

Research Article

Parameter Determination of Milling Process Using a Novel Teaching-Learning-Based Optimization Algorithm

Zhibo Zhai, Shujuan Li, and Yong Liu

School of Mechanical and Instrument Engineering, Xi'an University of Technology, 5 South Jinhua Road, Xi'an, Shaanxi 710048, China

Correspondence should be addressed to Shujuan Li; shujuanli@xaut.edu.cn

Received 24 July 2015; Revised 29 September 2015; Accepted 7 October 2015

Academic Editor: Anna Vila

Copyright © 2015 Zhibo Zhai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cutting parameter optimization dramatically affects the production time, cost, profit rate, and the quality of the final products, in milling operations. Aiming to select the optimum machining parameters in multitool milling operations such as corner milling, face milling, pocket milling, and slot milling, this paper presents a novel version of TLBO, TLBO with dynamic assignment learning strategy (DATLBO), in which all the learners are divided into three categories based on their results in “Learner Phase”: good learners, moderate learners, and poor ones. Good learners are self-motivated and try to learn by themselves; each moderate learner uses a probabilistic approach to select one of good learners to learn; each poor learner also uses a probabilistic approach to select several moderate learners to learn. The CEC2005 contest benchmark problems are first used to illustrate the effectiveness of the proposed algorithm. Finally, the DATLBO algorithm is applied to a multitool milling process based on maximum profit rate criterion with five practical technological constraints. The unit time, unit cost, and profit rate from the Handbook (HB), Feasible Direction (FD) method, Genetic Algorithm (GA) method, five other TLBO variants, and DATLBO are compared, illustrating that the proposed approach is more effective than HB, FD, GA, and five other TLBO variants.

1. Introduction

In modern manufacturing, determining optimal cutting parameters is of great importance to improve the quality of products, to reduce the machining costs, and to maximize the profit rate. The main cutting parameters in multitool milling operations include the feed per tooth, cutting velocity, and the radial and axial depths of cut. The conventional methods of selecting of cutting parameters mainly depend either on the operator experience or on machining data from handbooks. But it is a known fact that the cutting parameters obtained from these resources, in most cases, are extremely conservative. Consequently, it may not perform high productivity. So it is necessary to develop a new technique to investigate the cutting optimization problem.

There are many mathematical programming techniques to be used extensively for optimization of cutting parameter over the past few decades. In earlier studies, Gupta et al. [1] developed an integer programming for the determination of optimal subdivision of depth of cut in multipass turning with constraints. Subsequently, Wang et al. [2] used deterministic

graphical programming to optimize machining parameters of cutting conditions for single pass turning operations. Shin and Joo [3] used a dynamic programming for the determination of optimum of machining conditions with practical constraints. Petropoulos [4] developed a geometric programming model for the selection optimal selection of machining rate variables.

Although these mathematical programming techniques have been applied to solve the cutting parameter optimization problem, these studies have not involved some important cutting constraints. Considering the number of constraints such as surface roughness, cutting force, cutting velocity, machining power, and tool life, cutting parameter optimization problem is very complicated. The additional variables due to number of passes make the solution procedure more complicated. These mathematical programming techniques incline to obtain local optima and may be only useful for a specific problem.

Recently, nontraditional optimization approaches recently have been developed to solve the cutting parameter optimization problem. Shunmugam et al. [5] used a Genetic

Algorithm (GA) to optimize cutting parameters in multipass milling operations, exploiting total production cost as the objective function. Li et al. [6] developed a two-phase GA to optimize the spindle speed and feed and select the tools for drilling blind holes in parallel drilling operations to obtain the minimum completion time. Krimpenis and Vosniakos [7] used a GA to optimize rough milling for parts with sculptured surfaces and select process parameters such as federate, cutting speed, width of cut, raster pattern angle, spindle speed, and number of machining slices of variable thickness. Although GA has some advantages over traditional techniques, the successful application of GA depends on the population size and the diversity of individual solutions in the search space. If its diversity cannot be maintained before the global optimum is reached, it may prematurely converge to a local optimum. Liu and Wang [8] proposed a modified GA to optimize milling parameter selection. The operating domain is defined and changed to be around the optimal point in its evolutionary processes so that the convergence speed and accuracy are improved. Wang et al. [9] presented a parallel genetic simulated annealing to select optimal machining parameters for multipass milling operations. The Taguchi method was initially used to predict cutting parameter performance measures, and then the GA was utilized to optimize the cutting conditions. Subsequently, Öktem [10] discussed the utilization of Artificial Neural Network (ANN) and GA for predicting the best combinations of cutting parameters to provide the best surface roughness. Li et al. [11] suggested combining the ANN and GA to minimize the make-span in production scheduling problems. António et al. [12] used a GA based on an elitist strategy to minimize manufacturing costs of multipass cutting parameters in face milling operations. Zhou et al. [13] applied fuzzy particle swarm optimization algorithm (PSO) to select the machining parameters for milling operations. Zarei et al. [14] proposed a Harmony Search (HS) algorithm to define the optimum cutting parameters for a multipass face milling operation. Mahdavinjad et al. [15] developed a new hybrid optimization approach by combining the immune algorithm with ANN to predict the effect of milling parameters on the final surface roughness of Ti-6Al-4V work pieces. Briceno et al. [16] selected an ANN for modeling and simulating the milling process. Orthogonal design and specifically equally spaced dimensioning showed that ANN is a good method to define process parameters. Venkata Rao and Pawar [17] used an Artificial Bee Colony (ABC) algorithm to minimize production time of a multipass milling process to determine the optimal process parameters such as the number of passes, depth of cut for each pass, cutting velocity, and feed. Onwubolu [18] used a new optimization technique based on tribes to select the optimum machining parameters in multipass milling operations such as plain milling and face milling by simultaneously considering multipass rough machining and finish machining.

Although some improvements in optimizing machining parameters in milling operations have been made, these non-traditional optimization approaches require a lot of specific controlling parameters except for the common parameters such as number of generation and population size. For

instance, the GA involves crossover and mutation probability. Similarly, the HS algorithm includes harmony memory considering rate, bandwidth rate, and a random select rate. These specific controlling parameters affect significantly the performance of the above mentioned algorithms. Improper parameters of algorithms either raise total complexity of consumption time or fall into the local optimum. So there remains a need for efficient and effective optimization algorithms for the cutting parameters determination.

Very recently, Rao et al. proposed a Teaching-Learning-Based Optimization (TLBO) [19] algorithm. This algorithm does not need specific controlling parameters except for the common parameters such as number of generation and population size. As a stochastic search strategy, it is a new algorithm based on swarm intelligence having the characteristics of rapid convergence, simple computation, and no specific controlling parameters except for the common parameters such as number of generation and population size. However, it has some undesirable dynamical properties that degrade its searching ability [20]. One of the most important issues is that the population tends to be trapped in the local optima solution because of diversity loss. To improve the performance of the original TLBO, a few modified or improved algorithms are proposed in recent years, such as teaching-learning-based optimization with dynamic group strategy (DGSTLBO) [20], teaching-learning-based optimization with neighborhood search (NSTLBO) [21], an elitist teaching-learning-based optimization algorithm (ETLBO) [22], and a variant of teaching-learning-based optimization algorithm with differential learning (DLTLBO) [23]. These modified TLBOs have better performance than the original TLBO on classical benchmark functions. Although the abovementioned variants TLBO have some improvements, they never focus on correct assignment problem. That is to say, each learner should be assured correct assignment of learning objects in the "Learner Phase." To this aim, we present a novel version of TLBO, TLBO with dynamic assignment learning strategy (DATLBO), in which all the learners are divided into three categories in the "Learner Phase": good learners, moderate learners, and poor ones. Good learners are self-motivated and try to learn by themselves; each moderate learner uses a probabilistic approach to select one of good learners to learn; each poor learner also uses a probabilistic approach to select several moderate learners to learn. The modification tries to both enable the diversity of the population to be preserved in order to discourage premature convergence and achieve balance between the explorative and exploitative tendencies of achieving better solution. A case study in multitool milling operations is used to verify DATLBO. The results are compared with results from GA [24], the feasible direction method [25], handbook recommendations [26], and five other TLBO variants.

The paper is organized as follows. Section 2 gives a short introduction to modeling of milling operations. Original TLBO algorithm and the proposed algorithm, DATLBO, are described in Section 3. This case study of multitool milling parameter optimization is presented in Section 4, and the summary and conclusions are given in Section 5. The last section provides the nomenclature.

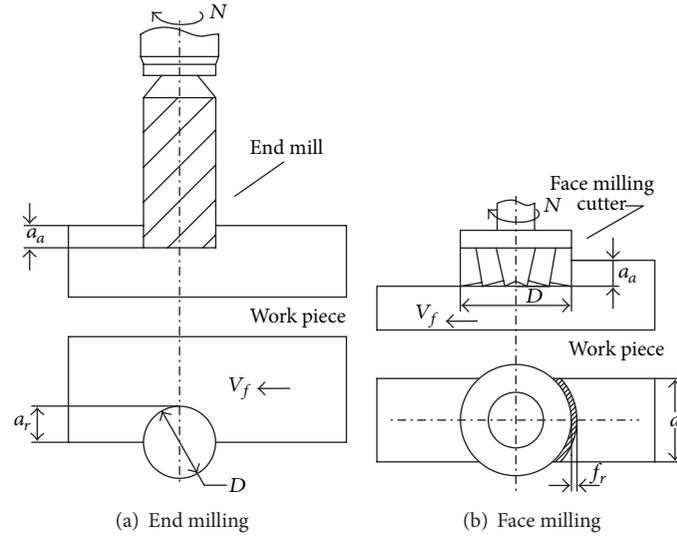


FIGURE 1: Milling operations.

2. Modeling of Milling Operations

Milling is a machining process which uses rotary multiple tooth cutters to remove material from a work piece. Figure 1 displays two kinds of milling operations: end milling and face milling. As the cutter rotates, each tooth removes a small amount of material from the advancing work piece during each spindle revolution.

In this study, face milling and end milling operations are considered. For multitool milling operations, it is very important to select the optimum process parameters such as depth of cut, feed per tooth, and cutting velocity. Depth of cut is usually predetermined by the work piece geometry and operation sequence. Hence, determining machining process parameters can be simplified to determining the proper cutting velocity and feed per tooth.

2.1. Modeling of Unit Time, Unit Cost, and Profit Rate. The mathematical model of multitool milling operations formulated in this study is based on the research of Tolouei-Rad and Bidhendi [25]. The decision variables considered for this model are cutting velocity (v) and feed per tooth f_z . The objective function is to maximize the profit rate. The unit production time for a single part in multitool milling

operations is the sum of setup time, machining time, and tool changing time. The unit production time is

$$T_u = t_s + \sum_{i=1}^m t_{mi} + \sum_{i=1}^m t_{cti}, \quad (1)$$

where the machining time is

$$t_{mi} = \frac{\pi d_i K}{1000 v_i f_{zi} z_i}, \quad i = 1, 2, \dots, m. \quad (2)$$

The unit cost for producing of the part in multitool milling operations is the sum of material cost, setup cost, machining cost, tool cost, and tool changing cost. The unit cost is

$$C_u = c_{\text{mat}} + (c_l + c_o) t_s + \sum_{i=1}^m (c_l + c_o) t_{mi} + \sum_{i=1}^m [(c_l + c_o) t_{ci} + c_{ti}] \frac{t_{mi}}{T_i}, \quad (3)$$

where the tool life is

$$T_i = \frac{60}{Q_i} \left\{ \frac{C_i [(a_{ei}/f_{zi})/5]^{g_i}}{(a_{ei} f_{zi})^{w_i} \pi d_i N_i / 1000} \right\}^{1/n_i}. \quad (4)$$

For multitool milling operations the profit rate is

$$P_r = \frac{S_p - \{c_{\text{mat}} + (c_l + c_o) t_s + \sum_{i=1}^m (c_l + c_o) t_{mi} + \sum_{i=1}^m [(c_l + c_o) t_{ci} + c_{ti}] (t_{mi}/T_i)\}}{t_s + \sum_{i=1}^m t_{mi} + \sum_{i=1}^m t_{cti}}. \quad (5)$$

2.2. Milling Process Constraints. Feed per tooth is in the range determined by the minimum and maximum feed per tooth of the machine tool:

$$f_{z\min} \leq f_z \leq f_{z\max}. \quad (6)$$

Optimum cutting velocity is in the range determined by the minimum and maximum cutting velocity of the machine tool:

$$v_{\min} \leq v \leq v_{\max}. \quad (7)$$

The total cutting force constraint is

$$\frac{C_F a_p^{x_F} f_z^{y_F} a_e^{u_F} z}{d^{q_F} N^{w_F}} K_{FC} - F_{c(\text{per})} \leq 0. \quad (8)$$

Machining power cannot exceed the effective maximum machining power. Therefore, the power constraint is

$$\frac{F_c v}{1000} - eP_m \leq 0. \quad (9)$$

The required surface roughness cannot exceed the maximum allowable surface roughness. Therefore the surface roughness constraint for end milling operations is [25]

$$318 \frac{f_z^2}{4d} - R_{a(\text{at})} \leq 0 \quad (10)$$

and for face milling constraint is [25]

$$318 \frac{f_z}{\tan(l_a) + \cot(c_a)} - R_{a(\text{at})} \leq 0. \quad (11)$$

3. Teaching-Learning-Based Optimization

3.1. TLBO Algorithm. Inspired by the philosophy of teaching and learning, Rao et al. presented a teaching-learning-based optimization (TLBO) [19]. It was developed based on the simulation of a classical learning process in a class. Like other population set-based methods such as GA, DE, and PSO, TLBO also uses a population of candidate solutions, called learners, with their positions initialized randomly from the search space. The teacher is generally referred to as a highly learned person who shares his or her knowledge with the learners in a class. The quality of a teacher affects the outcome (i.e., grads or marks) of learners. Furthermore, learners also learn from interaction between themselves.

In original TLBO algorithm, different design variables will be analogous to different subjects offered to learners and the learners' result is analogous to the "fitness," as in other population based optimization techniques. The learning process of TLBO is divided into two phases which consists of "Teacher Phase" and "Learner Phase."

Teacher Phase. During teaching phase, the best learner or teacher tries to bring the mean result of the class in a subject taught by him or her who depends on his or her capability. But in practice, a teacher can only move the mean of a class

up to close to his or her result to some extent. Suppose that T denotes the teacher and M denotes the mean at any iteration. If T moves M toward its own level mean M , the new mean will be T designated as M_{new} . The difference between the existing mean and the new mean is given as follows:

$$\text{Difference_Mean} = r (M_{\text{new}} - T_F M), \quad (12)$$

where r is a random vector in which each element is a random number in the range $[0, 1]$. T_F denotes a teaching factor which decides the value of mean to be changed, and the value of T_F can be either 1 or 2. Based on (12), the updating formula of the learning for a learner X_i in teacher phase is given by

$$X_{\text{new},i} = X_i + \text{Difference_Mean}. \quad (13)$$

Learner Phase. During learning phase, each learner interacts with other learners to improve his knowledge. The random interaction of learners is going with the help of formal communications, presentations, group discussions, and so forth. If the other learner have more knowledge than him or her, the learner learns something new. The learner mode is based on the following expression:

$$X_{\text{new},i} = \begin{cases} X_i + r (X_i - X_j), \\ X_i + r (X_j - X_i). \end{cases} \quad (14)$$

3.2. DATLBO Algorithm. In the original TLBO, each learner interacts randomly with other learners in "Learner Phase," which has certain blindness and does not assure correct assignment of learning objects to each learner. During the course of optimization, this situation results in slower convergence rate of optimization problem. Motivated by the fact, we propose a novel version of TLBO, TLBO with dynamic assignment learning strategy (DATLBO).

3.2.1. Dynamic Assignment Learning Strategy. Study reveals that birds employ different strategies to conduct mating process among their society. The ultimate success of a bird to raise a brood with superior features depends on an appropriate assignment strategy it uses [27]. It is well known that, in the original TLBO, each learner interacts randomly with other learners in "Learner Phase," which has a certain degree of blindness and does not assure correct assignment of learning objects to each learner. To be more specific, appropriate assignment strategy can play an important role in improving the results of the whole class. Inspired by the above bird mating process, all the learners are divided into three categories based on their results: good learners, moderate learners, and poor ones. The good learners are those learners that have the most result; the moderate learners are those learners that have the better result, and the poor ones are those learners that have bad result. Totally, each category has its own learning pattern. By means of this assignment, entire class is split into different categories of learners as per their level and each learner is assigned an appropriate learning object. The way by which each category produces a candidate solution will be explained below in detail.

```

(1) For  $c = 1$  to  $D$  do
(2)   If  $r_1 > sf$ 
(3)      $X_{new,i}^{(c)} = X_i^{(c)} + \eta \times (r_2 - r_3) + X_i^{(c)}$ ;
(4)   Else
(5)      $X_{new,i}^{(c)} = X_i^{(c)}$ ;
(6)   End If
(7) End For

```

PSEUDOCODE 1

```

(1)  $X_{new,i} = X_i + w \times r \times (X_j - X_i)$ ;
(2) If  $r_1 > sf$ 
(3)    $X_{new,i}^{(c)} = r_2 \times (l^{(c)} - u^{(c)})$ ;
(4) End If

```

PSEUDOCODE 2

3.2.2. *Each Category Learning Pattern in the Dynamic Assignment Learning Strategy.* Each good learner is able to self-learn without the help of others in “Learner Phase”; that is to say, good learners are self-motivated and try to learn by themselves. Therefore, each good learner tries to increase his or her own knowledge abilities of certain subject by making a small change in her subjects probabilistically. From the optimization view, exploitation of the best solutions found so far is performed by good learners. The self-learning pattern pseudocode of each good learner is implemented in Pseudocode 1, where $X_{new,i}$ is a newly generated learner according to X_i , D is the problem dimension, r_1 , r_2 , r_3 are uniformly distributed random numbers in the range $[0, 1]$, sf is the self-motivated factor of each good learner, and η denotes the step size.

Each moderate learner selects one of the whole good learners with a probabilistic approach and learns from his own interesting good learner. The assignment good learner has a better chance of being selected with more knowledge. The learning pattern pseudocode of each moderate learner is implemented in Pseudocode 2, where $X_{new,i}$ is a newly generated learner according to X_i , w is a time-varying weight to adjust the importance of the interesting good learner, r is a vector whose each element is distributed randomly in the range $[0, 1]$, X_j is the selected object from the whole good learners, sf is the self-motivated factor of each moderate learner, and u and l are the upper and lower bounds of the elements, respectively.

Each poor learner tends to learn from two or more moderate learners that are selected with a probabilistic approach. The learning pattern pseudocode of each poor learner is implemented in Pseudocode 3, where $X_{new,i}$ is a newly generated learner according to X_i , w is a time-varying weight to adjust the importance of the interesting moderate learner, n_i is the number of being selected moderate learners, r_i is a vector whose each element, distributed randomly in $[0, 1]$, X_j is the j th selected object from the whole moderate learners, r_2 is a random number in the range $[0, 1]$, sf is the

```

(1)  $X_{new,i} = X_i + w \times \sum_{m=1}^{n_i} r_i \times (X_j^m - X_i)$ ;
(2) If  $r_1 > sf$ 
(3)    $X_{new,i}^{(c)} = r_2 \times (l^{(c)} - u^{(c)})$ 
(4) End If

```

PSEUDOCODE 3

self-motivated factor of each poor learner, and u and l are the upper and lower bounds of the elements, respectively.

As explained above, the pseudocode of the dynamic assignment learning strategy is given in Pseudocode 4.

3.2.3. *Each Category Parameters Adjustment in the Dynamic Assignment Learning Strategy.* To apply the dynamic assignment learning strategy to DATLBO algorithm, the appropriate parameters have to be tuned. It seems that the most important parameter is the proportion of each learner from the class. It is suggested that the percentages of good learners, moderate learners, and bad learners are, respectively, set to 20, 50, and 30 of the class. Only one assigned good learner and two or three assigned moderate learners will be enough. Self-motivated factor is between 0 and 1. sf can be set between 0.9 and 1. Small values of this parameter may result in bad impact on the performance of the assignment learning strategy. It is better to select sf as an increasing linear function which changes from a small value nearby zero (e.g., 0.1) to a large one nearby 1 (e.g., 0.9). This behavior allows learners to change their subjects abilities at the beginning of the assignment learning strategy with high probability. This probability decreases during the generations and helps the learners to converge to the global solution. The step size η can be selected from the order of 10^{-2} or 10^{-3} . To provide a good balance between local and global search, w decreases linearly from a value nearby 2 to a small one nearby 0.

3.3. *The Pseudocode of DATLBO Algorithm.* By incorporating the dynamic assignment learning strategy in “Learner Phase” into the original TLBO framework, the DATLBO algorithm is developed. The pseudocode of DATLBO algorithm is presented in Pseudocode 5.

3.4. Experiments and Comparisons

3.4.1. *Benchmark Functions Used in Experiments.* To analyze and compare the performance and accuracy of DATLBO algorithm, a large set of CEC2005 tested benchmark functions are used to do the experiments. Based on the shape characteristics, the set of benchmark functions are grouped into unimodal functions (F_1 to F_5) and basic multimodal functions (F_6 to F_{12}). The brief descriptions of these benchmark functions are listed in Table 1. For more details about the definition of benchmark functions, refer to [28].

3.4.2. *Experimental Platform, Termination Criterion, and Parameter.* All experiments run in the same machine with

```

Begin
Set:  $sf = sf_{\min} - (sf_{\min} - sf_{\max}) * (t/t_{\max})$ ; [pp, mm] = sort(fitness); fitness = pp;  $X = X(\text{mm},:)$ ;
(1) For  $i = 1 : n_1$ 
(2)   For  $c = 1$  to  $D$  do
(3)     If  $r_1 > sf$ 
(4)        $X_{\text{new},i}^{(c)} = X_i^{(c)} + \eta \times (r_2 - r_3) + X_i^{(c)}$ ;
(5)     Else
(6)        $X_{\text{new},i}^{(c)} = X_i^{(c)}$ ;
(7)     End If
(8)   End For
(9) End For
(10)  $\text{index}_1 = \text{roulette wheel 1 (fitness, } n_1, n_2)$ ;
(11) For  $i = 1 : n_2$ 
(12)   For  $c = 1$  to  $D$  do
(13)      $X_{\text{new},n_1+i}^{(c)} = X_{n_1+i}^{(c)} + w \times r \times (X_{\text{index}_1}^{(c)} - X_{n_1+i}^{(c)})$ ;
(14)   End For
(15)    $m = \text{randi}(D)$ ;
(16)   If  $r_1 > sf$ 
(17)      $X_{\text{new},n_1+i}^{(m)} = l^{(m)} - r_2 \times (l^{(m)} - u^{(m)})$ ;
(18)   End If
(19) End For
(20)  $\text{index}_2 = \text{roulette wheel 2 (fitness, } n_1, n_3, \text{mm}_1)$ ;
(21) For  $i = 1 : n_3$ 
(22)   For  $c = 1$  to  $D$  do
(23)     For  $k = 1 : \text{mm}_1$ 
(24)       For  $c = 1$  to  $D$  do
(25)          $X_{3k}^{(c)} = \text{rand}(1, D) * (X_{\text{index}_1(i,k)}^{(c)} - X_{n_1+n_2+i}^{(c)})$ ;
(26)       End For
(27)     End For
(28)      $X_{\text{new},n_1+n_2+i}^{(c)} = X_{n_1+n_2+i}^{(c)} + w * \text{sum}(X_{3k})$ ;
(29)      $m = \text{randi}(D)$ ;
(30)     If  $r_1 > sf$ 
(31)        $X_{\text{new},n_1+n_2+i}^{(m)} = l^{(m)} - r_2 \times (l^{(m)} - u^{(m)})$ ;
(32)     End If
(33)   End For
(34) End For
End

```

PSEUDOCODE 4: The pseudocode of dynamic assignment learning strategy.

a Celoron2.26 GHz CPU, 2 GB memory, and windows XP operating system with Matlab7.9. For the purpose of decreasing statistical errors, all experiments independently run 25 times for twelve test functions of 30 variables and 300,000 function evaluations (FES) as the stopping criterion.

The parameter setting of DATLBO algorithm is as follows: $N = 50$, $K = 3$, and number of good learners, moderate learners, and poor learners is set to 10, 25, and 15, respectively; roulette wheel is used as the selection approach; number of being assigned good learners and moderate learners are set to 1 and 2, respectively; self-motivated factor $sf = 0.9$; step size $\eta = 10^{-3}$. The parameters of other algorithms agree well with the original papers.

3.4.3. Performance Metric. The mean value F_{mean} and standard deviation (SD) of the function error value $F(x) - F(x^*)$ are recorded to evaluate the performance of each algorithm, where $F(x)$ and $F(x^*)$ denote the best fitness value and the

real global optimization value of test problem, respectively. To verify whether the overall optimization performance of various algorithms is significantly different, statistical analysis method is used to compare the results obtained by algorithms for kinds of problems. Therefore, to statistically compare DATLBO algorithm with other five algorithms, the statistical tool Wilcoxon's rank sum test [29] at a 0.05 significance level is used to evaluate whether the median fitness values (F_{mean}) of two solutions from any two algorithms.

3.4.4. Numerical Experiments and Results. In this section, DATLBO algorithm is compared with five other TLBO variants. Each corresponding table presents the experimental results, and the last three rows of each table summarize the comparison results. The best results are shown in bold.

From the statistical results of Table 2, we can see that none of the algorithms can perfectly solve the twelve CEC2005 standard benchmark functions. From the Wilcoxon's rank

```

Input:  $N, D, K, e, m, t, \text{FESMAX}; n_1 = 0.2 * N; n_2 = 0.5 * N; n_3 = 0.3 * N;$ 
 $w_{\max} = 2.5; w_{\min} = 0.25 = 0.001; sf_{\max} = 0.9; sf_{\min} = 0.1; nm_1 = 2; K = 3;$ 
(1)  $t = 0;$ 
(2) Generate an initial population:  $X = \{x_1, x_2, \dots, x_N\};$ 
(3)  $\text{FES} = N; t_{\max} = \text{floor}((\text{FESMAX} - N)/N);$ 
(4) While  $\text{FES} \leq \text{FESMAX}$ 
(5) Evaluate the objective function values:  $f(X);$ 
(6)  $[kk, ll] = \text{min}(\text{fitness});$ 
(7) Find the best learner:  $gbest(t);$ 
(8)  $mbest = \text{mean}(X);$ 
(9) For  $i = 1 : N$ 
(10)  $T_F = \text{round}(1 + \text{rand});$ 
(11) For  $j = 1 : D$ 
(12)  $X_{\text{new},i}^{(j)} = X_i^{(j)} + \text{rand} \times (gbest^{(j)} - T_F \times mbest^{(j)});$ 
(13) If  $X_{\text{new},i}^{(j)} > lu_2^{(j)}$ 
(14)  $X_{\text{new},i}^{(j)} = \max(lu_1^{(j)}, 2 \times lu_2^j - X_{\text{new},i}^{(j)})$ 
(15) End If
(16) If  $X_{\text{new},i}^{(j)} < lu_1^{(j)}$ 
(17)  $X_{\text{new},i}^{(j)} = \max(lu_2^{(j)}, 2 \times lu_1^j - X_{\text{new},i}^{(j)});$ 
(18) End If
(19) End For
(20) End For
(21)  $sf = sf_{\min} - (sf_{\min} - sf_{\max}) * (t/t_{\max}); [pp, mm] = \text{sort}(\text{fitness}); \text{fitness} = pp; X = X(\text{mm},:)$ 
(22) For  $i = 1 : n_1$ 
(23) For  $c = 1$  to  $D$  do
(24) If  $r_1 > sf$ 
(25)  $X_{\text{new},i}^{(c)} = X_i^{(c)} + \eta \times (r_2 - r_3) + X_i^{(c)};$ 
(26) Else
(27)  $X_{\text{new},i}^{(c)} = X_i^{(c)};$ 
(28) End If
(29) End For
(30) End For
(31)  $\text{index}_1 = \text{roulette wheel 1}(\text{fitness}, n_1, n_2);$ 
(32) For  $i = 1 : n_2$ 
(33) For  $c = 1$  to  $D$  do
(34)  $X_{\text{new},n_1+i}^{(c)} = X_{n_1+i}^{(c)} + w \times r \times (X_{\text{index}_1}^{(c)} - X_{n_1+i}^{(c)});$ 
(35) End For
(36)  $m = \text{randi}(D);$ 
(37) If  $r_1 > sf$ 
(38)  $X_{\text{new},n_1+i}^{(m)} = l^{(m)} - r_2 \times (l^{(m)} - u^{(m)});$ 
(39) End If
(40) End For
(41)  $\text{index}_2 = \text{roulette wheel 2}(\text{fitness}, n_1, n_3, nm_1);$ 
(42) For  $i = 1 : n_3$ 
(43) For  $c = 1$  to  $D$  do
(44) For  $k = 1 : nm_1$ 
(45) For  $c = 1$  to  $D$  do
(46)  $X_{3k}^{(c)} = \text{rand}(1, D) * (X_{\text{index}_2(i,k)}^{(c)} - X_{n_1+n_2+i}^{(c)});$ 
(47) End For
(48) End For
(49)  $X_{\text{new},n_1+n_2+i}^{(c)} = X_{n_1+n_2+i}^{(c)} + w * \text{sum}(X_{3k});$ 
(50)  $m = \text{randi}(D);$ 
(51) If  $r_1 > sf$ 
(52)  $X_{\text{new},n_1+n_2+i}^{(m)} = l^{(m)} - r_2 \times (l^{(m)} - u^{(m)});$ 
(53) End If
(54) End For
(55) End For
End

```

PSEUDOCODE 5: The pseudocode of DATLBO algorithm.

TABLE 1: Benchmark functions definition.

Name	Definition	Range
F_1	Shifted sphere function	[50, 100]
F_2	Shifted Schwefel's problem 1.2	[50, 100]
F_3	Shifted rotated high conditioned elliptic function	[50, 100]
F_4	Shifted Schwefel's problem 1.2 with noise in fitness	[50, 100]
F_5	Schwefel's problem 2.6 with global optimum on bounds	[50, 100]
F_6	Shifted Rosenbrock's function	[50, 100]
F_7	Shifted rotated Griewank's function without bounds	[300, 600]
F_8	Shifted rotated Ackley's function with global optimum on bounds	[16, 32]
F_9	Shifted Rastrigin's function	[2.5, 5]
F_{10}	Shifted rotated Rastrigin's function	[2.5, 5]
F_{11}	Shifted rotated Weierstrass function	[0.25, 0.5]
F_{12}	Schwefel's problem 2.13	[50, 100]

sum test listed in the last three rows of Table 2, it is clear that DATLBO algorithm outperforms original TLBO algorithm on test functions except for functions F_5 , F_6 , and F_7 . Although DTLBO algorithm outperforms DATLBO algorithm on test function F_4 , DATLBO algorithm is significantly better than DTLBO algorithm on other test functions such as F_1 , F_3 , F_6 , F_8 , F_9 , F_{10} , F_{11} , and F_{12} . By the ensemble of assignment learning strategy, DATLBO algorithm achieves promising results on unimodal and multimodal functions. DATLBO algorithm outperforms TLBO, DGSTLBO, ETLBO, NSTLBO, and DTLBO algorithms over nine, nine, seven, nine, and nine out of twelve test functions, respectively. The convergence graphs comparison on twelve test functions for $D = 30$ derived from six relevant TLBO algorithms are shown in Figure 2. From the convergence graphs, we can see that our DATLBO algorithm has better convergence speed and solution quality in most cases than other five relevant TLBO algorithms. Therefore, it is interesting to note that the overall performance of DATLBO algorithm is significantly better than original TLBO, ETLBO, NSTLBO, DGSTLBO, and DTLBO algorithm.

4. Case Study for Milling Operation

4.1. Problem Description. In order to compare the performance of the DATLBO algorithm with the other algorithms, a case study from [25] is used to test. A work piece has four machined features, namely, a step, pocket, and two slots. The milling operation schematic is shown in Figure 3. The three-dimensional view is in Figure 3(a), the front view is in Figure 3(b), and the top view is in Figure 3(c). The objective is to find optimum machining parameters with the maximum profit rate. Table 4 shows the limits of the cutting velocity and feed per tooth for this case study. Five milling operations, face milling, corner milling, pocket milling, slot milling 1, and slot milling 2, respectively, are in Table 3. The data for the tools for each operation are in Table 5.

The machine tool is a vertical CNC milling machine; $P_m = 8.5$ kW and $e = 95\%$. The work piece material is 10L50l leaded steel with hardness = 225 BHN. Other data include $S_p = \$25$, $C_{\text{mat}} = \$0.55$, $C_o = 1.45$ \$/min, $C_l = 0.45$ \$/min, $t_s = 2$ min,

$t_{ct} = 0.5$ min, $C = 33.98$ for HSS tools, $w = 0.28$, $C = 100.05$ for carbide tools, $K_p = 2.24$, $n = 0.15$ for HSS tools, $n = 0.53$ for carbide tools, and $g = 0.14$, $C_F = 7900$, $x_F = 1.0$, $y_F = 0.75$, $u_F = 1.1$, $q_F = 1.3$, $w_F = 0.2$, and $K_{FC} = 0.25$; all the data are from [25].

4.2. Applications of DATLBO Algorithm to Case Study. In this case study, Deb's heuristic constrained handling method is used to handle the constraints with the DATLBO and five other TLBO variants. A tournament selection operator is used in Deb's heuristic method [30] to select and compare the two solutions. The following three heuristic rules are implemented:

Rule 1: if one solution is feasible and the other infeasible, the feasible solution is preferred.

Rule 2: if both the solutions are feasible, the solution having the better objective function value is preferred.

Rule 3: if both solutions are infeasible, the solution having the least constraint violation is preferred.

These rules are implemented at the end of the teacher and learner phases. Deb's constraint handling methods are used to select the new solution. For this case study, the DATLBO and five other TLBO variants are run for 25 times and the termination condition is a maximum number of 300,000 function evaluation. The population size and the dimension are set to 30 and 10, respectively.

Tables 6–11 show the optimal feed per tooth and cutting velocity obtained by the DATLBO and five other TLBO variants, respectively, for five operations when the maximum profit rate is reached. Figures 4–6 and Table 12 compare the unit cost, unit time, and profit rate obtained by the HB, FD, GA, DATLBO, and five other TLBO variants. Compared to the HB solution, the unit cost decreases by 38%, 39%, 40%, 41%, 43%, 41%, 45%, and 46%, the unit time decreases by 41%, 44%, 46%, 48%, 48%, 47%, 47%, and 51%, and the profit rate increases by 2.51, 2.73, 2.94, 3.00, 3.18, 2.96, 3.25, and 3.66 times for the FD, GA, TLBO, DGSTLBO, ETLBO, NSTLBO, DTLBO, and DATLBO, respectively. The maximum profit rate given by the DATLBO algorithm is 3.31 \$/min, which is

TABLE 2: Results of six algorithms over 25 independent times on 12 test functions of 30 variables with 300,000 FES.

Function	Result	TLBO	DGSTLBO	ETLBO	NSTLBO	DTLBO	DATLBO
F_1	F_{mean}	$2.63E - 27^-$	$1.69E - 09^-$	$1.49E - 27^-$	$9.31E - 20^-$	$6.32E - 27^-$	$1.86E - 28$
	SD	$1.22E - 27$	$2.34E - 09$	$2.11E - 27$	$1.24E - 19$	$2.55E - 26$	$4.02E - 29$
F_2	F_{mean}	$7.81E - 09^-$	$1.33E - 08^-$	$6.00E - 10^-$	$6.01E - 09^-$	$1.45E - 14^-$	$1.34E - 16$
	SD	$1.06E - 08$	$1.75E - 09$	$8.48E - 10$	$2.05E - 00$	$2.94E - 15$	$6.95E - 17$
F_3	F_{mean}	$1.30E - 06^-$	$2.12E + 06^-$	$2.73E + 05^+$	$1.81E + 06^-$	$1.09E + 06^-$	$3.66E + 03$
	SD	$7.28E - 05$	$8.88E + 05$	$4.57E + 05$	$4.57E + 05$	$3.67E + 04$	$2.68E + 03$
F_4	F_{mean}	$3.58E + 02^-$	$2.74E + 02^-$	$2.01E + 03^-$	$7.83E + 03^-$	$3.65E - 01^+$	$1.65E + 02$
	SD	$9.37E + 00$	$1.69E + 02$	$2.84E + 03$	$3.48E + 03$	$1.38E - 02$	$1.44E + 02$
F_5	F_{mean}	$2.46E + 03^+$	$4.26E + 03^-$	$8.42E + 02^-$	$4.24E + 03^+$	$3.27E + 04^-$	$5.81E + 03$
	SD	$1.05E + 03$	$8.32E + 02$	$1.88E + 03$	$1.22E + 03$	$9.02E + 02$	$1.35E + 03$
	-	4	4	4	4	4	
	+	1	1	1	1	1	
	\approx	0	0	0	0	0	
F_6	F_{mean}	$1.84E + 01^+$	$6.74E + 02^-$	$1.39E + 01^+$	$9.67E + 01^-$	$4.52E + 01^+$	$5.52E + 01$
	SD	$2.81E + 01$	$7.14E + 02$	$3.10E + 01$	$6.92E + 01$	$2.16E + 01$	$3.62E + 01$
F_7	F_{mean}	$4.70E + 03^{\approx}$	$4.70E + 03^{\approx}$	$4.70E + 03^{\approx}$	$4.70E + 03^{\approx}$	$4.70E + 03^{\approx}$	$4.75E + 03$
	SD	$1.45E - 12$	$6.78E - 13$	$1.49E + 03$	$2.23E - 12$	$3.35E - 14$	$9.59E - 13$
F_8	F_{mean}	$2.09E + 01^-$	$2.09E + 01^-$	$2.08E - 01^+$	$2.09E + 01^-$	$2.08E + 01^-$	$2.00E + 01$
	SD	$3.52E - 02$	$4.71E - 02$	$6.35E - 02$	$4.17E - 02$	$5.19E - 02$	$5.60E - 02$
F_9	F_{mean}	$8.86E + 01^-$	$6.11E + 01^+$	$2.07E + 01^-$	$1.06E + 02^{\approx}$	$2.28E + 01^-$	$9.95E - 00$
	SD	$1.38E + 01$	$1.17E + 01$	$4.63E + 01$	$2.48E + 01$	$2.36E + 01$	$4.45E - 00$
F_{10}	F_{mean}	$1.39E + 02$	$1.09E + 02^-$	$3.02E + 01$	$1.68E + 02$	$1.12E + 02^-$	$1.07E + 01$
	SD	$4.17E + 01$	$7.09E + 01$	$6.76E + 01$	$3.88E + 01$	$3.35E + 01$	$2.09E + 01$
F_{11}	F_{mean}	$3.25E + 01^-$	$1.87E + 01^{\approx}$	$6.10E - 00^-$	$3.32E + 01^-$	$6.15E - 00^-$	$6.04E - 00$
	SD	$6.06E - 00$	$1.90E - 00$	$1.36E - 01$	$4.29E - 00$	$1.52E - 01$	$4.45E - 00$
F_{12}	F_{mean}	$7.53E + 03^-$	$2.70E + 04^-$	$1.15E + 03^{\approx}$	$9.84E + 03^-$	$4.05E + 06^-$	$2.51E + 03$
	SD	$7.36E + 03$	$1.20E + 04$	$2.58E + 03$	$7.83E + 03$	$2.22E + 03$	$3.00E + 03$
	-	5	5	3	5	5	
	+	1	1	2	0	1	
	\approx	1	1	2	2	1	
	-	9	9	7	9	9	
	+	2	2	3	1	2	
	\approx	1	1	2	2	1	

“-”, “+”, and “ \approx ” denote that the performance of the corresponding algorithm is significantly worse than, significantly better than, and similar to that of DATLBO, respectively.

TABLE 3: Milling operations.

Operation	Operation type	Tool number	a_e (mm)	K (mm)	R_a (mm)	F_c (per)
1	Face milling	1	10	450	2	156,449.4
2	Corner milling	2	5	90	6	17,117.74
3	Pocket milling	2	10	450	5	17,117.74
4	Slot milling 1	3	10	32	—	14,264.78
5	Slot milling 2	3	5	84	1	14,264.78

better than all the other optimization methods used for the same model.

5. Summary and Conclusions

A novel version TLBO algorithm, the DATLBO algorithm, is proposed for cutting parameter selection of a multitool milling operation. In the proposed DATLBO algorithm, all the learners are divided into three categories based on their results in “Learner Phase”: good learners, moderate learners, and poor ones. Good learners are self-motivated

and try to learn by themselves; each moderate learner uses a probabilistic approach to select one of good learners to learn; each poor learner also uses a probabilistic approach to select several moderate learners to learn. The DATLBO is applied to a case study for a multitool milling selection problem based on maximum profit rate criterion with five practical technological constraints. Significant improvements are obtained with the DATLBO algorithm in comparison to the results by HB, FD, GA, TLBO, DGSTLBO, ETLBO, NSTLBO, and DTLBO algorithms. These results show that

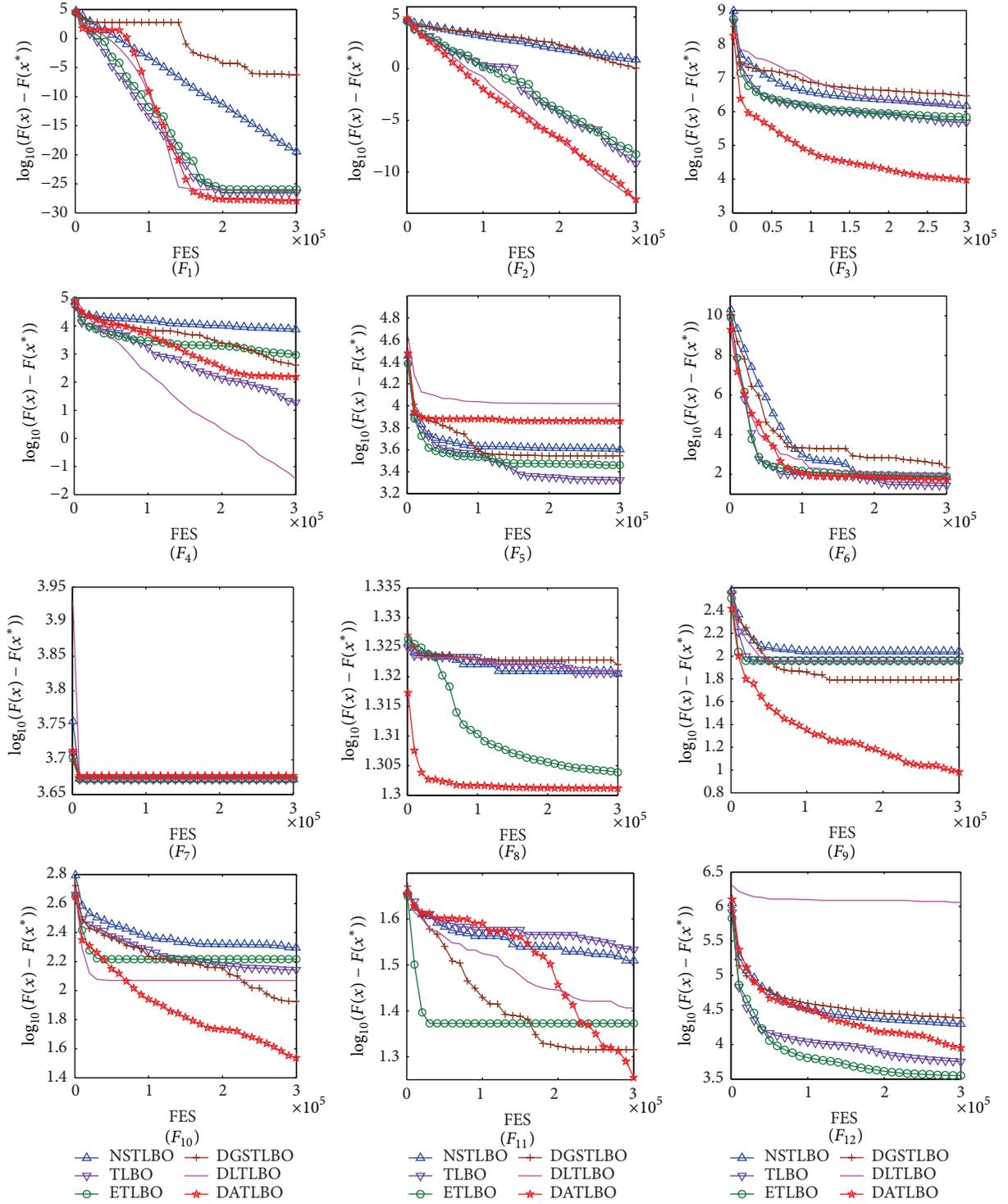


FIGURE 2: Convergence graph of the mean function error values derived from six algorithms versus the number of FES on twelve test functions.

the DATLBO algorithm is an important alternative for optimization of machining parameters in multitool milling operations.

Nomenclature

a_c : Tool clearance angle (deg.)
 a_e, a_p : Axial and radial depths of cut (mm)

a_l : Tool lead (corner) angle (deg.)
 A : Chip cross-sectional area (mm²)
 c_l, c_o : Labor cost and overhead costs (\$/min)
 c_m, c_{mat}, c_t : Machining cost, cost of raw material per part, and cutting tool cost (\$)
 c_u : Unit cost (\$)
 C : Cutting force equation constant
 C_F : Cutting force coefficient

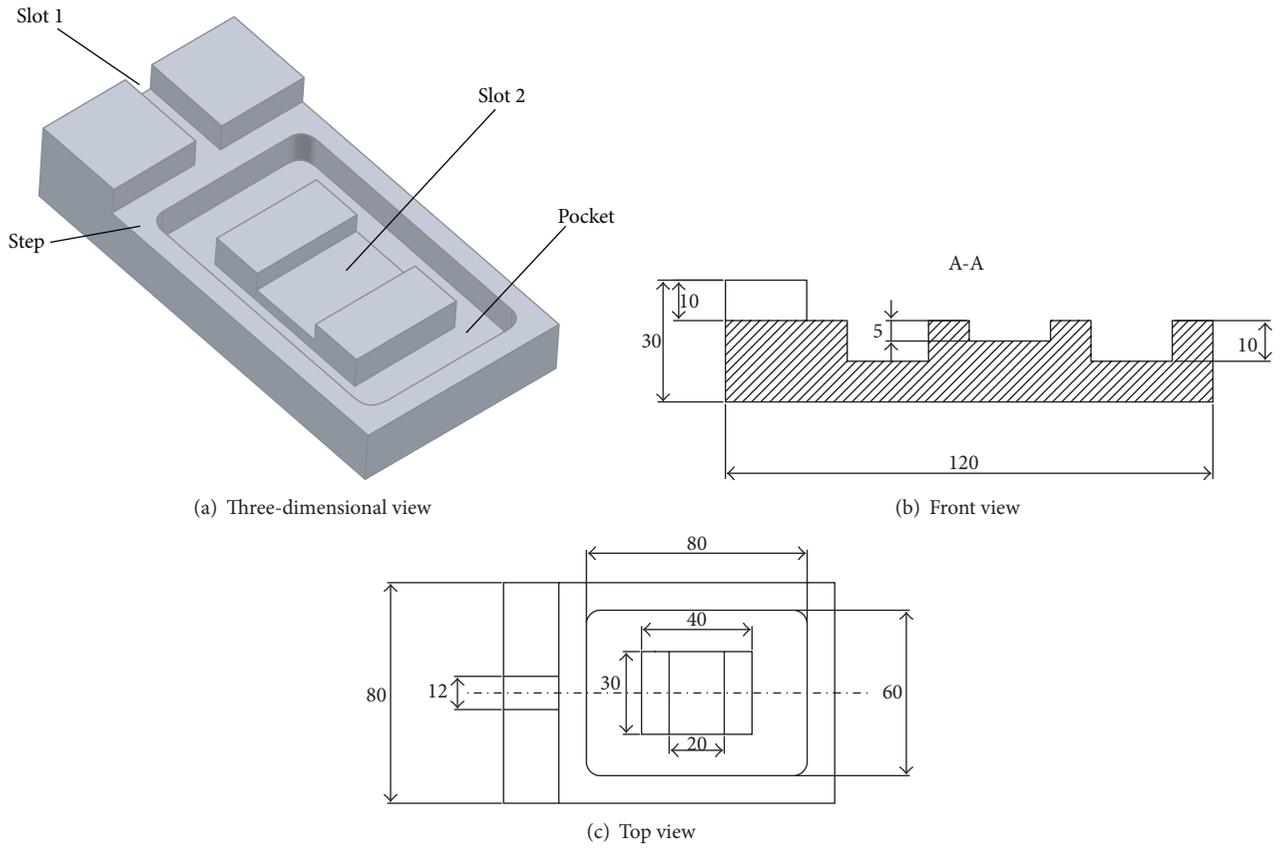


FIGURE 3: Milling operation schematic.

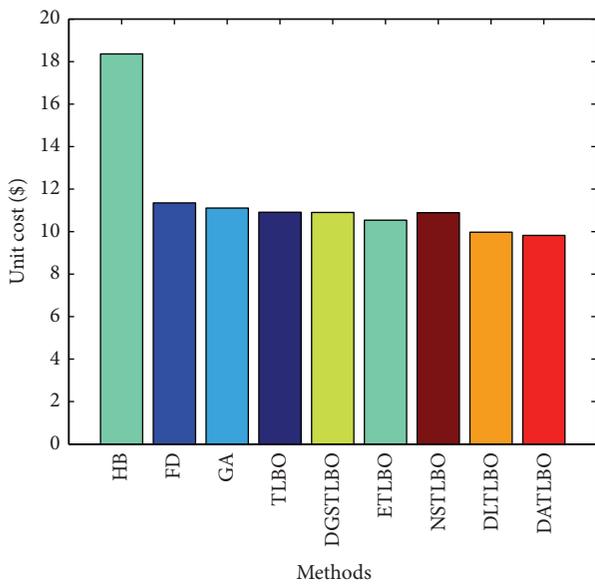


FIGURE 4: Comparison of unit cost obtained by various methods.

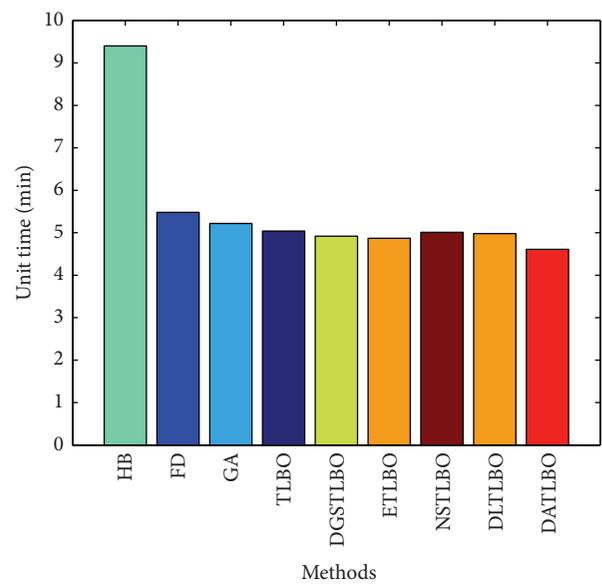


FIGURE 5: Comparison of unit time obtained by various methods.

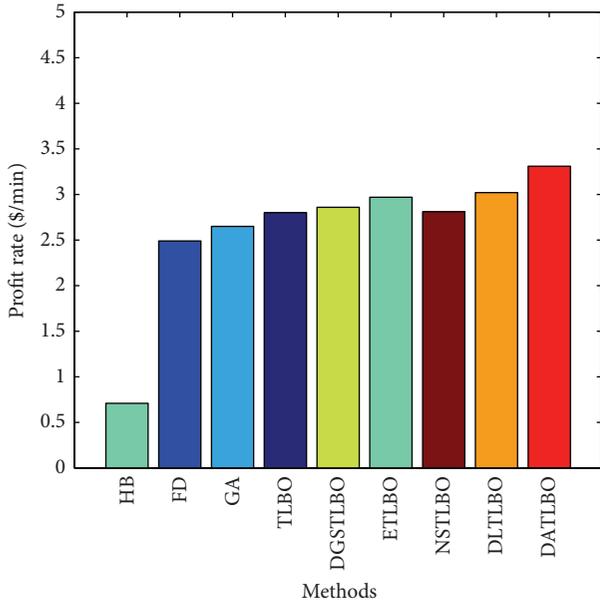


FIGURE 6: Comparison of profit rate obtained by various methods.

TABLE 4: Cutting velocity and feed per tooth ranges.

Operation	Operation type	Cutting velocity limits (m/min)	Feed per tooth limits (mm/tooth)
1	Face milling	60–120	0.05–0.4
2	Corner milling	40–70	0.05–0.5
3	Pocket milling	40–70	0.05–0.5
4	Slot milling 1	30–50	0.05–0.5
5	Slot milling 2	30–50	0.05–0.5

TABLE 5: Tools data.

Too	Tool type	Quality	d (mm)	z	Price (\$)	l_a	c_a
1	Face mill	Carbide	50	6	49.50	45	5
2	End mill	HSS	10	4	7.55	0	5
3	End mill	HSS	12	4	7.55	0	5

TABLE 6: Optimal results obtained by TLBO algorithm for milling case study.

Cutting velocity (m/min)	Feed per tooth (mm/tooth)	Unit cost (\$)	Unit time (min)	Profit rate (\$/min)
90.2536	0.3706			
60.8891	0.2196			
50.3567	0.2712	10.91	5.04	2.80
34.5682	0.1509			
39.3216	0.3991			

TABLE 7: Optimal results obtained by DGSTLBO algorithm for milling case study.

Cutting velocity (m/min)	Feed per tooth (mm/tooth)	Unit cost (\$)	Unit time (min)	Profit rate (\$/min)
89.3821	0.4872			
42.3654	0.4035			
41.2251	0.4902	10.90	4.92	2.86
42.8636	0.4662			
44.2648	0.4628			

TABLE 8: Optimal results obtained by ETLBO algorithm for milling case study.

Cutting velocity (m/min)	Feed per tooth (mm/tooth)	Unit cost (\$)	Unit time (min)	Profit rate (\$/min)
91.7539	0.4638			
71.3328	0.3898			
60.1682	0.4026	10.54	4.87	2.97
38.2543	0.4502			
35.3826	0.4026			

TABLE 9: Optimal results obtained by NSTLBO algorithm for milling case study.

Cutting velocity (m/min)	Feed per tooth (mm/tooth)	Unit cost (\$)	Unit time (min)	Profit rate (\$/min)
91.3682	0.3831			
58.9875	0.2537			
54.3568	0.3633	10.89	5.01	2.81
38.4828	0.2086			
41.3682	0.4128			

TABLE 10: Optimal results obtained by DLTLO algorithm for milling case study.

Cutting velocity (m/min)	Feed per tooth (mm/tooth)	Unit cost (\$)	Unit time (min)	Profit rate (\$/min)
90.4209	0.3905			
60.3259	0.2686			
52.6538	0.3562	9.97	4.98	3.02
35.3864	0.2482			
40.6637	0.5129			

TABLE 11: Optimal results obtained by DATLBO algorithm for milling case study.

Cutting velocity (m/min)	Feed per tooth (mm/tooth)	Unit cost (\$)	Unit time (min)	Profit rate (\$/min)
95.5829	0.3931			
43.6843	0.3902			
43.8542	0.4879	9.82	4.61	3.31
40.2535	0.3085			
41.9932	0.4873			

TABLE 12: Comparison of results for milling case study by various methods.

Method	C_u , unit cost (\$)	T_u , unit time (min)	P_r , profit rate (\$/min)
Handbook [26]	18.36	9.40	0.71
FD [25]	11.35	5.48	2.49
GA [24]	11.11	5.22	2.65
TLBO	10.91	5.04	2.80
DGSTLBO	10.90	4.92	2.86
ETLBO	10.54	4.87	2.97
NSTLBO	10.89	5.01	2.81
DLTLBO	9.97	4.98	3.02
DATLBO	9.82	4.61	3.31

- d : Tool diameter (mm)
- e : Machine tool efficiency factor
- f_z : Feed per tooth (mm/tooth)
- $F_c, F_c(\text{per})$: Cutting force and permitted cutting force (N)
- G, g : Slenderness ratio and exponent of slenderness ratio
- i : i th milling operation
- K : Distance traveled by tool to perform operation (mm)
- K_{FC} : Correction factor of cutting force under different experiment conditions
- m : Number of machining operations
- n, N : Tool life exponent and spindle speed (rpm)
- P, P_m : Required power for the operation and motor power (kW)
- P_r : Profit rate (\$/min)
- q_F : Influential exponent tool on cutting force
- Q : Contact proportion of cutting edge with work piece per revolution
- r : A vector which is distributed randomly in $[0, 1]$
- $R_a, R_{a(\text{at})}$: Value of surface roughness and attainable surface roughness (μm)
- S_p : Sale price of the product (\$)
- t_m, t_s, t_{ct} : Machining time, setup time, and tool changing time (min)
- Teacher*: The best learner
- T_F : Teaching factor
- T, T_u : Tool life (min) and unit production time (min)
- T_{max} : Maximum number of algorithm iterations
- u_F : Influential exponent axial depths of cut on cutting force
- $v, v_{\text{hb}}, v_{\text{opt}}$: Cutting velocity, recommended cutting velocity by handbook, and optimum cutting velocity (m/min)
- w : Tool wear exponent
- w_F : Influential exponent spindle speed on cutting force
- $X_{\text{new},i}$: i th learner
- x_F : Influential exponent radial depths of cut on cutting force

- y_F : Influential exponent feed per tooth on cutting force
- z : Number of tool teeth
- Ω : Feasible region
- τ : Arbitrarily small real number
- σ : Constant between zero and one
- ζ_n : Random vector sequence in probability space
- ζ^* : One convergence point with specific probability
- ζ_0 : One convergence point with probability 1.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors wish to acknowledge the financial support for this work from the National Natural Science Foundation of China (51575442 and 61402361) and the Shaanxi Province Education Department (11JS074).

References

- [1] R. Gupta, J. L. Batra, and G. K. Lal, "Determination of optimal subdivision of depth of cut in multipass turning with constraints," *International Journal of Production Research*, vol. 33, no. 9, pp. 2555–2565, 1995.
- [2] J. Wang, T. Kuriyagawa, X. P. Wei, and D. M. Guo, "Optimization of cutting conditions for single pass turning operations using a deterministic approach," *International Journal of Machine Tools & Manufacture*, vol. 42, no. 9, pp. 1023–1033, 2002.
- [3] Y. C. Shin and Y. S. Joo, "Optimization of machining conditions with practical constraints," *International Journal of Production Research*, vol. 30, no. 12, pp. 2907–2919, 1992.
- [4] P. G. Petropoulos, "Optimal selection of machining rate variables by geometric programming," *International Journal of Production Research*, vol. 11, no. 4, pp. 305–314, 1973.
- [5] M. S. Shunmugam, S. V. Bhaskara Reddy, and T. T. Narendran, "Selection of optimal conditions in multi-pass face-milling using a genetic algorithm," *International Journal of Machine Tools & Manufacture*, vol. 40, no. 3, pp. 401–414, 2000.
- [6] S. Li, Y. L. Liu, Y. Li, R. G. Landers, and L. Tang, "Process planning optimization for parallel drilling of blind holes using a two phase genetic algorithm," *Journal of Intelligent Manufacturing*, vol. 24, no. 4, pp. 791–804, 2013.
- [7] A. Krimpenis and G.-C. Vosniakos, "Rough milling optimisation for parts with sculptured surfaces using genetic algorithms in a Stackelberg gam," *Journal of Intelligent Manufacturing*, vol. 20, no. 4, pp. 447–461, 2009.
- [8] Y. Liu and C. Wang, "A modified genetic algorithm based optimisation of milling parameters," *International Journal of Advanced Manufacturing Technology*, vol. 15, no. 11, pp. 796–799, 1999.
- [9] Z. G. Wang, M. Rahman, Y. S. Wong, and J. Sun, "Optimization of multi-pass milling using parallel genetic algorithm and parallel genetic simulated annealing," *International Journal of Machine Tools & Manufacture*, vol. 45, no. 15, pp. 1726–1734, 2005.

- [10] H. Öktem, "An integrated study of surface roughness for modelling and optimization of cutting parameters during end milling operation," *International Journal of Advanced Manufacturing Technology*, vol. 43, no. 9-10, pp. 852-861, 2009.
- [11] S. Li, Y. Li, Y. Liu, and Y. Xu, "A GA-based NN approach for makespan estimation," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1003-1014, 2007.
- [12] C. A. C. António, C. F. Castro, and J. P. Davim, "Optimisation of multi-pass cutting parameters in face-milling based on genetic search," *International Journal of Advanced Manufacturing Technology*, vol. 44, no. 11-12, pp. 1106-1115, 2009.
- [13] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: a survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32-49, 2011.
- [14] O. Zarei, M. Fesanghary, B. Farshi, R. J. Saffar, and M. R. Razfar, "Optimization of multi-pass face-milling via harmony search algorithm," *Journal of Materials Processing Technology*, vol. 209, no. 5, pp. 2386-2392, 2009.
- [15] R. A. Mahdavi, N. Khani, and M. M. S. Fakhraee, "Optimization of milling parameters using artificial neural network and artificial immune system," *Journal of Mechanical Science and Technology*, vol. 26, no. 12, pp. 4097-4104, 2012.
- [16] J. F. Briceno, H. El-Mounayri, and S. Mukhopadhyay, "Selecting an artificial neural network for efficient modeling and accurate simulation of the milling process," *International Journal of Machine Tools & Manufacture*, vol. 42, no. 6, pp. 663-674, 2002.
- [17] R. Venkata Rao and P. J. Pawar, "Parameter optimization of a multi-pass milling process using non-traditional optimization algorithms," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 445-456, 2010.
- [18] G. C. Onwubolu, "Performance-based optimization of multi-pass face milling operations using Tribes," *International Journal of Machine Tools and Manufacture*, vol. 46, no. 7-8, pp. 717-727, 2006.
- [19] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems," *Information Sciences*, vol. 183, no. 1, pp. 1-15, 2012.
- [20] F. Zou, L. Wang, X. Hei, D. Chen, and D. Yang, "Teaching-learning-based optimization with dynamic group strategy for global optimization," *Information Sciences*, vol. 273, pp. 112-131, 2014.
- [21] L. Wang, F. Zou, X. Hei, D. Yang, D. Chen, and Q. Jiang, "An improved teaching-learning-based optimization with neighborhood search for applications of ANN," *Neurocomputing*, vol. 143, pp. 231-247, 2014.
- [22] R. V. Rao and V. Patel, "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 3, no. 4, pp. 535-560, 2012.
- [23] F. Zou, L. Wang, D. Chen, and X. Hei, "An improved teaching-learning-based optimization with differential learning and its application," *Mathematical Problems in Engineering*, vol. 2015, Article ID 754562, 19 pages, 2015.
- [24] A. Riza Yildiz, "A novel hybrid immune algorithm for global optimization in design and manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 2, pp. 261-270, 2009.
- [25] M. Tolouei-Rad and I. M. Bidhendi, "On the optimization of machining parameters for milling operations," *International Journal of Machine Tools & Manufacture*, vol. 37, no. 1, pp. 1-16, 1997.
- [26] Machinability Data Center, *Machining Data Handbook*, vol. 1, Machinability Data Center, 3rd edition, 1980.
- [27] A. Askarzadeh, "Bird mating optimizer: an optimization algorithm inspired by bird mating strategies," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 4, pp. 1213-1228, 2014.
- [28] D. Sarkar and J. M. Modak, "Pareto-optimal solutions for multi-objective optimization of fed-batch bioreactors using nondominated sorting genetic algorithm," *Chemical Engineering Science*, vol. 60, no. 2, pp. 481-492, 2005.
- [29] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55-66, 2011.
- [30] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311-338, 2000.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

