AFM

Abrasive Flow Machining (AFM) process is used to finish difficult-to-reach surfaces by flowing abrasive-laden viscoelastic polymer over them. The objective of this work is to maximize the Material Removal Rate 'MRR' (mg/min) in AFM process while obeying the surface roughness constraint.

The optimization problem formulated in this work is based on the empirical models developed by Jain and Jain [2] for MRR and Ra in AFM process. The process parameters considered are: media flow 'v' (cm/min), percentage concentration of abrasives 'c', abrasive mesh size 'd', and number of cycles 'n'. The objective functions, process parameters, and process parameter bounds considered in this work are same as those considered by Jain and Jain [2].

3.1.1. Objective functions

The objective function is expressed by Eq. (2):

$$\text{Maximize MRR} = 5.285E - 7v^{1.6469}c^{3.0776}\, d^{-0.9371}n^{-0.1893}. \tag{2}$$

3.1.2. Constraint

The constraint on surface roughness is expressed as:

$$Ra \leq Ra_{\max}, \tag{3}$$

where Ra is the surface roughness in μm given by Eq. (4), and $Ra_{\max}$ is the maximum allowable value of surface roughness:

$$Ra = 282751.0v^{-1.8221}c^{-1.3222}d^{0.1368}n^{-0.2258}. \tag{4}$$

3.1.3. Parameter bounds

The bounds on the process parameters are expressed as:

$$40 \leq v \leq 85, \tag{5}$$

$$33 \leq v \leq 45, \tag{6}$$

$$100 \leq d \leq 240, \tag{7}$$

$$20 \leq n \leq 120, \tag{8}$$

Jain and Jain [2] applied GA to solve the optimization problem considering a population size equal to 50, maximum number of generations equal to 200 (i.e., maximum number of function evaluations equal to 10000), total string length equal to 40, crossover probability equal to 0.8, and mutation probability equal to 0.01. Now, the same problem is solved using the Jaya algorithm in order to see whether or not improvement in the results can be achieved. For the purpose of fair comparison of results, the maximum number of function evaluations considered by Jaya algorithm is maintained as 10000. For this purpose, a population size of 10 and number of generations equal to 1000 are chosen for the Jaya algorithm after conducting several trials with different values of population sizes. A computer code for Jaya algorithm is developed in MATLAB r2009a. A computer system with a 2.93GHz processor and 4 GB random access memory is used for execution of the program.

The results obtained using Jaya algorithm for different values of maximum allowable surface roughness (i.e., $Ra_{\max} = 0.7$, 0.6, 0.5, and 0.4) are reported in Table 1. The comparison of results obtained using Jaya algorithm and GA is shown in Table 2. The values of MRR provided by Jaya algorithm are 6.41%, 6.27%, and 5.68% which are higher than the values of MRR provided by GA for $Ra_{\max} = 0.7$, 0.6, and 0.5, respectively. It can be observed from Figure 2(a)-(d) that the convergence graph for Jaya algorithm rises continuously until it reaches the maximum value of MRR, and then remains stable. This shows that the Jaya algorithm is robust and does not get trapped in local optima. The Jaya algorithm has shown a better performance in terms of convergence rate and objective function value as compared to GA.

3.2. *Optimization of process parameters of R-AFF process*

In the Rotational Abrasive Flow Finishing (R-AFF) process, in addition to the back and forth motions of the abrasive medium, a rotary motion is given to the workpiece in order to enhance the performance of the process. The objective of this work is to maximize the improvement in surface roughness 'ΔRa' (μm) in R-AFF process.

The optimization problem formulated in this work is based on the empirical models developed by Sankar et al. [6] for ΔRa in R-AFF process. The process parameters considered are process oil %wt in the medium 'M', extrusion pressure 'P' (MPa), number of cycles 'N', and rotational speed 'R' (rpm). Separate mathematical models for 'ΔRa' were developed considering three different work-piece materials such as Al alloy, Al alloy/SiC (10%), and Al alloy/SiC (15%). The objective functions, process parameters, and process parameter bounds considered in this work are same as those considered by Sankar et al. [6].

3.2.1. *Objective functions*

The objective functions are expressed by Eqs. (9)-(11):

$$\begin{aligned}\text{Maximize } \Delta Ra_{\text{Al alloy}} &= 0.098M + 0.875P + 0.002N \\ &+ 0.05R - 0.006M^2 - 0.068P^2 - 9.6E \\ &- 7N^2 - 0.002R^2,\end{aligned} \tag{9}$$

Table 1. Optimum combination of process parameters for AFM process obtained using Jaya algorithm.

Sr. no.	$Ra_{\max}$ (μm)	v	c	D	n	Ra (μm)	MRR (mg/min)	Computational time (s)
1	0.7	85	45	100	20	0.5367	0.738	0.160
2	0.6	85	45	100	20	0.5367	0.738	0.158
3	0.5	85	45	100	27.3758	0.5	0.6954	0.241
4	0.4	85	45	100	73.5436	0.4	0.5768	0.206

Table 2. Comparison of results obtained using Jaya algorithm and GA [2] for AFM process.

Sr. no.	$Ra_{\max}$	GA [2]		Jaya algorithm		% Increase in MRR
		Ra (μm)	MRR (mg/min)	Ra (μm)	MRR (mg/min)	
1	0.7	0.5433	0.6935	0.5367	**0.738**	6.41
2	0.6	0.5113	0.6944	0.5367	**0.738**	6.27
3	0.5	0.4812	0.6580	0.5	**0.6954**	5.68
4	0.4	0.4171*	0.5803	0.4	**0.5768**	—

* Constraint violated by GA [2].

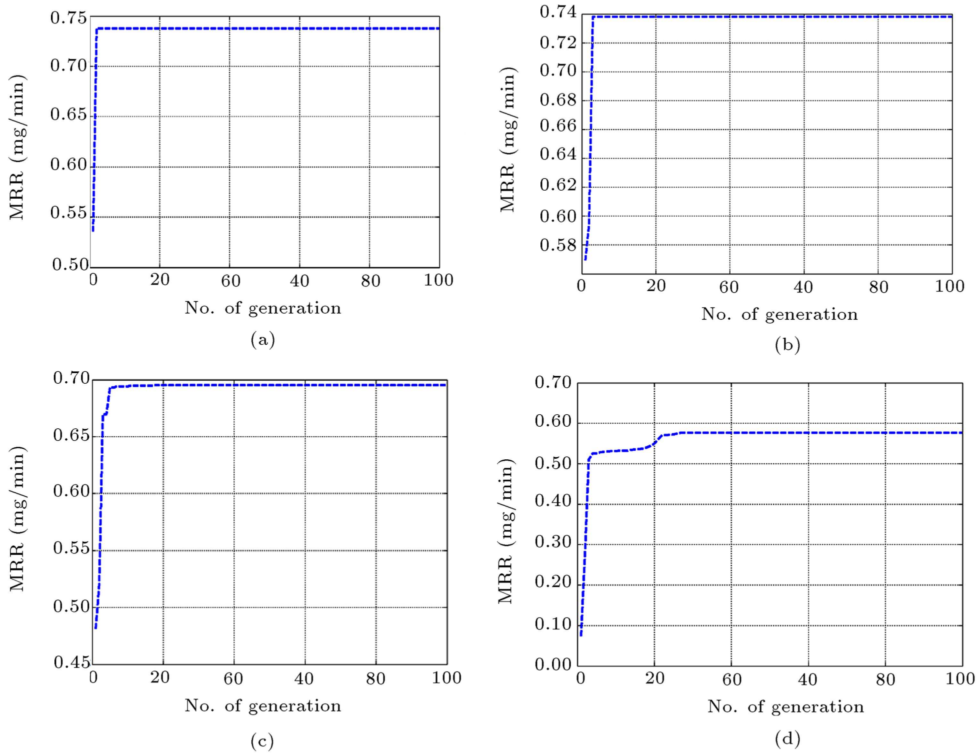


Figure 2. (a) Convergence graph of Jaya algorithm for AFM ($Ra_{\max} < 0.7$). (b) Convergence graph of Jaya algorithm for AFM ($Ra_{\max} < 0.6$). (c) Convergence graph of Jaya algorithm for AFM ($Ra_{\max} < 0.5$). (d) Convergence graph of Jaya algorithm for AFM ($Ra_{\max} < 0.4$).

Table 3. Optimum combination of process parameters for maximization of $\Delta Ra_{\text{Al alloy}}$, $\Delta Ra_{\text{Al/SiC(10\%)}}$, $\Delta Ra_{\text{Al/SiC(15\%)}}$ obtained using the Jaya algorithm for R-AFF process.

Sr. no.	M	P	N	R	$\Delta Ra_{\text{Al alloy}}$	$\Delta Ra_{\text{Al alloy/SiC(10\%)}}$	$\Delta Ra_{\text{Al alloy/SiC(15\%)}}$	Computational time (s)
1	8.1671	6.434	727.998	10	3.5281	—	—	0.697
2	9.833	6.2009	727.997	7.75	—	3.2835	—	0.51471
3	10.978	6.7163	728.000	10.0	—	—	3.3732	0.54269

$$\text{Maximize } \Delta Ra_{\text{Al alloy/SiC(10\%)}} = 0.118M + 0.831P + 0.001N + 0.031R - 0.006M^2 - 0.067P^2 - 1.2E - 6N^2 - 0.002R^2, \tag{10}$$

$$\text{Maximize } \Delta Ra_{\text{Al alloy/SiC(15\%)}} = 0.101M + 0.767P + 0.002N + 0.043R - 0.0046M^2 - 0.0571P^2 - 8.28E - 7N^2 - 0.002R^2. \tag{11}$$

3.2.2. Parameter bounds

The bounds on the process parameters are expressed as:

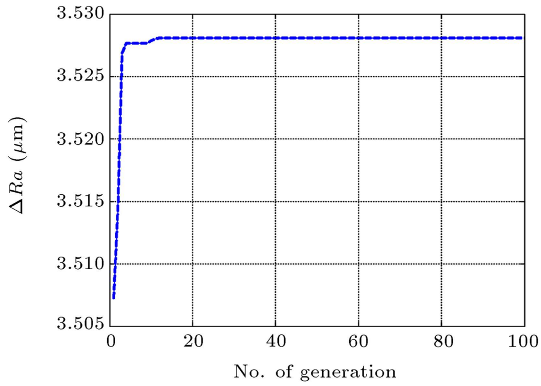
$$7 \leq M \leq 13, \tag{12}$$

$$5.35 \leq P \leq 7.15, \tag{13}$$

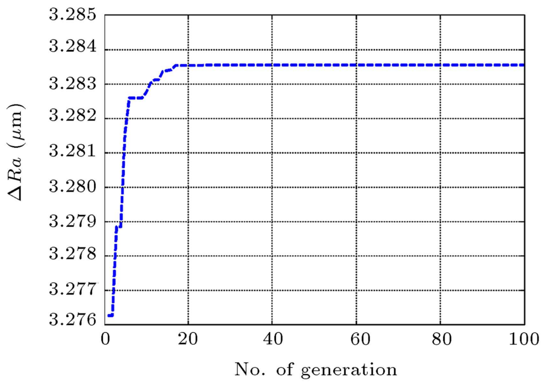
$$372 \leq N \leq 728, \tag{14}$$

$$2 \leq R \leq 10. \tag{15}$$

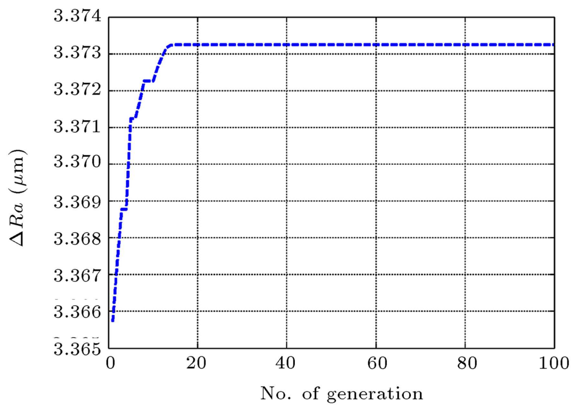
The optimum combination of process parameters obtained using the Jaya algorithm for maximization of $\Delta Ra_{\text{Al alloy}}$, $\Delta Ra_{\text{Al alloy/SiC(10\%)}}$, and $\Delta Ra_{\text{Al alloy/SiC(15\%)}}$ in R-AFF process is reported in Table 3. Figure 3(a)-(c) shows the convergence graphs of the Jaya algorithm. The number of function evaluations required by the Jaya algorithm to obtain the maximum value of $\Delta Ra_{\text{Al alloy}}$, $\Delta Ra_{\text{Al alloy/SiC(10\%)}}$, and $\Delta Ra_{\text{Al alloy/SiC(15\%)}}$ are 120, 170, and 130, respectively. Figure 3(a)-(c) present the convergence graph for the Jaya algorithm.



(a)



(b)



(c)

Figure 3. (a) Convergence graph of Jaya algorithm for R-AFF process (Al alloy). (b) Convergence graph of Jaya algorithm for R-AFF process (Al alloy SiC10 %). (c) Convergence graph of Jaya algorithm for R-AFF process (Al alloy SiC 15 %).

3.3. Optimization of process parameters of R-MRAFF

In the Rotational Magnetorheological Abrasive Flow Finishing (R-MRAFF) process, the objective is to maximize the percentage improvement in surface roughness '$\%\Delta Ra$' of the workpiece. The optimization problem formulated in this work is based on the empirical models developed by Das et al. [8] for $\%\Delta Ra$. They considered the process parameters to be: hydraulic extrusion pressure 'P' (bar), number of finishing cycles 'N', rotational speed of magnet 'S' (RPM), and volume ratio of CIP/SiC 'R'. The objective functions, process parameters, and process parameter bounds considered in this work are same as those considered by Das et al. [8].

3.3.1. Objective functions

The objective functions are expressed by Eqs. (16) and (17):

$$\begin{aligned}\text{Maximize}\%\Delta Ra_{SS} &= -403 + 17.66P + 0.2N \\ &+ 0.83S + 10.38R + 1.89E - 4PN \\ &- 1.67E - 3PS - 3.56E - 3PR \\ &- 6.53E - 5NS - 7E - 3NR + 8.46E \\ &- 3SR - 0.23P^2 - 1.39E - 4N^2 \\ &- 5.56E - 3S^2 - 1.35R^2, \end{aligned} \tag{16}$$

$$\begin{aligned}\text{Maximize}\%\Delta Ra_{BR} &= -912.47 + 39.27P + 0.41N \\ &+ 1.67S + 28.49R - 4.02E - 3PN \\ &- 0.01PS - 0.08PR + 3.93E - 4NS \\ &- 2.76E - 3NR - 0.10SR - 0.46P^2 \\ &-2.07E - 4N^2 - 9.26E - 3S^2 - 3.83R^2. \end{aligned} \tag{17}$$

3.3.2. Process parameters

The bounds on the process parameters are expressed as:

$$32.5 \leq P \leq 42.5, \tag{18}$$

$$400 \leq N \leq 800, \tag{19}$$

$$20 \leq S \leq 100, \tag{20}$$

$$0.34 \leq R \leq 4. \tag{21}$$

Das et al. [8] applied desirability function approach to determine the optimum combination of process parameters for R-MRAFF process. The maximum value of $\%\Delta Ra_{SS}$ and $\%\Delta Ra_{BR}$ obtained by Das et al. [8] using desirability function approach is reported in Tables 4 and 5, respectively. Now, the same problem is solved using Jaya algorithm by considering a population size of 10 and maximum number of function evaluations as 1000. The optimum value of $\%\Delta Ra_{SS}$ and $\%\Delta Ra_{BR}$ obtained by Jaya algorithm along with the optimum combination of process parameters of R-MAFF process is also reported in Tables 4 and 5, respectively. The Jaya algorithm achieved a better value of $\%\Delta Ra_{SS}$

Table 4. Optimum combination of process parameters for maximization of $\%\Delta Ra_{SS}$ obtained using Jaya algorithm and desirability function approach [8] for R-MRAFF process.

Sr. no.	Algorithm	P (bar)	N	S (RPM)	R	$\%\Delta Ra_{SS}$	Computational time (s)
1	Desirability function approach	38.92	660	6	2.3	45.27 (42.73*)	—
2	Jaya algorithm	38.4084	673.03	66.637	2.2577	**42.822**	0.415

* Corrected value

Table 5. Optimum combination of process parameters for maximization of $\%\Delta Ra_{BR}$ obtained using Jaya algorithm and desirability function approach [8] for R-MRAFF process.

Sr. no.	Algorithm	P (bar)	N	S (RPM)	R	$\%\Delta Ra_{BR}$	Computational time (s)
1	Desirability function approach	38.03	690	73	2.22	70.95 (76.319*)	—
2	Jaya algorithm	38.801	667.55	71.8568	2.135	**76.6582**	0.280

*Corrected value.

and $\%\Delta Ra_{BR}$ in 310 and 210 function evaluations, respectively, as compared to the values of $\%\Delta Ra_{SS}$ and $\%\Delta Ra_{BR}$ obtained using the desirability approach.

All the optimization problems formulated in this work are based on the mathematical models developed by previous researchers based on experimentation. The confirmation experiments for the developed mathematical models were also conducted by the previous researchers for processes such as AFM [2], R-AFF [6], and R-MRAFF [8]. In addition, the previous researchers had solved the optimization problems using GA and desirability function approach. Now, the same mathematical models are solved using Jaya algorithm, and the results obtained using Jaya algorithm are compared with the results obtained by the previous researchers. The previous researchers had considered the process parameters in their continuous form. Therefore, all the process parameters considered in this work are in their continuous form only. However, in actual practice, the values allowed by the machining process which are closer to the suggested optimum values may be considered.

4. Conclusions

- In the present work, the optimization problems of the three advanced finishing processes, i.e., AFM, R-AFF, and R-MRAFF, are solved using the newly proposed Jaya algorithm;
- The performance of the Jaya algorithm is studied in terms of convergence rate and accuracy of the solution. Compared to other advanced optimization methods, the Jaya algorithm does not require selection of algorithm-specific parameters, and this feature makes the Jaya algorithm applicable to real-life optimization problems, easily and effectively;
- In the case of AFM process, maximization of MRR is considered as the objective function, while the constraint is on the allowable value of surface roughness. The process parameter combination, as suggested by Jaya algorithm, increases the MRR by 6.41%, 6.27%, and 5.68% as compared to the MRR provided by GA for $Ra_{\max} = 0.7$, 0.6, and 0.5, respectively;
- In the case of R-AFF process, maximization of improvement in surface roughness is considered as the objective function. The Jaya algorithm obtained a maximum value of $\Delta Ra_{\text{Al alloy}}$, $\Delta Ra_{\text{Al alloy/SiC(10\%)}}$, and $\Delta Ra_{\text{Al alloy/SiC(15\%)}}$ in 120, 170, and 130 function evaluations, respectively, without getting trapped into local optima;
- In the case of R-MRAFF process, maximization of percentage improvement in surface roughness is considered as the objective function. The Jaya algorithm achieved a better value of $\%\Delta Ra_{SS}$ and $\%\Delta Ra_{BR}$ in 310 and 210 function evaluations, respectively, as compared to the values of $\%\Delta Ra_{SS}$ and $\%\Delta Ra_{BR}$ obtained by using the desirability approach;
- The results reported in this work show that the convergence accuracy and its speed are very high. The results obtained by Jaya algorithm are found to be better in terms of objective function values as compared to those obtained by using GA and desirability function approach and have also demonstrated the ability of Jaya algorithm to handle the constraints.

Acknowledgement

The authors are grateful to the Department of Science and Technology (DST), Ministry of Science & Technology of the Republic of India, and the Slovenian Research Agency (ARRS), Ministry of Education, Science and Sport of the Republic of Slovenia for providing the financial support for the project entitled "Optimization of Sustainable Manufacturing Processes using Advanced Techniques".

References

1. Rao, R.V. and Kalyankar, V.D. "Optimization of modern machining processes using advanced optimization techniques: a review", *Int. J. Adv. Manuf. Technol.*, **73**, pp. 1159-1188 (2014).
2. Jain, R.K. and Jain, V.K. "Optimum selection of machining conditions in abrasive flow machining using neural network", *J. Mater. Process Tech.*, **108**, pp. 62-67 (2000).
3. Jain, N.K., Jain, V.K. and Jha, S. "Parametric optimization of advanced fine-finishing processes", *Int. J. Adv. Manuf. Technol.*, **34**, pp. 1191-1213 (2007).
4. Girma, B., Joshi, S.S., Raghuram, M.V.G.S. and Balasubramaniam, R. "An experimental analysis of magnetic abrasives finishing of plane surfaces", *Mach. Sci. Technol.*, **10**, pp. 323-340 (2007).
5. Sidpara, A., Das, M. and Jain, V.K. "Rheological characterization of magnetorheological finishing fluid", *Mater. Manuf. Processes*, **24**, pp. 1467-1478 (2009).
6. Sankar, M.R., Jain, V.K. and Ramkumar, J. "Experimental investigations into rotating workpiece abrasive flow finishing", *Wear*, **267**, pp. 43-51 (2009).
7. Mali, H.S. and Manna, A. "Simulation of surface generated during abrasive flow finishing of Al/SiCp-MMC using neural networks", *Int. J. Adv. Manuf. Technol.*, **61**, pp. 1263-1268 (2012).
8. Das, M., Jain, V.K. and Ghoshdastidar, P.S. "Nano-finishing of flat workpieces using rotational-magnetorheological abrasive flow finishing (R-MRAFF) process", *Int J Adv. Manuf. Technol.*, **62**, pp. 405-420 (2012).
9. Sidpara, A. and Jain, V.K. "Theoretical analysis of forces in magnetorheological fluid based finishing process", *Int. J. Mech. Sci.*, **56**, pp. 50-59 (2012).
10. Sidpara, A. and Jain, V.K. "Nano-level finishing of single crystal silicon blank using magnetorheological finishing process", *Tribol. Int.*, **47**, pp. 159-166 (2012).
11. Jain, V.K., *Micromachining*, 2nd Edn., Narosa Publishing House, New Delhi (2014).
12. Brar, B.S., Walia, R.S. and Singh, V.P. "Electrochemical-aided abrasive flow machining (ECA2FM) process: a hybrid machining process", *Int. J. Adv. Manuf. Technol.*, **79**(1-4), pp. 329-342 (2015).
13. Sharma, A.K., Venkatesh, G., Rajesha, S. and Kumar, P. "Experimental investigations into ultrasonic-assisted abrasive flow machining (UAAFM) process", *Int. J. Adv. Manuf. Technol.*, **80**(1-4), pp. 477-493 (2015).
14. Rao, R.V., Savsani, V.J. and Vakharia, D.P. "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems", *Comput.-Aided Des.*, **43**, pp. 303-315 (2011).
15. Rao, R.V., *Teaching-Learning-Based Optimization (TLBO) Algorithm and Its Engineering Applications*, London, Springer-Verlag (2015).
16. Rao, R.V. "Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems", *Decision Science Letters*, **5**, pp. 1-30 (2016).
17. Rao, R.V. "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems", *Int. J. Industrial Eng. Comput.*, **7**(1), pp. 19-34 (2016).

Biographies

Ravipudi Venkata Rao is a Professor in the Department of Mechanical Engineering, S.V. National Institute of Technology, Surat, India. He has more than 25 years of teaching and research experience. He was deputed thrice to the Asian Institute of Technology, Bangkok, Thailand as a visiting Professor. He gained his BTech in 1988, MTech in 1991, PhD in 2002 and DSc in 2016. Rao's research interests include CAD/CAM, CIMS, advanced optimization algorithms, and fuzzy multiple attribute decision making methods. He has published about 300 research papers in national and international journals and conference proceedings and received national and international awards for best research work. He has been a reviewer to many national and international journals and on the editorial boards of few International journals. He had authored five books, and all these books were published by Springer Verlag, London, UK.

Dhiraj Praveen Rai is a research scholar and pursuing PhD programme at S.V. National Institute of Technology, Surat. His research interests include optimization of advanced manufacturing processes. He has 2 years of teaching experience and authored 2 research papers in international journals.

Joze Balic is a Professor of Manufacturing Technology and Systems in the Faculty of Mechanical Engineering, Maribor, Slovenia. He has authored more than 100 research papers published in reputed international journals and conference proceedings and received national and international awards for best research work. He has 6 patents and authored 10 text books for students. He is the founder and head of the laboratory for intelligent manufacturing systems. He is the editor, co-editor or member of publishing board of several Slovenian and international scientific journals.