

Multi-objective optimization of abrasive waterjet machining process using Jaya algorithm and PROMETHEE Method

R. Venkata Rao¹ · Dhiraj P. Rai¹ · J. Balic²

Received: 4 January 2017 / Accepted: 7 November 2017 / Published online: 24 November 2017
© Springer Science+Business Media, LLC, part of Springer Nature 2017

Abstract In this work, the process parameters optimization problems of abrasive waterjet machining process are solved using a recently proposed metaheuristic optimization algorithm named as Jaya algorithm and its posteriori version named as multi-objective Jaya (MO-Jaya) algorithm. The results of Jaya and MO-Jaya algorithms are compared with the results obtained by other well-known optimization algorithms such as simulated annealing, particle swarm optimization, firefly algorithm, cuckoo search algorithm, blackhole algorithm and bio-geography based optimization. A hypervolume performance metric is used to compare the results of MO-Jaya algorithm with the results of non-dominated sorting genetic algorithm and non-dominated sorting teaching–learning-based optimization algorithm. The results of Jaya and MO-Jaya algorithms are found to be better as compared to the other optimization algorithms. In addition, a multi-objective decision making method named PROMETHEE method is applied in this work in order to select a particular solution out-of-the multiple Pareto-optimal solutions provided by MO-Jaya algorithm which best suits the requirements of the process planer.

Keywords Multiobjective decision making · Abrasive waterjet machining process · Jaya algorithm · PROMETHEE · Hypervolume

Introduction

Abrasive waterjet machining (AWJM) process is a widely used modern machining process which is characterized by a number of input process parameters which significantly influence the process output parameters. The input parameters are classified into four groups namely hydraulic parameters, mixing and acceleration parameters, abrasive parameters and cutting parameters. The hydraulic parameters are: pump pressure, water-orifice diameter, water flow rate, etc. The mixing and acceleration parameters are: focus diameter and focus length, workpiece material composition and hardness. The abrasive parameters are: particle hardness, particle shape, particle size distribution, particle diameter and mass flow rate. The cutting parameters are: impact angle, stand-off distance, number of passes, traverse rate, etc.

The performance of AWJM process is measured in terms of the amount of material removed from the work-piece per unit time, smoothness of the cut surface, geometry of the cut in terms of kerf and taper angle, degree of accuracy achieved, satisfying the prescribed geometrical tolerances while machining, rate of wear of orifice and focusing tube. These performance measures are significantly influenced by the input process parameters. For instance, the depth of cut is mainly influenced by abrasive flowrate, waterjet pressure, traverse speed and stand-off distance. Increase in abrasive flow rate increases the number of abrasive particles striking on the workpiece which increases the depth of cut. The velocity of abrasive particles increases with the increase in water jet pressure which increases the depth of cut. At high stand-off distance the kinetic energy of the abrasive particles decreases which reduces the depth of cut.

The waterjet pressure, stand-off distance and traverse speed have a synergistic effect on kerf. At high waterjet pressure and low stand-off distance the diameter of jet is almost

✉ R. Venkata Rao
ravipudirao@gmail.com

¹ Department of Mechanical Engineering, S. V. National Institute of Technology, Surat, Gujarat 395007, India

² University of Maribor, Maribor, Slovenia

equal to the diameter of the orifice which results in low kerf. A low waterjet pressure and high stand-off distance flaring increase the diameter of jet which increases the kerf. Increase in the traverse speed decreases the jet interaction on a given area of material, which leads to material erosion by fewer abrasive particles. This reduces the depth of cut and increases the kerf. However, very low traverse speed increases the production time.

The surface roughness is mainly influenced by abrasive flow rate, waterjet pressure, stand-off distance and size of abrasive particles. As the stand-off distance increases the cutting ability of the abrasive particles decreases due to decrease in the kinetic energy and this increases the surface roughness. Increase in abrasive flow rate reduces the surface roughness because more number of impacts and cutting edges are available per unit area. As the size of the abrasives increases the surface roughness increases due to formation of large size crater on the surface of the work piece surface. Similarly, the other performance measures of AWJM process such as taper angle, geometry of cut, machining accuracy, etc. are influenced by the input process parameters.

Thus, in the presence of multiple input process parameters, selecting the best combination based only on the past experience and judgement is difficult for the process planner. The improper selection of input process parameters may result in poor material removal rate (MRR), low depth of cut, poor geometrical accuracy, poor surface finish and high wear rate of orifice and focusing tube. This may affect the quality of the machined workpiece, effectiveness and efficiency of the process. Therefore, selection of optimal combination of input parameters for AWJM process is a matter of concern for process engineers and researchers. Thus, there arises a need to apply optimization algorithms for determining the optimal combination of input process parameters.

The output process parameters of AWJM process are expressed by means of regression models which are developed by the researchers based on the data gathered by means of actual experimentation on AWJM process. These regression models are mathematical functions of input process parameters which help in precise mapping of the output process parameters corresponding to a particular set of input parameters. Therefore, the researchers have identified the strong need for developing such rigorous mathematical models which form a relationship between input process parameters and output process parameters. Furthermore, these models can be effectively used to form optimization models. These optimization models may be solved using a number of optimization techniques. However, traditional optimization techniques tend to provide a local optimum solution in scenarios where the size of the search space is large, a number of variables are to be handled, multiple objectives are to be achieved and a number of constraints are to be satisfied simultaneously. Therefore, researchers have devel-

oped a number of population based optimization algorithms known as the advanced optimization algorithms such as genetic algorithm (GA), particle swarm optimization (PSO) algorithm, simulated annealing (SA), cuckoo search (CS) algorithm, fire fly (FF) algorithm, black hole (BH) algorithm, biogeography based optimization (BBO) algorithm, cuckoo search algorithm (CSA), teaching–learning-based optimization (TLBO) algorithm, etc. These algorithms are found to perform well in such complex optimization scenarios which involve a number of variables, multiple objectives and a number of constraints (Rao and Kalyankar 2014). AWJM process is a multi-input process which requires optimization of multiple objectives. Therefore, it is observed that number of researchers have used the advanced modeling techniques such as response surface methodology (RSM), Taguchi's method, artificial neural networks (ANN), fuzzy logic, etc. in order to formulate the process models and advanced optimization algorithms to optimize the input process parameters of AWJM processes. Such works carried out by various researchers are summarized in Table 1.

It is observed that a number of nature-inspired population based optimization algorithms such as GA, SA, ABC, PSO, CS, COA, FF, BH, BBO, TLBO, etc. have been applied by the researchers for optimizing the input process parameters of AWJM process. However, the main limitation of these nature-inspired population based optimization algorithms is the need for setting of common control parameters like population and number of generations. In addition to common control parameters, these nature-inspired optimization algorithms require setting of their own algorithm-specific parameters. For instance, the GA requires tuning of algorithm specific parameters such as cross-over probability, mutation probability, selection operator, etc. The PSO algorithm requires tuning of algorithm-specific parameters such as inertia weight, cognitive and social coefficients. The SA algorithm requires tuning of initial temperature and cooling rate. The ABC algorithm requires algorithm-specific parameters such as number of scout bees, number of onlooker bees, number of employed bees, limit, etc. The CS algorithm requires tuning of levy flight constant and selection probability. The BBO algorithm requires tuning of algorithm-specific parameters such as habitat modification probability, mutation probability, maximum species count, maximum migration rates and maximum mutation rate. The FF algorithm requires tuning of algorithm-specific parameters such as attractiveness at source, randomness coefficient and absorptivity index.

The common control parameters and algorithm-specific parameters for any algorithm are required to be selected meticulously to achieve effective performance of the algorithm, to avoid convergence at local optima and to minimize the computational effort. In order to determine the best combination of common control parameters and algorithm-

Table 1 Summary of AWJM process parameters optimization review

| S. No. | Author/year | Objective(s) | Methodology |
|--------|-----------------------------|-----------------------|---|
| 1 | Jain et al. (2007) | Material removal rate | GA |
| 2 | Jegaraj and Babu (2007) | Surface roughness | Taguchi’s method, Neuro-fuzzy approach |
| | | Kerf width | |
| 3 | Srinivasu and Babu (2008) | Depth of cut | Neuro-genetic control strategy |
| 4 | Parikh and Lam (2009) | Material removal rate | ANN |
| | | Taper angle | |
| 5 | Kok et al. (2011) | Surface roughness | Genetic expression programming |
| 6 | Zain et al. (2011a) | Surface roughness | ANN-SA |
| 7 | Zain et al. (2011b) | Surface roughness | SA-GA |
| 8 | Vundavilli et al. (2012) | Depth of cut | Fuzzy-GA |
| 9 | Pawar and Rao (2013) | Material removal rate | TLBO |
| 10 | Aydin et al. (2014) | Kerf taper angle | ANN |
| 11 | Liu et al. (2014) | Depth of penetration | RSM |
| | | Surface roughness | |
| 12 | Yue et al. (2014) | Material removal rate | RSM |
| 13 | Yusup et al. (2014) | Surface roughness | ABC algorithm |
| 14 | Ergur and Oysal (2015) | Cutting speed | ANN |
| 15 | Mohamad et al. (2015) | Surface roughness | CS algorithm |
| 16 | Huang et al. (2015) | Material removal rate | TLBO algorithm, CS algorithm |
| 17 | Jagadish and Ray (2015) | Surface roughness | Fuzzy logic |
| 20 | Santhanakumar et al. (2015) | Surface roughness | RSM |
| | | Taper angle | |
| 21 | Jagadish and Ray (2016) | Surface roughness | RSM |
| | | Process time | |
| 22 | Mellal and Williams (2016) | Surface roughness | COA, Hoopoe heuristic |
| 23 | Santhanakumar et al. (2016) | Surface roughness | RSM, grey relational analysis |
| | | Striation zone | |
| | | Striation angle | |
| 24 | Shukla and Singh (2016a) | Kerf top width | Taguchi’s method, PSO algorithm, FA, SA, CS, BH algorithm, BBO algorithm and NSGA |
| | | Surface roughness | |
| 25 | Shukla and Singh (2016a) | Kerf | FA |
| | | Surface roughness | |

specific parameters for any optimization algorithm the user is required to perform a number of computational trials. Therefore, there is a need to develop an advanced optimization algorithm which is free from algorithm-specific parameters.

Recently, Rao (2016) has proposed a new metaheuristic optimization algorithm which does not require tuning of any algorithm-specific parameters for its working and it is named as Jaya algorithm. In order to implement the Jaya algorithm the user is only required to tune the common control parameters like population size and number of generations. In addition to algorithm specific parameter-less control, other distinctive features of Jaya algorithm are simplicity and ease of implementation, as the solutions are updated using a single

equation. The effectiveness of Jaya algorithm was proved on a number of constrained and unconstrained benchmark functions and engineering optimization problems (Rao 2016; Rao and Waghmare 2016).

The AWJM process is a multi-input process requiring simultaneous optimization of the multiple responses. (Rao 2011; Rostami and Neri 2017). Therefore, there arises a need to formulate the multi-objective optimization problems (MOOPs) and determine the Pareto-efficient set of solutions. These MOOPs can be effectively solved using a priori approach such as weighted sum, ϵ -constraint method, achievement scalarizing function, etc. However, in a priori approach, in order to obtain a set of distinct solutions,

it is required to run the algorithm independently for each set of weights. Another approach to solve MOOPs is, a posteriori approach, in such approach it is not required to assign the weights to the objective functions prior to the simulation run and multiple tradeoff (Pareto-efficient) solutions for a MOOP is obtained in a single run of simulation. The decision maker can then select one solution from the set of Pareto-efficient solutions based on the requirement or order of importance of objectives. Keeping in view of the advantages of the posteriori approach, researchers have developed posteriori versions of well-known optimization algorithms such as multi-objective differential evolution (MODE), non-dominated sorting genetic algorithm (NSGA), multi-objective particle swarm optimization (MOPSO), etc. These posteriori multi-objective optimization algorithms have been widely applied by the researchers to solve the multi-objective optimization problems in the field of engineering and science (Zhou et al. 2011). Chandrasekaran et al. (2010) proposed a posteriori multi-objective optimization algorithm based on clonal selection for solving engineering optimization problems. Falco et al. (2016) used multi-objective evolutionary algorithm for optimizing personalized touristic itineraries. Yu et al. (2016) proposed a prediction-based multi-objective evolutionary algorithm (MOEA) with NSGA-II for oil purchasing and distribution.

In this work a posteriori population based multi-objective version of the Jaya algorithm is applied to solve the MOOP of abrasive water-jet machining processes and is named as “Multi-objective Jaya algorithm (MO-Jaya)” algorithm. Similar to the Jaya algorithm, the MO-Jaya algorithm does not require tuning of any algorithm-specific parameters for its working. This averts the risk of algorithm getting trapped into local optima or slow convergence rate due to improper tuning of algorithm-specific parameters by the user. Furthermore, in the MO-Jaya algorithm the solutions are updated only in a single phase using a single equation. Therefore, the MO-Jaya algorithm is simple in implementation. The MO-Jaya algorithm has already proved its effectiveness in improving the performance of a number of modern machining processes and has shown better performance as compared to other optimization algorithms such as GA, NSGA, NSGA-II, BBO, NSTLBO, PSO, SQP and Monte Carlo simulations (Rao et al. 2017). Therefore, in the present work the MO-Jaya algorithm is applied to solve the input process parameters optimization problems of AWJM process.

The Jaya and MO-Jaya algorithms are described in “Optimization methodology” section. A computer code for MO-Jaya algorithm is developed in MATLAB R2009a. A computer system with a 2.93 GHz processor and 4 GB RAM is used for execution of the program.

Optimization methodology

Jaya algorithm

The Jaya algorithm mimics the behavior exhibited by biological species to survive and succeed in their respective habitats. There are a million of different species available in our eco-system. However, a common behavior exhibited by most of the species is to imitate the most successful member in the group at the same move away from the unsuccessful member of the group. Similarly, in the Jaya algorithm the solutions in the current population are analogous to a group of certain species. The most successful member in the group is analogous to the best solution, while, the most unsuccessful member in the group is analogous to the worst solution.

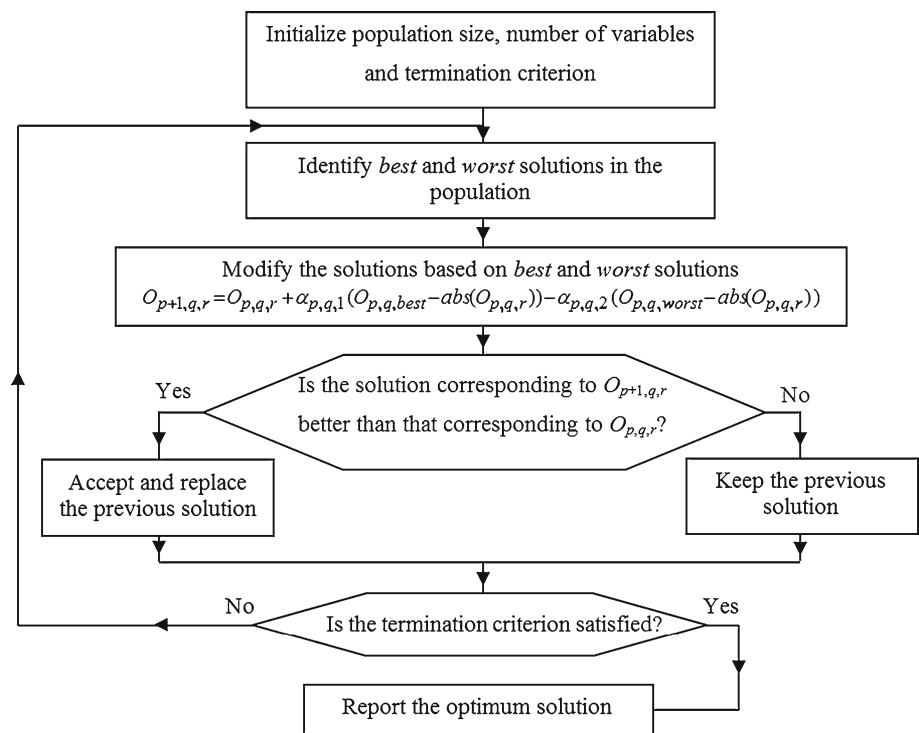
In the Jaya algorithm P initial solutions are randomly generated obeying the upper and lower bounds of the process variables. Thereafter, each variable of every solution is stochastically updated using Eq. (1). The best solution is the one with maximum fitness (i.e. best value of objective function) and the worst solution is the one with lowest fitness (i.e. worst value of objective function).

$$O_{p+1,q,r} = O_{p,q,r} + \alpha_{p,q,1}(O_{p,q,best} - abs(O_{p,q,r})) - \alpha_{p,q,2}(O_{p,q,worst} - abs(O_{p,q,r})) \quad (1)$$

Here *best* and *worst* represent the index of the best and worst solutions among the population. p, q, r are the index of iteration, variable, and candidate solution. $O_{p,q,r}$ means the q th variable of r th candidate solution in p th iteration. $\alpha_{p,q,1}$ and $\alpha_{p,q,2}$ are numbers generated randomly in the range of [0, 1]. The random numbers $\alpha_{p,q,1}$ and $\alpha_{p,q,2}$ act as scaling factors and ensure exploration. The absolute value of the variable (instead of a signed value) also ensures exploration. Figure 1 shows the flowchart for Jaya algorithm.

An important feature of the Jaya algorithm which ensures convergence is the selection procedure. Once every solution is modified based on Eq. (1), the fitness of new solutions (O_{p+1}) is evaluated based on the objective function. Thereafter, the fitness of every new solution is compared with its respective parent solution (O_p) and the new solution is selected for the next iteration, if and only if its fitness is better than the parent solution. If the fitness of the new solution is better than the parent solution then the parent solution is discarded and replaced by the new solution. However, if the fitness of the new solution is not better than the parent solution then it is discarded and the parent solution is kept as it is and taken to the next iteration. This type of selection procedure ensures selection of only superior solutions in every iteration of the Jaya algorithm. Therefore, in each iteration of Jaya algorithm the solutions always move closer to the best solution and thus convergence is achieved.

Fig. 1 Flowchart for Jaya algorithm



MO-Jaya algorithm

The MO-Jaya algorithm is a posteriori version of Jaya algorithm for solving MOOPs. The solutions in the MO-Jaya algorithm are updated in the similar manner as in the Jaya algorithm based on Eq. (1). In the interest of handling problems in which more than one objective co-exist the MO-Jaya algorithm is embedded with dominance ranking approach and crowding distance evaluation approach (Deb et al. 2002; Rao 2015).

In the MO-Jaya algorithm, the superiority among the solutions is decided according to the non-dominance rank and value of the density estimation parameter i.e. crowding distance (ξ). The solution with highest rank (i.e. rank = 1) and largest value of ξ is chosen as the *best* solution. On the other hand the solution with the lowest rank and lowest value of ξ is selected as the *worst* solution. Such a selection scheme is adopted so that solution in less populous region of the objective space may guide the search process. Once the *best* and *worst* solutions are selected, the solutions are updated based on the Eq. (1).

After all the solutions are updated, the updated solutions are combined with the initial population so that a set of $2P$ solutions (where P is the size of initial population) is formed. These solutions are again ranked and the ξ value for every solution is computed. Based on the new ranking and ξ value P good solutions are chosen. The flowchart of MO-Jaya algorithm is given in Fig. 2.

In the case of MO-Jaya algorithm, in each iteration, all the initial solutions in the population are updated according to

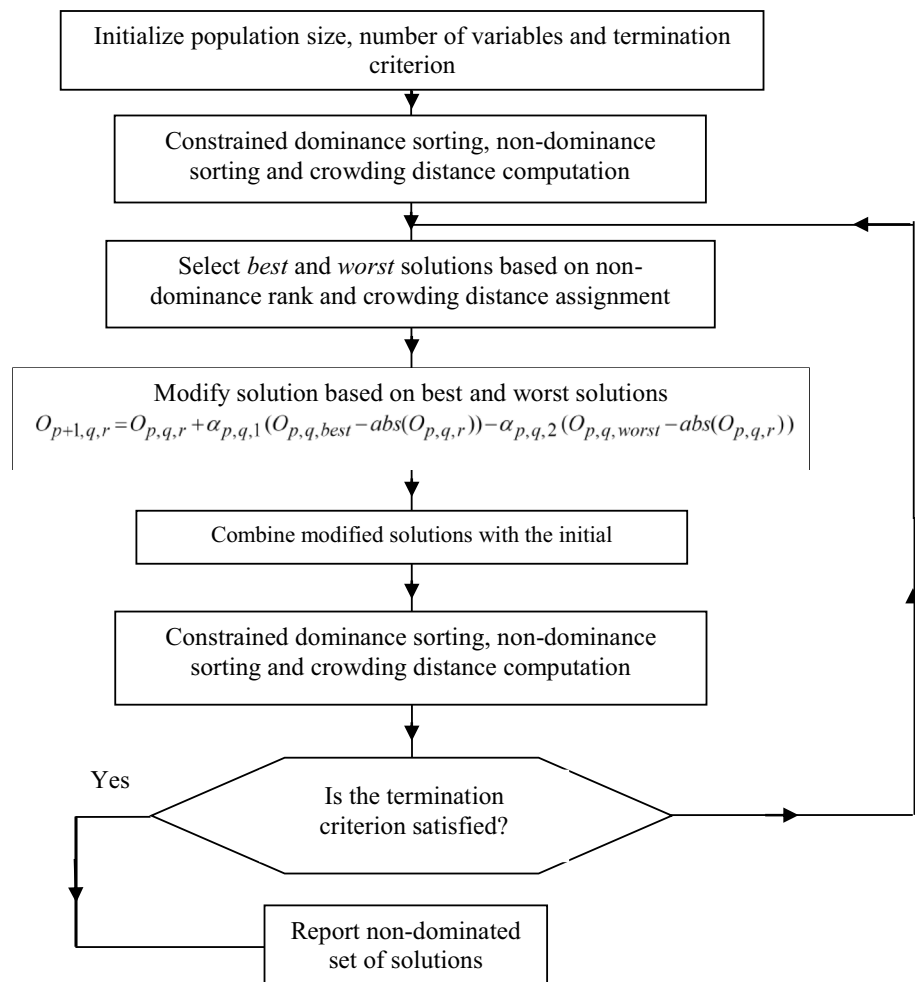
Eq. (1) to obtain the new solutions. Thereafter, a combined population is formed by combining initial solutions and the new solutions. Now from the combined population, good solutions are selected based on the non-dominance rank and crowding distance value and only the good solutions are taken to the next iteration. Such a selection procedure prevents the inferior solutions (i.e. lower rank solutions) from entering into the next iteration. As the algorithm progresses, the number of non-dominated solutions in the population increases. This ensures that, in every iteration of MO-Jaya algorithm, the solutions always move towards the Pareto-optimal set in every iteration. Finally, the algorithm converges at the Pareto-optimal set.

For every candidate solution the MO-Jaya algorithm evaluates the objective function only once in each iteration. Therefore, the total no. of function evaluations required by MO-Jaya algorithm = population size \times no. of iterations. However, when the algorithm is run more than once, then the number of function evaluations is to be calculated as: no. of function evaluations = no. of runs \times population size \times number of iterations. The methodology used for ranking of solutions, computing the crowding distance and crowding comparison operator are described in the following subsections.

Ranking methodology

The approach used for ranking of solutions is based on the non-dominance relation between solutions and is described

Fig. 2 Flowchart for MO-Jaya algorithm



as follows. In an M objective optimization problem, P is the set of solutions to be sorted and $n = |P|$.

Domination A solution x_1 is said to dominate another solution x_2 if and only if $f_i(x_1) \leq f_i(x_2)$ for all $1 \leq i \leq M$ and $f_i(x_1) < f_i(x_2)$ for at least one i , where $i \in \{1, \dots, M\}$ (when all objectives are to be minimized).

Non-dominance A solution x^* in P is non-dominated if there does not exist any solution x_j in P which dominates x^* .

Similarly, every solution in P is competes with every other solution and the not dominated solution are removed from P and assigned rank one. The remaining solutions in P are again sorted in the same way and the not dominated solutions are removed and assigned rank two. Unless all the solutions in P receive a rank this procedure is continued. A group of solutions with same rank is known as front (F)

Computing the crowding distance

The crowding distance (ξ_j) is an estimate of the density of the solutions in the vicinity of a particular solution j . For a

particular front F , let $l = |F|$ then for each member in F , ξ is calculated as follows.

Step 1: Initialize $\xi_j = 0$

Step 2: Sort all solutions in F the set in the worst order of objective function value f_m .

Step 3: In the sorted list of m th objective assign infinite crowding distance to solutions at the extremes of the sorted list (i.e. $\xi_1 = \xi_l = \infty$), for $j = 2$ to $(l - 1)$, calculate ξ_j as follows:

$$\xi_j = \xi_j + \frac{f_m^{j+1} - f_m^{j-1}}{f_m^{\max} - f_m^{\min}} \quad (2)$$

where, j represents a solution in the sorted list, f_m is the objective function value of m th objective of j th solution, and are the highest and the lowest values of the m th objective function in the current population. Likewise, ξ is computed for all the solutions in all F s

In the case of MOOPs there exists more than one optimal solution. Therefore, the aim is to find a set of Pareto-efficient solutions. In MO-Jaya algorithm in order to avoid clustering of solutions about a single good (higher rank) solution,

the good solutions in the isolated region of the search space are identified based on the ξ value, and a solution with a higher rank and higher ξ value is considered as the best solution in the next generation. Thus, the other solutions in the population will be directed towards the good solution which lies in the less populous (isolated) region of the search space in the next generation. This will prevent the algorithm from converging to single optimum solution and ensure diversity among the solutions. For this purpose a solution from the more isolated region of search space is given more preference than the solution in the crowded region of the search space. In the MO-Jaya algorithm, among the two competing solutions i and j , primarily, the solution with a higher rank is preferred. If the two solutions have equal rank then the solution with a higher ξ value is preferred.

Improved preference ranking organization method for enrichment evaluations

Selecting one solution out-of the multiple Pareto-optimal solutions provided by the MO-Jaya algorithm, based on the decision maker’s order of importance to the objectives, is a multi-criteria decision making (MCDM) problem. The multiple Pareto-optimal solutions are treated as alternatives and the objectives form the selection criteria. Rao and Patel (2010) proposed a multi-criteria decision making method named as improved preference ranking organization method for enrichment evaluations (improved PROMETHEE) and demonstrated its effectiveness in solving various decision making problems in the manufacturing environment.

Now in this work the improved PROMETHEE method is employed to select the best solution out-off the set of multiple solutions provided by MO-Jaya algorithm for optimization problems of modern machining processes. Like other MCDM techniques, the improved PROMETHEE method also employs pairwise comparison of the alternatives with respect to a particular criterion. If $a1$ and $a2$ are the two alternatives, then, the improved PROMETHEE method uses a preference function (P_i) to translate the difference between the evaluations obtained by two alternatives ($a1$ and $a2$) in terms of a particular criterion into a preference degree ranging from 0 to 1 which is mathematically expressed by Eqs. (3) and (4).

$$P_{i,a1a2} = G_i[c_i(a1) - c_i(a2)] \tag{3}$$

$$0 \leq P_{i,a1,a2} \leq 1 \tag{4}$$

where, G_i is a non-decreasing function of the observed deviation ($c_i(a1) - c_i(a2)$) between two alternatives $a1$ and $a2$ over the criterion c_i ; i represents a particular criterion. Further, the multiple criteria preference index is evaluated as the weighted average of preference functions P_i which is mathematically expressed by Eq. (5).

$$\Pi_{a1a2} = \sum_{i=1}^M w_i P_{i,a1,a2} \tag{5}$$

where, w_i is the weights assigned by the decision maker to each criterion and M is the total number of selection criteria. Π_{a1a2} is the overall preference index of alternative $a1$ over alternative $a2$. The next step is to evaluate the leaving flow, entering flow and the net flow for a particular alternative a , which belongs to the set of alternatives A , using the following equations.

$$\varphi^+(a) = \sum_{x \in A} \prod_{xa} \tag{6}$$

$$\varphi^-(a) = \sum_{x \in A} \prod_{xa} \tag{7}$$

$$\varphi(a) = \varphi^+(a) - \varphi^-(a) \tag{8}$$

where, $\varphi^+(a)$ is called the leaving flow, $\varphi^-(a)$ is called the entering flow and $\varphi(a)$ is called net flow. An alternative $a1$ out-ranks alternative $a2$ if $\varphi(a1) > \varphi(a2)$. The improved PROMETHEE method provides complete ranking of alternatives from best to worst using the net flows.

Performance measure

In order to compare the quality of Pareto-fronts obtained by optimization algorithms the hypervolume metric is used in this work. Hypervolume gives the volume (area in the case of bi-objective optimization problems) of the search space which is dominated by a Pareto-front obtained by a particular algorithm with respect to a given reference point. Therefore for a particular algorithm a higher value of hypervolume is desirable which indicates the quality of the Pareto-front obtained by the algorithm.

Mathematically, for a Pareto-front containing Q solutions, for each solution i belongs to Q , a hypervolume v_i is constructed with reference point W and the solution i as the diagonal corners of the hypercube. Thereafter the union of these hypercubes is found and its hypervolume is calculated as follows.

$$HV = volume \left(\bigcup_{i=1}^{|Q|} v_i \right) \tag{9}$$

A superior Pareto-front is characterized by three important criteria which are as follows: (1) quality of solutions i.e. solutions in a Pareto-front should be superior in terms of objective function values, (2) diversity among the solutions i.e. the solutions in a Pareto-front must be uniformly distributed as much as possible and (3) pertinence i.e. a Pareto-front must contain as many number of solutions as possible so as to give maximum possible choices to the decision maker. The

hypervolume is one such performance indicator which compares two sets of Pareto-optimal solutions on the basis of the all three aforementioned performance criteria simultaneously (Zitzler and Thiele 1999; Rostami and Neri 2017). Therefore, in the present work, the hypervolume indicator is used for comparison of Pareto-fronts obtained by MO-Jaya algorithm with the Pareto-fronts obtained by other optimization algorithms.

Case studies

This section describes three optimization case studies of AWJM process. Single objective and multi-objective optimization problems are formulated based on the regression models developed by previous researchers and the same are solved using Jaya algorithm and multi-objective Jaya algorithm, respectively, and the results are reported.

Case study 1

In the case of AWJM process, Kerf is an important performance characteristic as it directly governs the geometry of the cut produced by the abrasive waterjet. For superior quality of the machined components it is required to achieve a uniform cut within the prescribed geometrical tolerances. Thus, the minimization of Kerf is desirable. On the other hand a superior surface finish in the machined components is desirable in order to minimize the effect of wear and friction. For this purpose minimization of mean surface roughness R_a in machined components is desirable. Kechagias et al. (2012) showed that Kerf and R_a are mutually conflicting in nature and are largely influenced by input process parameters such as material thickness ‘ A ’ (mm), nozzle diameter ‘ B ’ (mm), stand-off distance ‘ C ’ (mm), traverse speed ‘ D ’ (mm/min). Kechagias et al. (2012) developed precise regression models based on actual experimental data in order to map the relationship between the input process parameters with Kerf and R_a . However, as Kerf and R_a govern the quality of the machined components in AWJM process, besides developing regression models, it is important to determine the combination of input process parameters that will provide a trade-off between Kerf and R_a . For this purpose, in the present case study, the regression models developed by Kechagias et al. (2012) are used to formulate a multi-objective optimization problem and the same is solved using MO-Jaya algorithm.

Objective functions

$$\begin{aligned} \text{Minimize Kerf} = & 0.86068 - 0.068333 \times A + 0.61131 \times B \\ & + 0.0021518 \times C - 0.0020416 \times D \\ & - 0.3596 \times A \times B + 0.0030106 \end{aligned}$$

$$\begin{aligned} & \times B \times C + 0.0019143 \\ & \times B \times D - 1.0143E - 5 \times C \times D \quad (10) \end{aligned}$$

$$\begin{aligned} \text{Minimize } R_a = & -11.106 + 10.211 \times A + 15.169 \\ & \times B + 0.0047057 \times C + 0.011813 \times D \\ & - 10.331 \times A \times B + 0.01012 \times B \\ & \times C - 0.0074535 \\ & \times B \times D + 7.1924E - 6 \times C \times D \quad (11) \end{aligned}$$

Process parameters

$$0.9 \leq A \leq 1.25 \quad (12)$$

$$0.95 \leq B \leq 1.5 \quad (13)$$

$$20 \leq C \leq 96 \quad (14)$$

$$200 \leq D \leq 600 \quad (15)$$

The Kerf and R_a are mutually conflicting performance measures as the process parameter setting which may give a minimum value of Kerf may increase R_a . The process parameter setting which may lead to minimum R_a may simultaneously increase Kerf. Therefore, it is necessary to achieve a trade-off between Kerf and R_a in order to achieve the best performance of the AWJM process. Therefore, in this work MO-Jaya algorithm is applied to simultaneously optimize Kerf and R_a and achieve multiple trade-off solutions.

In order to set the population size for MO-Jaya algorithm, computational experiments are conducted by varying the population size from 10 to 70 in a step size of 10. It is observed that the population size of 50 and maximum number of iterations equal to 100 provided the best result for MO-Jaya algorithm (i.e. maximum number of function evaluations = $50 \times 100 = 5000$). The Pareto-efficient set of solutions obtained by MO-Jaya algorithm in a single simulation run is reported in Table 2. Fifty optimal combinations of input process parameters like material thickness, nozzle diameter, stand-off distance and traverse speed are provided in Table 2 along with the corresponding values of Kerf and R_a .

Now, in order to show the effectiveness of MO-Jaya algorithm, the results of MO-Jaya algorithm are compared with the non-dominated sorting teaching-learning-based optimization algorithm (NSTLBO) algorithm. The NSTLBO algorithm is a posteriori version of TLBO algorithm. Similar, to the MO-Jaya algorithm the NSTLBO algorithm does not require tuning of any algorithm-specific parameters and provides a set of multiple Pareto-optimal at the end of every simulation run. Furthermore, the NSTLBO algorithm has been found successful in solving the multi-objective optimization problems of machining processes such as turning process, wire-electric-discharge machining process, focused ion beam micro-milling process, micro wire-electric discharge machining process and laser cutting process (Rao

Table 2 Pareto-efficient solutions obtained using MO-Jaya algorithm for case study I

| S. No. | A (mm) | B (mm) | C (mm) | D (mm/min) | Kerf (mm) | R_a (μm) |
|--------|--------|--------|---------|------------|-----------|-------------------------|
| 1 | 1.25 | 0.95 | 96 | 600 | 0.6921 | 8.4285 |
| 2 | 1.25 | 0.95 | 92.7855 | 600 | 0.6955 | 8.3686 |
| 3 | 1.25 | 0.95 | 87.8395 | 600 | 0.7008 | 8.2764 |
| 4 | 1.25 | 0.95 | 83.6259 | 600 | 0.7054 | 8.1979 |
| 5 | 1.25 | 0.95 | 71.2818 | 600 | 0.7186 | 7.9679 |
| 6 | 1.25 | 0.95 | 67.2073 | 600 | 0.723 | 7.892 |
| 7 | 1.25 | 0.95 | 62.2831 | 600 | 0.7283 | 7.8002 |
| 8 | 1.25 | 0.95 | 59.2725 | 600 | 0.7315 | 7.7441 |
| 9 | 1.25 | 0.95 | 56.6518 | 600 | 0.7343 | 7.6953 |
| 10 | 1.25 | 0.95 | 52.4845 | 600 | 0.7388 | 7.6176 |
| 11 | 1.25 | 0.95 | 46.6444 | 600 | 0.7451 | 7.5087 |
| 12 | 1.25 | 0.95 | 45.5529 | 600 | 0.7463 | 7.4884 |
| 13 | 1.25 | 0.95 | 37.8658 | 600 | 0.7545 | 7.3452 |
| 14 | 1.25 | 0.95 | 28.8233 | 600 | 0.7642 | 7.1767 |
| 15 | 1.25 | 0.95 | 24.5764 | 600 | 0.7688 | 7.0975 |
| 16 | 1.25 | 0.95 | 20 | 600 | 0.7737 | 7.0122 |
| 17 | 1.25 | 0.95 | 20 | 570.634 | 0.7862 | 6.8691 |
| 18 | 1.2499 | 0.95 | 20 | 555.663 | 0.7926 | 6.796 |
| 19 | 1.25 | 0.95 | 20.2853 | 546.765 | 0.7962 | 6.7579 |
| 20 | 1.2498 | 0.95 | 20 | 532.189 | 0.8027 | 6.6815 |
| 21 | 1.25 | 0.95 | 20 | 519.823 | 0.8078 | 6.6213 |
| 22 | 1.2499 | 0.95 | 20 | 502.323 | 0.8153 | 6.5359 |
| 23 | 1.25 | 0.95 | 20 | 498.555 | 0.8169 | 6.5176 |
| 24 | 1.25 | 0.95 | 20 | 482.706 | 0.8236 | 6.4403 |
| 25 | 1.2499 | 0.95 | 20.9497 | 475.656 | 0.8269 | 6.4227 |
| 26 | 1.2495 | 0.95 | 20 | 461.017 | 0.8331 | 6.3343 |
| 27 | 1.25 | 0.95 | 20 | 446.646 | 0.839 | 6.2645 |
| 28 | 1.25 | 0.95 | 20 | 420.606 | 0.8501 | 6.1375 |
| 29 | 1.25 | 0.95 | 20 | 410.757 | 0.8543 | 6.0895 |
| 30 | 1.2499 | 0.95 | 20 | 395.311 | 0.8609 | 6.0142 |
| 31 | 1.2499 | 0.95 | 20 | 391.606 | 0.8625 | 5.9961 |
| 32 | 1.25 | 0.95 | 20 | 376.661 | 0.8688 | 5.9232 |
| 33 | 1.25 | 0.95 | 20 | 366.559 | 0.8731 | 5.874 |
| 34 | 1.2499 | 0.95 | 20 | 343.304 | 0.8831 | 5.7606 |
| 35 | 1.2449 | 0.95 | 20 | 303.481 | 0.9021 | 5.5644 |
| 36 | 1.2499 | 0.95 | 20 | 286.065 | 0.9074 | 5.4815 |
| 37 | 1.25 | 0.95 | 20 | 262.191 | 0.9176 | 5.3651 |
| 38 | 1.25 | 0.95 | 20 | 246.143 | 0.9244 | 5.2868 |
| 39 | 1.2498 | 0.95 | 20 | 225.369 | 0.9333 | 5.1855 |
| 40 | 1.25 | 0.95 | 20 | 203.816 | 0.9424 | 5.0804 |
| 41 | 1.2197 | 0.95 | 20 | 200 | 0.9565 | 5.0498 |
| 42 | 1.1761 | 0.95 | 20 | 200 | 0.9743 | 5.0325 |
| 43 | 1.1571 | 0.95 | 20 | 200 | 0.9821 | 5.025 |
| 44 | 1.1294 | 0.95 | 20 | 200 | 0.9935 | 5.014 |
| 45 | 1.0803 | 0.95 | 20 | 200 | 1.0136 | 4.9945 |
| 46 | 1.0408 | 0.95 | 20 | 200 | 1.0298 | 4.9789 |
| 47 | 0.9931 | 0.95 | 20 | 200 | 1.0494 | 4.96 |

Table 2 continued

| S. No. | A (mm) | B (mm) | C (mm) | D (mm/min) | Kerf (mm) | R_a (μm) |
|--------|--------|--------|--------|------------|-----------|-------------------------|
| 48 | 0.9808 | 0.95 | 20 | 200 | 1.0544 | 4.9551 |
| 49 | 0.9456 | 0.95 | 20 | 200 | 1.0688 | 4.9411 |
| 50 | 0.9042 | 0.95 | 20 | 200 | 1.0858 | 4.9247 |

et al. 2016). The NSTLBO algorithm had provided better results as compared to the other well-known optimization algorithms such as GA, PSO, NSGA-II and iterative search method (Rao et al. 2016). Therefore, considering the effectiveness of NSTLBO algorithm in solving the multi-objective optimization problems of machining processes, in this work the NSTLBO algorithm is applied to solve the optimization problems of AWJM process and the results are compared with the results of MO-Jaya algorithm.

For the purpose of comparison, the same problem is solved using NSTLBO algorithm with a population size of 50 and maximum number of function evaluations equal to 5000. The Pareto-fronts obtained by MO-Jaya and NSTLBO algorithms are shown in Fig. 3. The MO-Jaya algorithm achieved a higher hypervolume as compared to NSTLBO algorithm ($HV_{MO-Jaya} = 1.4235$; $HV_{NSTLBO} = 1.352$). The MO-Jaya and NSTLBO algorithms required 1800 and 2200 function evaluations, respectively to converge at the Pareto-optimal set of solutions. The CPU time required by MO-Jaya and NSTLBO algorithms is 5.822 and 8.6183 s, respectively to perform 5000 function evaluations.

The MO-Jaya algorithm has provided 50 Pareto-optimal solutions for the optimization problem considered in this case study. The user is now required to choose one solution out-of the 50 solutions provided by MO-Jaya algorithm based on his order of preference to $Kerf$ and R_a . This forms a decision making problem with 50 alternatives and two selection criteria i.e. $Kerf$ and R_a . Both the criteria are considered as non-beneficial criteria. To demonstrate the applicability of the improved PROMETHEE method, three different sets of weights are considered and the best choice corresponding to each of the considered set of weights provided by improved PROMETHEE method is shown in Table 3. The steps of the improved PROMETHEE method for evaluating the alternatives are shown in the ‘‘Appendix A’’. In Table 3, three optimal combinations of input process parameters are provided which correspond to different sets of weights; w_1 and w_2 represent the weights assigned to $Kerf$ and R_a , respectively. By changing the values of weights three decision making scenarios are formed. Scenario 1 and scenario 2 are formed by assigning high importance to $Kerf$ and R_a , respectively. Scenario 3 is formed by assigning equal importance to both the criteria. The choice of the best alternative provided by improved PROMETHEE method out-of the 100 alternatives in each of the decision making scenario is shown in Table 3. The

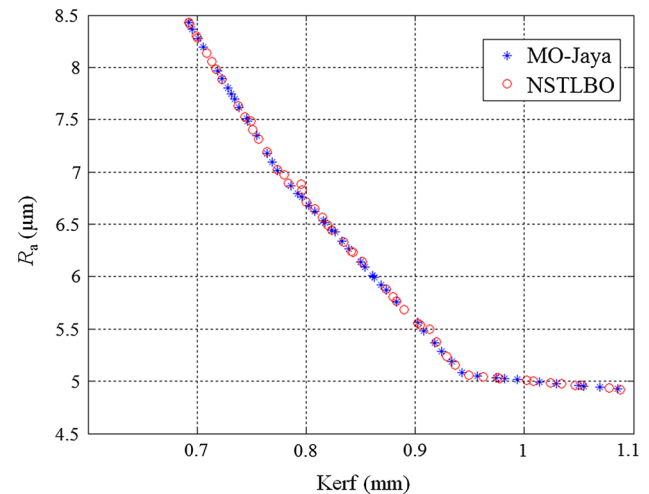


Fig. 3 Pareto-fronts obtained by MO-Jaya and NSTLBO algorithms for case study 1

weights assigned to the criteria in this work are only for the purpose of demonstration. However, the decision maker may choose any set of weights based on his interests and may select the best alternative using the improved PROMETHEE method.

Case study 2

In the case of AWJM process, depth of penetration ‘ DOP ’ and surface roughness ‘ R_a ’ are considered as important performance parameters. This is mainly because DOP directly corresponds to the material removal rate which governs the machining time and efficiency. On the other hand, R_a of the machined components governs the production quality. Therefore, maximization of depth of penetration and minimization of surface roughness is desirable in order to achieve production efficiency and product quality. However, Liu et al. (2014) showed that depth of penetration and surface roughness are mutually conflicting in nature and depend upon a number of process parameters such as traverse speed ‘ A ’ (mm/s), pressure ‘ B ’ (MPa), stand-off distance ‘ C ’ (mm), tilt angle ‘ D ’ (degree), surface speed ‘ E ’ (m/s) and abrasive flow rate ‘ F ’ (g/s). Liu et al. (2014) had developed precise regression models for $DOP(\mu\text{m})$ and $R_a(\mu\text{m})$ considering the aforementioned process parameters based on actual experimental data. Now considering the importance of DOP and R_a and

Table 3 The best choice provided by PROMTHEE method corresponding to different set of weights for case study 1

| S. No. | Weights assigned to the criteria | | Best choice provided by PROMTHEE method | | | | | |
|--------|----------------------------------|-------|---|------|-----|---------|--------------------|--------|
| | | | Process parameters | | | | Criteria/responses | |
| | w_1 | w_2 | A | B | C | D | $Kerf$ | R_a |
| 1 | 0.9 | 0.1 | 1.25 | 0.95 | 96 | 600 | 0.6921 | 8.4285 |
| 2 | 0.1 | 0.9 | 1.2197 | 0.95 | 20 | 200 | 0.9565 | 5.0498 |
| 3 | 0.5 | 0.5 | 1.25 | 0.95 | 20 | 203.816 | 0.9424 | 5.0804 |

their mutually conflicting nature, it is important to achieve a trade-off between the two. Therefore, in the present work, a multi-objective optimization problem is formulated based on the regression models developed by Liu et al. (2014) and the same is solved using MO-Jaya algorithm in order to obtain trade-off between DOP and R_a .

Objective functions

$$\begin{aligned}
 \text{Maximize } DOP = & 671.17 - 1107.08A - 1.03B - 8.49D \\
 & + 13.27F - 2.11AB - 35.74AC \\
 & - 42.93AF + 0.028BD + 0.89CF \\
 & + 2666.77A^2 \tag{16}
 \end{aligned}$$

$$\begin{aligned}
 \text{Minimize } R_a = & 33.17 + 0.023E \\
 & + 1.53F - 1.49AC - 0.12AD - 1.87AF \\
 & + 0.00084BD - 0.0029BE - 0.0051BF \\
 & + 0.0086CD + 0.066CE \\
 & + 0.061CF + 0.011DE \\
 & - 0.014DF - 0.081EF + 8.23A^2 \\
 & - 0.00019B^2 - 0.029C^2 + 0.0027D^2 \\
 & + 0.079F^2 \tag{17}
 \end{aligned}$$

Process parameters

$$0.1 \leq A \leq 0.5 \tag{18}$$

$$200 \leq B \leq 320 \tag{19}$$

$$2 \leq C \leq 10 \tag{20}$$

$$60 \leq D \leq 90 \tag{21}$$

$$2 \leq E \leq 8 \tag{22}$$

$$5 \leq F \leq 11.67 \tag{23}$$

Desirability function approach was used by Liu et al. (2014) in order to optimize DOP and R_a simultaneously and two trade-off solutions were provided. In the present work the MO-Jaya algorithm is applied for simultaneous optimization of DOP and R_a . In order to see whether any improvement in the results is obtained, the solutions obtained by Liu et al. (2014) using desirability approach are considered as it is and the same are compared with the solutions obtained

using MO-Jaya algorithm. In order to select a population size for MO-Jaya algorithm computational experiments are conducted by varying the population size from 20 to 70 with a step size of 10. A population size of 50, a maximum number of iterations equal to 100 (i.e. maximum number of function evaluations equal to 50,000) is chosen for MO-Jaya algorithm. The Pareto-efficient set of solutions obtained using MO-Jaya algorithm in a single simulation run is shown in Table 4.

For the purpose of comparison the same problem is solved by NSTLBO algorithm. The Pareto-fronts obtained by MO-Jaya and NSTLBO algorithms are shown in Fig. 4 and the quality of the Pareto-fronts is judged based on the hypervolume metric. Although, the Pareto-fronts obtained by MO-Jaya and NSTLBO algorithms overlap each other The MO-Jaya algorithm obtained a higher value of hypervolume ($HV_{MO-Jaya} = 15, 335$) as compared to NSTLBO algorithm ($HV_{NSTLBO} = 15, 100$). MO-Jaya and NSTLBO algorithms required 1250 and 1500 function evaluations, respectively, to converge at the Pareto-efficient set of solutions. The MO-Jaya and NSTLBO algorithms required a CPU time of 9.305 and 10.789 s, respectively, to perform 5000 function evaluation.

Table 5 gives optimum process parameter combination suggested by desirability approach and its comparison with the results of MO-Jaya algorithm. It is observed that the results of MO-Jaya algorithm are better than the results obtained using desirability approach in terms of DOP and R_a .

The MO-Jaya algorithm has provided 50 Pareto-optimal solutions for the multi-objective optimization problem considered in this case study. The decision maker is now required to choose one out-of the 50 alternative solutions with DOP and R_a as selection criteria. This forms a decision making problem with two criteria and 50 alternatives where DOP is the beneficial criteria and R_a as non-beneficial criteria. To demonstrate the applicability of improved PROMETHEE method, three different set of weights are considered. The steps in the improved PROMETHEE method are similar to those discussed in ‘‘Optimization methodology’’ section. The best choice corresponding to each of the considered set of weights provided by improved PROMETHEE method is shown in Table 6.

Table 4 Pareto-efficient solutions obtained using MO-Jaya algorithm for case study 2

| S. No. | <i>A</i> (mm/s) | <i>B</i> (MPa) | <i>C</i> (mm) | <i>D</i> (°) | <i>E</i> (m/s) | <i>F</i> (g/s) | <i>DOP</i> (μm) | <i>R_α</i> (μm) |
|--------|-----------------|----------------|---------------|--------------|----------------|----------------|-----------------|---------------------------|
| 1 | 0.5 | 320 | 2 | 60 | 7.9998 | 11.6667 | 34.7405 | 19.2594 |
| 2 | 0.5 | 319.977 | 2 | 60 | 7.9768 | 11.3381 | 36.8575 | 19.479 |
| 3 | 0.5 | 319.954 | 2 | 60 | 7.9821 | 10.2838 | 43.6302 | 20.2079 |
| 4 | 0.5 | 319.877 | 2 | 60 | 7.9859 | 8.5269 | 54.9327 | 21.8237 |
| 5 | 0.5 | 319.929 | 2 | 60 | 7.9773 | 7.7453 | 59.9251 | 22.6951 |
| 6 | 0.3175 | 320 | 2 | 60 | 7.9974 | 11.6486 | 66.5803 | 23.8781 |
| 7 | 0.3021 | 319.893 | 2 | 60 | 7.9464 | 11.6608 | 77.4081 | 24.3598 |
| 8 | 0.2909 | 320 | 2 | 60 | 8 | 11.67 | 86.1082 | 24.5893 |
| 9 | 0.286 | 320 | 2 | 60 | 7.9963 | 11.6404 | 90.0126 | 24.731 |
| 10 | 0.2582 | 319.936 | 2 | 60 | 7.9937 | 11.67 | 115.171 | 25.503 |
| 11 | 0.2513 | 320 | 2 | 60.0095 | 7.9966 | 11.648 | 121.982 | 25.6894 |
| 12 | 0.2445 | 320 | 2 | 60 | 7.9895 | 11.67 | 129.094 | 25.8799 |
| 13 | 0.2188 | 320 | 2 | 60 | 8 | 11.67 | 157.873 | 26.5937 |
| 14 | 0.2172 | 320 | 2 | 60 | 8 | 11.6697 | 159.756 | 26.6386 |
| 15 | 0.2051 | 320 | 2 | 60 | 8 | 11.67 | 174.669 | 26.9848 |
| 16 | 0.1993 | 320 | 2 | 60 | 8 | 11.6698 | 181.977 | 27.149 |
| 17 | 0.185 | 319.871 | 2 | 60 | 7.9673 | 11.67 | 201.058 | 27.6179 |
| 18 | 0.1704 | 319.798 | 2 | 60 | 7.9691 | 11.5537 | 220.579 | 28.0439 |
| 19 | 0.164 | 320 | 2 | 60 | 8 | 11.67 | 230.94 | 28.174 |
| 20 | 0.1537 | 320 | 2 | 60 | 8 | 11.67 | 246.487 | 28.4771 |
| 21 | 0.1477 | 320 | 2 | 60 | 7.9999 | 11.6694 | 255.85 | 28.6554 |
| 22 | 0.143 | 320 | 2 | 60.0249 | 8 | 11.67 | 263.182 | 28.8055 |
| 23 | 0.1358 | 320 | 2 | 60 | 7.9998 | 11.67 | 274.859 | 29.0083 |
| 24 | 0.129 | 319.888 | 2 | 60.3551 | 8 | 11.5482 | 285.044 | 29.3982 |
| 25 | 0.1203 | 319.737 | 2 | 60 | 7.9945 | 11.3572 | 297.6 | 29.4782 |
| 26 | 0.1124 | 319.906 | 2 | 60 | 8 | 11.67 | 314.298 | 29.7213 |
| 27 | 0.1029 | 320 | 2 | 60 | 7.9857 | 11.018 | 324.377 | 29.9233 |
| 28 | 0.1 | 319.718 | 2 | 60 | 7.9999 | 11.6699 | 336.471 | 30.1268 |
| 29 | 0.1 | 320 | 2.4625 | 60 | 7.9987 | 11.67 | 339.747 | 30.768 |
| 30 | 0.1 | 320 | 3.1277 | 60 | 8 | 11.67 | 344.278 | 31.7277 |
| 31 | 0.1 | 320 | 3.7105 | 60 | 8 | 11.67 | 348.248 | 32.5487 |
| 32 | 0.1 | 320 | 4.4709 | 60 | 8 | 11.6698 | 353.426 | 33.5901 |
| 33 | 0.1 | 320 | 5.2604 | 60 | 8 | 11.67 | 358.807 | 34.6359 |
| 34 | 0.1 | 320 | 6.2743 | 60 | 8 | 11.6676 | 365.679 | 35.9249 |
| 35 | 0.1 | 320 | 7.1021 | 60 | 7.9985 | 11.6685 | 371.33 | 36.9354 |
| 36 | 0.1 | 320 | 8.2773 | 60.933 | 8 | 11.67 | 379.798 | 38.8396 |
| 37 | 0.1 | 320 | 10 | 60 | 8 | 11.6471 | 390.686 | 40.1389 |
| 38 | 0.1 | 320 | 10 | 62.5995 | 8 | 11.67 | 392.316 | 41.71 |
| 39 | 0.1 | 320 | 10 | 64.9763 | 8 | 11.6509 | 393.091 | 43.1527 |
| 40 | 0.1 | 320 | 10 | 67.8263 | 8 | 11.67 | 394.773 | 44.9483 |
| 41 | 0.1 | 320 | 10 | 71.5128 | 8 | 11.67 | 396.505 | 47.3211 |
| 42 | 0.1 | 320 | 10 | 72.3868 | 8 | 11.67 | 396.916 | 47.8944 |
| 43 | 0.1 | 320 | 10 | 74.7276 | 8 | 11.67 | 398.016 | 49.4502 |
| 44 | 0.1 | 320 | 10 | 76.7611 | 8 | 11.6689 | 398.953 | 50.8252 |
| 45 | 0.1 | 320 | 10 | 76.9298 | 8 | 11.6699 | 399.049 | 50.9408 |
| 46 | 0.1 | 320 | 10 | 82.3488 | 8 | 11.67 | 401.598 | 54.7204 |
| 47 | 0.1 | 320 | 10 | 83.5923 | 8 | 11.67 | 402.183 | 55.6101 |

Table 4 continued

| S. No. | A (mm/s) | B (MPa) | C (mm) | D (°) | E(m/s) | F (g/s) | DOP (μm) | R _a (μm) |
|--------|----------|---------|--------|---------|--------|---------|----------|---------------------|
| 48 | 0.1 | 320 | 10 | 85.4051 | 8 | 11.67 | 403.035 | 56.922 |
| 49 | 0.1 | 320 | 10 | 88.2527 | 8 | 11.6697 | 404.367 | 59.0187 |
| 50 | 0.1 | 320 | 10 | 90 | 8 | 11.67 | 405.194 | 60.3269 |

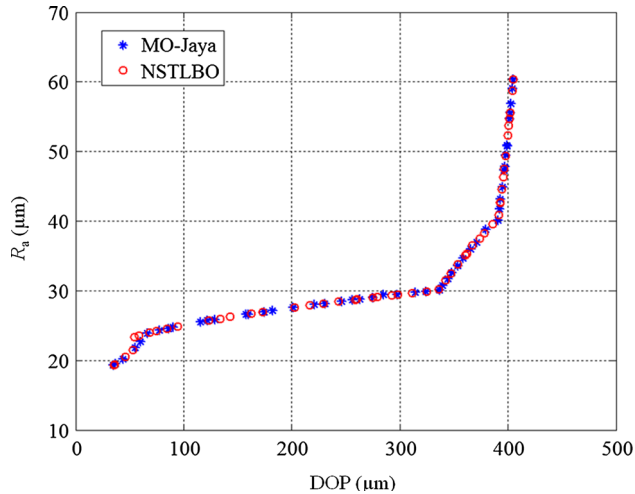


Fig. 4 Pareto-fronts obtained by MO-Jaya and NSTLBO algorithms for case study 2

Case study 3

In the case of AWJM process, in order to improve the quality of the machined components it is important to achieve a uniform cut within the prescribed geometrical tolerances.

The geometry of the cut is mostly governed by Kerf top width ‘Ktw’ and taper angle. Thus, in order to achieve a uniform cut it is important to maximize the Ktw and minimize the taper angle. Shukla and Singh (2016a) studied the influence of input process parameters such as traverse speed ‘x₁’ (mm/min), stand-off distance ‘x₂’ (mm), and mass flow rate ‘x₃’ (gm/sec) on Ktw and taper angle and developed regression models. Shukla and Singh (2016a) used these regression models to formulate an optimization problem. Shukla and Singh (2016a) optimized Ktw and taper angle separately using PSO, SA, FA, BH, CSA and BBO algorithms. Furthermore, Shukla and Singh (2016a) optimized Ktw and taper angle, simultaneously, using NSGA algorithm. Now in order to see whether any further improvement in the results can be achieved this case study is considered in the present work. The same optimization problem is solved using Jaya and MO-Jaya algorithms and the results are compared with the results reported by Shukla and Singh (2016a).

Objective functions

$$\begin{aligned} \text{Maximize } Ktw = & 12.7538 - 0.0974x_1 + 0.0913x_2 \\ & - 1.0424x_3 + 0.0002x_1^2 - 0.0289x_2^2 \end{aligned}$$

Table 5 Comparison of results obtained by desirability approach and MO-Jaya algorithm for case study 2

| S. No. | Technique | A | B | C | D | E | F | DOP | R _a |
|--------|---|------------|------------|-----------|----------------|----------|--------------|----------------|----------------|
| 1. | Desirability approach (Liu et al. 2014) | 0.1 | 317.75 | 9.78 | 72.96 | 4.63 | 11.15 | 384.69 | 49.298 |
| | MO-Jaya | 0.1 | 320 | 10 | 72.3868 | 8 | 11.67 | 396.916 | 47.8944 |
| | % Improvement obtained by MO-Jaya algorithm | | | | | | | 3.17% | 2.84% |
| 2. | Desirability approach (Liu et al. 2014) | 0.1 | 320 | 3.36 | 82.18 | 7.59 | 11.67 | 356.2852 | 45.5338 |
| | MO-Jaya | 0.1 | 320 | 10 | 67.8263 | 8 | 11.67 | 394.773 | 44.9483 |
| | % Improvement obtained by MO-Jaya algorithm | | | | | | | 10.80% | 1.28% |

The values in bold indicate better performance of the algorithm

Table 6 The best choice provided by PROMTHEE method corresponding to different set of weights for case study 2

| S. No. | Weights assigned to the criteria | | Best choice provided by PROMTHEE method | | | | | | | |
|--------|----------------------------------|----------------|---|---------|----|----|--------|---------|--------------------|----------------|
| | w ₁ | w ₂ | Process parameters | | | | | | Criteria/responses | |
| | | | A | B | C | D | E | F | DOP | R _a |
| 1 | 0.9 | 0.1 | 0.1 | 320 | 10 | 60 | 8 | 11.6471 | 390.686 | 40.1389 |
| 2 | 0.1 | 0.9 | 0.5 | 320 | 2 | 60 | 7.9998 | 11.6667 | 34.7405 | 19.2594 |
| 3 | 0.5 | 0.5 | 0.1 | 319.718 | 2 | 60 | 7.9999 | 11.6699 | 336.471 | 30.1268 |

Table 7 Results of single objective optimization for case study 3

| Optimization method | Performance parameter | Optimum value | Traverse speed | Standoff distance | Mass flow rate | No. of iterations |
|---------------------|-----------------------|---------------|----------------|-------------------|----------------|-------------------|
| PSO | Ktw | 2.8840 | 162.1396 | 4.9354 | 3.4964 | 20 |
| | Taper angle | 1.0916 | 185.7086 | 1.0044 | 2.2047 | 220 |
| FA | Ktw | 2.9127 | 160.4309 | 5 | 3.5 | 180 |
| | Taper angle | 1.0672 | 197.382 | 1.0002 | 2.2509 | 370 |
| SA | Ktw | 2.9136 | 160.3618 | 5 | 3.5 | 680 |
| | Taper angle | 1.0720 | 193.1769 | 1.017576 | 2.1858 | 270 |
| CSA | Ktw | 2.9184 | 160.0015 | 5 | 3.4990 | 5 |
| | Taper angle | 1.0692 | 193.8616 | 1.0178 | 2.2549 | 760 |
| BH | Ktw | 2.8698 | 160.8471 | 4.6944 | 3.4480 | 150 |
| | Taper angle | 1.0794 | 193.8199 | 1.0311 | 2.2647 | 620 |
| BBO | Ktw | 2.9187 | 160.0017 | 5 | 3.5 | 10 |
| | Taper angle | 1.0621 | 194.7495 | 1.0066 | 2.2479 | 15 |
| Jaya | Ktw | 2.9187 | 160 | 5 | 3.5 | 2 |
| | Taper angle | 1.0539 | 192.679 | 1 | 2.2957 | 6 |

The values in bold indicate better performance of the algorithm. The results of PSO, FA, SA, CSA, BH and BBO are obtained from Shukla and Singh (2016a)

$$+ 0.12011x_3^2 + 0.0016x_1x_2 + 0.0032x_1x_3 \quad (24)$$

$$\begin{aligned} \text{Minimize taperangle} = & 26.0879 - 0.2401x_1 + 0.5024x_2 \\ & - 2.2168x_3 + 0.0006x_1^2 - 0.0884x_2^2 \\ & 0.3657x_3^2 + 0.0024x_1x_2 \\ & + 0.0028x_1x_3 \quad (25) \end{aligned}$$

Process parameter bounds

$$160 \leq x_1 \leq 200 \quad (26)$$

$$1 \leq x_2 \leq 5 \quad (27)$$

$$1.5 \leq x_3 \leq 3.5 \quad (28)$$

Shukla and Singh (2016a) optimized *Ktw* and *taper angle* individually using PSO, SA, FA, BH, CSA and BBO algorithms. Now Jaya algorithm is applied to individually optimize *Ktw* and *taper angle*. For this purpose a population size of 25, maximum number of iterations equal to 20 (i.e. maximum number of function evaluations equal to 500) and the number of independent runs equal to 10 are selected for Jaya algorithm. Shukla and Singh (2016a) did not provide the size of initial population used for PSO, FA, BH, CSA and BBO algorithms, therefore, the maximum number of function evaluations required by these algorithms cannot be determined.

The results of Jaya algorithm are compared with the results of algorithms such as PSO, FA, BH, CSA and BBO (Shukla and Singh 2016a) and are reported in Table 7. The optimal combination of input process parameters such as traverse speed, stand-off distance and mass flow rate obtained by different algorithms are shown in Table 7 along with the cor-

responding optimum values of *Ktw* and *taper angle*. It can be observed that the Jaya algorithm could achieve the best value of *Ktw* in only 2 iterations which is much better as compared to the number of iterations required by PSO, SA, FA, BH, CSA and BBO algorithms.

The Jaya algorithm could achieve a value of *taper angle* which is better than the best value of *taper angle* obtained by Shukla and Singh (2016a). The Jaya algorithm required only 6 iterations to obtain the minimum value of *taper angle* which is much better as compared to the number of iterations required by PSO, SA, FA, BH, CSA and BBO algorithms (refer Table 7). Furthermore, the Jaya algorithm was executed 10 times independently and the mean, standard deviation (SD) and CPU time required by Jaya algorithm for 10 independent runs is recorded and the same is compared with mean, SD and CPU time required by BBO algorithm in Table 8. It is observed that the Jaya algorithm achieved the same value of *Ktw* and *taper angle* for all 10 independent runs. Therefore, Jaya algorithm has shown more consistency and robustness and required less CPU time as compared to BBO algorithm. Figures 5 and 6 give the convergence graph of Jaya algorithm for *Ktw* and *taper angle*, respectively.

In order to achieve multiple trade-off solutions for *Ktw* and *taper angle* in AWJM process Shukla and Singh (2016a) used NSGA considering a population size of 50 and maximum number of iterations as 1000 (i.e. maximum number of function evaluations as 50,000). The algorithm specific parameters required by NSGA were chosen by Shukla and Singh (2016a) as follows: sharing fitness 1.2, dummy fitness 50. However, the values of crossover and mutation parameters required by NSGA were not reported by Shukla and Singh (2016a). Now, the same problem is solved using MO-

Table 8 Mean, standard deviation and CPU time required by BBO and Jaya algorithms for case study 3

| | K _{tw} | | Taper angle | |
|----------|-----------------|---------------|-------------|---------------|
| | BBO | Jaya | BBO | Jaya |
| Mean | 2.9135 | 2.9187 | 1.1019 | 1.0539 |
| SD | 0.0045 | 0 | 0.0035 | 0 |
| CPU time | 0.5112 | 0.417 | 1.1220 | 0.4166 |

The values in bold indicate better performance of the algorithm. The results of BBO are obtained from Shukla and Singh (2016a)

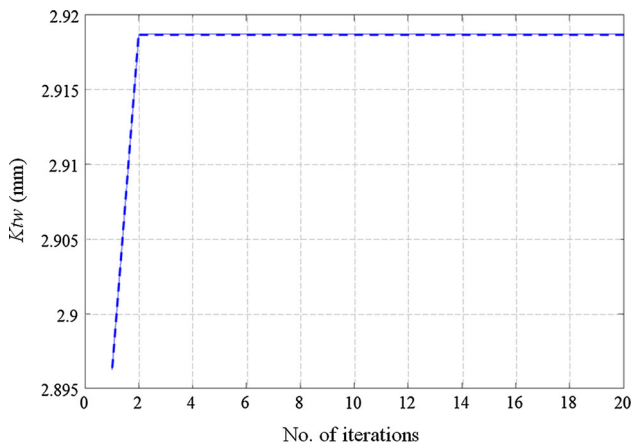


Fig. 5 Convergence graph of Jaya algorithm for maximization of *K_{tw}* (case study 3)

Jaya algorithm and NSTLBO algorithm (Rao et al. 2016). For fair comparison of results the maximum number of function evaluations for MO-Jaya and NSTLBO algorithms are maintained same as that of NSGA. For this purpose a population size of 100 and maximum number of iterations equal to 500 and 250 are chosen for MO-Jaya and NSTLBO algorithms, respectively. The Pareto-efficient set of solutions obtained by MO-Jaya algorithm in a single simulation run is reported in Table 9. In Table 9, optimal combinations of input process parameters such as traverse speed, stand-off distance and mass flow rate are provided, each solution corresponds to a trade-off between *K_{tw}* and *taper angle*. Figure 7 shows the Pareto-front obtained by NSGA, MO-Jaya and NSTLBO algorithms. In order to judge the quality of the Pareto-front obtained by the three algorithms the hypervolume (HV) metric is used (Zitzler and Thiele 1999). The value of HV metric achieved by MO-Jaya algorithm is higher than the NSTLBO and NSGA ($HV_{MO-Jaya} = 2.6028$; $HV_{NSTLBO} = 2.5995$ and $HV_{NSGA} = 2.5516$).

NSGA used 50,000 function evaluations to converge at the Pareto-efficient set of solutions. Only for fair comparison of results the maximum number of function evaluations for MO-Jaya and NSTLBO algorithms are maintained as 50,000. However, the MO-Jaya and NSTLBO algorithms required 700 and 1600 function evaluations, respectively to con-

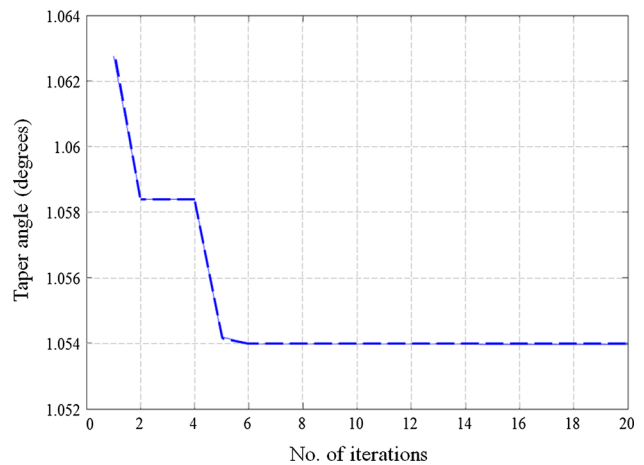


Fig. 6 Convergence graph of Jaya algorithm for minimization of taper angle (case study 3)

verge at the Pareto-efficient set of solutions. The CPU time required by MO-Jaya and NSTLBO algorithms to perform 50,000 function evaluations is 77.086 and 80.67 s, respectively. However, the CPU time required NSGA is not reported by Shukla and Singh (2016a).

The MO-Jaya algorithm has provided 100 Pareto-optimal solutions for the multi-objective optimization problem of AWJM process. The decision maker is now required to choose one out-of the 100 alternative solutions with *K_{tw}* and *taper angle* as selection criteria. This forms a decision making problem with two criteria and 100 alternatives considering *K_{tw}* and beneficial criteria and *taper angle* as non-beneficial criteria. To demonstrate the applicability of improved PROMETHEE method, three different set of weights are considered. The steps in the improved PROMETHEE method are similar to those discussed in “Optimization methodology” section. The best choice corresponding to each of the considered set of weights provided by improved PROMETHEE method is shown in Table 10.

Results and discussion

In the first case study, the results provided by MO-Jaya algorithm are justifiable and analogous with the working of AWJM process. In the AWJM process, at constant pressure as the nozzle diameter increases the kinetic energy of the abrasive particles decreases. Therefore, the surface roughness and *Kerf* increases with the increase in nozzle diameter. Therefore in order to obtain a lower value of surface roughness and *Kerf* the MO-Jaya algorithm has maintained the nozzle diameter to its lower bound value in order to ensure that the abrasive particles are discharged with a higher kinetic energy from the nozzle.

Table 9 Pareto-efficient solutions obtained using MO-Jaya algorithm for case study 3

| S. No. | x_1 | x_2 | x_3 | Taper angle (°) | Ktw (mm) |
|--------|----------|--------|--------|-----------------|----------|
| 1 | 192.8341 | 1 | 2.2969 | 1.054 | 1.4364 |
| 2 | 190.3569 | 1 | 2.3256 | 1.0575 | 1.4692 |
| 3 | 189.0174 | 1 | 2.3893 | 1.0646 | 1.4941 |
| 4 | 184.8933 | 1 | 2.4494 | 1.0963 | 1.5571 |
| 5 | 183.2294 | 1 | 2.4553 | 1.1135 | 1.5817 |
| 6 | 181.7402 | 1 | 2.3581 | 1.126 | 1.5926 |
| 7 | 183.1224 | 1 | 2.6216 | 1.14 | 1.6088 |
| 8 | 179.2901 | 1 | 2.3349 | 1.1615 | 1.6297 |
| 9 | 178.3591 | 1 | 2.4828 | 1.1835 | 1.6612 |
| 10 | 177.8894 | 1 | 2.536 | 1.1976 | 1.6759 |
| 11 | 176.6961 | 1 | 2.5357 | 1.2189 | 1.6958 |
| 12 | 177.2182 | 1 | 2.6921 | 1.2393 | 1.7109 |
| 13 | 175.1161 | 1 | 2.5117 | 1.2469 | 1.7202 |
| 14 | 175.9205 | 1 | 2.7512 | 1.2787 | 1.7427 |
| 15 | 174.0249 | 1 | 2.6695 | 1.2961 | 1.7614 |
| 16 | 171.902 | 1 | 2.5579 | 1.3245 | 1.7846 |
| 17 | 170.3026 | 1 | 2.3523 | 1.3534 | 1.7963 |
| 18 | 173.3885 | 1 | 2.8888 | 1.376 | 1.8122 |
| 19 | 169.3082 | 1 | 2.5461 | 1.39 | 1.8335 |
| 20 | 170.1231 | 1 | 2.7713 | 1.4141 | 1.8491 |
| 21 | 167.8296 | 1 | 2.5368 | 1.4308 | 1.8625 |
| 22 | 168.6643 | 1 | 2.7213 | 1.4397 | 1.8692 |
| 23 | 167.3267 | 1 | 2.6468 | 1.4618 | 1.8856 |
| 24 | 165.781 | 1 | 2.4384 | 1.4866 | 1.897 |
| 25 | 165.6612 | 1 | 2.548 | 1.4981 | 1.9091 |
| 26 | 164.5781 | 1 | 2.4693 | 1.527 | 1.9256 |
| 27 | 164.9521 | 1 | 2.6523 | 1.5363 | 1.9358 |
| 28 | 163.6484 | 1 | 2.5488 | 1.5646 | 1.953 |
| 29 | 161.9246 | 1 | 2.4196 | 1.6184 | 1.9825 |
| 30 | 163.1274 | 1 | 2.7874 | 1.6282 | 1.9934 |
| 31 | 160.903 | 1 | 2.5471 | 1.6628 | 2.0153 |
| 32 | 160 | 1 | 2.4136 | 1.6911 | 2.0277 |
| 33 | 160 | 1 | 2.7105 | 1.7223 | 2.053 |
| 34 | 160 | 1 | 2.8204 | 1.7502 | 2.0677 |
| 35 | 160.3105 | 1.1206 | 2.6007 | 1.7757 | 2.0684 |
| 36 | 160.0922 | 1 | 3.001 | 1.8118 | 2.0961 |
| 37 | 160 | 1 | 3.0497 | 1.8369 | 2.1077 |
| 38 | 160 | 1 | 3.111 | 1.8665 | 2.1206 |
| 39 | 160 | 1 | 3.153 | 1.8884 | 2.1299 |
| 40 | 160.4974 | 1 | 3.2392 | 1.9194 | 2.1398 |
| 41 | 160.2468 | 1 | 3.2538 | 1.9373 | 2.1487 |
| 42 | 160 | 1.1717 | 3.0692 | 1.9652 | 2.1606 |
| 43 | 160 | 1 | 3.3132 | 1.9839 | 2.1694 |
| 44 | 160.9715 | 1.2477 | 3.1271 | 2.0104 | 2.1735 |
| 45 | 160.3978 | 1 | 3.4275 | 2.0493 | 2.193 |
| 46 | 160 | 1 | 3.4407 | 2.0733 | 2.2052 |
| 47 | 160.1419 | 1.2956 | 3.1936 | 2.1078 | 2.2194 |
| 48 | 160 | 1.441 | 3.0533 | 2.1343 | 2.2305 |

Table 9 continued

| S. No. | x_1 | x_2 | x_3 | Taper angle (°) | Ktw (mm) |
|--------|----------|--------|--------|-----------------|----------|
| 49 | 160 | 1.1261 | 3.428 | 2.1519 | 2.2375 |
| 50 | 160 | 1.2261 | 3.3659 | 2.1754 | 2.2477 |
| 51 | 160 | 1.5655 | 3.0326 | 2.202 | 2.2588 |
| 52 | 160 | 1.4557 | 3.2306 | 2.2374 | 2.2742 |
| 53 | 160 | 1.4891 | 3.2233 | 2.2539 | 2.2811 |
| 54 | 160 | 1.2508 | 3.5 | 2.2913 | 2.2939 |
| 55 | 160 | 1.6316 | 3.2604 | 2.3633 | 2.327 |
| 56 | 160 | 1.7969 | 3.1443 | 2.3931 | 2.3403 |
| 57 | 160 | 1.9638 | 3.0159 | 2.4234 | 2.3532 |
| 58 | 160 | 1.5122 | 3.5 | 2.4592 | 2.3638 |
| 59 | 160 | 1.8912 | 3.2031 | 2.4785 | 2.3766 |
| 60 | 160 | 1.8239 | 3.2803 | 2.4874 | 2.3796 |
| 61 | 160 | 1.6459 | 3.4822 | 2.5264 | 2.3926 |
| 62 | 160 | 2.117 | 3.1381 | 2.5628 | 2.4139 |
| 63 | 160 | 2.2039 | 3.1078 | 2.591 | 2.4265 |
| 64 | 160 | 1.7834 | 3.5 | 2.6206 | 2.4322 |
| 65 | 160 | 2.0761 | 3.3131 | 2.6451 | 2.4474 |
| 66 | 160 | 2.6073 | 2.8463 | 2.6702 | 2.4622 |
| 67 | 160 | 2.7077 | 2.7865 | 2.6946 | 2.4729 |
| 68 | 160 | 2.5349 | 3.0574 | 2.7214 | 2.4856 |
| 69 | 160 | 2.2855 | 3.3213 | 2.7553 | 2.4959 |
| 70 | 160 | 2.9505 | 2.7807 | 2.7868 | 2.5168 |
| 71 | 160 | 3.0759 | 2.672 | 2.8067 | 2.5249 |
| 72 | 160 | 2.1918 | 3.5 | 2.839 | 2.5271 |
| 73 | 173.4034 | 5 | 2.6611 | 2.8604 | 2.5525 |
| 74 | 160 | 3.3135 | 2.6889 | 2.8864 | 2.5655 |
| 75 | 160 | 2.5849 | 3.3567 | 2.9157 | 2.5674 |
| 76 | 160 | 3.2944 | 2.8314 | 2.9162 | 2.5814 |
| 77 | 169.481 | 5 | 2.6634 | 2.9194 | 2.6011 |
| 78 | 167.0056 | 5 | 2.4094 | 2.9389 | 2.609 |
| 79 | 168.88 | 5 | 2.763 | 2.954 | 2.624 |
| 80 | 166.0315 | 5 | 2.667 | 2.9868 | 2.6491 |
| 81 | 163.9872 | 5 | 2.4796 | 3.0097 | 2.6606 |
| 82 | 165.0937 | 5 | 2.7381 | 3.0232 | 2.6724 |
| 83 | 163.3434 | 5 | 2.6442 | 3.0445 | 2.6867 |
| 84 | 162.0704 | 5 | 2.4139 | 3.0569 | 2.6876 |
| 85 | 163.0084 | 5 | 2.787 | 3.0851 | 2.7107 |
| 86 | 160 | 5 | 2.4514 | 3.1155 | 2.7254 |
| 87 | 160 | 5 | 2.7439 | 3.1538 | 2.7527 |
| 88 | 160.4855 | 5 | 2.8104 | 3.1577 | 2.7539 |
| 89 | 160 | 4.9389 | 2.9001 | 3.1995 | 2.7721 |
| 90 | 160 | 4.9319 | 2.9085 | 3.2024 | 2.7731 |
| 91 | 160 | 5 | 3.0448 | 3.2586 | 2.8024 |
| 92 | 161.5897 | 5 | 3.1958 | 3.2944 | 2.8118 |
| 93 | 160 | 5 | 3.1407 | 3.3059 | 2.8227 |
| 94 | 160 | 5 | 3.1851 | 3.3301 | 2.8329 |
| 95 | 162.227 | 4.8823 | 3.3429 | 3.369 | 2.8335 |
| 96 | 160 | 5 | 3.2683 | 3.3792 | 2.8533 |

Table 9 continued

| S. No. | x_1 | x_2 | x_3 | Taper angle (°) | K _{rw} (mm) |
|--------|-------|--------|--------|-----------------|----------------------|
| 97 | 160 | 4.9694 | 3.2964 | 3.3969 | 2.8587 |
| 98 | 160 | 5 | 3.3313 | 3.4198 | 2.8698 |
| 99 | 160 | 5 | 3.3725 | 3.448 | 2.8811 |
| 100 | 160 | 5 | 3.4936 | 3.5378 | 2.9168 |

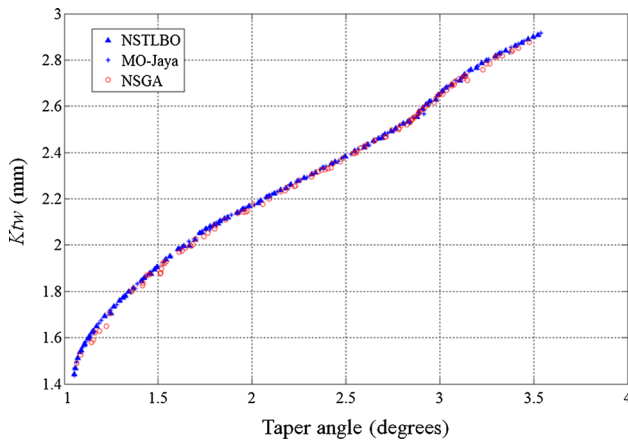


Fig. 7 Pareto-fronts obtained by NSGA (Shukla and Singh 2016a), MO-Jaya and NSTLBO for case study 3

As the traverse speed increase the faster passing of waterjet results in less number of abrasive particles impinging on the workpiece surface, resulting into formation of a narrow slot which reduces the *Kerf*. However, at low traverse speed more number of abrasive particles are available for eroding the material from the workpiece which results in a lower surface roughness. Accordingly, the MO-Jaya algorithm has selected the lower bound value of traverse speed (i.e. 200 mm/min) to achieve a minimum value of *Kerf* (i.e. 0.691 mm) and the MO-Jaya algorithm has selected the upper bound value of traverse speed (i.e. 600 mm/min) to achieve minimum value of surface roughness (i.e. 4.9247 μm). In order to obtain the intermediate trade-off solutions the MO-Jaya algorithm has appropriately selected the values of traverse speed within the working range.

Shukla and Singh (2016b) developed their own regression models for *Kerf* and R_a using the experimental of Kechagias et al. (2012). However, the regression model developed by

Shukla and Singh (2016b) for *Kerf* provides negative value of *Kerf* for the certain combination of process parameters the values which lie on their respective bounds which is practically incorrect. In the case of R_a the regression model developed by Shukla and Singh (2016b) is solved using Jaya algorithm. The Jaya algorithm minimum value of R_a obtained by Jaya algorithm is 4.431729 μm which is lower than the value of R_a obtained by Shukla and Singh (2016b) which is 4.443 μm. Furthermore, Shukla and Singh (2016b) did not report the number of iterations and computational time required by firefly (FA) algorithm to achieve the minimum value of R_a . However, Jaya algorithm required only 6 iterations with a population size of 25 and maximum number of iterations equal to 20 and a computational time of 0.052 s was required by Jaya algorithm to obtain the minimum value of R_a (i.e. for $x_1 = 0.9000$ mm; $x_2 = 0.9500$ mm; $x_3 = 20$ mm; $x_4 = 200$ mm/min.; $R_{a-} = 4.431729$ μm) using regression model developed by Shukla and Singh (2016b).

In the second case study, the results of MO-Jaya algorithm show that, *DOP* increases as the traverse speed decreases. This is mainly because at higher value of traverse speed the abrasive particles that impinge on a target area are very less which tend to reduce the *DOP*. Therefore to achieve maximum value of *DOP* (i.e. 405.194 μm) the MO-Jaya algorithm has maintained the traverse speed to its lower bound (i.e. 0.1 mm/s). According to Bernoulli’s principle as the waterjet pressure increases the energy also increase which results in higher *DOP*. Therefore to achieve the higher value of *DOP* the MO-Jaya algorithm as maintained the waterjet pressure to its upper bound value (i.e. 320 MPa). At lower value of stand-off distance the particle interference increases which reduces the energy and velocity of the particles impinging on the workpiece surface. Therefore, in order to achieve a high *DOP* the MO-Jaya algorithm has maintained the stand-off distance to its upper bound value (i.e. 10 mm).

Table 10 The best choice provided by PROMTHEE method corresponding to different set of weights for case study 3

| S. No. | Weights assigned to the criteria | | Best choice provided by PROMTHEE method | | | | |
|--------|----------------------------------|-------|---|-------|--------|--------------------|-------------|
| | w_1 | w_2 | Process parameters | | | Criteria/responses | |
| | | | x_1 | x_2 | x_3 | K _{tw} | Taper angle |
| 1 | 0.9 | 0.1 | 192.8341 | 1 | 2.2969 | 1.054 | 1.4364 |
| 2 | 0.1 | 0.9 | 160 | 5 | 3.4936 | 3.5378 | 2.9168 |
| 3 | 0.5 | 0.5 | 160 | 1 | 2.7105 | 1.7223 | 2.053 |

As the abrasive flow rate increases the number of abrasive particles available for material removal also increases. Therefore to achieve a higher *DOP* the MO-Jaya algorithm has maintained the abrasive flow rate mostly close to the upper bound. Although, the effect of surface speed and tilt angle seems to be negligible on *DOP*. However, tilt angle and waterjet pressure have a synergistic effect on *DOP*. It is observed from the results of MO-Jaya algorithm that maximum *DOP* is achieved at higher tilt angle and higher pressure.

At higher peripheral speed the workpiece chips are cleaned effectively. This reduces the damage caused to the workpiece surface due to formation of chips. Therefore, to achieve minimum surface roughness the MO-Jaya algorithm has maintained a higher value of peripheral speed (i.e. 7.99 m/s). As the pressure increases the jet diameter increases, and the overlapping of larger effective jet produces smoother surface. As waterjet pressure is beneficial for a *DOP* as well as R_a the Mo-Jaya algorithm has maintained waterjet pressure to its upper bound value (i.e. 320 MPa). The traverse speed and abrasive flow rate have a synergistic effect on R_a . The results of MO-Jaya algorithm show that minimum value of R_a is achieved at higher value of traverse speed and higher abrasive flow rate (refer Table 4).

In the third case study, the results provided by MO-Jaya algorithm are logical and analogous with the working of AWJM process. In the AWJM process, as the stand-off distance increases the divergence of jet before impingement on the work-piece also increases due to the external drag from the surrounding environment. Therefore, the kinetic energy of the abrasive particles at the periphery of the jet is less as compared to abrasive particles at the centerline of the jet. Therefore, the material removal at the centerline is higher as compared to the material removal at the periphery. Thus, the kerf top width and taper angle increases with increase in stand-off distance. Accordingly, the results of MO-Jaya algorithm in Table 9 show that as the *taper angle* and *K_{tw}* increases stand-off distance increases from 1 to 5 mm.

At a constant nozzle diameter, as the mass flow rate increases the kinetic energy of the abrasive particles increases. This reduces flaring of abrasive waterjet and a jet of uniform diameter is produced. High kinetic energy results in a narrow slot of uniform width. Therefore, as the mass flow rate increases the kerf top width and taper angle decrease. However, as the mass flow rate increases beyond a certain critical value kerf top width and taper angle increase with the increase in mass flow rate. Therefore, in order to achieve a trade-off between kerf top width and taper angle the MO-Jaya algorithm has appropriately varied the mass flow rate within the given range.

In the AWJM process as the traverse speed increases the faster passing of waterjet allows less number of abrasive par-

ticles to strike on the target material generating a narrow slot which reduces the kerf top width and taper angle. As the traverse speed decreases the number of abrasive particles striking the target material also increases. This increases the erosion of the workpiece material at the walls of the slot especially near the top surface which increases the *K_{tw}* and *taper angle*. The results of MO-Jaya algorithm (refer Table 9) show that minimum value of taper angle is achieved for the value of traverse speed close to the upper bound and a maximum value of kerf top width is achieved for the value of traverse speed close to the lower bound.

The maximum value of *K_{tw}* achieved by NSGA is 2.8768 mm. However, the maximum value of *K_{tw}* achieved by MO-Jaya algorithm is 2.9168 mm which is 1.39% higher than NSGA. The minimum value of taper angle achieved by NSGA is 1.0884. The minimum value of taper angle achieved by MO-Jaya algorithm is 1.054 which is 3.16% lower than NSGA.

Conclusions

The single-objective and multi-objective optimization aspects of abrasive waterjet machining process are considered in this work. Three optimization case studies are formulated based on the regression models developed by previous researchers. The performance measures such as depth of cut, surface roughness, kerf geometry and taper angle are optimized using Jaya and MO-Jaya algorithms.

The Jaya algorithm and MO-Jaya algorithm have provided better results as compared to other optimization algorithms such as SA, PSO, FA, CSA, BH, BBO, NSGA and NSTLBO algorithms. In the case of multi-objective optimization problems the Pareto-fronts provided by MO-Jaya algorithm are compared with the Pareto-fronts provided by NSGA and NSTLBO algorithms. The MO-Jaya algorithm achieved a higher value of hypervolume metric as compared to NSGA and NSTLBO algorithms. Thus the quality of Pareto-fronts provided by MO-Jaya algorithm is better than that of NSGA and NSTLBO algorithms for the optimization problems considered in this work.

In case study 1, the MO-Jaya algorithm obtained better results as compared to the NSTLBO algorithm requiring less CPU time for its execution. In case study 2, the MO-Jaya algorithm obtained better solutions as compared to NSTLBO algorithm and desirability function approach. The CPU time required by the MO-Jaya algorithm is less as compared to NSTLBO algorithm. In case study 3, the Jaya algorithm obtained better results as compared to BBO algorithm and the CPU time required by the Jaya algorithm is lower than that of BBO algorithm. The MO-Jaya algorithm obtained a better Pareto-front as compared to NSGA within a very less computational time.

It is observed that, the working of Jaya and MO-Jaya algorithms is dependent only on common control parameters like population size and number of generations and there is no need of tuning of any algorithm-specific parameters. The computational complexity of MO-Jaya algorithm is less as compared to NSTLBO, NSGA and BBO algorithms because in these algorithms the solutions are updated in multiple phases. However, in the Jaya and MO-Jaya algorithms the solutions are updated in a single phase using a single equation. Therefore, the two main advantages of Jaya and MO-Jaya algorithms are: (1) the Jaya and MO-Jaya algorithms do not burden the user with the task of tuning the algorithm-specific parameters. (2) The Jaya and MO-Jaya algorithms are simple in implementation and require less CPU time for their working.

The effects of the best and worst solutions in the current population are considered simultaneously which gives a high convergence speed to MO-Jaya algorithm without trapping into local optima. The ranking mechanism based on the concept of non-dominance relation between the solutions helps MO-Jaya algorithm to maintain the good solutions in every generation and guides the search process towards the Pareto-optimal set.

Furthermore, an improved PROMETHEE method is used to select a suitable solution from multiple Pareto-optimal solutions provided by MO-Jaya algorithm based on his/her order of preference. The improved PROMETHEE is a more objective and logical selection approach as it allows the decision maker to systematically assign the values of relative importance to the criteria based on his/her preferences. In the improved PROMETHEE method the alternatives are evaluated with better accuracy as the values of the criteria and their relative importance are considered simultaneously.

The Pareto optimal set of solutions provided by MO-Jaya algorithm for the three case studies considered in this work are helpful to the process planner as it contains a wide range of optimal values. This enables the process planner to choose a particular solution from the Pareto optimal set that corresponds to a specific order of importance of objectives, in order to achieve the desired outcome from the machining process even in situations where the order of importance of objectives is subjected to frequent changes. Thus, the results presented in this work are useful for real production systems.

Acknowledgements The Authors are thankful to the Department of Science and Technology (DST), India and the Slovenian Research Agency (ARRS), Slovenia for providing the financial support for the project entitled “Optimization of Sustainable Advanced Manufacturing Processes”.

Appendix A

Now the steps of the improved PROMETHEE method for selection of an optimal solution from the set of Pareto-optimal solution based on the order preference of the decision maker are demonstrated. For the purpose of demonstration the multi-objective optimization problem of AWJM process described in “Optimization methodology” section is considered.

- Step 1** A multi-objective decision making problem for AWJM process is formulated by considering the 50 Pareto-optimal solutions provided by MO-Jaya algorithm as alternatives and the multiple objectives such as $Kerf$ and R_a are considered as selection criteria. Both the criteria are considered as non-beneficial criteria. Table 11 shows the objective data for multi-objective decision problem of AWJM process. In Table 11, the fifty solutions are considered as alternatives while $Kerf$ and R_a are considered as two attributes.
- Step 2** The next step is to assign the weights of the criteria based on the order preference of decision maker. For the purpose of demonstration equal importance is assigned to all the criteria (i.e. $w_1 = w_2 = 0.5$). However, if a decision maker wants to assign unequal weights of relative importance to the criteria then he/she may use the analytical hierarchy process (AHP) method for determining weights of relative importance, explained by Rao and Patel (2010).
- Step 3** After assigning the weights of relative importance to the criteria next step is to select a preference function. Rao and Patel (2010) had suggested six preference functions namely, usual function, U-shape function, V-shape function, level function, linear function and Gaussian function. In this work the V-shape function is used as the preference function. For more details about calculating the values of preference indices using V-shape preference function the readers may refer to Rao and Patel (2010).
- Step 4** Now each alternative is to be compared with every other alternative and the preference indices are to be calculated considering each criterion separately. For the purpose of demonstration, the preference indices resulting from pairwise comparison of 50 alternative solutions with respect to the criteria $Kerf$ and R_a are shown in Tables 12 and 13, respectively.

Table 11 Objective data for multi-objective decision making problem of AWJM process (case study 1)

| S. No. | <i>Kerf</i> (mm) | <i>R_a</i> (μm) |
|--------|------------------|---------------------------|
| 1 | 0.12453 | 0.00965 |
| 2 | 0.12865 | 0.00986 |
| 3 | 0.13199 | 0.00996 |
| 4 | 0.14191 | 0.01162 |
| 5 | 0.14245 | 0.02215 |
| 6 | 0.25179 | 0.02405 |
| 7 | 0.30999 | 0.02672 |
| 8 | 0.50426 | 0.04827 |
| 9 | 0.55058 | 0.05499 |
| 10 | 0.61636 | 0.06041 |
| 11 | 0.90452 | 0.10027 |
| 12 | 0.94074 | 0.1871 |
| 13 | 1.00145 | 0.11314 |
| 14 | 1.1047 | 0.13154 |
| 15 | 1.1201 | 0.1547 |
| 16 | 1.13644 | 0.13979 |
| 17 | 1.1776 | 0.18677 |
| 18 | 1.27649 | 0.15735 |
| 19 | 1.42355 | 0.19259 |
| 20 | 1.63417 | 0.21735 |
| 21 | 1.71198 | 0.23377 |
| 22 | 1.86777 | 0.3184 |
| 23 | 1.95525 | 0.27339 |
| 24 | 2.03185 | 0.46091 |
| 25 | 2.04793 | 0.41102 |
| 26 | 2.20527 | 0.34706 |
| 27 | 2.31194 | 0.35922 |
| 28 | 2.37081 | 0.36683 |
| 29 | 2.48563 | 0.57193 |
| 30 | 2.59749 | 0.42886 |
| 31 | 2.68204 | 0.44199 |
| 32 | 2.68784 | 14.11848 |
| 33 | 2.68928 | 16.81049 |
| 34 | 2.7032 | 15.3657 |
| 35 | 2.72314 | 15.59124 |
| 36 | 2.75357 | 16.42039 |
| 37 | 2.76576 | 17.6157 |
| 38 | 2.79858 | 20.06127 |
| 39 | 2.83591 | 20.83648 |
| 40 | 2.8454 | 18.59188 |
| 41 | 2.95533 | 20.57847 |
| 42 | 2.96352 | 20.85368 |
| 43 | 2.98885 | 22.47545 |
| 44 | 3.0121 | 21.81782 |
| 45 | 3.042 | 22.8327 |
| 46 | 3.06445 | 23.34057 |

Table 11 continued

| S. No. | <i>Kerf</i> (mm) | <i>R_a</i> (μm) |
|--------|------------------|---------------------------|
| 47 | 3.07501 | 24.64366 |
| 48 | 3.08257 | 24.33128 |
| 49 | 3.08293 | 26.09632 |
| 50 | 3.10207 | 25.64056 |

Step 5 Once the preference values resulting from pairwise comparison of each alternative with every other alternative is calculated for each criteria separately, the next step is to determine the weighted average of the preference functions according to Eq. (5). The weighted average of preference function values are shown in Table 14.

Step 6 The values of leaving flow, entering flow and net flow are calculated for each alternative according to Eqs. (6), (7) and (8). Considering, the values of net flow rank is assigned to each alternative following the higher-the-better approach. The values of leaving flow, entering flow, net flow and rank of each alternative is shown in Table 14.

Table 14 shows that for equal importance to both the criteria (i.e. $w_1 = w_2 = 0.5$) the improved PROMETHEE method has suggested alternative solution no. 40 as the first choice. Mainly for the purpose of demonstration equal weights are considered for all the criteria. However, if a decision maker wants to assign unequal weights to the criteria then he/she may use the analytical hierarchy process (AHP) method for determining weights of relative importance, explained by Rao and Patel (2010).

Appendix B

In order to demonstrate the working of MO-Jaya algorithm, the multi-objective optimization problem formulated in “Optimization methodology” section is considered in this work. The objective functions are expressed by Eqs. (10) and (11). The input process parameters and their respective upper and lower bounds are expressed by Eqs. (12) to (15). For the purpose of demonstration, a population size of five is considered and two iterations of MO-Jaya algorithm are shown now. The randomly generated initial population is shown in Table 15. The ranks are assigned to each solution based on the ranking methodology discussed in “Ranking methodology” section and crowding distance (CD) is computed as discussed in “Computing the crowding distance” section. The solution

Table 14 Leaving, entering, net flow values and ranking of fifty alternatives

| Alternatives | $\varphi^+(a)$ | $\varphi^-(a)$ | $\varphi(a)$ | Rank |
|--------------|----------------|----------------|--------------|------|
| 1 | 9.4612 | 14.6649 | -5.2037 | 49 |
| 2 | 9.2563 | 14.3016 | -5.0454 | 48 |
| 3 | 8.9376 | 13.7388 | -4.8012 | 47 |
| 4 | 8.6601 | 13.2633 | -4.6032 | 46 |
| 5 | 7.9303 | 11.6296 | -3.6993 | 44 |
| 6 | 7.712 | 11.0881 | -3.376 | 43 |
| 7 | 7.4722 | 10.4505 | -2.9782 | 40 |
| 8 | 7.3377 | 10.0702 | -2.7326 | 39 |
| 9 | 7.2169 | 9.7439 | -2.527 | 38 |
| 10 | 7.0294 | 9.2445 | -2.2151 | 37 |
| 11 | 6.7924 | 8.5574 | -1.765 | 36 |
| 12 | 6.7493 | 8.4347 | -1.6853 | 34 |
| 13 | 6.5015 | 7.5886 | -1.0871 | 33 |
| 14 | 6.3084 | 6.6487 | -0.3403 | 31 |
| 15 | 6.2493 | 6.2387 | 0.0106 | 30 |
| 16 | 6.2063 | 5.8295 | 0.3768 | 29 |
| 17 | 5.9927 | 5.28 | 0.7127 | 27 |
| 18 | 5.9116 | 5.0497 | 0.8618 | 26 |
| 19 | 5.8677 | 4.947 | 0.9207 | 25 |
| 20 | 5.8275 | 4.7411 | 1.0864 | 24 |
| 21 | 5.8096 | 4.5953 | 1.2143 | 23 |
| 22 | 5.7873 | 4.4253 | 1.362 | 21 |
| 23 | 5.7864 | 4.3909 | 1.3955 | 20 |
| 24 | 5.8024 | 4.2524 | 1.55 | 18 |
| 25 | 5.7594 | 4.2723 | 1.487 | 19 |
| 26 | 5.8751 | 4.1179 | 1.7572 | 17 |
| 27 | 5.9645 | 4.057 | 1.9075 | 15 |
| 28 | 6.1646 | 4.0169 | 2.1477 | 14 |
| 29 | 6.2549 | 4.0152 | 2.2397 | 13 |
| 30 | 6.42 | 4.0376 | 2.3825 | 12 |
| 31 | 6.4601 | 4.0443 | 2.4158 | 11 |
| 32 | 6.6469 | 4.0796 | 2.5673 | 10 |
| 33 | 6.7866 | 4.1181 | 2.6685 | 9 |
| 34 | 7.123 | 4.2476 | 2.8754 | 8 |
| 35 | 7.8158 | 4.6487 | 3.1671 | 6 |
| 36 | 8.2109 | 4.7418 | 3.4691 | 5 |
| 37 | 8.743 | 5.0614 | 3.6815 | 4 |
| 38 | 9.1207 | 5.3631 | 3.7576 | 2 |
| 39 | 9.5917 | 5.8414 | 3.7503 | 3 |
| 40 | 10.0937 | 6.3341 | 3.7596 | 1 |
| 41 | 10.1203 | 7.1329 | 2.9874 | 7 |
| 42 | 10.0506 | 8.197 | 1.8536 | 16 |
| 43 | 10.0315 | 8.6805 | 1.351 | 22 |
| 44 | 10.0088 | 9.4044 | 0.6044 | 28 |
| 45 | 10.0016 | 10.7089 | -0.7072 | 32 |
| 46 | 10.0403 | 11.8021 | -1.7618 | 35 |

Table 14 continued

| Alternatives | $\varphi^+(a)$ | $\varphi^-(a)$ | $\varphi(a)$ | Rank |
|--------------|----------------|----------------|--------------|------|
| 47 | 10.1252 | 13.1616 | -3.0364 | 41 |
| 48 | 10.154 | 13.4979 | -3.3439 | 42 |
| 49 | 10.2364 | 14.4448 | -4.2084 | 45 |
| 50 | 10.3351 | 15.5388 | -5.2037 | 50 |

with the highest rank and highest CD value is the best solution and the solution with the lowest rank is selected as the worst solution.

Now the new values of the variables are calculated according to Eq. (1). For the purpose of demonstration, the random numbers α_1 and α_2 for variable *A* are considered as 0.91 and 0.15, respectively. The random numbers α_1 and α_2 for variable *B* are considered as 0.67 and 0.5, respectively. The random numbers α_1 and α_2 for variable *C* are considered as 0.25 and 0.65, respectively. The random numbers α_1 and α_2 for variable *D* are considered as 0.36 and 0.75, respectively. The new values of variables *A*, *B*, *C* and *D* are calculated and are shown in Table 16 along with the corresponding values of *Kerf* and *R_a*.

Now the new solutions are combined with the initial solutions and a combined population is formed. The combined population is shown in Table 17. The ranks and CD value are determined for all the solutions in the combined population. Now based on the ranks and CD value five good solutions are selected from the combined population. These five solutions which are shown in Table 18 will act as initial population for the next iteration.

Iteration 2

The solutions selected at the end of the first iteration, shown in Table 18 are used as initial population for the second iteration. Now in the second iteration, the initial solutions (Table 18) are modified according to Eq. (1) and are shown in Table 19. For the purpose of demonstration the random numbers α_1 and α_2 for variable *A* are considered as 0.54 and 0.29, respectively. The random number α_1 and α_2 for variable *B* are considered as 0.48 and 0.37, respectively. The random numbers α_1 and α_2 for variable *C* are considered as 0.62 and 0.14, respectively. The random numbers α_1 and α_2 for variable *D* are considered as 0.73 and 0.55, respectively.

The combined population is shown in Table 20. Five good solutions are selected from the combined population based on the ranks and CD which are shown in Table 21. These five solutions will act as initial solutions for the third iteration of MO-Jaya algorithm.

Table 15 Initial population

| S. No. | A | B | C | D | Kerf | R _a | Rank | CD | Status |
|--------|--------|--------|---------|----------|--------|----------------|------|----|--------|
| 1 | 1.1852 | 1.0036 | 30.7834 | 263.0452 | 1.0109 | 5.5867 | 1 | ∞ | Best |
| 2 | 1.2170 | 1.1032 | 52.0539 | 588.2371 | 0.9848 | 7.3432 | 1 | ∞ | |
| 3 | 0.9444 | 1.2508 | 89.5959 | 582.8668 | 1.3421 | 8.6899 | 2 | ∞ | |
| 4 | 1.2197 | 1.4766 | 80.2078 | 394.1503 | 1.5503 | 7.2622 | 2 | ∞ | |
| 5 | 1.1213 | 1.4807 | 92.9214 | 520.1122 | 1.6285 | 8.2329 | 3 | ∞ | Worst |

Table 16 New values of the variables and objective functions

| S. No. | A | B | C | D | Kerf | R _a |
|--------|--------|-------------------|-----------------|------------------|--------|----------------|
| 1 | 1.1948 | 0.95 ^a | 20 ^a | 200 ^a | 0.9667 | 5.04 |
| 2 | 1.2024 | 0.95 ^a | 20.1724 | 522.262 | 0.8263 | 6.617 |
| 3 | 1.1370 | 0.9702 | 72.7312 | 514.797 | 0.8737 | 7.51 |
| 4 | 1.2031 | 1.1576 | 59.588 | 252.481 | 1.2126 | 6.241 |
| 5 | 1.1794 | 1.1610 | 77.3869 | 427.568 | 1.1762 | 7.264 |

^aThis value has crossed the given range of the variable and hence it is assigned the bound value

Table 17 Combined population

| S. No. | A | B | C | D | Kerf | R _a | Rank | CD |
|--------|--------|--------|---------|----------|--------|----------------|------|---------|
| 1 | 1.1852 | 1.0036 | 30.7834 | 263.0452 | 1.0109 | 5.5867 | 2 | ∞ |
| 2 | 1.2170 | 1.1032 | 52.0539 | 588.2371 | 0.9848 | 7.3432 | 2 | 0.69798 |
| 3 | 0.9444 | 1.2508 | 89.5959 | 582.8668 | 1.3421 | 8.6899 | 4 | ∞ |
| 4 | 1.2197 | 1.4766 | 80.2078 | 394.1503 | 1.5503 | 7.2622 | 4 | ∞ |
| 5 | 1.1213 | 1.4807 | 92.9214 | 520.1122 | 1.6285 | 8.2329 | 5 | – |
| 6 | 1.1948 | 0.95 | 20 | 200 | 0.9667 | 5.04 | 1 | ∞ |
| 7 | 1.2024 | 0.95 | 20.1724 | 522.262 | 0.8263 | 6.617 | 1 | ∞ |
| 8 | 1.1370 | 0.9702 | 72.7312 | 514.797 | 0.8737 | 7.51 | 2 | ∞ |
| 9 | 1.2031 | 1.1576 | 59.588 | 252.481 | 1.2126 | 6.241 | 3 | ∞ |
| 10 | 1.1794 | 1.1610 | 77.3869 | 427.568 | 1.1762 | 7.264 | 3 | ∞ |

Table 18 Selection of candidate solutions based on non-dominance rank and crowding distance

| S. No. | A | B | C | D | Kerf | R _a | Rank | CD | Status |
|--------|--------|--------|---------|----------|--------|----------------|------|---------|--------|
| 1 | 1.1948 | 0.95 | 20 | 200 | 0.9667 | 5.04 | 1 | ∞ | Best |
| 2 | 1.2024 | 0.95 | 20.1724 | 522.262 | 0.8263 | 6.617 | 1 | ∞ | |
| 3 | 1.1852 | 1.0036 | 30.7834 | 263.0452 | 1.0109 | 5.5867 | 2 | ∞ | |
| 4 | 1.1370 | 0.9702 | 72.7312 | 514.797 | 0.8737 | 7.51 | 2 | ∞ | |
| 5 | 1.2170 | 1.1032 | 52.0539 | 588.2371 | 0.9848 | 7.3432 | 2 | 0.69798 | Worst |

Table 19 New values of the variables and objective functions

| S. No. | A | B | C | D | Kerf | R _a |
|--------|--------|-------------------|-----------------|------------------|--------|----------------|
| 1 | 1.1883 | 0.95 ^a | 20 ^a | 200 ^a | 0.969 | 5.0373 |
| 2 | 1.1941 | 0.95 ^a | 20 ^a | 250.7244 | 0.945 | 5.2870 |
| 3 | 1.1812 | 0.95 ^a | 21.1198 | 200 ^a | 0.9756 | 5.0522 |
| 4 | 1.1450 | 0.95 ^a | 42.9326 | 244.603 | 1.0261 | 5.6064 |
| 5 | 1.2050 | 1.0297 | 32.1804 | 304.824 | 1.0097 | 5.8179 |

^aThis value has crossed the given range of the variable and hence it is assigned the bound value

Table 20 Combined population

| S. No. | A | B | C | D | Kerf | R _a | Rank | CD |
|--------|--------|--------|---------|----------|--------|----------------|------|---------|
| 1 | 1.1948 | 0.95 | 20 | 200 | 0.9667 | 5.04 | 1 | 0.2211 |
| 2 | 1.2024 | 0.95 | 20.1724 | 522.262 | 0.8263 | 6.617 | 1 | ∞ |
| 3 | 1.1852 | 1.0036 | 30.7834 | 263.0452 | 1.0109 | 5.5867 | 3 | ∞ |
| 4 | 1.1370 | 0.9702 | 72.7312 | 514.797 | 0.8737 | 7.51 | 2 | ∞ |
| 5 | 1.2170 | 1.1032 | 52.0539 | 588.2371 | 0.9848 | 7.3432 | 3 | ∞ |
| 6 | 1.1883 | 0.95 | 20 | 200 | 0.969 | 5.0373 | 1 | ∞ |
| 7 | 1.1941 | 0.95 | 20 | 250.7244 | 0.945 | 5.2870 | 1 | 1.34046 |
| 8 | 1.1812 | 0.95 | 21.1198 | 200 | 0.9756 | 5.0522 | 2 | ∞ |
| 9 | 1.1450 | 0.95 | 42.9326 | 244.603 | 1.0261 | 5.6064 | 4 | ∞ |
| 10 | 1.2050 | 1.0297 | 32.1804 | 304.824 | 1.0097 | 5.8179 | 3 | 0.84098 |

Table 21 Selection of candidate solutions based on non-dominance rank and crowding distance

| S. No. | A | B | C | D | Kerf | R _a | Rank | CD | Status |
|--------|--------|------|---------|----------|--------|----------------|------|---------|--------|
| 1 | 1.2024 | 0.95 | 20.1724 | 522.262 | 0.8263 | 6.617 | 1 | ∞ | Best |
| 2 | 1.1883 | 0.95 | 20 | 200 | 0.969 | 5.0373 | 1 | ∞ | |
| 3 | 1.1941 | 0.95 | 20 | 250.7244 | 0.945 | 5.2870 | 1 | 1.34046 | |
| 4 | 1.1948 | 0.95 | 20 | 200 | 0.9667 | 5.04 | 1 | 0.2211 | |
| 5 | 1.1812 | 0.95 | 21.1198 | 200 | 0.9756 | 5.0522 | 2 | ∞ | Worst |

References

- Aydin, G., Karakurt, I., & Hamzacebi, C. (2014). Artificial neural network and regression models for performance prediction of abrasive waterjet in rock cutting. *International Journal of Advanced Manufacturing Technology*, 75, 1321–1330.
- Chandrasekaran, M., Muralidhar, M., Krishna, C. M., & Dixit, U. S. (2010). Application of soft computing techniques in machining performance prediction and optimization? A literature review. *International Journal of Advanced Manufacturing Technology*, 46, 445–464.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 182–197.
- Ergur, H. S., & Oysal, Y. (2015). Estimation of cutting speed in abrasive water jet using an adaptive wavelet neural network. *Journal of Intelligent Manufacturing*, 26, 403–413.
- Falco, I. D., Scafuri, U., & Tarantino, E. (2016). Optimizing personalized touristic itineraries by a multiobjective evolutionary algorithm. *International Journal of Information Technology & Decision Making*, 15, 1269–1312.
- Huang, J., Gao, L., & Li, X. (2015). An effective teaching-learning-based cuckoo search algorithm for parameter optimization problems in structure designing and machining processes. *Applied Soft Computing*, 36, 349–356.
- Jagadish, B. S., & Ray, A. (2016). Prediction and optimization of process parameters of green composites in AWJM process using response surface methodology. *International Journal of Advanced Manufacturing Technology*, 87, 1359–1370.
- Jagadish, B. S., & Ray, A. (2015). Prediction of surface roughness quality of green abrasive water jet machining: A soft computing approach. *Journal of Intelligent Manufacturing*. <https://doi.org/10.1007/s10845-015-1169-7>.
- Jain, N. K., Jain, V. K., & Deb, K. (2007). Optimization of process parameters of mechanical type advanced machining processes using genetic algorithms. *International Journal of Machine Tools & Manufacture*, 47, 900–919.
- Jegaraj, J. J. R., & Babu, N. R. (2007). A soft computing approach for controlling the quality of cut with abrasive waterjet cutting system experiencing orifice and focusing tube wear. *Journal of Materials Processing Technology*, 185, 217–227.
- Kechagias, J., Petropoulos, G., & Vaxevanidis, N. (2012). Application of Taguchi design for quality characterization of abrasive water jet machining of TRIP sheet steels. *International Journal of Advanced Manufacturing Technology*, 62, 635–643.
- Kok, M., Kanca, E., & Eyercioglu, O. (2011). Prediction of surface roughness in abrasive waterjet machining of particle reinforced MMCs using genetic expression programming. *International Journal of Advanced Manufacturing Technology*, 55, 955–968.
- Liu, D., Huang, C., Wang, J., Zhu, H., Yao, P., & Liu, Z. W. (2014). Modeling and optimization of operating parameters for abrasive waterjet turning alumina ceramics using response surface methodology combined with Box-Behnken design. *Ceramics International*, 40, 7899–7908.
- Mellal, M. A., & Williams, E. J. (2016). Parameter optimization of advanced machining processes using cuckoo optimization algorithm and hoopoe heuristic. *Journal of Intelligent Manufacturing*, 27, 927–942.
- Mohamad, A., Zain, A. M., Bazin, N. E. N., & Udin, A. (2015). A process prediction model based on Cuckoo algorithm for abrasive waterjet machining. *Journal of Intelligent Manufacturing*, 26, 1247–1252.
- Parikh, P. J., & Lam, S. S. (2009). Parameter estimation for abrasive water jet machining process using neural networks. *International Journal of Advanced Manufacturing Technology*, 40, 497–502.
- Pawar, P. J., & Rao, R. V. (2013). Parameter optimization of machining processes using teaching-learning-based optimization algorithm. *International Journal of Advanced Manufacturing Technology*, 67, 995–1006.
- Rao, R. V. (2011). *Advanced modeling and optimization of manufacturing processes*. London: Springer.
- Rao, R. V. (2015). *Teaching learning based optimization algorithm and its engineering applications*. Cham: Springer.

- Rao, R. V. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, 7, 19–34.
- Rao, R. V., & Kalyankar, V. D. (2014). Optimization of modern machining processes using advanced optimization techniques: A review. *International Journal of Advanced Manufacturing Technology*, 73, 1159–1188.
- Rao, R. V., & Patel, B. K. (2010). Decision making in the manufacturing environment using an improved PROMETHEE method. *International Journal of Production Research*, 48, 4665–4682.
- Rao, R. V., & Waghmare, G. G. (2016). A new optimization algorithm for solving complex constrained design optimization problems. *Engineering Optimization*. <https://doi.org/10.1080/0305215X.2016.1164855>.
- Rao, R. V., Rai, D. P., & Balic, J. (2016). Multi-objective optimization of machining and micro-machining processes using non-dominated sorting teaching-learning-based optimization algorithm. *Journal of Intelligent Manufacturing*. <https://doi.org/10.1007/s10845-016-1210-5>.
- Rao, R. V., Rai, D. P., & Balic, J. (2017). A multi-objective algorithm for optimization of modern machining processes. *Engineering Applications of Artificial Intelligence*, 61, 103–125.
- Rostami, S., & Neri, F. (2017). A fast hypervolume driven selection mechanism for many-objective optimisation problems. *Swarm and Evolutionary Computation*, 34, 50–67.
- Santhanakumar, M., Adalarasan, R., & Rajmohan, M. (2015). Experimental modelling and analysis in abrasive waterjet cutting of ceramic tiles using grey-based response surface methodology. *Arabian Journal of Science and Engineering*, 40, 3299–3311.
- Santhanakumar, M., Adalarasan, R., & Rajmohan, M. (2016). Parameter design for cut surface characteristics in abrasive waterjet cutting of Al/SiC/Al₂O₃ composite using grey theory based RSM. *Journal of Mechanical Science and Technology*, 30, 371–379.
- Shanmugam, D. K., Wang, J., & Liu, H. (2008). Minimisation of kerf tapers in abrasive waterjet machining of alumina ceramics using a compensation technique. *International Journal of Machine Tools and Manufacture*, 48, 1527–1534.
- Shukla, R., & Singh, D. (2016a). Experimentation investigation of abrasive water jet machining parameters using Taguchi and Evolutionary optimization techniques. *Swarm and Evolutionary Computation*. <https://doi.org/10.1016/j.swevo.2016.07.002>.
- Shukla, R., & Singh, D. (2016b). Selection of parameters for advanced machining processes using firefly algorithm. *Engineering Science and Technology: An International Journal*. <https://doi.org/10.1016/j.jestch.2016.06.001>.
- Srinivasu, D. S., & Babu, N. R. (2008). An adaptive control strategy for the abrasive waterjet cutting process with the integration of vision-based monitoring and a neuro-genetic control strategy. *International Journal of Advanced Manufacturing Technology*, 38, 514–523.
- Vundavilli, P. R., Parappagoudar, M. B., Kodali, S. P., & Benguluri, S. (2012). Fuzzy logic-based expert system for prediction of depth of cut in abrasive water jet machining process. *Knowledge-Based Systems*, 27, 456–464.
- Yu, L., Yang, Z., & Tang, L. (2016). Prediction based multi-objective optimization for oil purchasing and distribution with the NSGA-II algorithm. *International Journal of Information Technology & Decision Making*, 15, 423–451.
- Yue, Z., Huang, C., Zhu, H., Wang, J., Yao, P., & Liu, Z. W. (2014). Optimization of machining parameters in the abrasive waterjet turning of alumina ceramic based on the response surface methodology. *International Journal of Advanced Manufacturing Technology*, 71, 2107–2114.
- Yusup, N., Sarkheyli, A., Zain, A. M., Hashim, S. Z. M., & Ithnin, N. (2014). Estimation of optimal machining control parameters using artificial bee colony. *Journal of Intelligent Manufacturing*, 25, 1463–1472.
- Yusup, N., Zain, A. M., & Hashim, S. Z. M. (2012). Evolutionary techniques in optimizing machining parameters: Review and recent applications. *Expert Systems with Applications*, 39, 9909–9927.
- Zain, A. M., Haron, H., & Sharif, S. (2011a). Estimation of the minimum machining performance in the abrasive waterjet machining using integrated ANN-SA. *Expert Systems with Applications*, 38, 8316–8326.
- Zain, A. M., Haron, H., & Sharif, S. (2011b). Optimization of process parameters in the abrasive waterjet machining using integrated SA-GA. *Applied Soft Computing*, 11, 5350–5359.
- Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., & Zhang, Q. (2011). Multi-objective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1, 32–49.
- Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3, 257–271.
- Zohoor, M., & Nourian, S. H. (2012). Development of an algorithm for optimum control process to compensate the nozzle wear effect in cutting the hard and tough material using abrasive water jet cutting process. *International Journal of Advanced Manufacturing Technology*, 61, 1019–1028.