

HOSTED BY



ELSEVIER

Contents lists available at [ScienceDirect](http://www.sciencedirect.com)

Engineering Science and Technology, an International Journal

journal homepage: <http://www.elsevier.com/locate/jestch>

Full Length Article

Optimization of fused deposition modeling process using teaching-learning-based optimization algorithm

R. Venkata Rao ^{*}, Dhiraj P. Rai

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

ARTICLE INFO

Article history:

Received 15 July 2015

Received in revised form

27 August 2015

Accepted 16 September 2015

Available online 3 November 2015

Keywords:

Rapid prototyping

Fused deposition modeling

Teaching-learning-based-optimization

A posteriori approach

NSGA-II

ABSTRACT

The performance of rapid prototyping (RP) processes is often measured in terms of build time, product quality, dimensional accuracy, cost of production, mechanical and tribological properties of the models and energy consumed in the process. The success of any RP process in terms of these performance measures entails selection of the optimum combination of the influential process parameters. Thus, in this work the single-objective and multi-objective optimization problems of a widely used RP process, namely, fused deposition modeling (FDM), are formulated, and the same are solved using the teaching-learning-based optimization (TLBO) algorithm and non-dominated Sorting TLBO (NSTLBO) algorithm, respectively. The results of the TLBO algorithm are compared with those obtained using genetic algorithm (GA), and quantum behaved particle swarm optimization (QPSO) algorithm. The TLBO algorithm showed better performance as compared to GA and QPSO algorithms. The NSTLBO algorithm proposed to solve the multi-objective optimization problems of the FDM process in this work is a posteriori version of the TLBO algorithm. The NSTLBO algorithm is incorporated with non-dominated sorting concept and crowding distance assignment mechanism to obtain a dense set of Pareto optimal solutions in a single simulation run. The results of the NSTLBO algorithm are compared with those obtained using non-dominated sorting genetic algorithm (NSGA-II) and the desirability function approach. The Pareto-optimal set of solutions for each problem is obtained and reported. These Pareto-optimal set of solutions will help the decision maker in volatile scenarios and are useful for the FDM process.

Copyright © 2015, The Authors. Production and hosting by Elsevier B.V. on behalf of Karabuk University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

In recent years, due to globalization, the market scenario for the manufacturing industries has become extremely competitive and volatile. To survive in such a dynamic market scenario, it is inevitable for the manufacturing industries not only to manufacture products with highest quality at a lowest possible cost, but also fulfill the fast-changing customer desires, consider significance of aesthetics and conform to environmental norms. In order to achieve these goals, manufacturing industries are constrained to adopt flexibility in the production system and minimize time-to-market of their products. In the pursuit of these objectives, manufacturing industries have opted to implement advanced and automated machine tools. In addition to this, the manufacturing industries are also adopting a new paradigm of technology known as the Rapid Prototyping (RP).

RP is a process in which physical objects are directly produced from computer-aided design (CAD) data. RP uses a process in which a physical model is created by selectively adding material in the form of thin cross-sectional layers. Hence, RP is also referred to as additive manufacturing.

RP allows engineers to produce tangible prototypes quickly rather than mere two-dimensional pictures, these prototypes can be used for various important purposes from communicating ideas to co-workers and customers to testing of different aspects of a prototype. Besides this, RP offers a plethora of other advantages such as unambiguous data handling and storage, ability to create complex shapes and interlocking structures, free from tool/workpiece debris, absence of molds, dies, fixtures and patterns, mass customization and democratized manufacturing.

Owing to these advantages, nowadays, RP processes are being widely used in the manufacturing industries not only for production of prototypes but also for large-scale production of biomedical, aeronautical and mechanical models.

The dominant RP processes currently available in the market are fused deposition modeling (FDM), stereolithography (SL), selective laser sintering (SLS), laminated object manufacturing (LOM),

^{*} Corresponding author Tel.: +912612201982, fax: +912612227334.

E-mail address: ravipudirao@gmail.com (R.V. Rao).

Peer review under responsibility of Karabuk University.

3D printing and solid ground curing (SGC). However, the performance of any RP process is measured in terms of build time, quality characteristics such as surface roughness and dimensional accuracy, mechanical and tribological properties, cost of production and energy consumption. These performance measures of RP processes are significantly influenced by their process parameters. Due to this reason, many studies have been directed toward determining the optimum combination of process parameters for RP processes using traditional and advanced optimization techniques.

Pandey et al. [1] applied multi-criteria genetic algorithm (GA) to determine the optimum part deposition orientation in order to minimize the build time and improve the average surface quality of the FDM models. Lee et al. [2] applied Taguchi's method to optimize the process parameters of FDM to achieve the optimum elastic performance of the compliant acrylonitrile butadiene styrene prototype. Byun and Lee [3] applied GA to determine the optimum part deposition orientation in layered manufacturing (LM) in order to minimize the average weighted surface roughness, build time and support structure.

Thrimurthulu et al. [4] applied GA to determine the optimum part deposition orientation in FDM in order to minimize the average weighted surface roughness and build time of the models. Singhal et al. [5] determined the optimum part deposition orientation in SL process using the trust region method in order to achieve the best overall surface quality of the models. Chockalingam et al. [6] used design of experiments in order to optimize the SL process parameters to achieve maximum part strength. Raghunath and Pandey [7] applied Taguchi's method to optimize the SLS process in order to improve the accuracy through shrinkage modeling.

Tyagi et al. [8] used an advanced stickers-based algorithm inspired by the characteristics of deoxyribonucleic acid (DNA) as a tool to achieve the optimal orientation during fabrication of models in LM process. Singhal et al. [9] determined the optimum part deposition orientation for SL and SLS considering multiple objectives simultaneously, such as overall surface quality, build time and support structure of the models. The optimization problem was solved using an algorithm based on the trust region method. Rongji et al. [10] used artificial neural networks (ANN) to formulate the process model for SLS. GA was applied to optimize the process parameters of SLS in order to achieve higher level of accuracy.

Canellidis et al. [11] applied GA to solve the multi-objective optimization problem in SL to improve the fabrication accuracy, minimize the cost and build time. Sood et al. [12] investigated the effect of process parameters on the dimensional accuracy of the FDM models. The optimum combination of process parameters to minimize the dimensional inaccuracy of the models was determined using gray relational analysis (GRA). Sood et al. [13] investigated the effect of process parameters on the mechanical properties of the FDM models. Empirical equations for tensile strength, flexural strength and impact strength of the FDM models were developed using response surface methodology (RSM) and desirability function approach was used to predict the optimum combination of process parameters. Paul and Anand [14] investigated the relationship between the cylindricity tolerance and part build orientation in RP process. Mathematical models were developed and optimum build orientation was determined using a graphical technique.

Paul and Anand [15] presented mathematical analysis of laser energy required for manufacturing parts using SLS process. An optimization model was presented to determine the minimum energy required for manufacturing parts using the SLS process. Sood et al. [16] developed an empirical model for compressive strength of the FDM model, and optimum process parameter setting was predicted using the quantum behaved particle swarm optimization (QPSO) algorithm. Sood et al. [17] investigated the effect of process parameters on the sliding wear of the FDM models, and empirical equation for sliding wear was developed and solved using QPSO al-

gorithm to predict the optimum combination of process parameters for minimizing the sliding wear of the models.

Phatak and Pande [18] applied GA to determine the optimum part orientation in order to minimize the build time and material used and improve the part quality in the RP process. Singh et al. [19] used RSM and desirability function approach to improve the mechanical properties of polyamide parts in SLS process. Li and Zhang [20] applied multi-criteria GA for Pareto based optimization of RP process. Theoretical volume deviation and part height were optimized simultaneously. Boschetto et al. [21] used feed forward neural networks to predict the surface roughness in FDM, and the evaluation function developed was used to find the best solution.

Noriega et al. [22] used ANN to improve the dimensional accuracy of the FDM prismatic parts. Peng et al. [23] applied RSM in combination with fuzzy inference system to develop process models for the FDM process. GA was applied to optimize the responses such as the dimensional error, warp deformation and build time by formulating a single comprehensive response. Gurralla and Regalla [24] applied non-dominated sorting genetic algorithm (NSGA) for optimization of part strength and volumetric shrinkage in the FDM parts.

Rayegani and Onwubolu [25] applied differential evolution (DE) to determine the optimum combination of process parameters in order to improve the tensile strength of the FDM parts. Vijayaraghavan et al. [26] used an improved evolutionary computational approach for the process characterization of 3D printed components. Paul and Anand [27] analyzed the effect of part orientation on cylindricity and flatness error in parts manufactured using the LM process. An algorithm to provide the optimal part orientation to minimize the cylindricity and flatness error was proposed and tested.

Most of the RP process optimization problems involve complex functions and large number of process parameters. In such problems, traditional optimization techniques may get caught into local optima. In addition, traditional optimization techniques require an excellent initial guess of the optimal solution, and the results and the rate of convergence are very sensitive to this guess. In order to overcome these problems and to search a near optimum solution for complex problems, many population-based heuristic algorithms based on evolutionary and swarm intelligence have been developed by researchers in the past two decades. These optimization algorithms require common control parameters like population size, number of generations, elite size, etc. Besides the common control parameters, different algorithms require their algorithm-specific parameters. For example, GA uses mutation rate and crossover rate; particle swarm optimization (PSO) algorithm uses inertia weight, social cognitive parameters, maximum velocity; artificial bee colony (ABC) algorithm uses number of bees (scout, onlooker and employed) and limit; biogeography based optimization (BBO) algorithm requires habitat modification probability, mutation probability, maximum species count, maximum immigration rate, maximum emigration rate, maximum mutation rate, generation count limit and number of genes in each population member; heat transfer search (HTS) algorithm requires conduction factor, convection factor and radiation factor.

Proper tuning of these algorithm-specific parameters is a very crucial factor that affects the performance of the abovementioned algorithms. 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, the common control parameters also need to be tuned which further enhances the effort.

Considering this fact, Rao et al. [28] 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. [29]. Jaya algorithm is also a powerful algorithm-specific parameter-less algorithm but its multi-objective version is not yet developed [30].

Ghasemi et al. [31] proposed a hybrid algorithm of imperialist competitive algorithm and TLBO. The performance of the hybrid algorithm was tested on optimal power flow problem. Chen et al. [32] applied the TLBO algorithm to solve global optimization problems. In order to improve the performance of the TLBO algorithm local learning and self learning methods were assigned. Ghasemi et al. [33] proposed an improved TLBO algorithm using Levy mutation strategy to solve non-smooth optimal power flow problem. The Levy mutation TLBO was effective in solving the optimal power flow problem. Li et al. [34] proposed a discrete TLBO algorithm for realistic flowshop rescheduling problems. The discrete TLBO algorithm showed high searching quality, robustness and efficiency.

Most of the real-world optimization problems are multi-objective in nature, involving multiple conflicting objectives to be satisfied simultaneously. As the RP processes involve more than one performance characteristic, in the case of RP processes, there also arises a need to formulate and solve optimization problems that are multi-objective in nature.

Researchers have solved the multi-objective optimization problem of RP processes, but most of these works are based on a priori approach [35]. 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 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 RP 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.

Therefore, in this work a parameter-less posteriori multi-objective optimization algorithm based on the TLBO algorithm is proposed to solve the multi-objective optimization problems of the FDM process and is named as “Non-dominated Sorting Teaching-Learning-Based Optimization (NSTLBO)” algorithm. 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 toward 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 sections 2 and 3, respectively.

In this work, three single-objective optimization problems and two multi-objective optimization problems of the FDM process are considered. The single-objective and the multi-objective optimization problems of the FDM process are solved using the TLBO algorithm and NSTLBO algorithm, respectively, for the first time.

A computer program for the TLBO algorithm and 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.

2. 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 not only 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_i 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 . Fig. 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/>.

3. 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

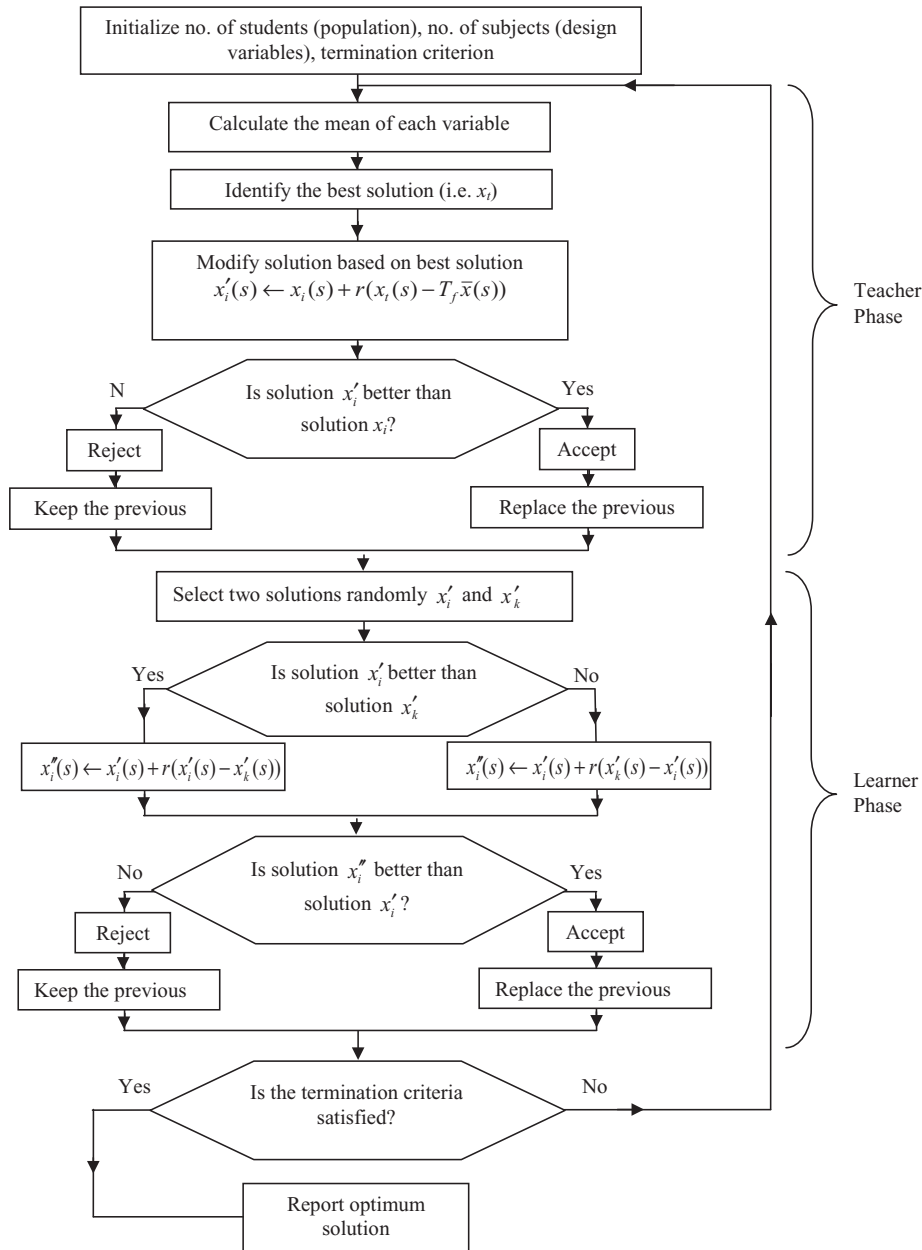


Fig. 1. Flow chart for the TLBO algorithm.

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. [36]. Balasubbareddy et al. [37] have used the non-dominated sorting approach with a hybrid cuckoo search algorithm to solve the multi-objective power flow problem, and Pareto-optimal set of solutions were successfully obtained.

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 toward the good solutions and the population is pushed toward 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 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 $2NP$ 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.

3.1. 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.

3.2. 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, an 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 $m = 1, 2, \dots, M$, sort the set in the worst order of f_m .
- Step 3: For $m = 1, 2, \dots, M$, 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:

$$CD_j = CD_j + \frac{f_m^{j+1} - f_m^{j-1}}{f_m^{\max} - f_m^{\min}} \quad (4)$$

where j is a solution in the sorted list, f_m is the objective function value of m th objective, f_m^{\max} and f_m^{\min} are the population-maximum and population-minimum values of the m th objective function.

3.3. 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-dominance 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-dominance 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.

3.4. 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 not be good with respect to the other objective and vice versa. Thus, in the case of multi-objective optimization problems with mutually conflicting objectives, there may exist not a single but multiple learners 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, 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 to improve the diversity among the learners (as the class is influenced by multiple teachers at the same time). The normalized Euclidean distance between a teacher and a learner is calculated according to Eq. (5).

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

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)

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

3.5. Constraint handling

In order to effectively handle the constraints, a constrained dominance concept [36] is introduced in the proposed approach. In the presence of constraints a solution i is said to dominate solution j if any of the following conditions is true.

1. Solution i is feasible and solution j is not.
2. Solution i and j both are infeasible, but overall constraint violation of solution i is less than overall constraint violation of solution j .
3. Solution i and j both are feasible but solution i dominates solution j .

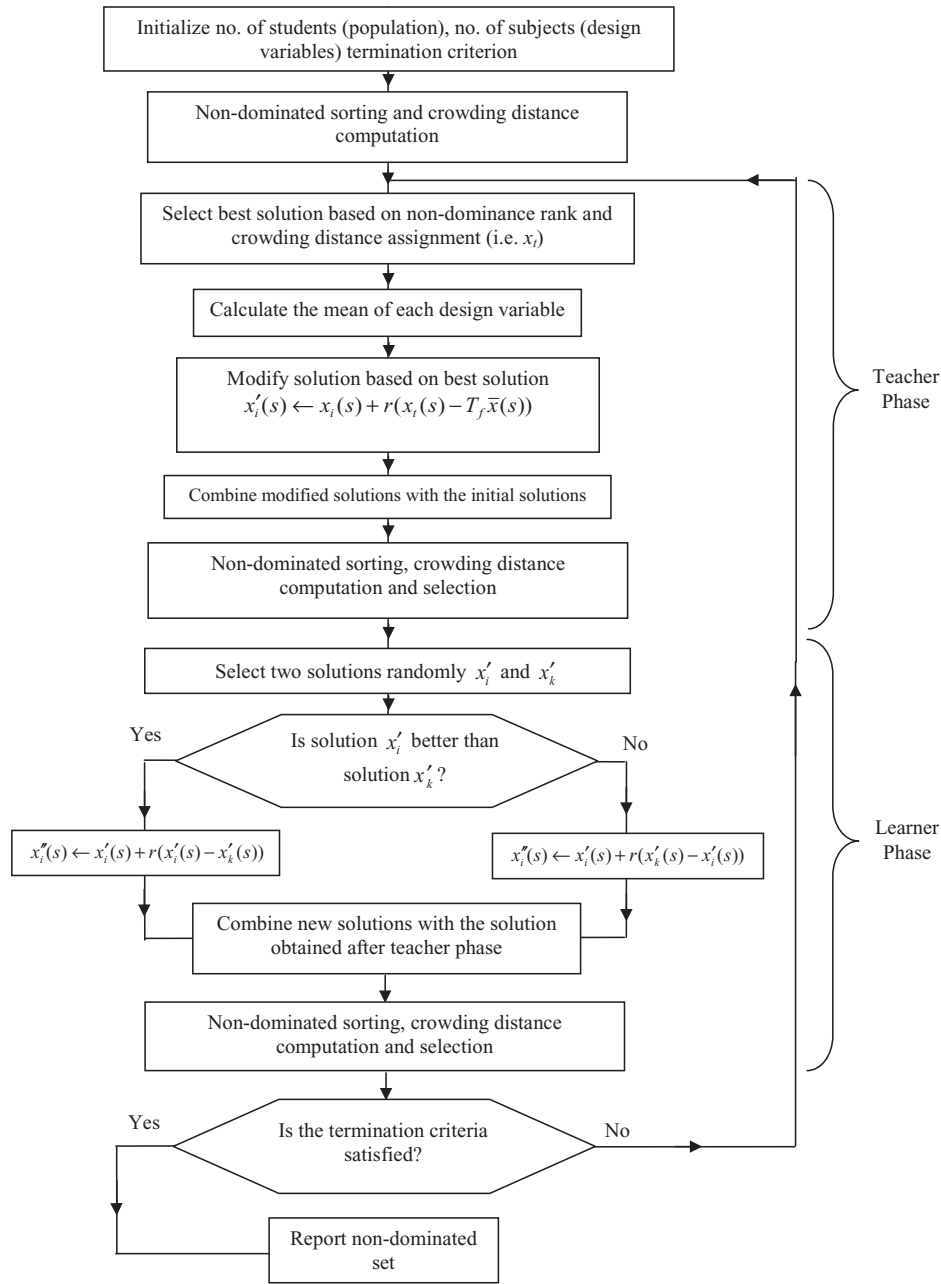


Fig. 2. Flowchart for the NSTLBO algorithm.

This constrained-dominance approach ensures better non-domination rank to feasible solutions as compared to infeasible solutions. The flowchart of NSTLBO algorithm is given in Fig. 2.

3.6. Performance measures

The main aim behind adopting a posteriori approach to solve multi-objective optimization problems is to obtain a diverse set of Pareto optimal solutions. Thus, in order to assess the performance of any multi-objective optimization algorithm two performance measures can be adopted.

3.6.1. Coverage to two sets

This performance measure was proposed by Zitzler et al. [38] and it compares two sets of non-dominated solutions (A, B), and it

gives the percentage of individuals of one set dominated by the individuals of the other set. It is defined as follows:

$$C(A, B) = \frac{|\{b \in B | \exists a \in A : a \preceq b\}|}{|B|} \tag{7}$$

where A and B are the two non-dominated set of solutions under comparison; $a \preceq b$ means a dominated or is equal to b.

The value $C(A, B) = 1$ means that all points in B are dominated by or equal to all points in A. $C(A, B) = 0$ represents the situation when none of the solutions in B are covered by the set A. Here, it is imperative to consider both $C(A, B)$ and $C(B, A)$, since $C(A, B)$ is not necessarily equal to $1 - C(B, A)$. When $C(A, B) = 1$ and $C(B, A) = 0$ then, it is said that the solutions in A completely dominate the solutions in B (i.e. this is the best possible performance of A). $C(A, B)$

represent the percentage of solutions in set B which are either inferior or equal to the solutions in set A ; $C(B, A)$ represent the percentage of solutions in set A which are either inferior or equal to the solutions in set B .

3.6.2. Spacing

This performance measure was proposed by Schott [39], and it quantifies the spread of solutions (i.e. how uniformly distributed the solutions are) along a Pareto front approximation. It is defined as follows:

$$S = \sqrt{\frac{1}{|n-1|} \sum_{i=1}^n (\bar{d} - d_i)^2} \tag{8}$$

where n is the number of non-dominated solutions.

$$d_i = \min_{i,j} \sum_{m=1}^k |f_m^i - f_m^j|, i, j = 1, 2, \dots, n \tag{9}$$

where k denotes the number of objectives and f_m is the objective function value of the m th objective.

$$\bar{d} = \sum_{i=1}^n d_i / |n| \tag{10}$$

$S = 0$, implies that all the solutions are uniformly spread (i.e. best possible performance).

The next section describes the optimization case studies on the FDM process and the same are solved using the TLBO algorithm and NSTLBO algorithm.

4. Case studies

Fused deposition modeling is the most widely used RP process. The system fabricates, layer by layer, components by depositing acrylonitrile butadiene styrene (ABS) in filament form. A temperature-controlled extrusion head is fed with thermoplastic modeling material that is heated to a semi-liquid state. The head extrudes, crushes and directs the filament with precision in ultra-thin layers onto a fixtureless base. Since the FDM process can be used for variety of applications including production of biomedical, aeronautical and mechanical models, many studies have been dedicated to improving the performance of the FDM process by selecting the optimum combination of process parameters. In order to select the optimum combination of process parameters for the FDM process, researchers have applied statistical techniques, heuristic optimization algorithm, fuzzy logic and neural network based optimization techniques [34]. In this work, the optimization problems in the FDM process are solved using the TLBO algorithm and NSTLBO algorithm.

4.1. Case study 1

The optimization problem formulated in this case study is based on the empirical model developed by Sood et al. [16] for prediction of the compressive strength ‘CS’ (MPa) of the FDM models. The objective function, process parameters and their bounds considered in this work are same as those considered by Sood et al. [16] and the process parameters are in the continuous form. The process parameters are: layer thickness ‘A’ (mm), orientation ‘B’ (degree), raster angle ‘C’ (degree), raster width ‘D’ (mm) and air gap ‘E’ (mm). The FDM vantage SE machine was used for fabrication of test specimens. Acrylonitrile butadiene styrene (ABS P400) was used as material for fabrication of test specimen [16].

Table 1 Factors and their levels [13] (case study 5).

Factor	Name	Units	Levels		
			-1	0	1
A	Layer thickness	Mm	0.127	0.1780	0.2540
B	Orientation	Degree	0.000	15.000	30.000
C	Raster angle	Degree	0.000	30.000	60.000
D	Raster width	mm	0.4064	0.4564	0.5064
E	Air gap	mm	0.000	0.0040	0.0080

4.1.1. Objective function

The objective function in terms of coded values of process parameters is expressed by Eq. (11). The coded values at different levels of process parameters are given in Table 1.

$$\begin{aligned} \text{maximize CS} = & 12.0164 + 0.6673 \times A - 1.7123 \times B + 0.3743 \times C \\ & + 0.0396 \times D - 0.3618 \times E + 0.395 \times A^2 + 1.61 \times B^2 \\ & - 0.11 \times C^2 - 0.615 \times D^2 - 0.345 \times E^2 + 0.2914 \times AB \\ & + 0.8326 \times AC - 0.3526 \times AD + 0.0151 \times AE \\ & + 0.1399 \times BC - 0.2124 \times BD - 0.8251 \times BE \\ & - 0.3211 \times CD - 1.1339 \times CE + 0.2364 \times DE \end{aligned} \tag{11}$$

4.1.2. Parameter bounds

The bounds on process parameters are expressed by Eqs. (12) to (16).

$$0.127 \leq A \leq 0.254 \tag{12}$$

$$0 \leq B \leq 30 \tag{13}$$

$$0 \leq C \leq 60 \tag{14}$$

$$0.4064 \leq D \leq 0.5064 \tag{15}$$

$$0 \leq E \leq 0.008 \tag{16}$$

Sood et al. [16] solved the optimization problem for maximization of CS using the QPSO algorithm, considering a population size of 50 and maximum number of generations equal to 500 (i.e. maximum number of function evaluations equal to 25000). Now, the same problem is solved using the TLBO algorithm. For the purpose of fair comparison of results, the maximum number of function evaluations for the TLBO algorithm is maintained as 25,000. The effect of population size on the performance of the TLBO algorithm is now evaluated considering different population sizes such as 10, 20, 30, 40 and 50. For each value of population size the TLBO algorithm is run 30 times, independently maintaining the maximum number of function evaluations as 25,000. Table 2 gives the best, mean, worst, standard deviation, mean function evaluations and mean computational time required by the TLBO algorithm over 30 independent run for maximization of compressive strength.

Table 2 The performance of TLBO algorithm over 30 independent runs for case study 1.

Sr. No.	P	Best	Mean	Worst	SD	Mean FE	Mean CT
1	10	17.998	17.775	17.341	0.3019	2883.0	3.17
2	20	17.998	17.960	17.341	0.1452	4621.3	2.31
3	30	17.998	17.938	17.341	0.183	3437.51	2.28
4	40	17.998	17.972	17.341	0.120	5705.33	2.01
5	50	17.998	17.994	17.895	0.018	4575.00	2.27

P is the population size; SD is the standard deviation; FE is the number of function evaluations required to achieve the best solution; CT is the computational time required to perform 25,000 function evaluations.

Table 3
Optimum solution obtained using TLBO algorithm for population size of 50 (case study 1).

Process parameters	Units	QPSO [16]	TLBO	% Improvement in the objective function value
Layer thickness	A mm	0.254	0.254	
Orientation	B degree	0.036	0	
Raster angle	C degree	59.44	60	
Raster width	D mm	0.422	0.4268	
Air gap	E mm	0.00026	0	
Objective function	CS MPa	17.4751	17.998	2.99%

It is observed that the TLBO algorithm achieved the best (maximum) value for compressive strength equal to 17.998 MPa for all population sizes. However, the TLBO algorithm showed best performance for the population size of 50, with the lowest value of standard deviation and mean function evaluations required to achieve the maximum value of compressive strength is 4535. The mean computational time required by the TLBO algorithm did not change significantly with the change in population size.

Table 3 gives the optimum combination of process parameters for maximization of compressive strength (CS) obtained using TLBO algorithm for a population size of 50 along with the solution obtained by the QPSO algorithm [16]. Fig. 3 shows the convergence graphs for the TLBO algorithm. The convergence graph of the TLBO algorithm rises continuously without getting caught into local optima until maximum value of CS is achieved. Fig. 4 shows of convergence graph of the QPSO algorithm [16]. It is observed from Fig. 4 that the number of function evaluations required by the QPSO algorithm to obtain the maximum value of CS is 7850 (i.e. 157 generations). Further, the convergence graph for QPSO [16] does not show a continuous trend, but rises in steps. This shows that the QPSO algorithm gets trapped into local optima and requires considerable number of function evaluations to recover from the same. This is mainly because performance of QPSO algorithm depends upon the tuning of algorithm-specific parameter called contraction-expansion coefficient (β). Improper tuning of algorithm-specific parameter adversely affects the convergence rate of the algorithm. On the other hand, the TLBO algorithm does not require any algorithm-specific parameters for its working. Hence, the TLBO algorithm has shown a higher convergence rate as compared to the QPSO algorithm.

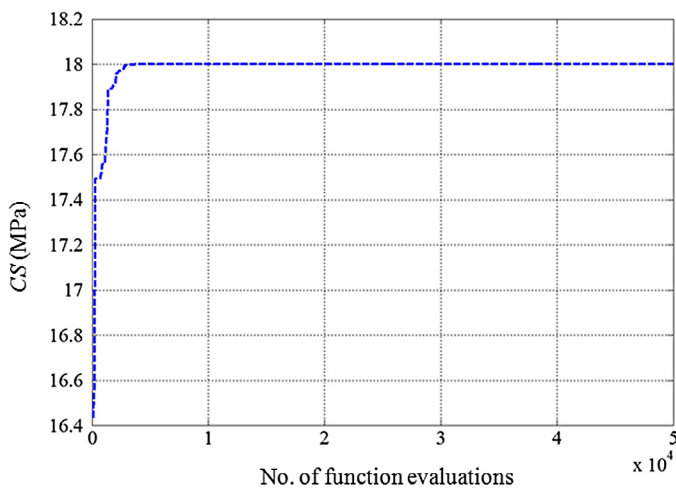


Fig. 3. Convergence graphs for TLBO algorithm for population size of 50 (case study 1).

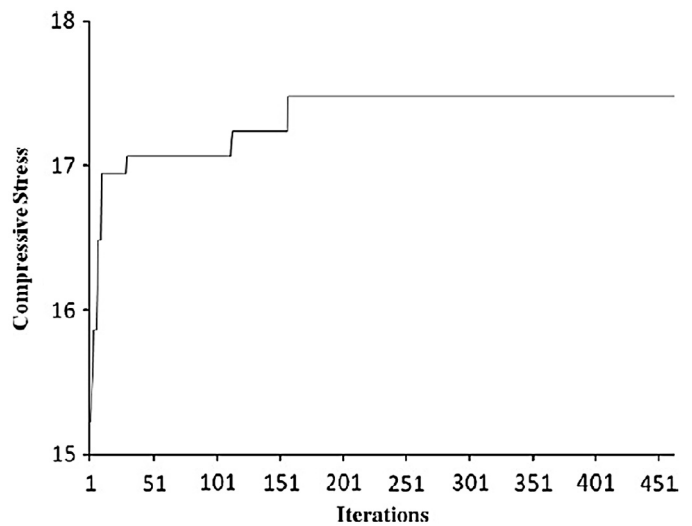


Fig. 4. Convergence graph for QPSO algorithm [16].

The results obtained by the TLBO algorithm are well supported by the experimental observations reported by Sood et al. [16] and they are as follows. The compressive stress decreases with decrease in layer thickness or increase in part build orientation [16]; therefore, to obtain a higher value of compressive stress a high value of layer thickness and low value of part build orientation is desirable. Thus, the TLBO algorithm has provided a value of layer thickness equal to the upper bound (i.e. 0.254 mm) and value of part build orientation equal to the lower bound (i.e. 0 degree). The compressive stress improves with an increase in raster angle [16], thus a high value of raster angle is desirable. Therefore, the TLBO algorithm has provided a value of raster angle equal to the upper bound (i.e. 60 degrees). Further, as a maximum value of compressive strength was observed at a value of raster width and air gap in between the respective lower and upper bounds [16], therefore, the values of raster width (i.e. 0.4268 mm) which lies in between lower and upper bound and a value of air gap equal to the lower bound as (i.e. 0 mm) provided by the TLBO algorithm are logical. On the other hand, the value of part build orientation, raster angle and air gap provided by QPSO (i.e. 0.036 degrees, 60 degrees and 0.00026 mm, respectively) are only close to the respective bounds when choosing bound values for these parameters was more desirable. Thus, the process parameter combination for maximization of compressive strength provided by the TLBO algorithm is more logically supported by experimental observations reported by Sood et al. [16] as compared to the combination of process parameters provided by QPSO.

The best value of compressive strength obtained by the TLBO algorithm (i.e. 17.998 MPa) is 2.99% higher as compared to the value of compressive strength obtained using QPSO algorithm (i.e. 17.475 MPa). The computational time required by the TLBO algorithm to perform 25,000 function evaluations is 2.27 s. However, the computational time required by the QPSO algorithm and the value of contraction-expansion coefficient (β) required by QPSO algorithm is not reported by Sood et al. [16].

4.2. Case study 2

The optimization problem formulated in this case study is based on the empirical model developed by Sood et al. [17] for predicting the sliding wear (mm^3/m) of the models built by the FDM process. The objective function, process parameters and their bounds considered in this work are same as those considered by Sood et al. [17], and the process parameters are in the continuous form. The

Table 4
The performance of TLBO algorithm over 30 independent runs for case study 2.

Sr. No.	P	Best	Mean	Worst	SD	Mean FE	Mean CT
1	10	0.0033	0.0039	0.0076	0.00131	422.0	9.633
2	20	0.0033	0.0033	0.0033	0.0	1504.0	8.832
3	30	0.0033	0.0034	0.0055	0.00041	2387.0	8.467
4	40	0.0033	0.0033	0.0033	0.0	1966.0	7.838
5	50	0.0033	0.0033	0.0033	0.0	2777.0	8.387

P is the population size; SD is the standard deviation; FE is the number of function evaluations required to achieve the best solution; CT is the computational time required to perform 100,000 function evaluations (s).

process parameters are: layer thickness ‘A’ (mm), part built orientation ‘B’ (degree), raster angle ‘C’ (degree), raster width ‘D’ (mm) and air gap ‘E’. The bounds on the process parameters are same as those expressed by Eqs. (12) to (16).

4.2.1. Objective function

The objective function in terms of coded values of process parameters is expressed by Eq. (17). The coded values at different levels of process parameters are same as those given in Table 1.

$$\begin{aligned}
 \text{minimize sliding wear} = & 0.032993 - 0.002136 \times B - 0.005261 \times D \\
 & + 0.002193 \times E - 0.005330 \times A^2 \\
 & - 0.008242 \times B^2 - 0.002150 \times A \times B \quad (17) \\
 & + 0.002602 \times A \times C + 0.003702 \times A \times E \\
 & + 0.003583 \times B \times D + 0.003902 \times C \times E
 \end{aligned}$$

Sood et al. [17] solved the optimization problem for minimization of sliding wear in the FDM models using the QPSO algorithm. A population size of 50 and maximum number of generations equal to 2000 (i.e. maximum number of function evaluations equal to 100,000) was considered by the QPSO algorithm.

Now, the same problem is solved using the TLBO algorithm in order to see whether any improvement in result can be achieved. For the purpose of fair comparison of results, the maximum number of function evaluations for the TLBO algorithm is maintained as 100,000.

The effect of population size on the performance of the TLBO algorithm is now evaluated considering different population sizes such as 10, 20, 30, 40 and 50. For each value of population size the TLBO algorithm is run 30 times independently maintaining the maximum number of function evaluations as 100,000. Table 4 gives the best, mean, worst, standard deviation, mean function evaluations and mean computational time required by the TLBO algorithm over 30 independent runs for minimization of sliding wear.

It is observed that the TLBO algorithm achieved the best (minimum) value of sliding equal to 0.0033 mm³/min for all population sizes. However, the TLBO algorithm showed the best performance for the population size of 20 with the value of standard deviation as 0.0; mean function evaluations required by TLBO algorithm to achieve the best solution for population size of 20 is equal to 1504. The computational time required by the TLBO algo-

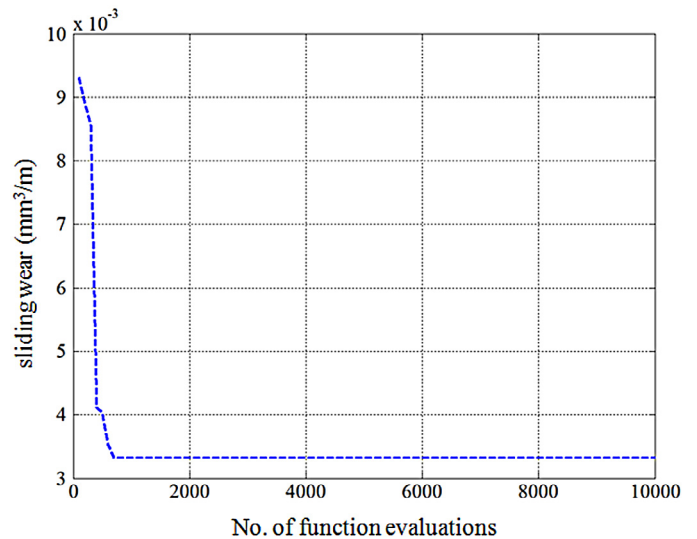


Fig. 5. Convergence graphs for TLBO algorithm for population size of 20 (case study 2).

gorithm did not change significantly with the change in population size.

Table 5 gives the optimum combination of process parameters for minimization of sliding wear obtained using TLBO algorithm for a population size of 20 along with the solution obtained by the QPSO algorithm [17]. Fig. 5 shows the convergence graphs for the TLBO algorithm. The convergence graph of the TLBO algorithm falls steeply without getting caught into local optima until minimum value of sliding wear is achieved. Fig. 6 shows the convergence graph of the QPSO algorithm [17]. It is observed from Fig. 6 that the number of function evaluations required by the QPSO algorithm to obtain the minimum value of sliding wear is 45,850 (i.e. 917 generations). Further, the convergence graph for QPSO [16] does not show a continuously decreasing trend, rather it reduces in steps. This shows that the QPSO algorithm gets trapped into local optima and requires considerable number of function evaluations to recover from the same. This is mainly because performance of QPSO algorithm depends upon the tuning of algorithm-specific parameter called contraction-expansion coefficient (β). Improper tuning of algorithm-specific parameters adversely affects the convergence rate of the algorithm. On the other hand, the TLBO algorithm does not require any algorithm-specific parameters for its working. Hence, the TLBO algorithm has shown a higher convergence rate as compared to the QPSO algorithm.

The results obtained by the TLBO algorithm are well supported by the experimental observation reported by Sood et al. [17] and they are as follows. Wear rate initially increases and then decreases as layer thickness or orientation increases [17]. Therefore a high value of layer thickness and orientation are desirable to achieve low sliding wear. Therefore, the value of layer thickness and

Table 5
Optimum solution obtained by TLBO algorithm for a population size of 20 (case study 2).

Process parameters	Units	QPSO [17]	TLBO	% Improvement in the objective function value
Layer thickness	A mm	0.253	0.254	
Orientation	B degree	0.145	30	
Raster angle	C degree	59.19	60	
Raster width	D mm	0.435	0.5064	
Air gap	E mm	0.00669	0	
Objective function	Wear mm ³ /m	0.007 (0.0358*)	0.0033	90.78%

Asterisk indicates corrected value.

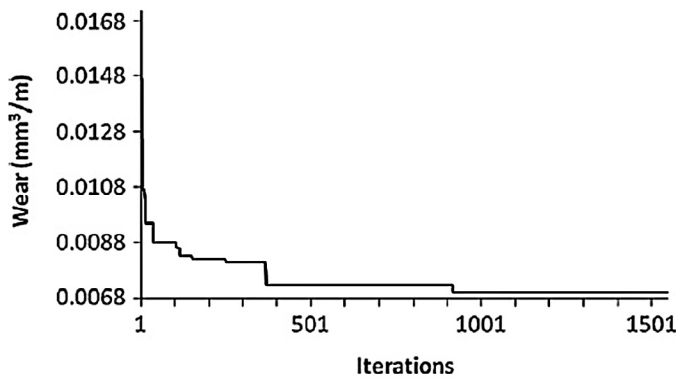


Fig. 6. Convergence graph for QPSO algorithm [17].

orientation provided by the TLBO algorithm equal to their respective upper bounds (i.e. 0.254 mm and 60 degrees, respectively) is logical. However, the value of orientation provided by QPSO (i.e. 0.145 degrees) is close to the lower bound which in fact increases the sliding wear. Wear decreases with increase in raster angle at low level of air gap [17]. Thus, a high value of raster angle must be complimented with a low value of air gap to achieve a minimum value of sliding wear. Therefore, the TLBO algorithm has provided a value of raster angle equal to the upper bound (i.e. 60 degrees) and a value of air gap equal to lower bound (i.e. 0 mm). Sliding wear decreases with increase in raster width [17]. Therefore, the value of raster width provided by TLBO (i.e. 0.5064 mm) is more logical as compared to the value of raster width provided by QPSO (i.e. 0.435 mm).

The best value of sliding wear obtained using the TLBO algorithm (i.e. 0.0033 mm³/m) for a population size of 20 is 90.78% less as compared to the value of sliding wear obtained using QPSO algorithm (i.e. 0.0358 mm³/m). The computational time required by the TLBO algorithm to perform 100,000 function evaluations is 8.387 s. However, the computational time required by the QPSO algorithm and the value of contraction-expansion coefficient (β) required by QPSO algorithm is not reported by Sood et al. [17].

4.3. Case study 3

The optimization problem formulated in this case study is based on the empirical relation developed by Peng et al. [23] considering the effect of controllable factors such as line width compensation ' x_1 ' (mm), extrusion velocity ' x_2 ' (mm/s), filling velocity ' x_3 ' (mm/s), layer thickness ' x_4 ' (mm) on the responses such as dimensional errors, wrap deformation and build time. The experiments were conducted on rapid prototyping machine MEM 300. The test prototypes were made of ABS. Further, Peng et al. [23] converted the three responses into a single compressive response using a fuzzy inference system. The process parameters and their bounds considered in this case study are the same as those considered by Peng et al. [23]. The objective function, process parameters and their bounds are same as those considered by Peng et al. [23], and all the process parameters are in the continuous form.

During fabrication, due to the width of filament, the actual contour path exceeds theoretical contour path by a certain value; therefore an offset is provided, called line width compensation. Extrusion velocity is the velocity at which the molten filament is extruded through the heated nozzle, depending on the filament feeding speed and extrusion pressure. Filling velocity is the moving speed of the nozzle. A very low value of filling velocity leads to the lower fabrication efficiency, the fabricated layers may get burnt because of the searing heat nozzle, and even form knots in the

Table 6
Parameter setting for GA [23].

Parameter	Value/setting
Population type	Double vector
Creation function	Uniform
Fitness scaling	Rank
Elite count	2
Crossover fraction	0.8
Crossover function	Scattered
Mutation function	Gaussian
Mutation fraction	0.2

extreme cases, while a very high filling velocity creates mechanical vibrations in the nozzle, deteriorating part accuracy. A filling velocity is far greater than the extrusion velocity causing thinning of the filament diameter due to dragging and may result in fabrication failure. On the other hand, if the filling velocity is much less than the extrusion velocity, the filament diameter expands with the increase in extrusion velocity, and finally, after increasing to a certain degree, the extruded filament adheres to the outer conical surface of nozzle, resulting in fabrication failure. Therefore, it is important to select an appropriate combination of filling velocity and extrusion velocity in FDM [23].

4.3.1. Objective function

The objective function is expressed by Eq. (18).

$$\begin{aligned} \text{maximize } y = & 806 - 33763.1x_1 + 225.30x_2 + 14.81x_3 - 2759.88x_4 \\ & + 67560.93x_1^2 - 3.57x_2^2 + 0.17x_3^2 + 10607.87x_4^2 \\ & + 17.16x_1x_2 + 172.22x_1x_3 - 1.86x_2x_3 \\ & + 5.57x_2x_4 - 61.53x_3x_4 \end{aligned} \quad (18)$$

4.3.2. Parameter bounds

The parameter bounds are expressed by Eqs. (19) to (22).

$$0.17 \leq x_1 \leq 0.25 \quad (19)$$

$$20 \leq x_2 \leq 30 \quad (20)$$

$$20 \leq x_3 \leq 40 \quad (21)$$

$$0.15 \leq x_4 \leq 0.30 \quad (22)$$

The optimization problem was solved by Peng et al. [23] using genetic algorithm (GA) toolbox in the Matlab software. A population size of 20 and maximum number of generations equal to 100 (i.e. maximum number of function evaluations equal to 2000) were used by GA. The algorithm-specific parameters required by GA and their corresponding values are reported in Table 6. Now, the optimization problem of the FDM process is solved using TLBO algorithm, in order to see whether any improvement in results can be achieved.

For fair comparison of results, the maximum number of function evaluations considered by the TLBO algorithm is maintained as 2000. The effect of population size on the performance of the TLBO algorithm is now evaluated considering different population sizes such as 10, 20, 30, 40 and 50. For each value of population size the TLBO algorithm is run 30 times independently maintaining the maximum number of function evaluations as 2000. Table 7 gives the best, mean, worst, standard deviation, mean function evaluations and mean computational time required by the TLBO algorithm over 30 independent run for maximization of objective function (y).

It is observed that the TLBO algorithm achieved the best (maximum) value for objective function equal to 334.65 for all population sizes. However, the TLBO algorithm showed best performance for the population size of 50, with standard deviation equal to 0.

Table 7
The performance of TLBO algorithm over 30 independent runs for case study 3.

Sr. No.	P	Best	Mean	Worst	SD	Mean FE	Mean CT
1	10	334.65	326.94	285.83	14.48	723.33	0.159
2	20	334.65	332.09	326.14	3.96	908.66	0.148
3	30	334.65	332.50	326.14	3.63	939.00	0.157
4	40	334.65	332.54	326.14	3.65	881.33	0.144
5	50	334.65	334.65	334.65	0.0	905.00	0.141

P is the population size; SD is the standard deviation; FE is the number of function evaluations required to achieve the best solution; CT is the computational time required to perform 2000 function evaluations (s).

The value of mean function evaluations required by the TLBO algorithm to achieve the maximum value of objective function for a population size of 50 is 905. The mean computational time required by the TLBO algorithm did not change significantly with the change in population size.

Table 8 gives the optimum combination of process parameters for maximization of objective function (y) obtained using TLBO algorithm for a population size of 50 along with the solution obtained by the GA [23]. Fig. 7 shows the convergence graphs for the TLBO algorithm. The convergence graph of the TLBO algorithm rises linearly without getting caught into local optima until maximum value of objective function is achieved and then becomes stable.

The best objective function value obtained by the TLBO algorithm for a population size of 50 (i.e. 334.65) is 74.23% higher as compared to the objective function value obtained using GA (i.e. 192.0682), as in this case study the maximization of objective function is desirable. The TLBO algorithm required only 905 function evaluations to achieve the maximum value of objective function for a population size of 50. The number of generations required by GA to achieve its best value is greater than 65 (i.e. maximum number of function evaluations greater than 1300) [23]. This is mainly because the TLBO algorithm does not require tuning of algorithm-specific parameters. On the other hand, GA requires tuning of algorithm-specific parameters such as population type, creation function, fitness scaling, elite count, crossover fraction, crossover function, mutation function, and mutation fraction [23]. Improper selection of these algorithm-specific parameters may lead to stagnation of the algorithm at local optima or a low convergence rate which further increase the computational burden. The computational time required by the TLBO algorithm to perform 2000 function evaluations is 0.141 s. However, the number of function evaluations required by GA to converge at its best solution and the computational time is not reported by Peng et al. [23].

4.4. Case study 4

The multi-objective optimization problem formulated in this case study is based on the empirical models for strength ‘St’ (MPa) and volumetric shrinkage ‘VS’ (%) developed by Gurralla and Regalla [24]. The process parameters and their bounds considered in this work are same as those considered by Gurralla and Regalla [24], and all

Table 8
Optimum solution obtained by TLBO algorithm for a population size of 50 (case study 3).

Process parameters	Units	GA [23]	TLBO	% Improvement in the objective function value	
Line width compensation	x ₁	mm	0.1702	0.25	
Extrusion velocity	x ₂	mm/s	22.4908	21.8523	
Filling velocity	x ₃	mm/s	23.896	40	
Layer thickness	x ₄	mm	0.2875	0.15	
Objective function	y	-	132.2583 (192.0682*)	334.65	74.23%

* Corrected value.

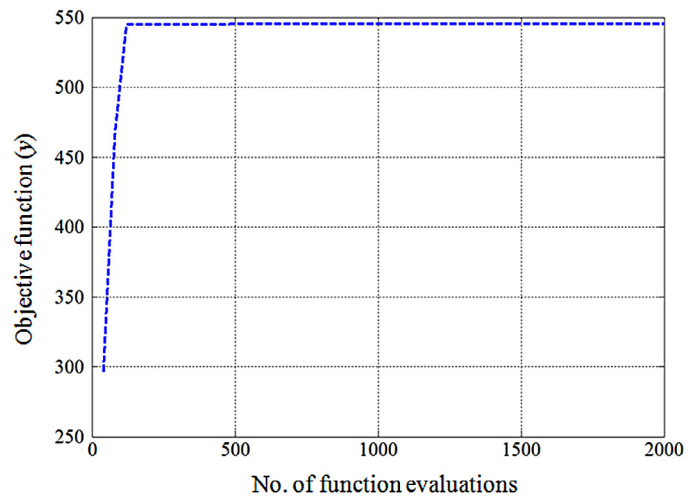


Fig. 7. Convergence graphs for TLBO algorithm for a population size of 50 (case study 3).

the process parameters are in the continuous form. The process parameters are: model interior ‘A’ (cubic cm), horizontal direction ‘B’ (degrees) and vertical direction ‘C’ (degrees). A Stratasys uPrint FDM machine was used to manufacture parts for the purpose of experimentation [24].

4.4.1. Objectives function

The objective functions in terms of coded values of process parameters are expressed by Eqs. (23) and (24). The coded levels of process parameters are given in Table 9.

$$\text{maximize (St)} = 17.51 + 7.19 \times A + 0.73 \times B - 0.37 \times C - 0.032 \times A \times B + 0.25 \times A \times C + 1.41 \times B \times C + 2.5 \times A^2 - 5.86 \times B^2 + 8.56 \times C^2 \quad (23)$$

$$\text{minimize (VS)} = 4.26 + 0.0076 \times A + 0.76 \times B - 0.49 \times C + 0.42 \times A \times B - 0.66 \times A \times C + 1.94 \times B \times C - 0.29 \times A^2 - 1.19 \times B^2 + 2.64 \times C^2 \quad (24)$$

4.4.2. Parameter bounds

$$14.43 \leq A \leq 22.72 \quad (25)$$

$$0 \leq B \leq 45 \quad (26)$$

$$0 \leq C \leq 90 \quad (27)$$

Gurralla and Regalla [24] solved the multi-objective optimization problem using NSGA-II and a non-dominated set of solution was obtained. A population size of 100 and maximum number of generations equal to 100 were considered (i.e. maximum number

Table 9
Process parameters and their coded levels [24] (case study 4).

Factor	Name	Units	Levels		
			-1	0	1
A	Model interior	Cubic cm	14.43	17.53	22.72
B	Horizontal direction	Degrees	0	22.5	45
C	Vertical direction	Degrees	0	45	90

of function evaluations equal to 10,000) for NSGA-II. Besides population size and number of generations, NSGA-II requires tuning of algorithm-specific parameters such as crossover probability and mutation probability which were set to 0.90 and 0.10, respectively [24].

Now, NSTLBO algorithm is applied to solve the multi-objective optimization problem of FDM, in order to see whether any improvements in results can be achieved. For fair comparison of results, the maximum number of function evaluations of NSTLBO algorithm is maintained as 10,000. Gurralla and Regalla [24] provided

a non-dominated set with 100 solutions using NSGA-II. Therefore to obtain a set of 100 non-dominated solutions using NSTLBO, a population size of 100 is chosen. In order to maintain the maximum number of function evaluations as 10,000, maximum number of generations equal to 50 are used by the NSTLBO algorithm. A non-dominated set of solutions is obtained using NSTLBO algorithm and is reported in Table 10.

The Pareto-front so obtained is shown in Fig. 8. It can be observed from Fig. 8 that there is a remarkable increase in the part strength at the expense of volumetric shrinkage. Thus, Fig. 8 clearly explains the mutually conflicting nature of the two objectives. It is also observed that the results obtained using NSTLBO algorithm are well supported by the experimental observations of Gurralla and Regalla [24] which are as follows. As the model interior (A) increases, the amount of volume of material embedded in the part increases, thereby giving good strength to the part [24]. Therefore, the NSTLBO algorithm has maintained the model interior at its upper bound (i.e. A = 22.72 cm³).

Table 10
Non-dominated set of solutions obtained using NSTLBO algorithm in terms of uncoded values of process parameters (case study 4).

Sr. No.	A	B	C	St	VS	Sr. No.	A	B	C	St	VS
1	22.72	0	71.3475	22.6808	0.7034	51	22.72	6.6758	90	31.2588	2.6846
2	22.72	0	36.2723	22.9122	0.7053	52	22.72	6.9413	44.973	31.3616	2.7378
3	22.72	0	36.7515	23.1072	0.7096	53	22.72	7.2113	44.991	31.4953	2.7969
4	22.72	0	37.6403	23.4882	0.7237	54	22.72	7.4588	44.9978	31.6104	2.8499
5	22.72	0	37.9508	23.6278	0.7306	55	22.72	8.1	44.9798	31.876	2.9808
6	22.72	0	38.7158	23.9859	0.7519	56	22.72	8.2598	45	31.9584	3.0163
7	22.72	0	39.0308	24.1381	0.7624	57	22.72	8.6468	45	32.1205	3.0954
8	22.72	0	39.258	24.251	0.7707	58	22.72	9.0113	44.9618	32.2436	3.1647
9	22.72	0	39.942	24.6005	0.7987	59	22.72	9.279	44.9618	32.3503	3.2183
10	22.72	0	40.2345	24.7544	0.8122	60	22.72	9.5873	45	32.5	3.285
11	22.72	0	40.5	24.8961	0.8252	61	22.72	9.7718	44.9955	32.5697	3.3214
12	22.72	0	40.8375	25.0808	0.8428	62	22.72	10.08	45	32.6899	3.3825
13	22.72	0	41.4855	25.4463	0.88	63	22.72	10.3545	45	32.7944	3.4366
14	22.72	0	41.841	25.6513	0.9021	64	22.72	10.7145	45	32.9279	3.5068
15	22.72	0	42.12	25.8162	0.9205	65	22.72	11.196	45	33.1017	3.5998
16	22.72	0	42.6915	26.1626	0.9607	66	22.72	11.5515	44.9978	33.2251	3.6673
17	22.72	0	42.8828	26.281	0.9749	67	22.72	11.8328	45	33.3237	3.721
18	22.72	0	43.182	26.4683	0.9979	68	22.72	12.3503	45	33.4964	3.8179
19	22.72	0	43.533	26.6914	1.026	69	22.72	12.546	44.9888	33.5519	3.8527
20	22.72	0	43.686	26.7905	1.0387	70	22.72	12.852	45	33.6587	3.911
21	22.72	0.0225	43.9538	26.9786	1.0666	71	22.72	13.2098	45	33.7703	3.9763
22	22.72	0	44.5005	27.3297	1.1102	72	22.72	13.5023	45	33.8601	4.0297
23	22.72	0	44.64	27.4244	1.1232	73	22.72	13.9185	44.9978	33.9826	4.1044
24	22.72	0	44.991	27.6658	1.1567	74	22.72	14.382	45	34.1168	4.1871
25	22.72	0.099	44.9933	27.7284	1.1813	75	22.72	14.4248	44.9978	34.1274	4.1944
26	22.72	0.5175	45	27.9863	1.2832	76	22.72	14.8793	45	34.2541	4.2745
27	22.72	0.7448	44.8898	28.0466	1.3273	77	22.72	15.4665	45	34.4085	4.3761
28	22.72	0.9112	44.9258	28.1708	1.3711	78	22.72	15.7298	45	34.4749	4.4209
29	22.72	1.0238	45	28.2897	1.4056	79	22.72	16.263	44.9843	34.593	4.5086
30	22.72	1.2555	44.9258	28.3749	1.4536	80	22.72	16.6028	45	34.6846	4.568
31	22.72	1.6088	45	28.6307	1.5447	81	22.72	17.1338	44.9753	34.7861	4.6518
32	22.72	1.8765	45	28.7849	1.6082	82	22.72	17.469	45	34.8755	4.7103
33	22.72	2.0543	44.9843	28.8747	1.6482	83	22.72	18.2408	44.9573	34.9994	4.8273
34	22.72	2.4953	44.982	29.1208	1.751	84	22.72	18.54	45	35.0877	4.8818
35	22.72	2.6595	44.9595	29.1958	1.7867	85	22.72	19.0688	45	35.1822	4.9641
36	22.72	2.808	45	29.306	1.8253	86	22.72	19.125	45	35.192	4.9729
37	22.72	3.0195	45	29.4216	1.874	87	22.72	19.719	45	35.2901	5.0639
38	22.72	3.2243	44.9618	29.507	1.9174	88	22.72	20.2995	45	35.3778	5.1511
39	22.72	3.3953	44.9798	29.6117	1.9584	89	22.72	20.826	45	35.4507	5.2288
40	22.72	3.6675	45	29.7698	2.0223	90	22.72	21.3098	45	35.5122	5.2993
41	22.72	4.1243	44.7143	29.8106	2.094	91	22.72	22.212	45	35.6121	5.4276
42	22.72	4.0478	44.9955	29.9663	2.1079	92	22.72	22.7025	45	35.6585	5.4956
43	22.72	4.3178	44.955	30.0787	2.1643	93	22.72	23.1683	44.9978	35.6957	5.5587
44	22.72	4.581	44.9303	30.1949	2.2201	94	22.72	24.084	45	35.7594	5.6813
45	22.72	5.4563	45	30.6809	2.4215	95	22.72	24.8423	45	35.7959	5.7795
46	22.72	5.6363	44.9955	30.765	2.4602	96	22.72	25.6005	45	35.8192	5.875
47	22.72	5.8433	44.9933	30.8639	2.5052	97	22.72	26.2283	45	35.8284	5.9519
48	22.72	5.9445	45	30.9168	2.5279	98	22.72	26.5568	45	35.8296	5.9914
49	22.72	6.1335	45	31.0062	2.5685	99	22.72	23.5778	0	35.8324	7.7284
50	22.72	6.4755	44.9618	31.1401	2.6376	100	22.72	21.1455	0	35.9016	7.8091

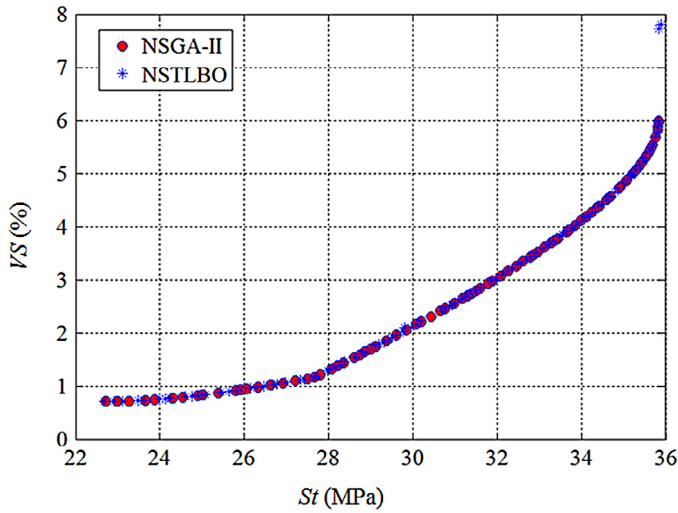


Fig. 8. The comparison of Pareto-fronts obtained using NSTLBO algorithm and NSGA-II [24] (case study 4).

The horizontal direction (*B*) and vertical direction (*C*), describe the angle of deposition of material in horizontal and vertical planes of the build platform, respectively. Therefore, as the interaction effect of *B* and *C* increases the specimens load bearing capacity also increases [24]. Therefore, it can be observed from Fig. 8 and Table 10 that part strength increases on the expense of volumetric shrinkage as *C* increases toward its upper bound (i.e. *C* = 90 degrees) while *B* is maintained to its lower bound (i.e. *B* = 0 degrees). The part strength further increases on the expense of volumetric shrinkage as *B* increases from its lower bound toward the mid value (i.e. 22.5 degrees) while *C* is maintained at its upper bound (i.e. *C* = 90 degrees).

Gurralla and Reggala [24] had stated that any change in parameter setting may not yield a better strength that 35.83 MPa and any improvement in shrinkage value below 0.77% is not possible. However, the extreme points in Fig. 8 show that, the maximum value of part strength achieved by NSTLBO algorithm is 35.90 MPa which is higher than the maximum value of part strength achieved by NSGA-II [24]. Also, the minimum value of volumetric shrinkage achieved by NSTLBO is 0.70% which is lower than the minimum value of volumetric shrinkage achieved by NSGA-II [24].

It is observed from Fig. 8 that the Pareto-front obtained using NSTLBO algorithm and NSGA-II overlap each other. Therefore, in order to quantify the performance of NSTLBO algorithm, two performance measures, that is, coverage and spacing of non-dominated sets have been adopted. The best, mean and worst values of coverage and spacing for NSTLBO algorithm obtained over 30 independent runs are reported in Table 11. In Table 11, *P* represents the non-dominated set of solutions obtained using NSTLBO algorithm; *Q* represents the non-dominated set of solutions obtained using NSGA-II.

The values of mean and standard deviation reported in Table 11 indicate that there is very less variation in the results obtained using the NSTLBO algorithm over 30 independent runs. The value of cov-

erage $C(P, Q)_{best} = 0.3267$ implies that, considering best values 32.67% of the non-dominated solutions obtained using NSGA-II are inferior to the non-dominated solutions obtained NSTLBO algorithm. On the other hand, the coverage value $C(P, Q)_{best} = 0.02$ implies that 2% of the non-dominated solutions obtained using NSTLBO are inferior to the solutions obtained using NSGA-II. The spacing for the non-dominated set of solutions obtained using NSGA-II is calculated as $S(Q) = 0.09564$. It is observed that the non-dominated set of solutions obtained using NSTLBO algorithm show better distribution as compared to the non-dominated set of solutions obtained using NSGA-II, as $S(P)_{best} < S(Q)$.

Thus, the values of performance indicators that is, coverage and spacing indicate that, the non-dominated set of solutions obtained using NSTLBO algorithm is better than the non-dominated set of solutions obtained using NSGA-II. It may be stated here that, the NSTLBO algorithm requires only common control parameters such as population size and number of generations and does not require tuning of any algorithm-specific parameters for its working.

The non-dominated set of solutions obtained using NSTLBO algorithm will be useful to the process planner. Since none of the solutions in the non-dominated set is absolutely better than any other, all of them are acceptable solutions. Each solution in the non-dominated set corresponds to a particular order of importance of objectives, and the choice of one solution over another depends upon the requirement of the process planner. The process planner may require high strength or low volumetric shrinkage, based on the requirement a suitable combination of process parameters can be selected from Table 10.

4.5. Case study 5

The optimization problem formulated in this case study is based on the empirical models developed by Sood et al. [13] relating mechanical properties such as tensile strength '*Ts*' (MPa), flexural strength '*Fs*' (MPa) and impact strength '*Is*' (MPa) of the models fabricated using the FDM process with the process parameters such as layer thickness '*A*' (mm), orientation '*B*' (degree), raster angle '*C*' (degree), raster width '*D*' (mm), air gap '*E*' (mm). The objective functions, process parameters and their bounds considered in this case study are same as those considered by Sood et al. [13], and all the process parameters are in the continuous form. An FDM machine by Stratasys Inc., USA was used for manufacturing parts for the purpose of experimentation.

4.5.1. Objective functions

The objective functions in terms of coded levels of process parameters are expressed by Eqs. (28) to (30). The coded levels of process parameters are same as those given in Table 1.

$$\begin{aligned} \text{maximize } (Ts) = & 13.5625 + 0.7156 \times A - 1.3123 \times B + 0.9760 \times C \\ & + 0.5183 \times E + 1.1671A^2 - 1.3014 \times B^2 \\ & - 0.4363(A \times C) + 0.4364(A \times D) - 0.4364(A \times E) \\ & + 0.4364(B \times C) + 0.4898(B \times E) - 0.5389(C \times D) \\ & + 0.5389(C \times E) - 0.5389(D \times E) \end{aligned} \quad (28)$$

$$\begin{aligned} \text{maximize } (Fs) = & 29.9178 + 0.8719 \times A - 4.8741 \times B + 2.4251 \times C \\ & - 0.9096 \times D + 1.6626E - 1.7199(A \times C) \\ & + 1.7412(A \times D) - 1.1275(A \times E) + 1.0621(B \times E) \\ & + 1.0621(C \times E) - 1.0408(D \times E) \end{aligned} \quad (29)$$

$$\begin{aligned} \text{maximize } (Is) = & 0.401992 + 0.034198 \times A + 0.008356B \\ & + 0.013673C + 0.021383A^2 + 0.008077(B \times D) \end{aligned} \quad (30)$$

Table 11 The best, mean and worst values of coverage for NSTLBO algorithm.

	Best	Mean	Worst	SD
<i>C</i> (<i>P</i> , <i>Q</i>)	0.3267	0.3280	0.3465	0.00431
<i>C</i> (<i>Q</i> , <i>P</i>)	0.0100	0.042	0.1700	0.03408
<i>S</i> (<i>P</i>)	0.0039	0.0066	0.0064	0.002512

P is the non-dominated set of solutions obtained using NSTLBO algorithm; *Q* is the non-dominated set of solutions obtained using NSGA-II; SD is the standard deviation.

Table 12
Optimum solution by desirability function approach in terms of uncoded values of process parameters [13] (case study 5).

A (mm)	B (degree)	C (degree)	D (mm)	E (mm)	T _s (MPa)	F _s (MPa)	I _s (MPa)
0.254	0.0589	60	0.4064	0.008	16.3405	37.6383	0.4710

4.5.2. Parameter bounds

The bounds on process parameters are expressed by Eqs. (31) to (35).

$$0.127 \leq A \leq 0.254 \tag{31}$$

$$0 \leq B \leq 30 \tag{32}$$

$$0 \leq C \leq 60 \tag{33}$$

$$0.4064 \leq D \leq 0.5064 \tag{34}$$

$$0 \leq E \leq 0.008 \tag{35}$$

Sood et al. [13] solved the multi-objective optimization problem using RSM and desirability function approach assigning equal importance to all three objectives. The unique optimum solution obtained by them is reported in Table 12. Now, the same problem is solved using NSTLBO algorithm in order to see whether any improvement in results can be obtained.

A population size of 50 and maximum number of generations equal to 100 are considered by the NSTLBO algorithm. The non-dominated set of solutions obtained using the NSTLBO algorithm is reported in Table 13. Figs. 9 and 10 show the graphical representations of the non-dominated set of solutions for the FDM process obtained using NSTLBO algorithm. It is observed from Figs. 9 and

Table 13
Non-dominated set of solutions obtained using NSTLBO algorithm (case study 5).

Sr. No.	A (mm)	B (degree)	C (degree)	D (mm)	E (mm)	T _s (MPa)	F _s (MPa)	I _s (MPa)
1	0.254	29.9847	60	0.5064	0.008	14.8395	29.5687	0.5036
2	0.254	29.6667	60	0.5064	0.008	14.9023	29.6495	0.5033
3	0.254	28.4383	60	0.5064	0.008	15.1336	29.9617	0.5019
4	0.254	27.7108	60	0.5064	0.008	15.2623	30.1466	0.5012
5	0.254	27.1774	60	0.5048	0.008	15.3704	30.2779	0.5004
6	0.254	26.0799	60	0.5064	0.008	15.5287	30.561	0.4994
7	0.1926	0	60	0.4064	0.008	15.6834	40.5129	0.4244
8	0.1994	0	60	0.4064	0.008	15.6906	40.1202	0.4293
9	0.1758	0	60	0.4064	0.0077	15.6949	41.2486	0.4142
10	0.1827	0	60	0.4064	0.008	15.7209	41.0971	0.418
11	0.1844	0.9234	59.7231	0.4064	0.0079	15.8064	40.666	0.4189
12	0.1703	0	60	0.4064	0.008	15.847	41.8194	0.4116
13	0.2243	0	60	0.4064	0.008	15.942	38.6507	0.4517
14	0.254	24.6297	60	0.4823	0.0079	15.9991	30.8327	0.4953
15	0.1602	0	60	0.4064	0.008	16.0168	42.4134	0.4075
16	0.231	0	60	0.4064	0.008	16.0666	38.2433	0.4588
17	0.1632	1.1156	60	0.4064	0.008	16.1131	41.9397	0.4086
18	0.1552	0	60	0.4064	0.008	16.1216	42.7041	0.4059
19	0.2538	23.5921	60	0.472	0.008	16.2474	31.0952	0.4932
20	0.2394	0	60	0.4064	0.008	16.2752	37.777	0.4684
21	0.2414	0	60	0.4064	0.008	16.3269	37.6489	0.4709
22	0.2391	0.6263	60	0.4064	0.008	16.3542	37.6272	0.4681
23	0.246	0	60	0.4064	0.008	16.461	37.39	0.4765
24	0.1393	0	60	0.4064	0.008	16.5538	43.6358	0.4025
25	0.243	1.9817	60	0.4064	0.008	16.6446	37.0601	0.4729
26	0.254	21.4979	60	0.4524	0.0079	16.6988	31.537	0.4906
27	0.1325	0	60	0.4064	0.008	16.7836	44.0347	0.4019
28	0.254	19.3104	60	0.4585	0.0079	16.8215	32.0996	0.4897
29	0.2443	3.7113	60	0.4064	0.008	16.8761	36.5412	0.4744
30	0.127	0	60	0.4064	0.008	16.9877	44.3551	0.4017
31	0.2488	4.46	60	0.4064	0.008	17.089	36.0865	0.4802
32	0.2509	4.5042	60	0.4064	0.008	17.1629	35.9593	0.4829
33	0.2539	2.4251	38.3051	0.496	0.0002	17.2412	36.9746	0.4649
34	0.254	5.3593	60	0.4064	0.008	17.3455	35.5597	0.487
35	0.254	16.647	60	0.4219	0.0079	17.3859	32.6796	0.4875
36	0.254	15.6237	60	0.4218	0.0079	17.4321	32.9533	0.4873
37	0.2537	10.9922	60	0.4083	0.008	17.6041	34.1173	0.4867
38	0.254	12.5104	60	0.4064	0.008	17.6632	33.7424	0.4872
39	0.254	11.4578	17.2222	0.5063	0	17.8162	34.7872	0.4639
40	0.254	13.0022	8.7288	0.5064	0	17.8566	34.53	0.4617
41	0.254	10.7913	12.1227	0.5064	0	18.004	35.2651	0.4608
42	0.254	9.8909	9.1738	0.5064	0	18.1552	35.7429	0.4585
43	0.254	8.5054	9.0396	0.5064	0	18.2603	36.2967	0.4569
44	0.254	10.0995	2.054	0.5064	0	18.3199	35.9536	0.4554
45	0.254	10.0635	0	0.5064	0	18.3757	36.0524	0.4545
46	0.254	0	6.9659	0.5064	0	18.4709	39.7481	0.4466
47	0.254	2.1679	7.3255	0.5064	0	18.499	38.8754	0.4492
48	0.254	7.7465	0	0.5064	0	18.5581	36.9693	0.4519
49	0.254	0.8232	0	0.5064	0	18.7332	39.7092	0.4443
50	0.254	2.0975	0	0.5064	0	18.7426	39.2049	0.4457

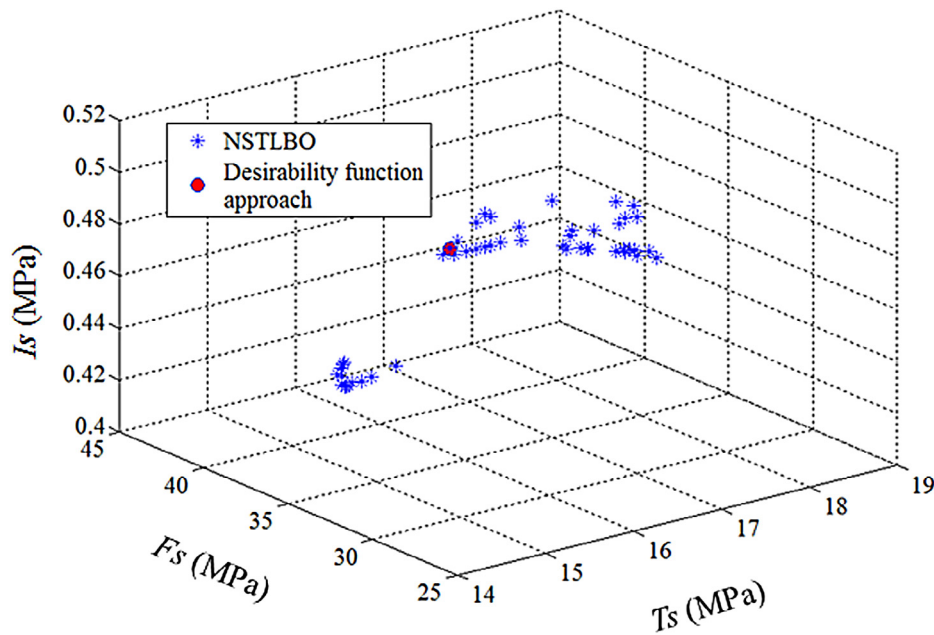


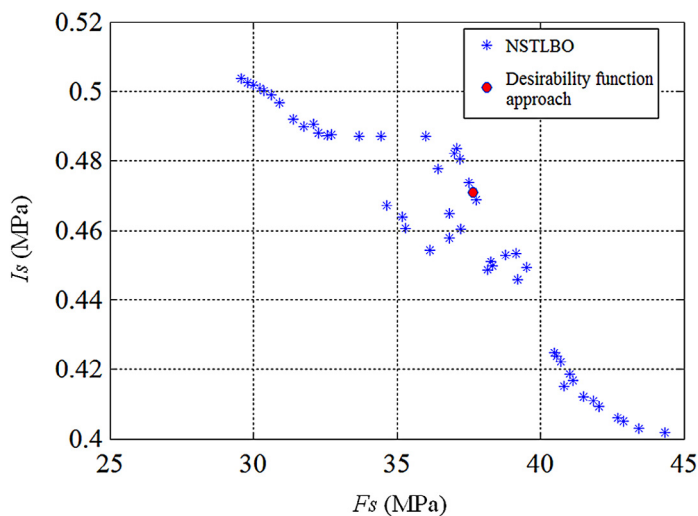
Fig. 9. Non-dominated set of solutions obtained by NSTLBO algorithm and the solution obtained using desirability function approach [13] (case study 5).

10 that the objectives considered in this case study are mutually conflicting in nature.

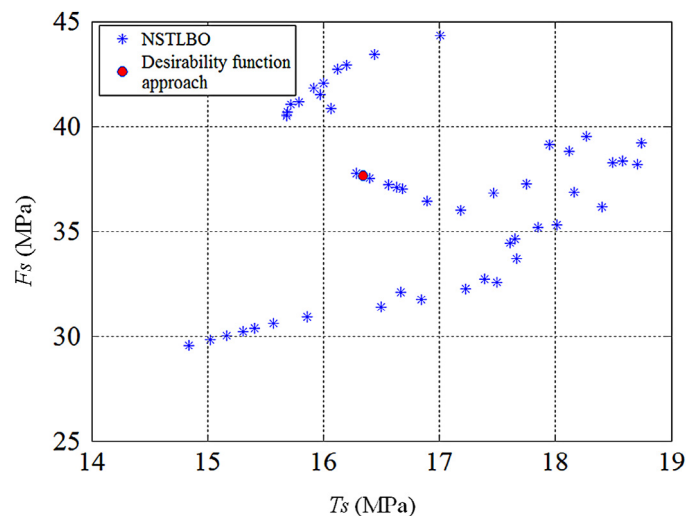
Sood et al. [13] have applied a statistical technique and have obtained a unique optimum solution to the problem. This unique solution corresponds to a specific set of weights assigned to the objectives (i.e. equal weightage to all objectives in this case). These weights depend upon the order of importance of objectives and may change if the order of importance of objective changes. The optimality of a unique solution may be invalid if the value of weights assigned to the objective changes. To mitigate this limitation, a non-dominated set of solutions (Pareto-optimal set) is obtained using NSTLBO algorithm. The non-dominated set consists of multiple solutions, and all the solutions are equally good. Each solution in the non-dominated set corresponds to a particular order of importance of objectives, thus giving flexibility to the process planner to

choose one solution from the non-dominated set which best suits the requirement. The non-dominated set of solutions obtained using NSTLBO algorithm is useful especially in volatile scenarios where the order of importance of objectives are subject to frequent change. Figs. 9 and 10 show that the NSTLBO algorithm has provided 50 non-dominated solutions in a single simulation run which also covers the unique solution provided by desirability function approach.

Now, the results obtained using the NSTLBO algorithm are co-ordinated with the experimental data reported by Sood et al. [13]. Solution no. 1 in the non-dominated set of solutions reported in Table 13 corresponds to a maximum importance to impact strength. It is observed that layer thickness, orientation, raster angle and raster width have a positive influence on the impact strength, therefore the values of these process parameters equal to their respective upper bounds (i.e. 0.254 mm, 2.9847 degrees, 60 degrees and 0.5064 mm,



(a) Flexural strength v/s Impact strength



(b) Tensile strength v/s Flexural strength

Fig. 10. (a, b) 2-D plots of non-dominated solutions obtained by NSTLBO algorithm and the solution obtained using desirability function approach [13] (case study 5).

respectively) is selected by the NSTLBO algorithm. Thus, the results obtained using the NSTLBO algorithm are well supported by the experimental observations.

Solution no. 30 in the non-dominated set reported in Table 13 corresponds to a maximum importance to flexural strength. Orientation, raster angle and air gap most predominantly affect the flexural strength. Orientation has a negative effect on flexural strength; therefore, the value of orientation equal to the lower bound (i.e. 0 degrees) is selected by the NSTLBO algorithm. Raster angle and air gap have a positive influence on flexural strength; therefore, the values of raster angle and air gap equal to their respective upper bound (i.e. 60 degrees and 0.008 mm, respectively) is selected by the NSTLBO algorithm. Layer thickness and raster width have a negative influence on flexural strength; therefore, the values of layer thickness and raster width equal to their respective lower bounds (i.e. 0.127 mm and 0.4064 mm) is selected by the NSTLBO algorithm.

Solution no. 50 in the non-dominated set of solutions reported in Table 13 corresponds to a maximum importance to tensile strength. Layer thickness and orientation are the most predominantly affected tensile strength. The value of layer thickness equal to the upper bound (i.e. 0.254 mm) is selected by the NSTLBO algorithm because layer thickness has a positive influence on tensile strength. However, the interactions of layer thickness with air gap and raster angle have a negative effect on tensile strength; therefore, the values of air gap and raster angle equal to their respective lower bounds (i.e. 0 degrees and 0 mm, respectively) are selected by the NSTLBO algorithm. The value of orientation close to the lower bound (i.e. 2.0975 degrees) is selected by the NSTLBO algorithm because orientation has a negative influence on tensile strength. The interaction of layer thickness and raster width has a positive influence on T_z ; therefore, the value of raster width equal to upper bound (i.e. 0.5064) is selected by the NSTLBO algorithm. The computational time required by the NSTLBO algorithm to obtain the Pareto-optimal set is 4.4851 s. The computational time required by desirability function approach to obtain the unique optimum solution is not reported by Sood et al. [13].

The optimization problems formulated in all the FDM process optimization case studies considered in section 4 are based on the mathematical models developed by previous researchers based on experimentation. The real data obtained from experimentation were used by the previous researchers for formulation of mathematical models. The confirmation experiments for the developed mathematical models were also conducted by the previous researchers such as Sood et al. [16,17], Peng et al. [23], Gurralla and Regalla [24] and Sood et al. [13] for case studies 1 to 5, respectively. In addition, the previous researchers had solved the optimization problems using techniques such as QPSO [16,17], GA [23], NSGA-II [24] and desirability function approach [13]. Now the same mathematical models have been solved using TLBO and NSTLBO algorithms, and the results obtained using TLBO and NSTLBO algorithms are compared with the results obtained by the previous researchers. Therefore, confirmation experiments for the results obtained using TLBO and NSTLBO algorithm are not required as the mathematical models used as objective functions in this work were already validated by previous researchers by conducting thorough experimentation. 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. Therefore, the optimization problems formulated in this work are continuous parameter optimization problems. [However, in actual practice, the values allowed by the FDM machine which are closer to the suggested optimum values may be considered.]

The results obtained using TLBO and NSTLBO algorithms are better than the results obtained by previous researchers using algorithms such as QPSO [16,17], GA [23], NSGA-II [24]. In addition,

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 rapid prototyping systems.

5. Conclusions

In this work single-objective and multi-objective optimization aspects of a widely used RP process, namely, FDM are considered. Three single-objective optimization problems and two multi-objective optimization problems pertaining to FDM are solved using the TLBO algorithm and NSTLBO algorithm, respectively.

The TLBO algorithm showed better performance as compared to GA and QPSO algorithms in terms of objective function value with a higher convergence rate. The NSTLBO algorithm showed better performance as compared to NSGA-II in terms coverage and spacing of the Pareto optimal set. Thus, the results presented in this work are useful for real RP systems. The results obtained in this work are well supported with the experimental data reported by the previous researchers.

In the present work the TLBO and NSTLBO algorithms are applied to solve the optimization problems of the FDM process only. These algorithms may also be applied to solve the optimization problems pertaining to other RP processes such as stereolithography, selective laser sintering, laminated object manufacturing, 3D printing, solid ground curing, etc.

References

- [1] P.M. Pandey, K. Thirumurthulu, N.V. Reddy, Optimal part deposition orientation in FDM by using a multicriteria genetic algorithm, *Int. J. Prod. Res.* 42 (2004) 4069–4089, doi:10.1080/00207540410001708470.
- [2] B.H. Lee, J. Abdullah, Z.A. Khan, Optimization of rapid prototyping parameters for production of flexible ABS object, *J. Mater. Process. Technol.* 169 (2005) 54–61, doi:10.1016/j.jmatprotec.2005.02.259.
- [3] H.S. Byun, K.H. Lee, Determination of the optimal part orientation in layered manufacturing using a genetic algorithm, *Int. J. Prod. Res.* 43 (2005) 2709–2724, doi:10.1080/00207540500031857.
- [4] K. Thirumurthulu, P.M. Pandey, N.V. Reddy, Optimum part deposition orientation in fused deposition modeling, *Int. J. Mach. Tools Manuf.* 44 (2004) 585–594, doi:10.1016/j.ijmactools.2003.12.004.
- [5] S.K. Singhal, A.P. Pandey, P.M. Pandey, A.K. Nagpal, Optimum part deposition orientation in stereolithography, *Comput. Aided Des. Appl.* 2 (2005) 319–328, doi:10.1016/j.ijmactools.2003.12.004.
- [6] K. Chockalingam, N. Jawahar, K.N. Ramanathan, P.S. Banerjee, Optimization of stereolithography process parameters for part strength using design of experiments, *Int. J. Adv. Manuf. Technol.* 29 (2006) 79–88, doi:10.1007/s00170-004-2307-0.
- [7] N. Raghunath, P.M. Pandey, Improving accuracy through shrinkage modelling by using Taguchi method in selective laser sintering, *Int. J. Mach. Tools Manuf.* 47 (2007) 985–995, doi:10.1016/j.ijmactools.2006.07.001.
- [8] S.K. Tyagi, A. Ghorpade, K.P. Karunakaran, M.K. Tiwari, Optimal part orientation in layered manufacturing using evolutionary stickers-based DNA algorithm, *Virtual Phys. Prototyp.* 2 (2007) 3–19, doi:10.1080/17452750701330968.
- [9] S.K. Singhal, P.K. Jain, P.M. Pandey, A.K. Nagpal, Optimum part deposition orientation for multiple objectives in SL and SLS prototyping, *Int. J. Prod. Res.* 47 (2009) 6375–6396, doi:10.1080/00207540802183661.
- [10] W. Rong-ji, L. Xin-hua, W. Qing-ding, W. Lingling, Optimizing process parameters for selective laser sintering based on neural network and genetic algorithm, *Int. J. Adv. Manuf. Technol.* 42 (2009) 1035–1042, doi:10.1007/s00170-008-1669-0.
- [11] V. Canellidis, J. Giannatsis, V. Dedoussis, Genetic-algorithm-based multi-objective optimization of the build orientation in stereolithography, *Int. J. Adv. Manuf. Technol.* 45 (2009) 714–730, doi:10.1007/s00170-009-2006-y.
- [12] A.K. Sood, R.K. Ohdar, S.S. Mahapatra, Improving dimensional accuracy of fused deposition modelling processed part using grey Taguchi method, *Mater. Des.* 30 (2009) 4243–4252, doi:10.1016/j.matdes.2009.04.030.
- [13] A.K. Sood, R.K. Ohdar, S.S. Mahapatra, Parametric appraisal of mechanical property of fused deposition modelling processed parts, *Mater. Des.* 31 (2010) 287–295, doi:10.1016/j.matdes.2009.06.016.
- [14] R. Paul, S. Anand, Optimal part orientation in Rapid Manufacturing process for achieving geometric tolerances, *J. Manuf. Syst.* 30 (2011) 214–222, doi:10.1016/j.jmsy.2011.07.010.

- [15] R. Paul, S. Anand, Process energy analysis and optimization in selective laser sintering, *J. Manuf. Syst.* 31 (2012) 429–437, doi:10.1016/j.jmsy.2012.07.004.
- [16] A.K. Sood, R.K. Ohdar, S.S. Mahapatra, Experimental investigation and empirical modelling of FDM process for compressive strength improvement, *J. Adv. Res.* 3 (2012) 81–90, doi:10.1016/j.jare.2011.05.001.
- [17] A.K. Sood, A. Equbal, V. Toppo, R.K. Ohdar, S.S. Mahapatra, An investigation on sliding wear of FDM built parts, *CIRP J. Manuf. Sci. Technol.* 5 (2012) 48–54, doi:10.1016/j.cirpj.2011.08.003.
- [18] A.M. Phatak, S.S. Pande, Optimum part orientation in rapid prototyping using genetic algorithm, *J. Manuf. Syst.* 31 (2012) 395–402, doi:10.1016/j.jmsy.2012.07.001.
- [19] S. Singh, V.S. Sharma, A. Sachdeva, Optimization and analysis of shrinkage in selective laser sintered polyamide parts, *Mater. Manuf. Process.* 27 (2012) 707–714, doi:10.1080/10426914.2011.593247.
- [20] Y. Li, J. Zhang, Multi-criteria GA-based Pareto optimization of building direction for rapid prototyping, *Int. J. Adv. Manuf. Technol.* 69 (2013) 1819–1831, doi:10.1007/s00170-013-5147-y.
- [21] A. Boschetto, V. Giordano, F. Veniali, Surface roughness prediction in fused deposition modelling by neural networks, *Int. J. Adv. Manuf. Technol.* 67 (2013) 2727–2742, doi:10.1007/s00170-012-4687-x.
- [22] A. Noriega, D. Blanco, B.J. Alvarez, A. Garcia, Dimensional accuracy improvement of FDM square cross-section parts using artificial neural networks and an optimization algorithm, *Int. J. Adv. Manuf. Technol.* 69 (2013) 2301–2313, doi:10.1007/s00170-013-5196-2.
- [23] A. Peng, X. Xiao, R. Yue, Process parameter optimization for fused deposition modeling using response surface methodology combined with fuzzy inference system, *Int. J. Adv. Manuf. Technol.* 73 (2014) 87–100, doi:10.1007/s00170-014-5796-5.
- [24] P.K. Gurralla, S.P. Regalla, Multi-objective optimisation of strength and volumetric shrinkage of FDM parts, *Virtual Phys. Prototyp.* 9 (2014) 127–138, doi:10.1080/17452759.2014.898851.
- [25] F. Rayegani, G.C. Onwubolu, Fused deposition modelling (FDM) process parameter prediction and optimization using group method for data handling (GMDH) and differential evolution (DE), *Int. J. Adv. Manuf. Technol.* 73 (2014) 509–519, doi:10.1007/s00170-014-5835-2.
- [26] V. Vijayaraghavan, A. Garg, J.S.L. Lam, B. Panda, S.S. Mahapatra, Process characterisation of 3D-printed FDM components using improved evolutionary computational approach, *Int. J. Adv. Manuf. Technol.* (2014) doi:10.1007/s00170-014-6679-5.
- [27] R. Paul, S. Anand, Optimization of layered manufacturing process for reducing form errors with minimal support structures, *J. Manuf. Syst.* (2014) doi:10.1016/j.jmsy.2014.06.014.
- [28] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, *Comput. Aided Des.* 43 (2011) 303–315, doi:10.1016/j.cad.2010.12.015.
- [29] R.V. Rao, *Teaching-Learning-Based Optimization (TLBO) Algorithm and Its Engineering Applications*, Springer-Verlag, London, 2015.
- [30] R.V. Rao, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, *Int. J. Ind. Eng. Comput.* 7 (2016) 19–34, doi:10.5267/j.ijiec.2015.8.004.
- [31] M. Ghasemi, S. Ghavidel, S. Rahmani, A. Roosta, H. Falah, A novel hybrid algorithm of imperialist competitive algorithm and teaching learning algorithm for optimal power flow problem with non-smooth cost functions, *Eng. Appl. Artif. Intell.* 29 (2014) 54–69, doi:10.1016/j.engappai.2013.11.003.
- [32] D. Chen, F. Zou, Z. Li, J. Wang, S. Li, An improved teaching-learning-based optimization algorithm for solving global optimization problem, *Inf. Sci. (Ny)* 297 (2015) 171–190, doi:10.1016/j.ins.2014.11.001.
- [33] M. Ghasemi, S. Ghavidel, M. Gitizadeh, E. Akbari, An improved teaching-learning-based optimization algorithm using Lévy mutation strategy for non-smooth optimal power flow, *Int. J. Electr. Power Energy Syst.* 65 (2015) 375–384, doi:10.1016/j.ijepes.2014.10.027.
- [34] J. Li, Q. Pan, K. Mao, A discrete teaching-learning-based optimisation algorithm for realistic flowshop rescheduling problems, *Eng. Appl. Artif. Intell.* 37 (2015) 279–292, doi:10.1016/j.engappai.2014.09.015.
- [35] O.A. Mohamed, S.H. Masood, J.L. Bhowmik, Optimization of fused deposition modeling process parameters: a review of current research and future prospects, *Adv. Manuf.* (2015) 42–53, doi:10.1007/s40436-014-0097-7.
- [36] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2002) 182–197, doi:10.1109/4235.996017.
- [37] M. Balasubbareddy, S. Sivanagaraju, C.V. Suresh, Multi-objective optimization in the presence of practical constraints using non-dominated sorting hybrid cuckoo search algorithm, *Eng. Sci. Technol. Int. J.* (2015) doi:10.1016/j.jjestch.2015.04.005.
- [38] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, *Evol. Comput.* 8 (2000) 173–195, doi:10.1162/106365600568202.
- [39] J.R. Schott, *Fault tolerant design using single and multicriteria genetic algorithm optimization* (Master's thesis), Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.