J Intell Manuf (2018) 29:1715–1737
https://doi.org/10.1007/s10845-016-1210-5

Multi-objective optimization of machining and micro-machining processes using non-dominated sorting teaching–learning-based optimization algorithm

R. Venkata Rao[1] · **Dhiraj P. Rai**[1] · **J. Balic**[2]

Received: 25 August 2015 / Accepted: 16 March 2016 / Published online: 21 March 2016

Abstract Selection of optimum machining parameters is vital to the machining processes in order to ensure the quality of the product, reduce the machining cost, increasing the productivity and conserve resources for sustainability. Hence, in this work a posteriori multi-objective optimization algorithm named as Non-dominated Sorting Teaching–Learning-Based Optimization (NSTLBO) is applied to solve the multi-objective optimization problems of three machining processes namely, turning, wire-electric-discharge machining and laser cutting process and two micro-machining processes namely, focused ion beam micro-milling and micro wire-electric-discharge machining. The NSTLBO algorithm is incorporated with non-dominated sorting approach and crowding distance computation mechanism to maintain a diverse set of solutions in order to provide a Pareto-optimal set of solutions in a single simulation run. The results of the NSTLBO algorithm are compared with the results obtained using GA, NSGA-II, PSO, iterative search method and MOTLBO and are found to be competitive. The Pareto-optimal set of solutions for each optimization problem is obtained and reported. These Pareto-optimal set of solutions will help the decision maker in volatile scenarios and are useful for real production systems.

Keywords Sustainable machining processes · Micro-machining · Parameter optimization · Teaching–learning-based optimization algorithm · A posteriori approach

✉ 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

Introduction

Manufacturing is a process of converting raw material into finish goods for the satisfaction of human needs. Manufacturing sector is mostly the largest contributor in the gross domestic product of a nation that is an indicator of the performance of an economy and a quantum of a nation's prosperity. However, manufacturing of goods and products on a large scale has a negative impact on the ecology of our planet. Directly, in the form of disposal of solid, liquid or gaseous wastes generated as by-products of manufacturing processes. Indirectly, in the form of consumption of energy on a massive scale required to run these processes. However, in the recent times, stringent government norms and public awareness has made it obligatory for manufacturing industries to reduce their environmental footprint. This has steered the manufacturing industries towards implementing sustainable processes i.e. manufacturing products through economically sound processes while conserving energy and natural resources.

The fundamental challenge faced by the manufacturing industry today is to achieve economic goals by increasing the production rate, improving the quality of the product and lowering the production cost and simultaneously reducing its environmental impact through energy conservation, efficient material utilization and reduction in industrial waste. To achieve these goals it is important to utilize the machine tools to its fullest capabilities to achieve the best performance.

The performance of any machining process is significantly influenced by its process parameters. Thus, selecting optimum combination of these process parameters is essential for the success of the machining process. Determination of optimum combination process parameters of any machining process requires comprehensive knowledge of the process, empirical equations to develop realistic constraints, specification of machine tool capabilities, development of practical optimization criteria, and knowledge of mathematical and numerical optimization techniques. A human process planner selects proper machining process parameters using his own experience or machining tables. In most of the cases, the selected parameters are conservative and far from optimum. Selecting optimum combination of process parameters through experimentation is costly, time consuming and tedious. These factors have maneuvered the researchers towards applying numerical and heuristics based optimization techniques for process parameter optimization of machining processes.

In order to determine the optimum combination of process parameters, researchers have applied traditional optimization algorithms such as geometric programming, nonlinear programming, sequential programming, goal programming, dynamic programming, etc. (Mukherjee and Ray 2006). It is observed that some advanced optimization algorithms have also been applied by researchers for optimization of machining processes (Chandrasekaran et al. 2010; Yusup et al. 2012; Rao and Kalyankar 2014). Teimouri et al. (2014) applied imperialist competitive algorithm for multi-response optimization of ultrasonic machining process. Mellal and Williams (2014) applied cuckoo optimization algorithm and hoopoe heuristic for the optimization of modern machining process. Yusup et al. (2014) applied artificial bee colony (ABC) algorithm for optimization of optimal machining control parameters of abrasive water jet machining process. Zainal et al. (2014) applied glowworm swarm optimization for optimization of cutting parameters of end milling process. Mohanty et al. (2014) applied multi-objective particle swarm optimization (MOPSO) algorithm for multi-objective optimization of electric discharge machining process.

Depending on the nature of the phenomenon simulated by the algorithms, these population-based heuristic algorithms can be classified into two principal groups: Evolutionary Algorithms (EA) and swarm intelligence based algorithms. However, all evolutionary and swarm intelligence based optimization algorithms require common control parameters like population size, number of generations, elite size, etc. for their working. Besides the common control parameters, different algorithms require their own algorithm-specific parameters. For example, genetic algorithm (GA) uses mutation rate and crossover rate; particle swarm optimization (PSO) algorithm uses inertia weight, social cognitive parameters, maximum velocity; ABC algorithm uses number of bees (scout, onlooker and employed) and limit; However, the improper tuning of algorithm-specific parameters either increases the computational effort or yields to local optimal solution. In addition to the tuning of algorithm-specific parameters, common control parameters also need to be tuned which further enhances the effort.

Considering this fact, Rao et al. (2011) have introduced the teaching–learning-based optimization (TLBO) algorithm that does not require any algorithm-specific parameters. It requires only common control parameters like population size and number of generations for its working. The TLBO algorithm possesses excellent exploration and exploitation capabilities, it is less complex and has also proved its effectiveness in solving single-objective and multi-objective optimization problems. The TLBO algorithm has been widely applied by optimization researchers in various fields of engineering in order to solve continuous and discrete optimization problems in mechanical engineering, electrical engineering, civil engineering, computer science, etc. (Rao 2015, 2016a, b). Jaya algorithm is also a powerful algorithm-specific parameter-less algorithm but its multi-objective version is not yet developed (Rao 2016a, b).

Li et al. (2014) proposed a multi-objective TLBO algorithm for balancing of two sided assembly line with multiple constraints. Yu et al. (2014) proposed an improved TLBO algorithm with feedback phase, mutation crossover operation and chaotic perturbation mechanism to solve the numerical and engineering optimization problems. Kumar et al. (2015) applied TLBO algorithm for parametric appraisal and optimization of turning of CFRP composites using a single point cutting tool. Chen et al. (2015) applied TLBO algorithm with variable population scheme to artificial neural networks and global optimization. Zou et al. (2014) proposed a multi-objective version of the teaching–learning-based optimization (MOTLBO) algorithm. In order to handle multiple objectives simultaneously a sorting and ranking mechanism based on the non-dominance concept and crowding distance was adopted. A current archive and an external archive of solutions were maintained. The centroid of the non-dominated solutions in the current archive was selected as mean of the learners. Rao and Patel (2014) applied an improved version of TLBO algorithm for solving multi-objective optimization problem using a weighted sum approach. The concept of number of teachers, adaptive teaching factor and learning through tutorial were proposed to improve the performance of the TLBO algorithm. Similar work was reported by Patel and Savsani (2014) with the addition of Friedman's statistical test. Medina et al. (2014) proposed a multi-objective variant of TLBO algorithm using decomposition based approach. In the proposed algorithm the solution was updated in three phases. In addition to the teacher phase and learner phase a modified phase was proposed with crossover and polynomial mutation which requires tuning of crossover parameters and mutation distribution index for its working. Yu et al. (2015) proposed a self-adaptive multi-objective TLBO (SA-MTLBO) algorithm. The learners were allowed to select the mode of learning self adaptively based on their levels of knowledge in the classroom. Simulated binary cross-over and polynomial mutation strategies were used to improve the learners which require tuning of cross-over parameters and mutation probability. Sultana and Roy (2014) proposed a multi-objective version of TLBO algorithm by introducing the quasi oppositional based learning concept into the TLBO algorithm. However, this algorithm requires specifying the jump rate for opposite population jumping which is an algorithm-specific parameter that requires tuning by the user.

Researchers have already proposed various multi-objective versions of the TLBO algorithm. However, unlike previously proposed multi-objective versions of TLBO algorithm, "non-dominated sorting teaching learning based optimization" (NSTLBO) algorithm proposed in this work does not involve any algorithm specific parameters, requires only two phases, does not use solution archives, is simple to apply and requires less computational effort.

In the NSTLBO algorithm the non-dominance concept and mechanism of crowding distance computation (Deb et al. 2002) is adopted to handle multiple objectives simultaneously. However, unlike Zou et al. (2014), in order to maintain the simplicity of the algorithm neither the current archive nor external archive is used. The teacher is selected based on the rank and crowding distance from the current population. Further, unlike Zou et al. (2014), the mean of each decision variable in the current population is used to update the solutions in the teacher phase.

Most of the machining processes involve more than one machining process performance characteristic. This gives rise to the need to formulate and solve multi-objective optimization problems. There are basically two approaches to solve a multi-objective optimization problem and these are: a priori approach and a posteriori approach. In a priori approach, multi-objective optimization problem is transformed into a single objective optimization problem by assigning an appropriate weight to each objective. This ultimately leads to a unique optimum solution. However, the solution obtained by this process depends largely on the weights assigned to various objective functions. This approach does not provide a dense spread of the Pareto points. Furthermore, in order to assign weights to each objective the process planner is required to precisely know the order of importance of each objective in advance which may be difficult when the scenario is volatile. This drawback of a priori approach is eliminated in a posteriori approach, wherein it is not required to assign the weights to the objective functions prior to the simulation run. A posteriori approach does not lead to a unique optimum solution at the end of one simulation run but provides a dense spread of Pareto points (Pareto optimal solutions). The process planner can then select one solution from the set of Pareto optimal solutions based on the requirement or order of importance of objectives. The major advantage of a posteriori approach over a priori approach is that, a posteriori approach provides multiple tradeoff solutions for a multi-objective optimization problem in a single simulation run. On the other hand, as a priori approach provides only a single solution at the end of one simulation run, in order to achieve multiple tradeoff solutions using a priori approach the algorithm has to be run multiple times with different combination of weights. Thus, a posteriori approach is very suitable for solving multi-objective optimization problems in machining processes wherein taking into account volatility in the market and frequent change in customer desires is of paramount importance and determining the weights to be assigned to the objectives in advance is difficult.

The aim of this work is to further explore the potential of TLBO algorithm in solving multi-objective optimization problems as a posteriori approach. Therefore, in this work, multi-objective posteriori version of TLBO algorithm named as NSTLBO is applied to solve the optimization problems of selected machining and micro-machining processes.

In addition, a number of teachers concept is proposed in order utilize the expertise of multiple teachers simultaneously in training of learners in the teacher phase which was not discussed by the previous researchers. It is worthy to note that, in the NSTLBO algorithm, the number of teachers is not an algorithm-specific parameter as it is not required to be specified by the user, as in the case of Patel and Savsani (2014). The number of teachers depends upon the number of mutually conflicting objectives involved in the optimization problem. Therefore, unlike the previously proposed multi-objective versions of TLBO algorithm, the NSTLBO algorithm does not require tuning of any algorithm-specific parameters. Also, unlike Medina et al. (2014), in the NSTLBO algorithm the solution is updated in only two phases (i.e. teacher phase and learner phase).

In the NSTLBO algorithm, the teacher phase and learner phase maintain the vital balance between the exploration and exploitation capabilities and the teacher selection based on non-dominance rank of the solutions and crowding distance computation mechanism ensures the selection process towards better solutions with diversity among the solutions, in order to obtain a Pareto optimal set of solutions in a single simulation run. The TLBO and NSTLBO algorithms are described in detail in "Teaching learning based optimization algorithm" and "Non-dominated sorting teaching–learning-based optimization algorithm" sections, respectively.

In summary, there is a need to apply optimization algorithms that are free from algorithm-specific parameters for optimization of machining processes in order to reduce the burden of tuning the algorithm-specific parameters from the user. Further, there is also a need for optimization algorithms that can provide multiple trade-off solutions in a single simulation run for multi-objective optimization problems. Among several a posteriori optimization algorithms available in the literature the non-dominated sorting genetic algorithm (NSGA-II) is most widely used by the researchers for machining process optimization (Rao and Kalyankar 2014). However, NSGA-II also requires tuning of algorithm-specific parameters. Therefore, in this work the NSTLBO algorithm which is a multi-objective version of TLBO algorithm and which is an algorithm-specific parameter-less algorithm is applied to solve the multi-objective optimization problems of machining processes such as turning, wire-electric-discharge machining (WEDM) and micro-machining processes such as focused ion beam (FIB) micro-milling process, micro wire-electric-discharge machining process (micro-WEDM) and laser cutting. Pareto-fronts for each of the considered machining processes is obtained and the same are compared with the solutions obtained by other algorithms such as GA, NSGA-II, PSO and iterative search method and MOTLBO which were applied by previous researchers. Unlike previously proposed posteriori versions of TLBO algorithm, the NSTLBO algorithm requires only two phases for its working, does not involve any algorithm specific parameters and is simple to apply with less computational effort. The performance of the proposed NSTLBO is also compared with MOTLBO proposed by Zou et al. (2014). Just to identify the advantages of the proposed method.

A computer program for NSTLBO algorithm is developed in MATLAB r2009a. A computer system with a 2.93 GHz processor and 4 GB random access memory is used for execution of the program.

Teaching learning based optimization algorithm

The TLBO algorithm emulates the teaching learning process of a classroom. In each generation, the best solution is considered as the teacher, and other solutions are considered as learners. The learners mostly accept the instructions from the teacher, but also learn from each other. In the TLBO algorithm, an academic subject is analogous to an independent variable or candidate solution feature. The TLBO algorithm consists of two important phases i.e. the teacher phase and the learner phase. In the teacher phase, each independent variable s in each candidate solution x is modified according to Eqs. (1) and (2).

$$x_i'(s) \leftarrow x_i(s) + r(x_t(s) - T_f \bar{x}(s)) \quad (1)$$

$$\text{where } \bar{x}(s) = \frac{1}{N}\sum_{i=1}^{N} x_i(s) \quad (2)$$

for $i \in [1, N]$ and independent variable $s \in [1, n]$, where N is the population size, n is the total number of independent variables, x_t is the best individual in the population (i.e. the teacher), r is the random number taken from a uniform distribution on [0, 1], and T_f is the teaching factor and is randomly set equal to either 1 or 2 with equal probability. The new solution obtained after the teacher phase x_i' replaces the previous solution x_i if it is better than x_i.

As soon as the teacher phase ends the learner phase commences. The learner phase mimics the act of knowledge sharing among two randomly selected learners. The learner phase entails updating each learner based on another randomly selected learner as follows:

$$x_i''(s) \leftarrow \begin{cases} x_i'(s) + r(x_i'(s) - x_k'(s)) \text{ if } x_i' \text{ is better than } x_k' \\ x_i'(s) + r(x_k'(s) - x_i'(s)) \text{ otherwise} \end{cases} \quad (3)$$

For $i \in [1, N]$ and independent variable $s \in [1, n]$, where k is the random integer in $[1, N]$ such that $k \neq i$, and r is a random number taken from a uniform distribution on [0, 1]. Again, the new candidate solution obtained after the learner phase x_i'' replaces the previous solution x_i' if it is better than the previous solution x_i'. Figure 1 gives the flowchart for the TLBO algorithm. More details about the TLBO algorithm can be obtained from https://sites.google.com/site/tlborao/tlbo-code/.

Non-dominated sorting teaching–learning-based optimization algorithm

The NSTLBO algorithm is an extension of the TLBO algorithm. It is a posteriori approach for solving multi-objective optimization problems and maintains a diverse set of solutions. NSTLBO algorithm consists of teacher phase and learner phase similar to the TLBO algorithm. However, in order to handle multiple objectives, effectively and efficiently the NSTLBO algorithm is incorporated with non-dominated sorting approach and crowding distance computation mechanism proposed by Deb et al. (2002).

In the NSTLBO algorithm, the teacher phase and learner phase ensure good exploration and exploitation of the search space while non-dominated sorting approach makes certain that the selection process is always towards the good solutions and the population is pushed towards the Pareto-front in each generation. The crowding distance assignment mechanism ensures the selection of teacher from a sparse region of the search space with a view to avert premature convergence of the algorithm at local optima.

In the NSTLBO algorithm, the learners are updated according to the teacher phase and the learner phase of the TLBO algorithm. However, in case of single-objective optimization it is easy to decide which solution is better than the other based on the objective function value. But in the presence of multiple conflicting objectives determining the best solution from a set of solutions is difficult. In the NSTLBO

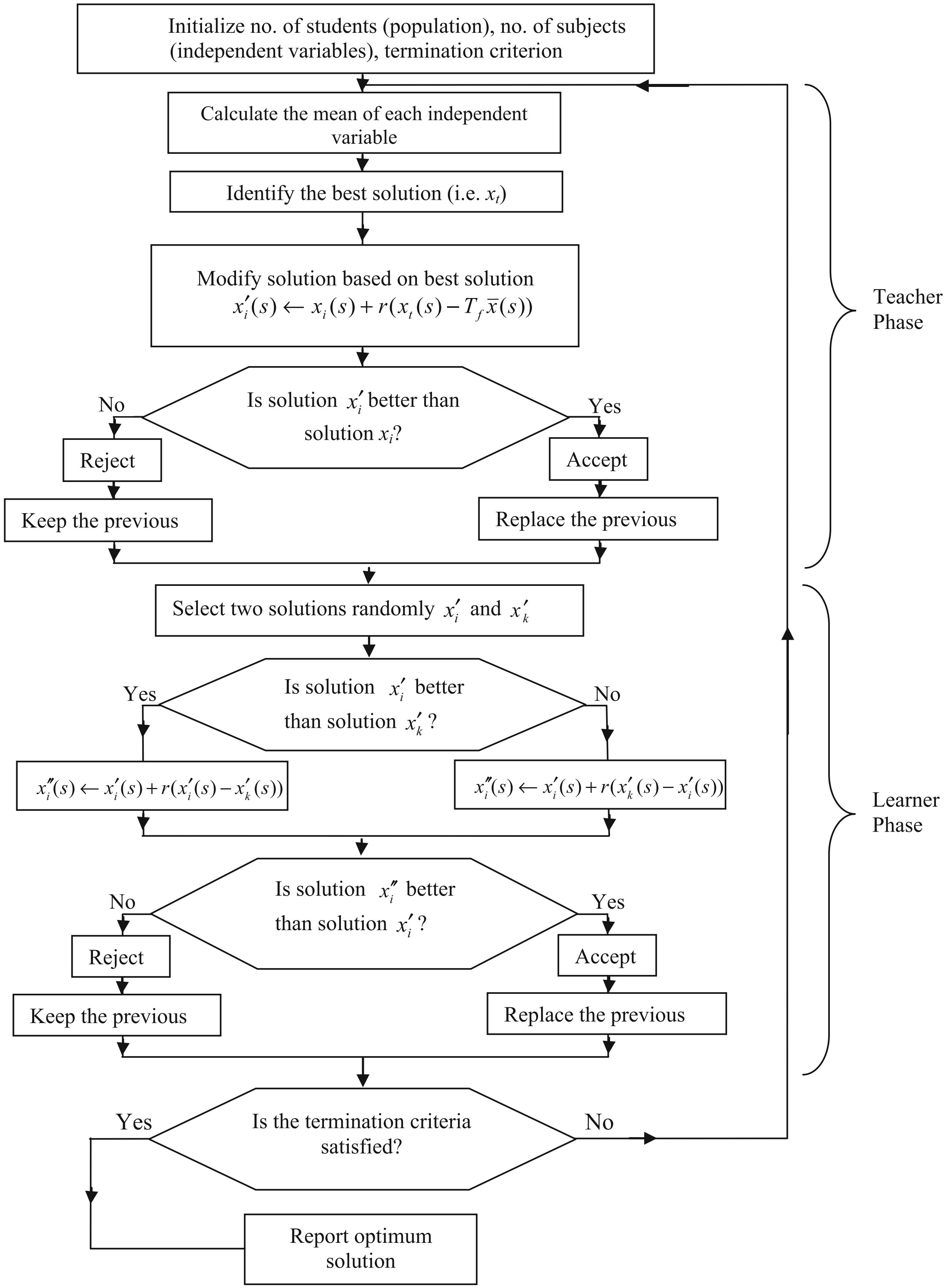

Fig. 1 Flow chart for the TLBO algorithm

algorithm, the task of finding the best solution is accomplished by comparing the rank assigned to the solutions based on the non-dominance concept and the crowding distance value.

In the beginning, an initial population is randomly generated with *NP* number of solutions (learners). This initial population is then sorted and ranked based on the non-dominance concept. The learner with the highest rank (rank = 1) is selected as the teacher of the class. In case, there exists more than one learner with the same rank then the learner with the highest value of crowding distance is selected as the teacher of the class. This ensures that the teacher is selected from the sparse region of the search space. Once the teacher is selected, learners are updated based on the teacher phase of the TLBO algorithm i.e. according to Eqs. (1) and (2).

After the teacher phase, the set of updated learners (new learners) is concatenated to the initial population to obtain a set of 2*NP* solutions (learners). These learners are again sorted and ranked based on the non-dominance concept and the crowding distance value for each learner is computed. Based on the new ranking and crowding distance value *NP* number of best learners are selected. These learners are further updated according to the learner phase of the TLBO algorithm i.e. according to Eq. (3).

The superiority among the learners is determined based on the non-dominance rank and the crowding distance value of the learners. A learner with a higher rank is regarded as superior to the other learner. If both the learners hold the same rank, then the learner with higher crowding distance value is seen as superior to the other.

After the end of the learner phase, the new learners are combined with the old learners and again sorted and ranked. Based on the new ranking and crowding distance value *NP* number of best learners are selected, and these learners are directly updated based on the teacher phase in the next iteration.

Non-dominated sorting of the population

In this approach the population is sorted into several ranks (fronts) based on the dominance concept as follows: a solution x_i is said to dominate other solution x_j if and only if solution x_i is no worse than solution x_j with respect to all the objectives and the solution x_i is strictly better than solution x_j in at least one objective. If any of the two conditions are violated, then solution x_i does not dominate solution x_j.

Among a set of solutions P, the non-dominated solutions are those that are not dominated by any solution in the set P. All such non-dominated solutions which are identified in the first sorting run are assigned rank one (first front) and are deleted from the set P. The remaining solutions in set P are again sorted and the procedure is repeated until all the solutions in the set P are sorted and ranked. In the case of constrained multi-objective optimization problems constrained-dominance concept (Deb et al. 2002) may be used in the proposed approach.

Crowding distance computation

The crowding distance is assigned to each solution in the population with an aim to estimate the density of solutions surrounding a particular solution i. Thus, average distance of two solutions on either side of solution i is measured along each of the M objectives. This quantity is called as the crowding distance (CD_i). The following steps may be followed to compute the CD_i for each solution i in the front F.

Step 1: Determine the number of solutions in front F as $l = |F|$. For each solution i in the set assign $CD_i = 0$.
Step 2: For each objective function $m = 1, 2, \ldots, M$, sort the set in the worst order of f_m.
Step 3: For $m = 1, 2, \ldots, M$, assign largest crowding distance to boundary solutions in the sorted list ($CD_1 = CD_l = \infty$), and for all the other solutions in the sorted list $j = 2$ to (l-1), assign crowding distance as follows:

$$CD_j = CD_j + \frac{\mathrm{f}_m^{j+1} - \mathrm{f}_m^{j-1}}{\mathrm{f}_m^{\max} - \mathrm{f}_m^{\min}} \tag{4}$$

Where, j is a solution in the sorted list, f_m is the objective function value of mth objective, $\mathrm{f}_m^{\max}$ and $\mathrm{f}_m^{\min}$ are the population-maximum and population-minimum values of the mth objective function.

Crowding-comparison operator

Crowding-comparison operator is used to identify the superior solution among two solutions under comparison, based on the two important attributes possessed by every individual i in the population i.e. non-domination rank ($Rank_i$) and crowding distance (CD_i). Thus, the crowded-comparison operator ($\prec_n$) is defined as follows:

$$i \prec_n j \text{ if } (Rank_i < Rank_j) \text{ or } ((Rank_i = Rank_j) \text{ and } (CD_i > CD_j))$$

That is, between two solutions (i and j) with differing non-domination ranks, the solution with lower or better rank is preferred. Otherwise, if both solutions belong to the same front ($Rank_i = Rank_j$), then the solution located in the lesser crowded region ($CD_i > CD_j$) is preferred.

Number of teachers concept

In the TLBO algorithm, the learner with best objective function value is selected as the teacher of the class. The onus of improving the mean result of the class is on the teacher. However, in the case of multi-objective optimization problems with mutually conflicting objectives, if a solution is good with respect to one objective, it may be equally bad with respect to the other objective and vice versa. Similarly, a solution which is best with respect to one objective, may be worst with respect to the other objective. Thus, in the case of multi-objective optimization problems with mutually conflicting objectives there may exists not a single but multiple learners which may be suitable to be selected as the teacher of the class, and number of such suitable learners will depend upon the number of objectives considered.

Thus, in this work, in order to take advantage of the expertise of multiple teachers simultaneously, a number of teachers concept is proposed. Instead of assigning a single teacher to the entire class, a teacher is assigned to each learner individually depending on the proximity of the learner to a particular teacher. This is achieved by calculating the normalized Euclidean distance between the learners and the teachers. Such an approach is adopted with a perspective of enhancing the exploitation capability of the algorithm (as a learner would be trained by the closest teacher) at the same time improve the diversity among the learners (as the class is been influenced by multiple teachers at the same time). Here, it is worthy to note that the *number of teachers* is not a parameter to the NSTLBO algorithm. In fact, the *number of teachers* in the NSTLBO is decided automatically based on the number of objectives considered in the optimization problem. The normalized Euclidean distance between a teacher and a learner is calculated according to Eq. (5).

$$E_{i,t} = \sqrt{\sum_{s=1}^{n}\left(\frac{x_t(s) - x_i(s)}{x_{\max}(s) - x_{\min}(s)}\right)^2} \tag{5}$$

Where n is the number of solution features or dimensions; N is the population size; $i \in [1 : N]$ N_t is the number of teachers; $t \in [1 : N_t]$; $E_{i,t}$ is the normalized Euclidean distance between a teacher (t) and a learner (i); $x_{\max}(s)$ and $x_{\min}(s)$ are the upper and lower bounds of solution feature (s).
Among all the teachers the teacher which is closest to a learner is assigned as the teacher to that learner, according to Eq. (6)

$$teacher_i = \min(E_{i,t}) \tag{6}$$

Figure 2 shows the flowchart for the NSTLBO algorithm.

Example 1 Optimization of turning process

The optimization problem formulated in this work is based on the empirical models developed by Palanikumar et al. (2009) for machining of glass fiber reinforced plastic using polycrystalline diamond cutting tool using an all geared lathe machine. The objective functions and process parameters considered in this work are same as those considered by Palanikumar et al. (2009). The objectives are: minimization of tool flank wear 'V_b' (mm), minimization of surface roughness 'R_a' (µm) and maximization of material removal rate '*MRR*' (mm^3/min). The process parameters considered in this work are: cutting speed 'v' (m/min), feed 'f' (mm/rev) and depth of cut 'd' (mm).

Objective functions

The objective functions are expressed by Eqs. (7), (8) and (9)

$$\begin{aligned}\text{minimize } V_b = {} & 0.09981 + 0.00069\, v + 1.41111\, f \\ & -0.17944\, d + 0.000001\, v^2 - 3.11111\, f^2 \\ & +0.00222\, d^2 - 0.00267\, v\, f \\ & +0.00007\, v\, d + 0.96667\, f d \end{aligned} \tag{7}$$

$$\begin{aligned}\text{minimize } R_a = {} & 1.9065 - 0.0103\, v + 11.1889\, f \\ & +0.3283\, d + 0.000001\, v^2 - 7.1111\, f^2 \\ & +0.0022\, d^2 + 0.0340\, v\, f \\ & -0.0015\, v\, d - 4.433\, f\, d \end{aligned} \tag{8}$$

$$\text{maximize} MRR = 1000\, v\, f\, d \tag{9}$$

Parameter bounds

The bounds on process parameters are expressed by Eqs. (10) to (12).

$$50 \le v \le 150 \tag{10}$$

$$0.1 \le f \le 0.20 \tag{11}$$

$$0.5 \le d \le 1.5 \tag{12}$$

The multi-objective optimization problem was solved by Palanikumar et al. (2009) using non-dominated sorting genetic algorithm-II (NSGA-II) considering a population size of 100 number of generations equal to 100 (i.e. number of function evaluations equal to 10,000). The algorithm-specific parameters used by NSGA-II are: crossover probability equal to 0.9, distribution index equal to 20 and mutation probability equal to 0.25 (Palanikumar et al. 2009).

Now, the same problem is solved using NSTLBO and MOTLBO (Zou et al. 2014) algorithms in order to see whether any improvement in the results can be achieved. For the purpose of fair comparison of results, the maximum number of function evaluations for NSTLBO and MOTLBO algorithms are maintained as 10,000. For this purpose a population size equal to 50 and maximum number of generations equal to 100 is considered by the NSTLBO and MOTLBO. The non-dominated set of solutions (Pareto-optimal solutions) obtained using NSTLBO algorithm is reported in Table 1. The non-dominated solutions obtained by NSGA-II (Palanikumar et al. 2009) are shown in Fig. 3 along with the non-dominated solutions obtained using NSTLBO and MOTLBO algorithms.

In order to compare the non-dominated set of solutions obtained using NSTLBO, MOTLBO and NSGA-II

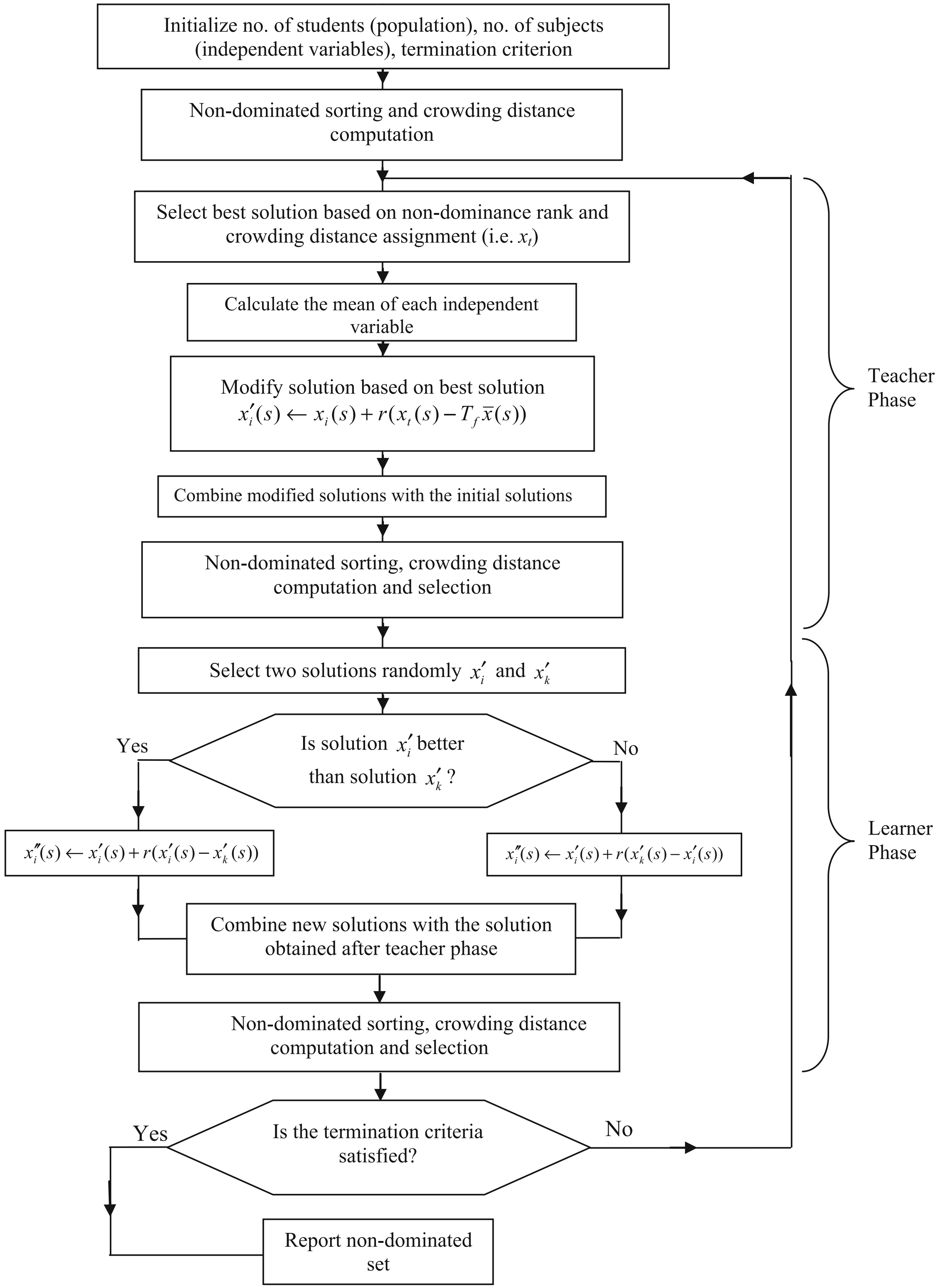

Fig. 2 Flowchart for the NSTLBO algorithm

Table 1 The non-dominated set of solutions for turning process obtained using NSTLBO

S. no.	v (m/min)	f (mm/rev)	d (mm)	V_b (mm)	R_a (µm)	MRR (mm^3/min)
1	54.2795	0.1	1.4994	0.1223	2.2931	8138.9121
2	58.3274	0.1	1.5	0.1248	2.2564	8749.1042
3	62.4693	0.1	1.5	0.1275	2.219	9370.3944
4	71.9101	0.1	1.4924	0.1343	2.1356	10,732.096
5	75.3031	0.1004	1.4986	0.1369	2.1057	11,325.274
6	79.2396	0.1004	1.5	0.1396	2.0703	11,933.641
7	83.2911	0.1005	1.4991	0.1426	2.0346	12,545.298
8	88.5514	0.1003	1.4933	0.1464	1.9879	13,265.659
9	92.4951	0.1003	1.4992	0.1487	1.951	13,906.875
10	102.2128	0.1	1.5	0.1551	1.8619	15,331.917
11	109.2442	0.1	1.4713	0.1622	1.8069	16,072.785
12	112.5882	0.1	1.5	0.1628	1.7692	16,888.234
13	117.1517	0.1	1.4928	0.1667	1.7306	17,488.753
14	121.8446	0.1	1.4921	0.1704	1.689	18,180.313
15	125.5378	0.1	1.5	0.1727	1.6538	18,830.675
16	130.8624	0.1	1.5	0.1769	1.6065	19,629.367
17	134.3337	0.1	1.4946	0.18	1.5773	20,077.852
18	136.3089	0.1	1.5	0.1812	1.5581	20,446.334
19	150	0.1	1.5	0.1924	1.4371	22,509.985
20	150	0.1029	1.5	0.1976	1.4601	23,141.492
21	150	0.1055	1.5	0.2024	1.482	23,745.633
22	150	0.108	1.4994	0.2069	1.5022	24,289.652
23	150	0.1098	1.4977	0.2102	1.5173	24,662.139
24	150	0.1136	1.5	0.2168	1.5472	25,559.907
25	148.4627	0.1176	1.5	0.2226	1.5921	26,196.464
26	150	0.1203	1.5	0.2284	1.6009	27,075.545
27	150	0.1241	1.5	0.2349	1.6308	27,926.109
28	150	0.1242	1.5	0.2349	1.6311	27,934.667
29	150	0.1296	1.489	0.2444	1.6787	28,943.341
30	150	0.1329	1.4992	0.2495	1.6998	29,888.757
31	150	0.1368	1.5	0.2558	1.7295	30,779.681
32	150	0.1395	1.5	0.2602	1.7506	31,397.552
33	150	0.1431	1.5	0.2658	1.7777	32,197.449
34	148.4572	0.1497	1.4984	0.2749	1.8392	33,292.613
35	150	0.1496	1.5	0.2758	1.8265	33,651.974
36	150	0.1545	1.4838	0.2835	1.873	34,392.181
37	149.5388	0.1604	1.4985	0.2917	1.9109	35,936.839
38	149.8541	0.1625	1.4976	0.2951	1.9252	36,474.722
39	148.1898	0.1666	1.5	0.2999	1.9647	37,042.167
40	149.8148	0.169	1.5	0.3042	1.9709	37,970.197
41	150	0.172	1.5	0.3086	1.9916	38,702.384
42	149.8909	0.175	1.494	0.3126	2.0174	39,179.928
43	149.9526	0.1797	1.4978	0.319	2.0483	40,360.413
44	150	0.1841	1.5	0.3249	2.0774	41,420.598
45	150	0.1858	1.5	0.3272	2.0895	41,806.908
46	149.1499	0.1888	1.4995	0.3305	2.1153	42,215.482
47	149.4195	0.1928	1.4998	0.3358	2.1412	43,195.272
48	150	0.1955	1.5	0.3396	2.1569	43,996.501

Table 1 continued

S. no.	v (m/min)	f (mm/rev)	d (mm)	V_b (mm)	R_a (µm)	MRR (mm^3/min)
49	150	0.197	1.5	0.3413	2.1665	44,313.77
50	150	0.2	1.4997	0.3451	2.1876	44,989.703

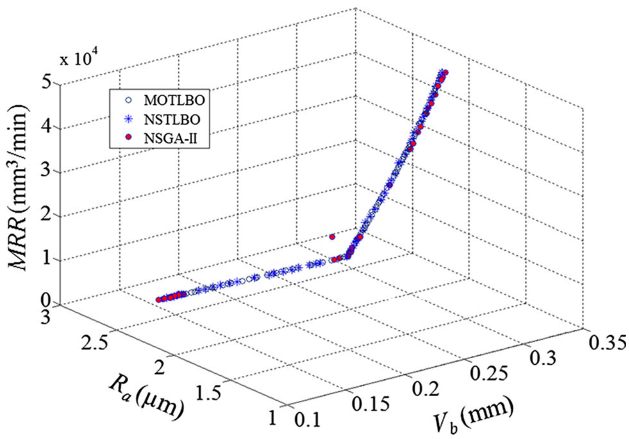

Fig. 3 Comparison of non-dominated solutions obtained using NSTLBO, MOTLBO and NSGA-II (Palanikumar et al. 2009) for turning process

Table 2 Best, mean and worst values of hypervolume obtained using NSTLBO and MOTLBO (example 1)

Algorithm	Best	Mean	Worst	SD
NSTLBO	9308.5	9258.39	9201.3	20.786
MOTLBO	9294.3	9273.03	9202.5	26.397

(Palanikumar et al. 2009), the hypervolume (HV) performance indicator proposed by Zitzler and Thiele (1999) is adopted in this work. Hypervolume is a metric that calculates the volume (or area in the case of bi-objective problems) of the objective space covered by the members of a Pareto-optimal set. Thus a higher value of hypervolume indicates better performance of a particular algorithm. In this work the method suggested in Deb (2001) is used to calculate the hypervolume. The NSTLBO and MOTLBO algorithms are run 30 times independently and the best, mean and worst values of hypervolume are reported in Table 2. It is observed that the hypervolume of the Pareto-front obtained using NSTLBO algorithm and MOTLBO algorithm (i.e. $HV_{NSTLBO} = 9308.5$ and $HV_{MOTLBO} = 9294.3$) is slightly higher than the hypervolume of the Pareto-front obtained using NSGA-II (i.e. $HV_{NSGA-II} = 9190.1$). This is mainly because, although the Pareto-fronts obtained using NSTLBO, MOTLBO and NSGA-II seem to overlap each other, the solutions obtained by NSTLBO and MOTLBO algorithms are uniformly distributed along the Pareto-front. On the other hand, clustering is observed in the Pareto-optimal solutions obtained using NSGA-II.

In order to justify the results provided by the NSTLBO algorithm for turning process three solutions from Table 1 are considered i.e. solution no. 1, which gives highest priority to minimization of flank wear 'V_b', solution no. 19, which gives highest priority to minimization of average surface roughness 'R_a' and solution no. 50 which gives highest priority to maximization of material removal rate '*MRR*'.

V_b increases as the cutting velocity and feed rate increase and reduces with the increase in depth of cut, therefore, a minimum value of cutting velocity and feed rate and a maximum value of depth of cut is desirable to minimize V_b. Accordingly, the NSTLBO algorithm has selected the values of cutting velocity and feed rate close to their respective lower bound, whereas, the value of depth of cut is selected to its upper bound (i.e. $v = 54.2795$ m/min, $f = 0.1$ mm/rev, $d = 1.4994$ mm) (refer solution 1 in Table 1). R_a reduces as the cutting velocity increases and increases with increase in feed rate. R_a decreases as the interaction between feed rate and depth of cut increases, therefore, a maximum value of cutting velocity, minimum value of feed rate and a maximum value of depth of cut are desirable to minimize R_a. Accordingly, the NSTLBO algorithm has selected the values of cutting velocity to its lower bound, whereas, the values of feed rate and depth of cut is selected to their respective upper bound (i.e. $v = 150$ m/min, $f = 0.1$ mm/rev, $d = 1.5$ mm) (refer solution 19 in Table 1).

Volume of material removed per unit time in turning process is the product of cutting speed, feed rate and depth of cut. Therefore, in order to achieve a maximum value of *MRR* the NSTLBO has selected the upper bound values of cutting speed, feed rate and depth of cut (i.e. $v = 150$ m/min, $f = 0.2$ mm/rev, $d = 1.4997$ mm) (refer solution 50 in Table 1). The intermediate solutions provided by the NSTLBO algorithm are also important and consistent with the experimental observations of Palanikumar et al. (2009). Depth of cut is beneficial for the three responses; namely tool flank wear, surface roughness and material removal rate therefore, it is maintained close to the upper bound for all the solutions. Cutting velocity and feed rate are non-beneficial for tool flank wear but beneficial for material removal rate, therefore, cutting velocity and feed rate increases gradually from lower bound value to upper bound value as tool flank wear and material removal rate increase from minimum to maximum value.

NSGA-II required 10,000 function evaluations to obtain the non-dominated set of solutions (Palanikumar et al. 2009). Only for the purpose of fair comparison of results the maximum number of function evaluations for NSTLBO and MOTLBO algorithms is maintained as 10,000. However, NSTLBO and MOTLBO algorithms could obtain the non-dominated set of solutions within in 1000 function evaluations. This is mainly because NSGA-II requires tuning of algorithm-specific parameters for its working. Improper tuning of these algorithm specific parameters may result in a low convergence rate or stagnation of the algorithm at the local optima. On the other hand, the NSTLBO and MOTLBO algorithms do not require tuning of any algorithm-specific parameters for its working. The computational time required by NSTLBO and MOTLBO algorithms for 10,000 function evaluations 6.07 and 24.931 s, respectively. However, the computational time required by NSGA-II for 10,000 function evaluations is not given in Palanikumar et al. (2009).

Example 2 Optimization of wire-electric-discharge machining process

The optimization problem formulated in this work is based on the mathematical models for cutting speed '*CS*' (mm/min) and surface roughness '*SR*' (µm) developed by Garg et al. (2012) using real experimental data. An Electronica 4 axis Sprintcut-734 CNC Wire Cut machine was used for experimentation using deionized water as dielectric fluid. Titanium alloy (Ti 6-2-4-2) was used as work material. The process parameters such as pulse on time 'T_{on}' (µs), pulse off time 'T_{off}' (µs), peak current '*IP*' (A), spark set voltage '*SV*' (V), wire feed '*WF*' (m/min) and wire tension '*WT*' (g) were considered. The objective functions and process parameters considered in this work are same as those considered by Garg et al. (2012).

Objective functions

The objective functions are expressed by Eqs. (13) and (14).

$$\begin{aligned} \text{maximize } CS = & -24.85563 + 0.29637 \times T_{on} + 0.12237 \\ & \times T_{off} + (6.53472E-4) \times IP \\ & +0.1454 \times SV + 0.060880 \times WT \\ & +(1.52323E-3) \times T_{off}^2 \\ & -(3.15625E-3) \times T_{on} \times T_{off} \\ & -(1.66667E-3) \times T_{on} \times SV \\ & +(7.84375E-4) \times T_{off} \times SV \\ & -(1.30312E-3) \times SV \times WT \end{aligned} \tag{13}$$

$$\begin{aligned} \text{minimize } SR = & \ 2.28046 + (0.014514 \times T_{on}) \\ & -0.01175 \times T_{off} - (7.54444E-3) \times IP \\ & -(4.466E-3) \times SV - 0.19140 \\ & \times WF - 0.8279 \times WT \\ & +(7.35417E-3) \times T_{on} \times WT \\ & +(1.08333E-3) \times IP \times WF \end{aligned} \tag{14}$$

Parameter bounds

The bounds on process parameters are expressed by Eqs. (15) to (20).

$$112 \le T_{on} \le 118 \tag{15}$$
$$48 \le T_{off} \le 56 \tag{16}$$
$$140 \le IP \le 200 \tag{17}$$
$$35 \le SV \le 55 \tag{18}$$
$$6 \le WF \le 10 \tag{19}$$
$$4 \le WT \le 8 \tag{20}$$

Garg et al. (2012) solved the multi-objective optimization problem of WEDM using NSGA-II considering a population size equal to 100 and maximum number of generations equal to 1000 (i.e. maximum number of function evaluations equal to 100,000), the crossover probability and mutation probability were selected as 0.9 and 0.116, respectively. Now the same problem is solved using the NSTLBO and MOTLBO (Zou et al. 2014) algorithm to see whether any improvement in the results can be achieved. Therefore, for the purpose of fair comparison of results the maximum number of function evaluations considered by NSTLBO and MOTLBO algorithms is maintained as 100,000. For this purpose, a population size of 50 and maximum number of generations equal to 1000 are chosen for the NSTLBO and MOTLBO algorithms. The non-dominated set of solutions obtained using NSTLBO algorithm is reported in Table 3. Figure 4 shows the non-dominated solutions obtained using NSGA-II (Garg et al. 2012), NSTLBO and MOTLBO algorithms. The Pareto-fronts obtained using NSTLBO algorithm, MOTLBO algorithm and NSGA-II are compared based on the hypervolume performance indicator. The best, mean and worst values of hypervolume of Pareto-fronts obtained using NSTLBO and MOTLBO algorithms over 30 independent runs are reported in Table 4. It is observed that, although the Pareto-fronts obtained using NSTLBO, MOTLBO and NSGA-II seem to overlap each other the best value of hypervolume obtained using NSTLBO and MOTLBO (i.e. HV_{NSTLBO} = 0.6599, HV_{MOTLBO} = 0.6599) are slightly higher than the hypervolume obtained using NSGA-II (i.e. $HV_{NSGA-II}$ = 0.6373). This is mainly because the Pareto-optimal solutions obtained using NSTLBO and MOTLBO algorithms are well distributed along the Pareto-front as compared to the Pareto-optimal solutions obtained using NSGA-II (Fig. 4).

Further, only for the purpose of fair comparison of results obtained using NSTLBO and MOTLBO algorithms with the results obtained by NSGA-II (Garg et al. 2012) the maximum number of function evaluations for NSTLBO are maintained are 100,000. However, the NSTLBO and MOTLBO could obtain the non-dominated set of solutions within 4000 function evaluations for WEDM process. The computational time required by NSTLBO and MOTLBO algorithms to per-

Table 3 The non-dominated set of solutions for WEDM obtained using NSTLBO algorithm

S. no.	T_{on} (μs)	T_{off} (μs)	*IP* (A)	*SV* (V)	*WF* (m/min)	*WT* (g)	*CS* (mm/min)	*SR* (μm)
1	112	56	140.0084	55	10	8	0.3227	1.515
2	112	55.7456	140	55	10	8	0.3272	1.518
3	112	54.1329	140.0662	55	10	8	0.3605	1.5371
4	112	53.1498	140	55	10	8	0.3847	1.5485
5	112	52.3656	140	55	10	8	0.406	1.5577
6	112	52.5899	140	53.8734	10	8	0.4115	1.5601
7	112	51.2675	140	55	10	8	0.4391	1.5706
8	112	54.1704	140	45.2386	10	8	0.4495	1.5801
9	112	50.3989	140	54.802	10	8	0.4703	1.5817
10	112	49.7906	140.0138	53.3606	10	8	0.5101	1.5953
11	112	48	140	55	10	8	0.5593	1.609
12	112	49.4588	140.0129	48.3331	10	8	0.5876	1.6217
13	112	48.6735	140.0537	49.6516	10	8	0.6042	1.6251
14	112	48	140	49.6813	10	8	0.634	1.6327
15	112	48	140	48.0341	10	8	0.6571	1.6401
16	112.0003	48	140	45.8241	10	8	0.6882	1.65
17	112.0051	48.1251	140.0646	41.8557	10	8	0.7378	1.6668
18	112	48	140.0031	39.9919	10	8	0.7701	1.676
19	112	48	140	38.3046	10	8	0.7937	1.6835
20	112.0425	48	140.0926	36.8229	10	8	0.8182	1.6936
21	112	48	140.0029	35	10	8	0.8401	1.6983
22	112.083	48	140.2117	35	10	8	0.8475	1.7051
23	112.6033	48	140	35	10	8	0.8924	1.7425
24	112.7933	48	140	35	10	8	0.9088	1.7565
25	113.1242	48	140	35	10	8	0.9374	1.7807
26	113.8887	48	140	35	10	4.5849	0.9514	1.8038
27	113.5755	48.0004	140	35.0075	10	7.4396	0.9678	1.8097
28	115.0893	48	140	35	10	4	1.0464	1.8509
29	115.2332	48	140	35	10	4	1.0589	1.8573
30	115.6821	48	140	35	10	4	1.0977	1.877
31	116.0263	48	140	35	10	4	1.1275	1.8921
32	116.3251	48	140	35	10	4	1.1533	1.9052
33	116.4661	48.0039	140.0295	35.0206	9.9857	4.2675	1.169	1.9196
34	116.8284	48	140	35	10	4	1.1969	1.9273
35	117.4169	48	140	35	10	4	1.2478	1.9532
36	117.5523	48	140.0048	35.0009	9.9969	4.0652	1.2605	1.9617
37	117.7382	48	140	35	10	4	1.2756	1.9673
38	118	48	140.0288	35	9.9924	4.309	1.303	1.9915
39	118	48	140.0343	35	9.9935	4.7765	1.3102	2.0102
40	118	48	142.806	35	9.9348	4.8035	1.3124	2.0225
41	118	48	140.0287	35.0079	9.9959	5.5565	1.3219	2.0411
42	118	48	147.1012	35	9.9345	5.4216	1.3246	2.061
43	118	48	200	35	6	4	1.3375	2.0751
44	118	48	140.0363	35.0059	9.9953	6.9113	1.3426	2.0952
45	118	48	200	35	6	5.0274	1.3532	2.1161
46	118	48	200	35	6	5.9286	1.3669	2.152
47	118	48	200	35	6	6.3818	1.3739	2.1701

Table 3 continued

S. no.	T_{on} (µs)	T_{off} (µs)	*IP* (A)	*SV* (V)	*WF* (m/min)	*WT* (g)	*CS* (mm/min)	*SR* (µm)
48	118	48	200	35	6	6.5716	1.3768	2.1777
49	118	48	200	35	6	7.5579	1.3918	2.217
50	118	48	200	35	6	7.9005	1.3971	2.2307

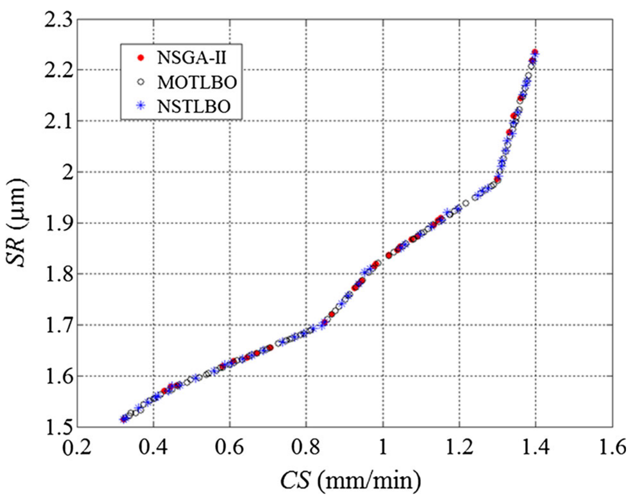

Fig. 4 The Pareto fronts obtained using NSTLBO, MOTLBO and NSGA-II (Garg et al. 2012) for WEDM

Table 4 Best, mean and worst values of hypervolume obtained using NSTLBO and MOTLBO (example 2)

Algorithm	Best	Mean	Worst	SD
NSTLBO	0.6599	0.6535	0.6478	0.0355
MOTLBO	0.6599	0.6309	0.6107	0.0128

form 100,000 function evaluations are 63.06 and 255.47 s, respectively. However, the computational time required by NSGA-II to perform 100,000 function evaluations is not given in Garg et al. (2012).

In order to justify the solutions provided by the NSTLBO algorithm two extreme solutions from Table 3 are taken into consideration i.e. solution no. 1, which gives a maximum priority to minimization of surface roughness '*SR*' and solution no. 50, which gives a maximum priority to maximization of cutting speed '*CS*'. From the main effect of process parameters of WEDM on *SR* (Garg et al. 2012) it is observed that *SR* increases with the increase in pulse on time and peak current but reduces with the increases in pulse off time, spark gap set voltage and wire feed. Therefore, to achieve a minimum value of *SR* the NSTLBO algorithm has selected lower bound values of pulse on time ($T_{\text{on}} = 112$ µs) and peak current (*IP* = 140 A) and upper bound values of pulse off time ($T_{\text{off}} = 56$ µs), spark gap set voltage (*SV* = 55 V) and wire feed (*WF* = 10 m/min) (refer solution no. 1 in Table 3).

From the main effect of process parameters of WEDM on *CS* (Garg et al. 2012) it is observed that *CS* increases with the increase in pulse on time and peak current but reduces with the increases in pulse off time and spark gap set voltage. The effect of wire tension on *CS* is insignificant. Therefore, to achieve a maximum value of *CS* the NSTLBO algorithm has selected upper bound values of pulse on time ($T_{\text{on}} = 118$ µs) and peak current (*IP* = 200 A) and a lower bound value of pulse off time ($T_{\text{off}} = 48$ µs) and spark gap set voltage (*SV* = 35 V) (refer solution no. 50 in Table 3). Similarly, the intermediate solutions provided by the NSTLBO algorithm (i.e. solution no. 2 to solution no. 49 in Table 3) are also important and are consistent with the experimental observations of Garg et al. (2012).

Example 3 Optimization of focused ion beam micro-milling process

The optimization problem formulated in this work is based on the empirical models for material removal rate '*MRR*' (µm^3/sec) and surface roughness 'R_a' (nm) in focused ion beam (FIB) micro-milling of cemented carbide developed by Bhavsar et al. (2015) based on real experimental data. A Quanta 3D FEG machine was used for experimentation. The process parameters such as extraction voltage 'x_1' (kV), angle of inclination, 'x_2' (degree), beam current 'x_3' (nA), dwell time 'x_4' (µs) overlap 'x_5' (%) were considered. The objective functions, process parameters and their bounds considered in this problem are same as those considered by Bhavsar et al. (2015).

Objective functions

The objective functions are expressed by Eqs. (21) and (22).

$$\begin{aligned}\text{maximize } MRR = {} & 0.0514 - 0.00506x_1 - 0.0269x_3 \\ & -0.000032x_2^2 - 0.00009x_5^2 \\ & -0.000103x_1x_2 \\ & +0.0036x_1x_3 + 0.000228x_1x_5 \\ & +0.000625x_2x_3 + 0.0001x_2x_5 \\ & +0.000514x_3x_5 \end{aligned} \quad (21)$$

$$\begin{aligned}\text{minimize } R_a = {} & 245 + 3.61x_2 - 5.38x_2 - 0.304x_1^2 \\ & +0.0428x_2^2 + 0.0735x_5^2 + 0.863x_1x_3 \\ & +0.144x_1x_5 - 0.17x_2x_3 - 0.139x_2x_5 \\ & +1.5x_3x_4 \end{aligned} \quad (22)$$

Parameter bounds

The bounds on the process parameters are expressed by Eqs. (23) to (27).

$$15 \le x_1 \le 30 \quad (23)$$

$$10 \le x_2 \le 70 \quad (24)$$

$$0.03 \le x_3 \le 3.5 \quad (25)$$

$$1 \le x_4 \le 10 \quad (26)$$

$$30 \le x_5 \le 75 \quad (27)$$

Bhavsar et al. (2015) solved the multi-objective optimization problem of FIB micro-milling NSGA-II considering a population size of 60 and a maximum number of generations equal to 1000 (i.e. maximum number of function evaluations equal to 60,000). The values algorithm-specific parameters required by NSGA-II are as follows: Pareto fraction equal to 0.7, cross-over probability equal to 0.9 (Bhavsar et al. 2015). The value of the mutation probability required by NSGA-II was not reported by Bhavsar et al. (2015). Now the same problem is solved using NSTLBO and MOTLBO (Zou et al. 2014) algorithms in order to see whether any improvement in the results can be achieved. For the purpose of fair comparison of results, the maximum number of function evaluations considered by NSTLBO is maintained as 60,000. For this purpose a population size of 50 and maximum number of generations equal to 600 is considered for the NSTLBO and MOTLBO algorithms.

The non-dominated set of solutions for FIB micro-milling process obtained by NSTLBO is reported in Table 5 and the Pareto-fronts obtained using NSTLBO, MOTLBO and NSGA-II are shown in Fig. 5. Table 6 gives the best, mean and worst values of hypervolume of the Pareto-

Table 5 The non-dominated set of solutions for FIB micro-milling process obtained using NSTLBO

S. no.	x_1 (kV)	x_2 (°)	x_3 (nA)	x_4 (µs)	x_5 (%)	*MRR* (µm^3/sec)	R_a (nm)
1	30	70	3.5	1	75	0.6302	92.2225
2	29.9996	70	3.4319	1	75	0.619	91.1702
3	30	70	3.3503	1	75	0.6057	89.9041
4	29.9833	70	3.3079	1	75	0.5985	89.3231
5	29.9908	70	3.1017	1	75	0.5649	86.0961
6	29.9947	70	3.0608	1	75	0.5583	85.4443
7	29.9986	70	3.0096	1	75	0.55	84.633
8	30	70	2.9277	1	75	0.5366	83.3571
9	29.9927	70	2.903	1	75	0.5325	83.0111
10	29.981	70	2.7613	1	75	0.5092	80.8762
11	29.9965	70	2.6704	1	75	0.4945	79.3892
12	29.9853	70	2.4893	1	75	0.4648	76.6445
13	29.974	70	2.3304	1	75	0.4387	74.2458
14	30	70	2.2896	1	75	0.4324	73.4736
15	29.9942	70	2.2138	1	75	0.4199	72.3304
16	29.9915	70	2.134	1	75	0.4068	71.1111
17	29.9944	70	2.0987	1	75	0.4011	70.548
18	29.9982	70	2.0105	1	75	0.3867	69.1607
19	29.9958	70	1.8975	1	75	0.3683	67.4248
20	29.9837	70	1.8087	1	75	0.3536	66.1203
21	29.9897	70	1.7772	1	75	0.3485	65.5972
22	29.9848	70	1.6681	1	75	0.3307	63.9378
23	30	70	1.5511	1	75	0.3117	62.0343
24	29.9955	70	1.4243	1	75	0.2909	60.0972
25	29.9972	70	1.3343	1	75	0.2762	58.693
26	29.9958	70	1.1532	1	75	0.2466	55.8972
27	29.9951	70	1.088	1	75	0.236	54.8923
28	29.9992	70	1.0091	1	75	0.2231	53.6433
29	29.9939	70	0.9527	1	75	0.2139	52.8057
30	30	10	1.7945	1.0783	30	0.2041	50.7416
31	30	10	1.6457	1.189	30	0.1888	47.1747

Table 5 continued

S. no.	x_1 (kV)	x_2 (°)	x_3 (nA)	x_4 (µs)	x_5 (%)	MRR (µm^3/sec)	R_a (nm)
32	30	10	1.6002	1.1248	30	0.1841	45.8376
33	30	10	1.5339	1.0705	30	0.1773	43.9982
34	30	10	1.47	1.1532	30	0.1642	40.646
35	30	10	1.3299	1.0384	30	0.1564	38.6717
36	30	10	1.2213	1.1013	30	0.1452	35.9904
37	30	10	1.1563	1.0046	30	0.1385	34.1442
38	30	10	1.0782	1	30	0.1305	32.13
39	30	10	1.0134	1.0599	30	0.1238	30.5557
40	30	10	0.919	1.0118	30	0.1141	28.0552
41	30	10	0.7803	1.009	30	0.0999	24.4857
42	30	10	0.7539	1.0126	30	0.0972	23.8108
43	30	10	0.6523	1.0393	30	0.0867	21.2268
44	30	10	0.571	1.1141	30	0.0784	19.1976
45	30	10	0.511	1.0697	30	0.0722	17.6112
46	30	10	0.4663	1.1138	30	0.0676	16.4895
47	30	10	0.3187	1	30	0.0524	12.6165
48	30	10	0.1957	1.1934	30	0.0398	9.5149
49	30	10	0.1179	1.008	30	0.0318	7.4608
50	30	10	0.03	1.0375	30	0.0228	5.2024

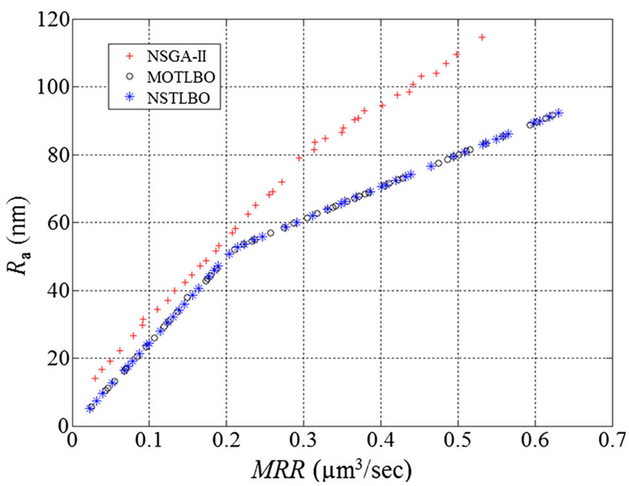

Fig. 5 Pareto fronts obtained using NSTLBO, MOTLBO and NSGA-II for FIB micro-milling process (Bhavsar et al. 2015)

Table 6 Best, mean and worst values of hypervolume obtained using NSTLBO and MOTLBO (example 3)

Algorithm	Best	Mean	Worst	SD
NSTLBO	55.633	55.263	54.9378	0.1749
MOTLBO	54.999	54.302	54.3795	1.133

front obtained using NSTLBO and MOTLBO algorithms. It is observed from Fig. 5 that the Pareto-front obtained using NSTLBO and MOTLBO algorithms are superior to the Pareto-front obtained using NSGA-II. This observation is also well supported by the values of hypervolume for NSTLBO, MOTLBO and NSGA-II algorithms (i.e. $HV_{NSTLBO} = 55.633$, $HV_{MOTLBO} = 54.999$ and $HV_{NSGA-II} = 28.6241$). The NSTLBO and MOTLBO algorithms achieved a higher value of hypervolume as compared to NSGA-II.

Further, even though the maximum number of function evaluations chosen for NSTLBO, MOTLBO and NSGA-II is 60,000, the NSGA-II required 7740 function evaluations to obtain the non-dominated set of solutions. On the other hand, NSTLBO and MOTLBO algorithms could obtain the non-dominated set of solutions within 2000 function evaluations for FIB micro-milling process. Thus, the NSTLBO and MOTLBO algorithms have shown a higher convergence rate as compared to NSGA-II. This is mainly because of the fact that NSGA-II requires tuning of algorithm-specific parameters and proper tuning of these algorithm-specific parameters is important to avert chances of entrapment in the local optima or a low convergence rate. On the other hand, NSTLBO and MOTLBO algorithms require only common control parameters and do not require any algorithm-specific parameters for its working. The NSTLBO and MOTLBO algorithms required 41.45 and 148.84 s, respectively, to perform 60,000 function evaluations. However, the computational time required by NSGA-II to perform 60,000 function evaluations was not given in Bhavsar et al. (2015).

From Table 5 it is observed that, the minimum value of R_a observed in the Pareto-front obtained by NSGA-II is 13.97

Table 7 Coded values of process parameters and their bounds (Kuriachen et al. 2015)

Factor	Parameter	Level 1 (−1)	Level 2 (0)	Level 3 (1)
A	Gap voltage (V)	100	125	150
B	Capacitance (μF)	0.01	0.1	0.4
C	Feed rate (μm/s)	3	6	9
D	Wire tension (gm)	(5 %) 4.125	(10 %) 8.25	(15 %) 12.375

nm and the maximum value of *MRR* is 0.5314 $\mu m^3/s$. The minimum value of R_a observed in the Pareto-front obtained by NSTLBO is 5.2024 nm which is 62.76 % better than the minimum value of R_a obtained using NSGA-II and the maximum value of *MRR* obtained by NSTLBO is 0.6302 $\mu m^3/s$ which is 15.67 % higher than the maximum value of *MRR* obtained by NSGA-II.

The results provided by the NSTLBO algorithm for FIB micro-milling process are well justified by experimental observations of Bhavsar et al. (2015). In Table 5, solution no. 1 corresponds to the maximum value of *MRR* (i.e. 0.6302 $\mu m^3/sec$). It is observed from the main effect plots of FIB process parameters (Bhavsar et al. 2015) that the values of extraction voltage, angle of incidence and beam current at their respective upper bounds and dwell at its lower bound are responsible for highest value of *MRR*. Accordingly, in order to achieve a maximum value of *MRR* the NSTLBO algorithm has selected the upper bound values of extraction voltage, angle of incidence, beam current and overlap and a lower bound value of dwell time (i.e. $x_1 = 30$ kV, $x_2 = 70°$, $x_3 = 3.5$ nA, $x_4 = 1$ μs and $x_5 = 75\,\%$).

Solution no. 50 in Table 5 corresponds to a minimum value of R_a provided by NSTLBO algorithm (i.e. $R_a = 5.2024$ nm). *MRR* and R_a are mutually conflicting responses as *MRR* increases R_a also increases. An increase in beam current increases the positive ions impinging the parent material thus increasing the *MRR*, a high value of angle of incidence also causes the *MRR* to increase which also increases R_a. Therefore, to achieve a lower value of R_a, a lower value of beam current and angle of incidence is desirable. Further, a lower value of dwell time results in less value of R_a, whereas, extraction voltage has no significant effect on R_a. Accordingly, NSTLBO algorithm has provided a minimum value of R_a by selecting the appropriate bound values of the process parameters of FIB micro-milling process (i.e. $x_1 = 30$ kV, $x_2 = 10°$, $x_3 = 0.03$ nA, $x_4 = 1.0375$ μs and $x_5 = 30\,\%$).

The non-dominated set of solutions provided by NSTLBO algorithm for the FIB micro-milling of cemented carbide gives flexibility to the process planner by allowing him to choose a solution from the non-dominated set reported in Table 5 which may satisfy his requirement of either high *MRR* or low R_a in order to comply with the customer specification.

Example 4 Optimization of micro wire-electric-discharge machining process

The optimization problem formulated in this work is based on the empirical models developed by Kuriachen et al. (2015) for material removal rate '*MRR*' (mm^3/min) and surface roughness '*SR*' (μm) for micro-WEDM of a titanium alloy (Ti–6Al–4V) using the real experimental data. The experiments were carried out in a micro-WEDM system using DT-110, multi-process micro-machining center. The process parameters such as gap voltage (V), capacitance (μF), feed rate (μm/s) and wire tension (gm) were considered. The objective functions, process parameters and the process parameter bounds considered in this problem are same as those considered by Kuriachen et al. (2015).

Objective functions

The objective functions in terms of coded values of process parameters are expressed by Eqs. (28) and (29). The bounds and coded values of process parameters as considered by Kuriachen et al. (2015) are reported in Table 7.

$$\begin{aligned} \text{maximize } \sqrt{\text{MRR}} = {} & 0.14 + 0.006812\,A + 0.024\,B \\ & +0.014\,C - 0.007979\,AB \\ & +0.00385\,BC - 0.039\,B^2 \end{aligned} \tag{28}$$

$$\begin{aligned} \text{minimize } SR = {} & 1.13 - 0.11\,A + 0.080\,B \\ & -0.17\,C - 0.16\,BC + 0.60\,A^2 \\ & +0.28\,C^2 \end{aligned} \tag{29}$$

Kuriachen et al. (2015) applied PSO algorithm to solve the multi-objective optimization problem by formulating a combined objective function using fuzzy logic. The unique solution obtained by PSO (Kuriachen et al. 2015) is reported in Table 8. Like all heuristic optimization algorithms, PSO also requires tuning of common control parameters. Besides common control parameters, PSO requires tuning of algorithm-specific parameters. However, the common control parameters and the algorithm-specific parameter used for PSO were not reported by Kuriachen et al. (2015). Now the NSTLBO and MOTLBO algorithms are applied to solve the multi-objective optimization problem of micro-WEDM in order to see whether any improvement in the results can be achieved and to obtain the Pareto-optimal set of solutions.

A population size of 50 and a maximum number of generations equal to 20 (i.e. maximum number of function

Table 8 Optimal solution obtained by PSO for micro-WEDM process (Kuriachen et al. 2015)

Gap voltage (V)	Capacitance (μF)	Feed rate (μm/s)	*MRR* (mm^3/min)	*SR* (μm)
113	0.26	9	0.02676 (0.0230*)	1.2796 (1.3386*)

*Corrected values

evaluations equal to 2000) are considered for the NSTLBO and MOTLBO algorithms. The non-dominated set of solutions obtained using NSTLBO algorithm are reported in Table 9. Figure 6 shows the Pareto-front for micro-WEDM obtained using NSTLBO and MOTLBO algorithms. It is observed that the unique solution obtained by PSO (Kuriachen et al. 2015) lies in the objective space which is dominated by the Pareto-front obtained by NSTLBO and MOTLBO algorithms. The NSTLBO and MOTLBO algorithms are run 30 times independently, and the best, mean and worst values of hypervolume thus obtained are reported in Table 10. The best value of hypervolume obtained by NSTLBO algorithm (i.e. $HV_{NSTLBO} = 0.1594$) is marginally higher than the best value obtained by MOTLBO algorithm (i.e. $HV_{MOTLBO} = 0.1594$). The NSTLBO and MOTLBO algorithms required 1.876 and 5.19 s, respectively, to perform

Table 9 The non-dominated set of solutions for micro-WEDM obtained using NSTLBO

S. no.	Gap voltage (V)	Capacitance (μF)	Feed rate (μm/s)	*MRR* (mm^3/min)	*SR* (μm)
1	127.5362	0.01	6.6043	0.0157	1.0799
2	127.5512	0.01	6.683	0.0158	1.0801
3	127.1506	0.0149	6.809	0.0163	1.082
4	126.3742	0.0275	6.3936	0.0166	1.0869
5	127.5622	0.0417	6.51	0.0177	1.0883
6	127.5709	0.0616	6.8391	0.0192	1.0904
7	126.7396	0.0651	6.9987	0.0195	1.0918
8	126.6504	0.0755	7.0133	0.0201	1.0933
9	127.4166	0.0873	7.0936	0.0208	1.0949
10	126.8775	0.1041	7.0221	0.0212	1.0967
11	127.2741	0.1133	7.1204	0.0217	1.0977
12	128.5621	0.1179	7.2287	0.0221	1.1001
13	127.7182	0.1491	7.3145	0.0227	1.101
14	127.5904	0.1558	7.4536	0.0229	1.102
15	127.7345	0.1661	7.6185	0.0232	1.1042
16	127.4851	0.1869	7.989	0.0237	1.1111
17	128.4227	0.1793	8.1458	0.024	1.1192
18	128.2071	0.1816	8.3251	0.0243	1.1267
19	128.7842	0.1806	8.4176	0.0245	1.1332
20	128.2862	0.1793	8.564	0.0247	1.1413
21	127.5773	0.1886	8.7611	0.0249	1.1513
22	128.578	0.1772	8.8675	0.0252	1.165
23	129.8908	0.1917	9	0.0254	1.1756
24	129.0233	0.1614	9	0.0255	1.1839
25	131.8662	0.1649	9	0.0256	1.1997
26	133.3393	0.1657	8.9985	0.0257	1.2143
27	133.9554	0.162	9	0.0257	1.2234
28	134.9952	0.1681	8.9986	0.0258	1.2352
29	136.2422	0.1684	9	0.0259	1.2552
30	137.6607	0.1633	8.9934	0.0259	1.2829
31	138.8097	0.1631	8.9836	0.026	1.3062
32	138.9056	0.1656	8.9857	0.026	1.3075
33	139.6854	0.1653	9	0.0261	1.3269
34	140.5523	0.1629	8.9907	0.0261	1.3484

Table 9 continued

S. no.	Gap voltage (V)	Capacitance (μF)	feed rate (μm/s)	*MRR* (mm^3/min)	*SR* (μm)
35	141.4885	0.1632	9	0.0262	1.3738
36	142.2547	0.1617	9	0.0262	1.3958
37	142.7115	0.1539	9	0.0262	1.4122
38	143.1763	0.1565	9	0.0263	1.4251
39	143.4845	0.1563	9	0.0263	1.4347
40	144.8025	0.1588	9	0.0264	1.4764
41	145.2812	0.1522	9	0.0264	1.4953
42	145.4714	0.1557	9	0.0264	1.5005
43	146.1907	0.1539	9	0.0264	1.5268
44	146.8804	0.1618	9	0.0265	1.5492
45	147.0261	0.156	9	0.0265	1.557
46	147.5998	0.1534	9	0.0265	1.58
47	148.4937	0.1529	9	0.0266	1.6159
48	148.9593	0.1513	9	0.0266	1.6357
49	149.4353	0.1545	9	0.0266	1.6544
50	150	0.1522	9	0.0267	1.6796

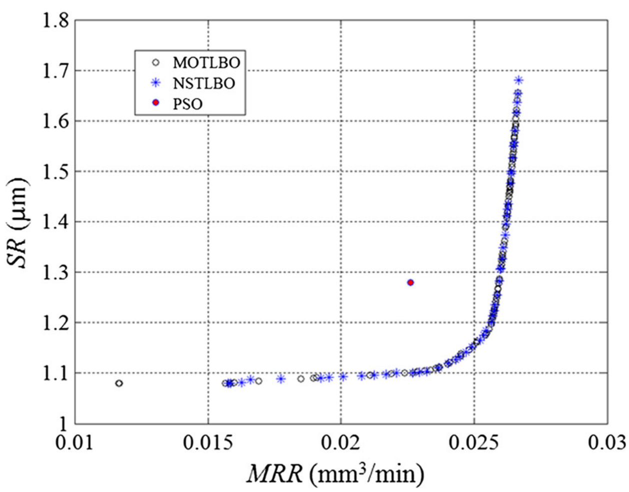

Fig. 6 Pareto-front for micro-WEDM process obtained using NSTLBO, MOTLBO and unique solution obtained using PSO (Kuriachen et al. 2015)

Table 10 Best, mean and worst values of hypervolume obtained using NSTLBO and MOTLBO (example 4)

Algorithm	Best	Mean	Worst	SD
NSTLBO	0.1594	0.1553	0.1530	0.00244
MOTLBO	0.1594	0.1555	0.1535	0.00255

2000 function evaluations. However, the computational time required by PSO was not given in Kuriachen et al. (2015).

The values of *MRR* and *SR* obtained by substituting the values of optimum process parameter combination obtained by PSO (Kuriachen et al. 2015) are 0.0230 mm^3/min and 1.3386 μm, respectively. Now, for a value of *MRR* higher than that provided by PSO, the corresponding value of *SR* provided by NSTLBO algorithm is 1.1042 μm (refer Table 9, solution no. 15), which is 17.51 % less than the value of *SR* provided by PSO. On the other hand, for a value of *SR* lower than that provided by PSO, the corresponding value of *MRR* provided by NSTLBO algorithm is 0.0261 mm^3/min (refer Table 9, solution no. 33), which is 13.91 % higher than the value of *MRR* obtained by PSO.

In order to justify the solutions provided by the NSTLBO algorithm two extreme solutions from Table 9 are taken into consideration i.e. solution no. 1, which gives highest priority to minimization of surface roughness '*SR*' and solution no. 50, which gives a maximum priority to maximization of material removal rate '*MRR*'. The solutions provided by the NSTLBO algorithm for micro-WEDM process are consistent with the experimental observations of the Somashekhar et al. (2010) and Kuriachen et al. (2015) which are as follows.

Increase in capacitance causes large energy dissipation forming large craters on the work-piece surface which increase the '*SR*'. Increase in feed rate causes *SR* to decrease until an intermediate is reached value beyond which it increases. *SR* reduces with increases in gap voltage until an intermediate is reached value beyond which it increases. Therefore, to achieve minimum *SR* the NSTLBO algorithm has selected a lowest value of capacitance (0.01 μF); intermediate value of feed rate (6.6043 μm/s); and intermediate value of gap voltage (127.5362 V) (refer solution 1 in Table 9).

As capacitance increases high energy is dissipated which increases the erosion of the work material increasing the *MRR*, however, further increase in capacitance causes the *MRR* to decrease due accumulation of debris causing unwanted sparking. Therefore, to achieve maximum *MRR* an intermediate value of capacitance is desirable. As feed rate increases the spark energy involved in material erosion is more which causes increase in *MRR*. At low gap voltage energy per spark is low, lowering the inter-electrode gap which reduces the *MRR* due to trapping of debris causing unwanted sparking. Accordingly, to achieve a maximum *MRR* the NSTLBO algorithm has selected the values of gap voltage (150 V) and feed rate (9 μm/s) at their respective upper bounds and an intermediate value of capacitance (0.1522 μF) (refer solution 50 in Table 9). Similarly, the intermediate solutions provided by the NSTLBO algorithm (i.e. solution no. 2 to solution no. 49 in Table 9) are also important and are consistent with the experimental observations of Somashekhar et al. (2010) and Kuriachen et al. (2015).

The NSTLBO and MOTLBO algorithms have provided multiple tradeoff solutions (Pareto-optimal solutions) for the multi-objective optimization problem in a single simulation run as compared to the single solution provided by PSO (Kuriachen et al. 2015). All the solutions in the Pareto-optimal set are equally good as each solution corresponds to a different order of importance of objectives. The Pareto-optimal set of solutions will serve as a ready reference to the process planner and allow him to choose a suitable solution based on his preferred order of importance of objectives.

Example 5 Optimization of laser cutting process

The optimization problem formulated in this work is based on the analysis given by Pandey and Dubey (2012). The process parameters considered in this work are same as those considered by Pandey and Dubey (2012) and they are as follows: gas pressure 'x_1' (kg/cm^2), pulse width 'x_2' (ms), pulse frequency 'x_3' (Hz) and cutting speed 'x_4' (mm/min) while minimization of surface roughness 'R_a' (μm) and kerf taper 'K_t' (degrees) are considered as objectives.

Objective functions

The objective functions are expressed by Eqs. (30) and (31)

$$\begin{aligned}\text{minimize } R_a = &-33.4550 + 7.2650x_1 + 12.1910x_2 \\ &+1.8114x_3 - 0.2813x_2^2 - 0.0371x_3^2 \\ &-0.7193x_1x_2 + 0.0108x_3x_4 + 0.0752x_1x_2\end{aligned} \tag{30}$$

$$\begin{aligned}\text{minimize } K_t = &-8.567 - 2.528x_1 + 0.2093x_1^2 \\ &+2.1318x_2^2 - 0.0371x_3^2 - 0.7193x_1x_2 \\ &+0.0108x_3x_4 + 0.0752x_1x_3\end{aligned} \tag{31}$$

Parameter bounds

The bounds on the process parameters are expressed by Eqs. (32)–(35).

$$5 \le x_1 \le 9 \tag{32}$$

$$1.4 \le x_2 \le 2.2 \tag{33}$$

$$6 \le x_3 \le 14 \tag{34}$$

$$15 \le x_4 \le 25 \tag{35}$$

Pandey and Dubey (2012) applied GA to solve the multi-objective optimization problem of laser cutting process, considering a population size of 200 and maximum number of generations equal to 800 (i.e. maximum number of function evaluations equal to 160,000). The crossover probability and mutation probability for GA were chosen as 0.8 and 0.07, respectively. Now the same problem is solved using NSTLBO and MOTLBO algorithms (Zou et al. 2014) in order to see whether any improvement in results can be achieved. For fair comparison of results, the maximum number of function evaluations for NSTLBO and MOTLBO algorithms are maintained as 160,000. For this purpose, a population size of 50 and maximum number of generations equal to 3200 is used by the NSTLBO and MOTLBO algorithms. The non-dominated set of solutions obtained using NSTLBO algorithm is reported in Table 11. The Pareto-front for laser cutting process obtained using NSTLBO and MOTLBO algorithms are shown in Fig. 7. It is observed from Fig. 7 that the Pareto-front obtained by GA lies in the objective space which is dominated by the Pareto-front obtained using NSTLBO and MOTLBO algorithms. The best, mean and worst values of the hypervolume of the Pareto-fronts obtained by NSTLBO and MOTLBO algorithms over 30 independent runs are reported in Table 12.

Kovacevic et al. (2014) solved the same problem using an iterative search method. However, the maximum number of function evaluations required by the iterative search method and the time taken by the iterative search method to obtain the Pareto-optimal set of solutions is not known. Nevertheless, for the purpose of fair comparison, the performances of NSTLBO, MOTLBO, GA and iterative search method on laser cutting optimization problem are compared based on the values of hypervolume achieved by the respective algorithms (i.e. HV_{NSTLBO} =15.370; HV_{MOTLBO} = 14.970; HV_{GA} = 13.5525; $HV_{iterativesearch}$ = 14.4603, respectively).

It is observed that the hypervolumes achieved by NSTLBO and MOTLBO algorithms are higher than the hypervolume achieved by GA. This indicates that the Pareto-front obtained using NSTLBO and MOTLBO algorithms are superior to the Pareto-front obtained using GA. The Pareto-fronts obtained using NSTLBO, MOTLBO and iterative search method seem to overlap each other. However, it is observed that hypervolume achieved by NSTLBO and MOTLBO algorithms are slightly higher than the hypervolume achieved by iterative search method.

The non-dominated solutions provided by NSTLBO algorithm are consistent with the experimental observations of Pandey and Dubey (2012) which are as follows. K_t decreases

Table 11 The non-dominated set of solutions for laser cutting obtained using NSTLBO

S. no.	x_1 (kg/cm^2)	x_2 (ms)	x_3 (Hz)	x_4 (mm/min)	K_t (degrees)	R_a (μm)
1	5.9314	1.4	14	15	0.3822	11.9862
2	5.7594	1.4	14	15	0.3883	11.7289
3	5.66	1.4	14	15	0.3974	11.5726
4	5.4986	1.4	14	15	0.4211	11.3071
5	5.4202	1.4	14	15	0.4365	11.1728
6	5.3159	1.4	14	15	0.4611	10.9888
7	5.238	1.4	14	15	0.4824	10.8473
8	5.2174	1.4	14	15	0.4884	10.8093
9	5.1373	1.4	14	15	0.5136	10.6594
10	5.0609	1.4	14	15	0.5402	10.5131
11	5.0021	1.4	14	15	0.5623	10.3982
12	5	1.4	14	15.2536	0.6015	10.3519
13	5	1.4	14	15.7384	0.6748	10.2693
14	5	1.4	14	15.9463	0.7062	10.233
15	5	1.4	14	16.8022	0.8356	10.0789
16	5	1.4	13.9719	17.4378	0.945	9.9653
17	5	1.4	14	18.0268	1.0208	9.8443
18	5	1.4	14	18.4189	1.0801	9.7657
19	5	1.4	14	19.3091	1.2147	9.581
20	5.007	1.4	14	19.9533	1.3094	9.4557
21	6.2167	1.4	6	15.8956	1.3575	9.3488
22	6.6796	1.4	6	19.3757	1.4049	9.2091
23	6.7303	1.4	6	20.5891	1.4695	8.9938
24	6.4991	1.4	6	19.9038	1.4979	8.9006
25	6.8437	1.4	6	22.2959	1.5526	8.7007
26	6.0754	1.4	6	18.4919	1.5979	8.6704
27	6.8644	1.4	6	23.2366	1.6091	8.4843
28	6.7509	1.4	6	23.2582	1.6371	8.3704
29	7.1756	1.4	6	25	1.6782	8.2763
30	6.6517	1.4	6	24.1883	1.725	8.0272
31	6.7185	1.4	6	25	1.7585	7.8759
32	6.5882	1.4	6	25	1.7974	7.7403
33	6.5087	1.4	6	25	1.8247	7.6528
34	6.36	1.4	6	25	1.8827	7.4797
35	6.3273	1.4	6	25	1.8967	7.4399
36	6.1826	1.4	6	25	1.9641	7.2568
37	6.0535	1.4	6	25	2.0316	7.0834
38	5.9673	1.4	6	25	2.0806	6.9626
39	5.8994	1.4	6	25	2.1213	6.8644
40	5.8653	1.4	6	25	2.1425	6.814
41	5.7703	1.4	6	25	2.2041	6.6705
42	5.7402	1.4	6	25	2.2244	6.6239
43	5.6759	1.4	6	25	2.269	6.5229
44	5.5282	1.4	6	25	2.3782	6.2817
45	5.4728	1.4	6	25	2.4214	6.1881
46	5.2683	1.4	6	25	2.5923	5.8276
47	5.1706	1.4	6	25	2.6801	5.6471

Table 11 continued

S. no.	x_1 (kg/cm^2)	x_2 (ms)	x_3 (Hz)	x_4 (mm/min)	K_t (degrees)	R_a (µm)
48	5.0765	1.4	6	25	2.7686	5.4681
49	5	1.4	6	25	2.8431	5.3189
50	5	1.4	6	25	2.8431	5.3189

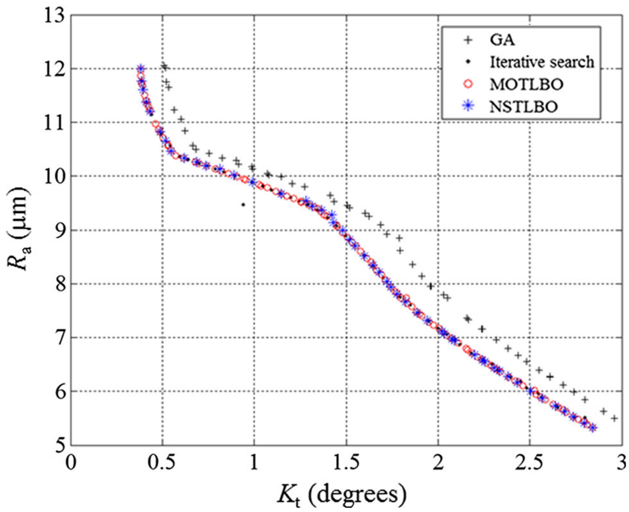

Fig. 7 Pareto-fronts for laser cutting process obtained using NSTLBO, MOTLBO, GA (Pandey and Dubey 2012) and iterative search method (Kovacevic et al. 2014)

Table 12 Best, mean and worst values of hypervolume obtained using NSTLBO and MOTLBO (example 5)

Algorithm	Best	Mean	Worst	SD
NSTLBO	15.370	15.230	14.872	0.182
MOTLBO	14.9706	14.950	14.872	0.0366

with increase in assist gas pressure at lower values of pulse width. However, at higher values of pulse width, K_t first decreases and then increases by increasing assist gas pressure. Accordingly, in Table 11, K_t increases with the increase in assist gas pressure while pulse width is maintained at its lower bound. K_t values increases at higher values of pulse width by increasing gas pressure. Therefore, as pulse width is non-beneficial for K_t it is maintained at is lower bound for all the solutions. Surface roughness 'R_a' increases with increase in cutting speed at upper as well as lower values of pulse frequency. Accordingly, in Table 11, surface roughness increases a cutting speed increases from lower bound value (i.e. $x_4 = 15$ mm/min) to upper bound value (i.e. $x_4 = 25$ mm/min), while pulse frequency is maintained at either upper bound (i.e. $x_3 = 14$ Hz) value or lower bound value (i.e. $x_3 = 6$ Hz). This is mainly because, at higher cutting speed, the heat available for the cutting is reduced due to which the melted material gets removed completely from the cutting front which gives smooth cut surface. At lower values of cutting speed, as pulse frequencies increases, the pulse energy also increases which results more melting of material. The melted material may not be removed completely and the remaining melted material may be re-solidified at the cutting edges, which increases R_a of the cutting edges. However, at higher cutting speed, the heat available for the cutting is reduced due to which the melted material is removed completely from the cutting front which gives smooth cut surface.

For the purpose of fair comparison of results, the maximum number of function evaluations used by NSTLBO and MOTLBO algorithms is maintained as 160,000 but the NSTLBO and MOTLBO algorithms could obtain the non-dominated set of solutions within 7000 function evaluations. The computational time required by NSTLBO and MOTLBO algorithms to perform 160,000 function evaluations are 102.94 and 406.292 s, respectively. However, the computational time required by GA (Pandey and Dubey 2012) and iterative search based algorithm (Kovacevic et al. 2014) are not available for comparison.

All the optimization problems formulated in this work are based on the mathematical models developed by previous researchers based on experimentation. The confirmation experiments for the developed mathematical models were also conducted by the previous researchers for machining processes such as turning (Palanikumar et al. 2009), WEDM (Garg et al. 2012), FIB micro-milling (Bhavsar et al. 2015), micro-WEDM (Kuriachen et al. 2015) and laser cutting (Pandey and Dubey 2012). In addition, the previous researchers had solved the optimization problems using advanced optimization algorithms such as GA, NSGA-II, PSO and iterative search method. Now the same mathematical models have been solved using NSTLBO and MOTLBO algorithms and the results obtained using NSTLBO and MOTLBO algorithms are compared with the results obtained by the previous researchers. The previous researchers had considered the process parameters in their continuous form. Therefore, all the process parameters considered in this work are in their continuous form only. [However, in actual practice, the values allowed by the machining process which are closer to the suggested optimum values may be considered.]

The Pareto-optimal set of solutions provided by NSTLBO algorithm contains a wide range of optimal values which will enable the process planner to choose a particular solution from the Pareto-set depending on his preference and imperativeness of the objectives. Therefore, the results reported in the present work are useful for real production systems.

Conclusions

Multi-objective optimization aspects of three machining processes namely turning, wire-electric-discharge machining and laser cutting and two micro-machining processes namely focused ion beam micro-milling and micro wire-electric-discharge machining are considered in this work. A posteriori multi-objective optimization algorithm based on teaching-learning-based optimization named "Non-dominated Sorting Teaching–Learning-Based Optimization" (NSTLBO) algorithm and "Multi-objective Teaching–Learning-Based Optimization" (MOTLBO) algorithm are applied to solve the respective machining process optimization problems. The same models were previously attempted by other researchers using GA, NSGA-II, PSO and iterative search method.

In the case of turning and wire-electric-discharge machining processes the results obtained using NSTLBO and MOTLBO are competitive. The NSTLBO and MOTLBO algorithms achieved the Pareto-optimal set of solutions in a very less number of function evaluations as compared to NSGA-II showing a higher convergence speed.

In the case of focused ion beam micro-milling the Pareto-front obtained using NSTLBO and MOTLBO algorithms are superior to the Pareto-front obtained using NSGA-II. In the case of micro-wire electric discharge machining process, the solutions in the Pareto-set obtained using NSTLBO and MOTLBO algorithms are better than the unique solution provided by PSO.

In the case of laser cutting process, the NSTLBO and MOTLBO algorithms obtained a superior Pareto-front as compared GA within comparatively less number of function evaluations. The Pareto-front obtained using NSTLBO and MOTLBO algorithms are not inferior to the Pareto-front obtained using the recently proposed iterative search method for laser cutting process.

The performance of NSTLBO algorithm is compared with MOTLBO algorithm for all the case studies considered in this work, based on hypervolume indicator and computational time. It is observed that the hypervolume achieved by NSTLBO algorithm is higher than the hypervolume achieved by MOTLBO algorithm for turning process, laser cutting and FIB micro-milling process. The values of hypervolume achieved by NSTLBO and MOTLBO algorithms are equal in the case of WEDM and micro-WEDM processes. However, the computational time required by MOTLBO algorithm is comparatively higher than the computational time required by NSTLBO algorithm for all the optimization case studies considered in this work. This may be because of additional computational effort required in maintaining the external archive and the need to allocate memory locations dynamically to store the best solution obtained after every generation in MOTLBO.

In the present work, NSTLBO and MOTLBO algorithms are applied to solve optimization problems of only selected machining processes. However, NSTLBO and MOTLBO algorithms may also be applied to solve optimization problems other traditional and modern machining processes. The NSTLBO and MOTLBO algorithms may also be applied to optimization problems of other manufacturing processes such as casting, welding, forming, etc.

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

References

Abhishek, K., Kumar, R. V., Datta, S., & Mahapatra, S. S. (2015). Parametric appraisal and optimization in machining of CFRP composites by using TLBO (teaching-learning based optimization algorithm). *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-015-1050-8.

Bhavsar, S. N., Aravindan, S., & Rao, P. V. (2015). Investigating material removal rate and surface roughness using multi-objective optimization for focused ion beam (FIB) micro-milling of cemented carbide. *Precision Engineering*, *40*, 131–138.

Chandrasekaran, M., Muralidhar, M., Krishna, M. C., & 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.

Chen, D., Lu, R., Zou, F., & Li, S. (2015). Teaching–learning-based optimization with variable-population scheme and its application for ANN and global optimization. *Neurocomputing*. doi:10.1016/j.neucom.2015.08.068.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. London: Wiley.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*, 182–197.

Garg, M. P., Jain, A., & Bhushan, G. (2012). Modelling and multi-objective optimization of process parameters of wire electrical-discharge machining using non-dominated sorting genetic algorithm-II. *Proceedings of Institution of Mechanical Engineers: Part B-Journal of Engineering Manufacture*, *226*(12), 1986–2001.

Kovacevic, M., Madic, M., Radovanovic, M., & Rancic, D. (2014). Software prototype for solving multi-objective machining optimization problems: Application in non-conventional machining processes. *Expert Systems with Applications*, *41*, 5657–5668.

Kuriachen, B., Somashekhar, K. P., & Mathew, J. (2015). Multiresponse optimization of micro-wire electrical discharge machining process. *International Journal of Advanced Manufacturing Technology*, *76*, 91–104.

Li, D., Zhang, C., Shao, X., & Lin, W. (2014). A multi-objective TLBO algorithm for balancing two-sided assembly line with multiple constraints. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0919-2.

Medina, M. A., Das, S., Coello, C. A. C., & Ramírez, J. M. (2014). Decomposition-based modern metaheuristic algorithms for multi-objective optimal power flow—A comparative study. *Engineering Applications of Artificial Intelligence*, *32*, 10–20.

Mellal, M. A., & Williams, E. J. (2014). Parameter optimization of advanced machining processes using cuckoo optimization algorithm and hoopoe heuristic. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0925-4.

Mohanty, C. P., Mahapatra, S. S., & Singh, M. R. (2014). A particle swarm approach for multi-objective optimization of electrical discharge machining process. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0942-3.

Mukherjee, I., & Ray, P. K. (2006). A review of optimization techniques in metal cutting processes. *Computers & Industrial Engineering*, *50*, 15–34.

Palanikumar, K., Latha, B., Senthilkumar, V. S., & Karthikeyan, R. (2009). Multiple performance optimization in machining of GFRP composites by a PCD tool using non-dominated sorting genetic algorithm (NSGA-II). *Metals and Materials International*, *15*(2), 249–258.

Pandey, A. K., & Dubey, A. K. (2012). Simultaneous optimization of multiple quality characteristics in laser cutting of titanium alloy sheet. *Optics and Laser Technology*, *44*, 1858–1865.

Patel, V. K., & Savsani, V. J. (2014). A multi-objective improved teaching-learning based. optimization algorithm (MO-ITLBO). *Information*. doi:10.1016/j.ins.2014.05.049.

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, *43*, 303–315.

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, V. (2014). A multi-objective improved teaching-learning based optimization algorithm for unconstrained and constrained optimization problems. *International Journal of Industrial Engineering Computations*, *5*, 1–22.

Rao, R. V. (2015). *Teaching–learning-based optimization (TLBO) algorithm and its engineering applications*. London: 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*(1), 19–34.

Rao, R. V. (2016). Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. *Decision Science Letters*, *5*, 1–30.

Somashekhar, K. P., Ramachandran, N., & Mathew, J. (2010). Material removal characteristics of microslot (kerf) geometry in μ-WEDM on aluminium. *International Journal of Advanced Manufacturing Technology*, *51*, 611–626.

Sultana, S., & Roy, P. K. (2014). Multi-objective quasi-oppositional teaching learning based optimization for optimal location of distributed generator in radial distribution systems. *Electrical Power and Energy Systems*, *63*, 534–535.

Teimouri, R., Baseri, H., & Moharami, R. (2014). Multi-responses optimization of ultrasonic machining process. *Journal of Intelligent Manufacturing*, *26*, 745–753.

Yu, K., Wang, X., & Wang, Z. (2014). An improved teaching–learning-based optimization algorithm for numerical and engineering optimization problems. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0918-3.

Yu, K., Wang, X., & Wang, Z. (2015). Self-adaptive multi-objective teaching–learning-based optimization and its application in ethylene cracking furnace operation optimization. *Chemometrics and Intelligent Laboratory Systems*, *146*, 198–210.

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.

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.

Zainal, N., Zain, A. M., Radzi, N. H. M., & Othman, M. R. (2014). Glowworm swarm optimization (GSO) for optimization of machining parameters. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0914-7.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, *3*(4), 257–271.

Zou, F., Wang, L., Hei, X., Chen, D., & Wang, B. (2014). Multi-objective optimization using teaching–learning-based optimization algorithm. *Engineering Applications of Artificial Intelligence*, *26*, 1291–1300.