Chapter 2
Teaching-Learning-Based Optimization Algorithm

Abstract This chapter introduces teaching-learning-based optimization (TLBO) algorithm and its elitist and non-dominated sorting multiobjective versions. Two examples of unconstrained and constrained benchmark functions and an example of a multiobjective constrained problem are presented to demonstrate the procedural steps of the algorithm.

2.1 Teaching-Learning-Based Optimization Algorithm

All evolutionary and swarm intelligence based optimization algorithms require common control parameters like population size, number of generations, elite size, etc. Besides the common control parameters, different algorithms require their own algorithm-specific parameters. For example, GA uses mutation probability and crossover probability and selection operator; PSO uses inertia weight and social and cognitive parameters; ABC algorithm uses number of bees (scout, onlooker, and employed) and limit; and NSGA-II requires crossover probability, mutation probability, and distribution index. Proper tuning of these algorithm-specific parameters is a very crucial factor which affects the performance of the algorithms. The improper tuning of algorithm-specific parameters either increases the computational effort or yields a 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. Thus, there is a need to develop an algorithm which does not require any algorithm-specific parameters and teaching-learning-based optimization (TLBO) is such an algorithm.

The TLBO algorithm is a teaching-learning process inspired algorithm proposed by Rao et al. (2011, 2012a, b) and Rao and Savsani (2012) based on the effect of influence of a teacher on the output of learners in a class. The algorithm describes two basic modes of the learning: (i) through teacher (known as teacher phase) and (ii) through interaction with the other learners (known as learner phase). In this optimization algorithm, a group of learners is considered as population and different

© Springer International Publishing Switzerland 2016

R.V. Rao, *Teaching Learning Based Optimization Algorithm*,
DOI 10.1007/978-3-319-22732-0_2

subjects offered to the learners are considered as different design variables of the optimization problem and a learner's result is analogous to the 'fitness' value of the optimization problem. The best solution in the entire population is considered as the teacher. The design variables are actually the parameters involved in the objective function of the given optimization problem and the best solution is the best value of the objective function.

The working of TLBO is divided into two parts, 'Teacher phase' and 'Learner phase'. Working of both the phases is explained below.

2.1.1 Teacher Phase

It is the first part of the algorithm where learners learn through the teacher. During this phase, a teacher tries to increase the mean result of the class in the subject taught by him or her depending on his or her capability. At any iteration i, assume that there are 'm' number of subjects (i.e., design variables), 'n' number of learners (i.e., population size, $k = 1, 2,\ldots, n$) and $M_{j,i}$ be the mean result of the learners in a particular subject 'j' ($j = 1, 2,\ldots, m$) The best overall result $X_{\text{total-kbest},i}$ considering all the subjects together obtained in the entire population of learners can be considered as the result of best learner kbest. However, as the teacher is usually considered as a highly learned person who trains learners so that they can have better results, the best learner identified is considered by the algorithm as the teacher. The difference between the existing mean result of each subject and the corresponding result of the teacher for each subject is given by,

$$\text{Difference\_Mean}_{j,k,i} = r_i \left(X_{j,\text{kbest},i} - T_F M_{j,i} \right) \tag{2.1}$$

where, $X_{j,\text{kbest},i}$ is the result of the best learner in subject j. T_F is the teaching factor which decides the value of mean to be changed, and r_i is the random number in the range [0, 1]. Value of T_F can be either 1 or 2. The value of T_F is decided randomly with equal probability as,

$$T_F = \text{round}[1 + \text{rand}(0,1)\{2 - 1\}] \tag{2.2}$$

T_F is not a parameter of the TLBO algorithm. The value of T_F is not given as an input to the algorithm and its value is randomly decided by the algorithm using Eq. (2.2). After conducting a number of experiments on many benchmark functions it is concluded that the algorithm performs better if the value of T_F is between 1 and 2. However, the algorithm is found to perform much better if the value of TF is either 1 or 2 and hence to simplify the algorithm, the teaching factor is suggested to take either 1 or 2 depending on the rounding up criteria given by Eq. (2.2). Based on the Difference_Mean$_{j,k,i}$, the existing solution is updated in the teacher phase according to the following expression.

$$X'_{j,k,i} = X_{j,k,i} + \text{Difference\_Mean}_{j,k,i} \tag{2.3}$$

where, $X'_{j,k,i}$ is the updated value of $X_{j,k,i}$. $X'_{j,k,i}$ is accepted if it gives better function value. All the accepted function values at the end of the teacher phase are maintained and these values become the input to the learner phase. The learner phase depends upon the teacher phase.

2.1.2 Learner Phase

It is the second part of the algorithm where learners increase their knowledge by interacting among themselves. A learner interacts randomly with other learners for enhancing his or her knowledge. A learner learns new things if the other learner has more knowledge than him or her. Considering a population size of 'n', the learning phenomenon of this phase is explained below.

Randomly select two learners P and Q such that $X'_{\text{total-}P,i} \neq X'_{\text{total-}Q,i}$ (where, $X'_{\text{total-}P,i}$ and $X'_{\text{total-}Q,i}$ are the updated function values of $X_{\text{total-}P,i}$ and $X_{\text{total-}Q,i}$ of P and Q, respectively, at the end of teacher phase)

$$X''_{j,P,i} = X'_{j,P,i} + r_i(X'_{j,P,i} - X'_{j,Q,i}),\ \mathit{If}\ X'_{\text{total}-P,i} < X'_{\text{total}-Q,i} \tag{2.4}$$

$$X''_{j,P,i} = X'_{j,P,i} + r_i(X'_{j,Q,i} - X'_{j,P,i}),\ \mathit{If}\ X'_{\text{total}-Q,I} < X'_{\text{total}-P,i} \tag{2.5}$$

$X''_{j,P,I}$ is accepted if it gives a better function value.

The Eqs. (2.4) and (2.5) are for minimization problems. In the case of maximization problems, the Eqs. (2.6) and (2.7) are used.

$$X''_{j,P,i} = X'_{j,P,i} + r_i(X'_{j,P,i} - X'_{j,Q,i}),\ \mathit{If}\ X'_{\text{total}-Q,i} < X'_{\text{total}-P,i} \tag{2.6}$$

$$X''_{j,P,i} = X'_{j,P,i} + r_i(X'_{j,Q,i} - X'_{j,P,i}),\ \mathit{If}\ X'_{\text{total}-P,i} < X'_{\text{total}-Q,i} \tag{2.7}$$

Teaching-learning-based optimisation (TLBO) is a population-based algorithm which simulates the teaching-learning process of the class room. This algorithm requires only the common control parameters such as the population size and the number of generations and does not require any algorithm-specific control parameters.

2.2 Demonstration of the Working of TLBO Algorithm on Unconstrained Optimization Problems

To demonstrate the working of TLBO algorithm, an unconstrained benchmark function of sphere is considered. The objective function is to find out the values of x_i that minimize the sphere function.

Benchmark function: Sphere
Minimize,

$$f(x_i) = \sum_{i=1}^{n} x_i^2 \tag{2.8}$$

Range of variables: $-100 \leq x_i \leq 100$

The known solution to this benchmark function is 0 for all x_i values of 0. Now to demonstrate the TLBO algorithm, let us assume a population size of 5 (i.e., number of learners), two design variables x_1 and x_2 (i.e., number of subjects) and one iteration as the termination criterion. The initial population is randomly generated within the ranges of the variables and the corresponding values of the objective function are shown in Table 2.1. The mean values of x_1 and x_2 are also shown. As it is a minimization function, the lowest value of $f(x)$ is considered as the best learner (and is considered as equivalent to teacher).

Now the teacher tries to improve the mean result of the class. Assuming random numbers $r_1 = 0.58$ for x_1 and $r_2 = 0.49$ for x_2, and $T_f = 1$, the difference_mean values for x_1 and x_2 are calculated as,

$$\text{difference\_Mean}\,(x_1) = 0.58 * (-18 - (-8.2)) = -5.684$$

$$\text{difference\_Mean}\,(x_2) = 0.49 * (-27 - (-1)) = -12.74$$

The value of difference_mean (x_1) is added to all the values under the x_1 column and the value of difference_mean (x_2) is added to all the values under the x_2 column of Table 2.1. Table 2.2 shows the new values of x_1 and x_2 and the corresponding values of the objective function.

Table 2.1 Initial population

	x_1	x_2	$f(x)$	
	–55	36	4321	
	0	41	1681	
	96	–86	16612	
	–64	31	5057	
	–18	–27	1053	Teacher
Mean	–8.2	–1		

Table 2.2 New values of the variables and the objective function (teacher phase)

x_1	x_2	$f(x)$
–60.684	23.26	4223.575
–5.684	28.26	830.9355
90.316	–98.74	17906.57
–69.684	18.26	5189.287
–23.684	–39.74	2140.199

Table 2.3 Updated values of the variables and the objective function based on fitness comparison (teacher phase)

x_1	x_2	$f(x)$
–60.684	23.26	4223.575
–5.684	28.26	830.9355
96	–86	16612
–64	31	5057
–18	–27	1053

Now, the values of $f(x)$ of Tables 2.1 and 2.2 are compared and the best values of $f(x)$ are considered and placed in Table 2.3. This completes the teacher phase of the TLBO algorithm.

Now, the learner phase (also called student phase) starts and any student can interact with any other student for knowledge transfer. This interaction can be done in random manner. In this example, interactions between learners 1 and 2, 2 and 4, 3 and 5, 4 and 1, and 5 and 3 are considered. It is to be noted that every learner has to interact with any other learner. That is why, in this example, five interactions are considered (i.e., one interaction for each learner). Table 2.4 shows the new values of x_1 and x_2 for the learners after the interactions and considering random numbers $r_1 = 0.81$ for x_1 and $r_2 = 0.92$ for x_2. For example, the new values of x_1 and x_2 for learner 1 are calculated as explained below.

As it is a minimization function, the value of $f(x)$ is better for learner 2 as compared to that of learner 1 and hence the knowledge transfer is from learner 2 to learner 1. Hence the new values of x_1 and x_2 for learner 1 is calculated as,

$$(x_1)\,\text{new for learner 1} = -60.684 + 0.81(-5.684 - (-60.684)) = -16.134$$
$$(x_2)\,\text{new for learner 1} = 23.26 + 0.92(28.26 - 23.26) = 27.86.$$

Similarly, the new values of x_1 and x_2 for learner 2 are calculated as explained below.

As it is a minimization function, the value of $f(x)$ is better for learner 2 as compared to that of learner 4 and hence the knowledge transfer is from learner 2 to learner 4. Hence the new values of x_1 and x_2 for learner 2 is calculated as,

$$(x_1)\,\text{new for learner 2} = -5.684 + 0.81(-5.684 - (-64)) = 41.552$$
$$(x_2)\,\text{new for learner 2} = 28.26 + 0.92(28.26 - 31) = 25.7392$$

Table 2.4 New values of the variables and the objective function (learner phase)

x_1	x_2	$f(x)$	Interaction
–16.134	27.86	1036.486	1 and 2
41.552	25.7392	2389.075	2 and 4
3.66	–31.72	1019.554	3 and 5
–61.314	23.879	4329.613	4 and 1
–100(–110.34[a])	27.28	10744.2	5 and 3

[a] This value has crossed the given range of the variable and hence it is assigned the bound value

Now, the values of $f(x)$ of Tables 2.3 and 2.4 are compared and the best values of $f(x)$ are considered and placed in Table 2.5. This completes the learner phase and one iteration of the TLBO algorithm.

It can be noted that the minimum value of the objective function in the randomly generated initial population is 1053 and it has been reduced to 830.9355 (shown bold) at the end of first iteration. If we increase the number of iterations then the known value of the objective function (i.e., 0) can be obtained within next few iterations. Also, it is to be noted that in the case of maximization function problems, the best value means the maximum value of the objective function and the calculations are to be proceeded accordingly in the teacher and learner phases.

Now, the same minimization function of sphere is attempted in a different way. Here the interactions in the learner phase are considered in such a manner that each learner interacts with all other learners. For example, learner 1 interacts with the learners 2, 3, 4, and 5; learner 2 interacts with the learners 1, 3, 4, and 5; learner 3 interacts with learners 1, 2, 4, and 5; and so on. Tables 2.6, 2.7, 2.8, 2.9, and 2.10 show these interactions. The best interactions are also shown in the tables.

Table 2.11 shows the updated values of the variables and the objective function based on the best fitness values obtained in Tables 2.6, 2.7, 2.8, 2.9, and 2.10.

Table 2.5 Updated values of the variables and the objective function based on fitness comparison (learner phase)

x_1	x_2	$f(x)$
–16.134	27.86	1036.486
–5.684	**28.26**	**830.9355**
3.66	–31.72	1019.554
–61.314	23.879	4329.613
–18	–27	1053

Table 2.6 First learner interacting with every other learner (learner phase)

x_1	x_2	$f(x)$	
–60.684	23.26	4223.575	
–16.134	27.86	1036.486	best
–100(–187.598[a])	100(123.779[a])	20000	
–57.998	16.1392	3624.242	
–26.11	-22.9792	1209.776	

[a]These values have crossed the given ranges of the variables and hence they are assigned the bound values

Table 2.7 Second learner interacting with every other learner (learner phase)

x_1	x_2	$f(x)$	
38.866	32.86	2590.346	
–5.684	28.26	830.9355	best
–88.048	100(133.379[a])	17752.45	
41.552	25.7392	2389.075	
4.292	79.0992	6275.105	

[a]This value has crossed the given range of the variable and hence it is assigned the bound value

Table 2.8 Third learner interacting with every other learner (learner phase)

x_1	x_2	$f(x)$	
–30.914	14.5192	1166.483	
13.636	19.119	551.4767	best
96	–86	16612	
–33.6	21.64	1597.25	
3.66	–31.72	1019.554	

Table 2.9 Fourth learner interacting with every other learner (learner phase)

x_1	x_2	$f(x)$	
–61.314	23.879	4329.613	
–16.764	28.4792	1092.097	best
–100(–193.6[a])	100(138.64[a])	20000	
–64	31	5057	
–26.74	–22.36	1214.997	

[a] These values have crossed the given ranges of the variables and hence they are assigned the bound values

Table 2.10 Fifth learner interacting with every other learner (learner phase)

x_1	x_2	$f(x)$	
16.574	–73.239	5638.649	
–8.024	23.8392	632.692	best
–100(–110.34[a])	27.28	10744.2	
19.26	–80.36	6828.677	
–18	–27	1053	

[a] This value has crossed the given range of the variable and hence it is assigned the bound value

Table 2.11 updated values of the variables and the objective function based on the best fitness values obtained (learner phase)

x_1	x_2	$f(x)$
–16.134	27.86	1036.486
–5.684	28.26	830.9355
13.636	**19.119**	**551.4767**
–16.764	28.4792	1092.097
–8.024	23.8392	632.692

It can be noted that the minimum value of the objective function in the randomly generated initial population is 1053 and it has been reduced to 551.4767 at the end of first iteration. It can be noted that by following the above approach of every learner interacting with all the remaining learners, the number of calculations get increased but this approach may provide the global optimum value (i.e., 0 in this example) in less number of iterations as compared to the approach of considering random interactions between the learners.

2.3 Demonstration of the Working of TLBO Algorithm on Constrained Optimization Problems

To demonstrate the working of TLBO algorithm, a constrained benchmark function of Himmelblau is considered. The objective function is to find out the values of x_1 and x_2 that minimize the Himmelblau function.

Benchmark function: Himmelblau

Minimize,

$$f(x_i) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{2.9}$$

Constraints:

$$g_1(x) = 26 - (x_1 - 5)^2 - x_2^2 \geq 0 \tag{2.10}$$

$$g_2(x) = 20 - 4x_1 - x_2 \geq 0 \tag{2.11}$$

Ranges of variables: $-5 \leq x_1, x_2 \leq 5$

The known solution to this benchmark function is 0 for $x_1 = 3$ and $x_2 = 2$ and $g_1(x) = 18$ and $g_2(x) = 6$. Now to demonstrate the TLBO algorithm, let us assume a population size of 5 (i.e., number of learners), two design variables x_1 and x_2 (i.e., number of subjects) and one iteration as the termination criterion. The initial population is randomly generated within the ranges of the variables and the corresponding values of the objective function are shown in Table 2.12. The mean values of x_1 and x_2 are also shown. As it is a minimization function, the lowest value of $f(x)$ is considered as the best learner (and is considered as equivalent to teacher). If the constraints are violated then penalties are assigned to the objective function. There are many ways to assign the penalties and in this example the penalty p_1 for violation of $g_1(x)$ is considered as $10 * (g_1(x))^2$ and the penalty p_2 for violation of $g_2(x)$ is considered as $10 * (g_2(x))^2$. As it is a minimization problem, the values of penalties are added to the value of the objective function $f(x)$ and the fitness function is $f'(x) = f(x) + 10 * (g_1(x))^2 + 10 * (g_2(x))^2$. The function $f'(x)$ is called the pseudo-objective function.

Table 2.12 Initial population

	x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$
	3.22	**0.403**	**13.13922**	**22.66919**	**0**	**6.717**	**0**	**13.13922**
	0.191	2.289	77.71054	–2.366	55.979	16.947	0	133.6902
	3.182	0.335	14.02423	22.58265	0	6.937	0	14.02423
	1.66	4.593	261.5732	–6.25125	390.781	8.767	0	652.3543
	2.214	0.867	43.64116	17.48652	0	10.277	0	43.64116
Mean	2.093	1.697						

Table 2.13 New values of the variables, objective function, and the penalties (teacher phase)

x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$
3.5017	–0.794	8.443577	23.12466	0	6.7872	0	8.443577
0.4727	1.092	122.2511	4.311091	0	17.0172	0	122.2511
3.4637	–0.862	7.820563	22.89674	0	7.0072	0	7.820563
1.9417	3.396	56.61739	5.113985	0	8.8372	0	56.61739
2.4957	–0.33	45.34465	19.61958	0	10.3472	0	45.34465

It may be noted that 10 * $(g_1(x))^2$ is used as the penalty function in this example for the violation in constraint $g_1(x)$ and 10 * $(g_2(x))^2$ is used as the penalty function for the violation in constraint $g_2(x)$. Higher penalties are desirable and one may use 50 * $(g_1(x))^2$ or 100 * $(g_1(x))^2$ or 500 * $(g_1(x))^2$ or any such penalty for violation of $g_1(x)$ and 50 * $(g_2(x))^2$ or 100 * $(g_2(x))^2$ or 500 * $(g_2(x))^2$ or any such penalty for violation of $g_2(x)$. The assignment of penalties for violations depends upon the designer/decision-maker/user. Sometimes, the penalty functions assigned may be different for different constraints depending upon the application as decided by the designer/decision-maker/user.

The mean values of x_1 and x_2 are also shown in Table 2.12. As it is a minimization function, the lowest value of $f'(x)$ is considered as the best learner (and is considered as equivalent to teacher).

Now the teacher tries to improve the mean result of the class. Assuming random numbers $r_1 = 0.25$ for x_1 and $r_2 = 0.925$ for x_2 and $T_f = 1$, the difference_mean values for x_1 and x_2 are calculated as,

$$\text{difference\_mean}(x_1) = 0.25 * (3.22 - (2.093)) = 0.2817$$

$$\text{difference\_mean}(x_2) = 0.925 * (0.403 - (1.697)) = -1.197$$

The value of difference_mean (x_1) is added to all the values under the x_1 column and the value of difference_mean (x_2) is added to all the values under the x_2 column of Table 2.12. Table 2.13 shows the new values of x_1 and x_2, penalties and the corresponding values of the objective function.

Now, the values of $f'(x)$ of Tables 2.12 and 2.13 are compared and the best values of $f'(x)$ are considered and placed in Table 2.14. This completes the teacher phase of the TLBO algorithm.

Table 2.14 Updated values of the variables, objective function, and the penalties based on fitness comparison (teacher phase)

x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$
3.5017	–0.794	8.443577	23.12466	0	6.7872	0	8.443577
0.4727	1.092	122.2511	4.311091	0	17.0172	0	122.2511
3.4637	–0.862	7.820563	22.89674	0	7.0072	0	7.820563
1.9417	3.396	56.61739	5.113985	0	8.8372	0	56.61739
2.214	0.867	43.64116	17.48652	0	10.277	0	43.64116

Table 2.15 New values of the variables, objective function and the penalties (learner phase)

x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$	Interaction
5(6.4095[a])	–2.0199	147.8492	21.92	0	2.0199	0	147.8492	1 and 2
1.8829	2.5896	26.19377	9.577659	0	9.8788	0	26.19377	2 and 4
4.6634	–1.9869	79.34047	21.93893	0	3.3333	0	79.34047	3 and 5
3.4393	0.6725	11.91628	23.11196	0	5.5703	0	11.91628	4 and 1
3.8856	0.7207	29.95277	24.2387	0	3.7369	0	29.95277	5 and 2

[a] This value has crossed the given range of the variable and hence it is assigned the bound value

Now, the learner phase (also called student phase) starts and any student can interact with any other student for knowledge transfer. This interaction can be done in random manner. In this example, interactions between learners 1 and 2, 2 and 4, 3 and 5, 4 and 1, and 5 and 2 are considered. Table 2.15 shows the new values of x_1 and x_2 for the learners after the interactions and considering random numbers $r_1 = 0.96$ for x_1 and $r_2 = 0.65$ for x_2. For example, the new values of x_1 and x_2 for learner 1 are calculated as explained below.

As it is a minimization function, the value of $f'(x)$ is better for learner 1 as compared to that of learner 2 and hence the knowledge transfer is from learner 1 to learner 2. Hence the new values of x_1 and x_2 for learner 1 is calculated as,

$$\begin{aligned}(x_1)\,\text{new for learner 1} &= 3.5017 + 0.96(3.5017 - (0.4727)) = 6.4095\\ (x_2)\,\text{new for learner 1} &= -0.794 + 0.65(-0.794 - 1.092) = -2.0199\end{aligned}$$

Similarly, the new values of x_1 and x_2 for learner 2 are calculated. The value of $f'(x)$ is better for learner 4 as compared to that of learner 2 and hence the knowledge transfer is from learner 4 to learner 2. Hence the new values of x_1 and x_2 for learner 2 is calculated as,

$$(x_1)\,\text{new for learner 2} = 0.4727 + 0.96(1.9417 - 0.4727) = 1.8829$$

$$(x_2)\,\text{new for learner 2} = 1.092 + 0.65(3.396 - 1.092) = 2.5896$$

Now, the values of $f'(x)$ of Tables 2.14 and 2.15 are compared and the best values of $f'(x)$ are considered and are placed in Table 2.16. This completes the learner phase and one iteration of the TLBO algorithm.

It can be noted that the minimum value of the objective function in the randomly generated initial population is –13.13922 and it has been reduced to 7.820563 at the end of the first iteration. If we increase the number of iterations then the known value of the objective function (i.e., 0) can be obtained within next few iterations.

It is to be noted that in the case of maximization problems, the best value means the maximum value of the objective function and the calculations are to be proceeded accordingly in the teacher and learner phases. The values of penalties are to

Table 2.16 Updated values of the variables, objective function, and the penalties based on fitness comparison (learner phase)

x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$
3.5017	–0.794	8.443577	23.12466	0	6.7872	0	8.443577
1.8829	2.5896	26.19377	9.577659	0	9.8788	0	26.19377
3.4637	–0.862	7.820563	22.89674	0	7.0072	0	7.820563
3.4393	0.6725	11.91628	23.11196	0	5.5703	0	11.91628
3.8856	0.7207	29.95277	24.2387	0	3.7369	0	29.95277

Table 2.17 First learner interacting with every other learner

x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$	
3.5017	–0.794	8.443577	23.12466	0	6.7872	0	8.443577	
5(6.4095[a])	–2.0199	147.8492	21.92	0	2.0199	0	147.8492	
3.465	–0.8382	8.05084	22.9412	0	6.9782	0	8.05084	best
4.999	–3.5175	217.2476	13.62719	0	3.5215	0	217.2476	
4.7378	–1.8737	93.20215	22.4205	0	2.9225	0	93.20215	

[a] This has crossed the given range of the variable and hence is assigned the bound value

be subtracted from the objective function in the case of maximization problems (i.e., $f'(x) = f(x) - 10 * (g_1(x))^2 - 10 * (g_2(x))^2$).

Now, the same minimization function of Himmelblau is attempted in a different way. Here the interactions in the learner phase are considered in such a manner that each learner interacts with all other learners. For example, learner 1 interacts with the learners 2, 3, 4, and 5; learner 2 interacts with the learners 1, 3, 4, and 5; learner 3 interacts with learners 1, 2, 4, and 5; and so on. Tables 2.17, 2.18, 2.19, 2.20, and 2.21 show these interactions. The best interactions are also shown in the tables.

Table 2.22 shows the updated values of the variables, objective function and the penalties based on the best fitness values obtained in Tables 2.17, 2.18, 2.19, 2.20 and 2.21

It can be noted that the minimum value of the objective function in the randomly generated initial population is –13.13922 and it has been reduced to 7.597071 at the end of first iteration. It can be noted that by following the above approach of every learner interacting with all the remaining learners, the number of calculations seem

Table 2.18 Second learner interacting with every other learner

x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$	
3.3805	–0.1339	13.05768	23.35929	0	6.6119	0	13.05768	best
0.4727	1.092	122.2511	4.311091	0	17.0172	0	122.2511	
3.3441	–0.1781	13.13471	23.22628	0	6.8017	0	13.13471	
1.8829	2.5896	26.19377	9.577659	0	9.8788	0	26.19377	
2.1443	0.9457	45.46327	16.95063	0	10.4771	0	45.46327	

Table 2.19 Third learner interacting with every other learner

x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$	
3.4272	–0.9062	7.597071	22.7051	0	7.1974	0	7.597071	best
5(6.3351[a])	–2.1321	147.3284	21.45415	0	2.1321	0	147.3284	
3.4637	–0.862	7.820563	22.89674	0	7.0072	0	7.820563	
4.9248	–3.6297	215.8199	12.81962	0	3.9305	0	215.8199	
4.6634	–1.9859	79.34521	21.9429	0	3.3323	0	79.34521	

[a] This has crossed the given range of the variable and hence is assigned the bound value

Table 2.20 Fourth learner interacting with every other learner

x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$	
3.4393	0.6725	11.91628	23.11196	0	5.5703	0	11.91628	
3.3519	4.8936	438.3633	–0.66355	4.4030	1.6988	0	442.7664	
3.4028	0.6283	11.71331	23.05419	0	5.7605	0	11.71331	best
1.9417	3.396	56.61739	5.113985	0	8.8372	0	56.61739	
2.2031	1.7521	22.29212	15.1075	0	9.4355	0	22.29212	

Table 2.21 Fifth learner interacting with every other learner

x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$	
3.4502	–0.2127	12.75966	23.55288	0	6.4119	0	12.75966	
3.8856	0.7207	29.95277	24.2387	0	3.7369	0	29.95277	
3.4137	–0.2569	12.5497	23.41765	0	6.6021	0	12.5497	best
2.4754	–0.7769	47.28898	19.02282	0	10.8753	0	47.28898	
2.214	0.867	43.64116	17.48652	0	10.277	0	43.64116	

Table 2.22 Updated values of the variables, objective function, and the penalties based on the best fitness values (learner phase)

x_1	x_2	$f(x)$	$g_1(x)$	p_1	$g_2(x)$	p_2	$f'(x)$
3.465	–0.8382	8.05084	22.9412	0	6.9782	0	8.05084
3.3805	–0.1339	13.05768	23.35929	0	6.6119	0	13.05768
3.4272	–0.9062	7.597071	22.7051	0	7.1974	0	7.597071
3.4028	0.6283	11.71331	23.05419	0	5.7605	0	11.71331
3.4137	–0.2569	12.5497	23.41765	0	6.6021	0	12.5497

to be increased but this approach may provide the global optimum value (i.e., 0 in this example) in less number of iterations as compared to the approach of considering random interactions between the learners. The two demonstrations of working of the TLBO algorithm clearly prove its simplicity and effectiveness in solving the unconstrained and constrained optimization problems. The flowchart of the TLBO algorithm is shown in Fig. 2.1.

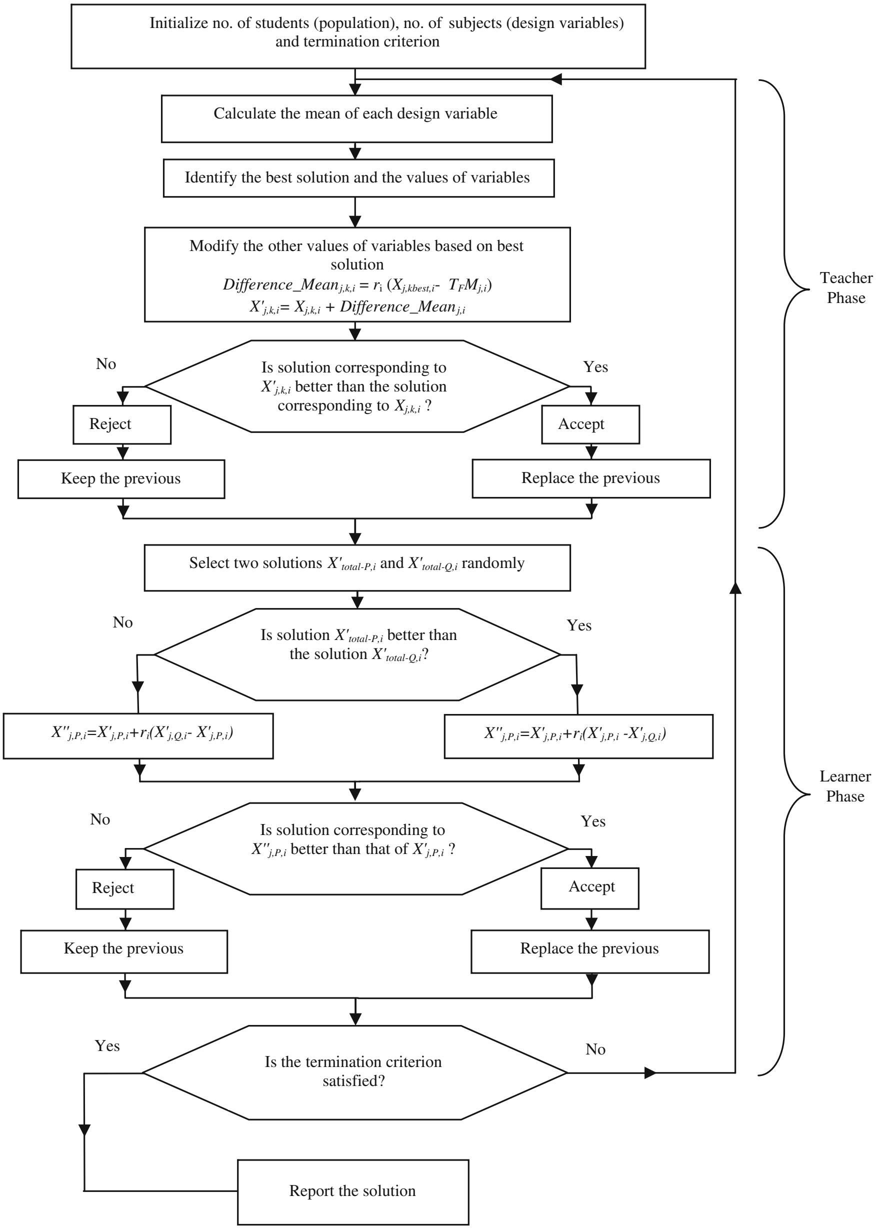
Initialize no. of students (population), no. of subjects (design variables) and termination criterion
Calculate the mean of each design variable
Identify the best solution and the values of variables
Modify the other values of variables based on best solution
$Difference\_Mean_{j,k,i} = r_i\ (X_{j,kbest,i} -\ T_F M_{j,i})$
$X'_{j,k,i} = X_{j,k,i} + Difference\_Mean_{j,i}$
No
Yes
Is solution corresponding to $X'_{j,k,i}$ better than the solution corresponding to $X_{j,k,i}$?
Reject
Accept
Keep the previous
Replace the previous
Teacher Phase
Select two solutions $X'_{total-P,i}$ and $X'_{total-Q,i}$ randomly
No
Yes
Is solution $X'_{total-P,i}$ better than the solution $X'_{total-Q,i}$?
$X''_{j,P,i}=X'_{j,P,i}+r_i(X'_{j,Q,i}- X'_{j,P,i})$
$X''_{j,P,i}=X'_{j,P,i}+r_i(X'_{j,P,i} -X'_{j,Q,i})$
Learner Phase
No
Yes
Is solution corresponding to $X''_{j,P,i}$ better than that of $X'_{j,P,i}$?
Reject
Accept
Keep the previous
Replace the previous
Yes
No
Is the termination criterion satisfied?
Report the solution

Fig. 2.1 Flowchart of TLBO algorithm

2.4 Elitist TLBO Algorithm

In the works on TLBO algorithmby Rao et al. (2011) and Rao and Savsani (2012), the aspect of ‘elitism’ was not considered and only two common controlling parameters, i.e., population size and number of generations were used. Rao and

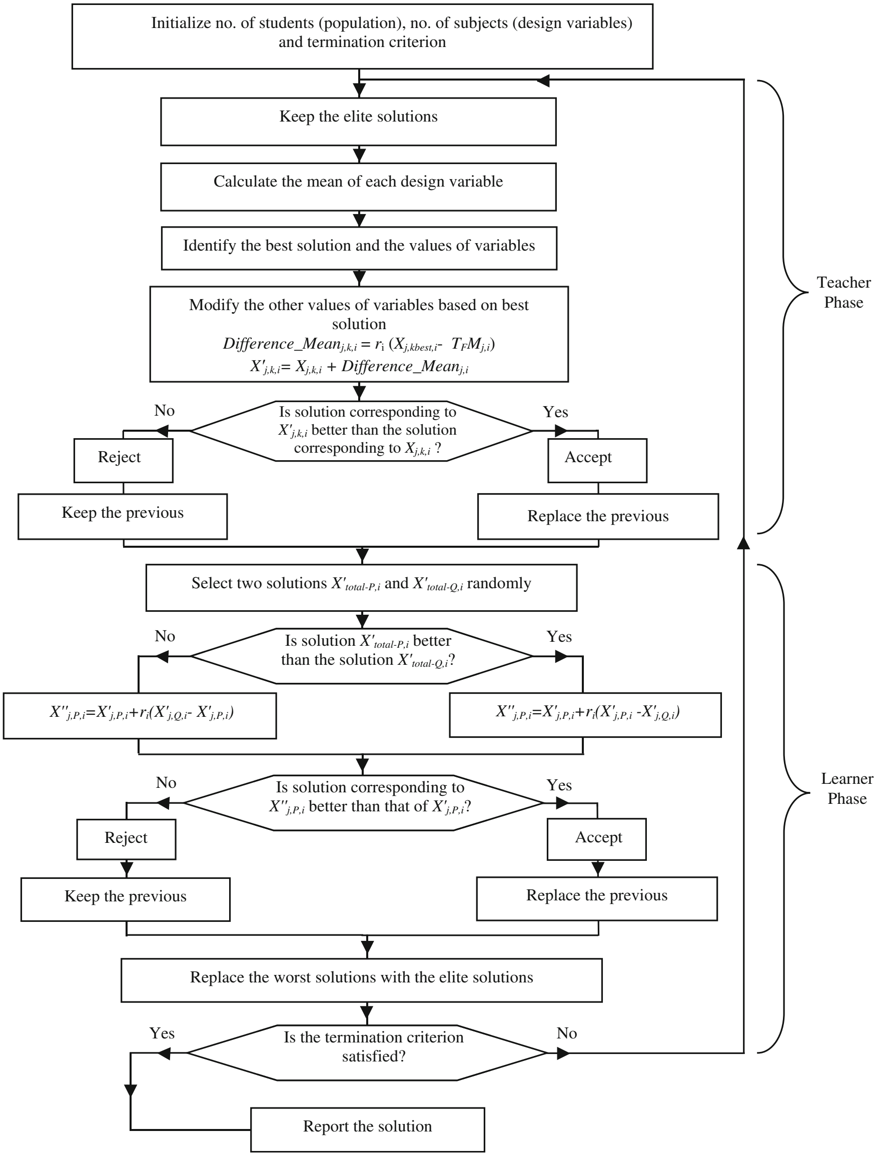
Initialize no. of students (population), no. of subjects (design variables) and termination criterion
Keep the elite solutions
Calculate the mean of each design variable
Identify the best solution and the values of variables
Modify the other values of variables based on best solution
$Difference\_Mean_{j,k,i} = r_i (X_{j,kbest,i} - T_F M_{j,i})$
$X'_{j,k,i} = X_{j,k,i} + Difference\_Mean_{j,i}$
Is solution corresponding to $X'_{j,k,i}$ better than the solution corresponding to $X_{j,k,i}$?
No
Yes
Reject
Accept
Keep the previous
Replace the previous
Select two solutions $X'_{total-P,i}$ and $X'_{total-Q,i}$ randomly
Is solution $X'_{total-P,i}$ better than the solution $X'_{total-Q,i}$?
No
Yes
$X''_{j,P,i}=X'_{j,P,i}+r_i(X'_{j,Q,i}- X'_{j,P,i})$
$X''_{j,P,i}=X'_{j,P,i}+r_i(X'_{j,P,i} -X'_{j,Q,i})$
Is solution corresponding to $X''_{j,P,i}$ better than that of $X'_{j,P,i}$?
No
Yes
Reject
Accept
Keep the previous
Replace the previous
Replace the worst solutions with the elite solutions
Is the termination criterion satisfied?
Yes
No
Report the solution
Teacher Phase
Learner Phase

Fig. 2.2 Flowchart of ETLBO algorithm

Patel (2012) introduced ‘elitism’ in the TLBO algorithm to identify its effect on the exploration and exploitation capacities of the algorithm. The concept of elitism is utilized in most of the evolutionary and swarm intelligence algorithms where during every generation the worst solutions are replaced by the elite solutions. The flowchart of the elitist TLBO algorithm is shown in Fig. 2.2.

2.5 Non-dominated Sorting TLBO Algorithm for Multiobjective Optimization

Multiobjective optimization is an area of multiple criteria decision-making that is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously. Multiobjective optimization has been applied in many fields of science, engineering, economics, and logistics where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Minimizing cost while maximizing comfort while buying a car, and maximizing performance while minimizing fuel consumption, and emission of pollutants of a vehicle are examples of multiobjective optimization problems involving two and three objectives, respectively. In practical problems there can be more than three objectives.

For a multiobjective optimization problem, there does not exist a single solution that simultaneously optimizes each objective. In that case, the objective functions are said to be conflicting and there exist a large number of Pareto optimal solutions. A solution is called non-dominated, Pareto optimal, Pareto efficient, or noninferior, if none of the objective functions can be improved in value without degrading some of the other objective values. Researchers study multiobjective optimization problems from different viewpoints and, thus, there exist different solution philosophies and goals when setting and solving them. The goal may be to find a representative set of Pareto optimal solutions, and/or quantify the trade-offs in satisfying the different objectives, and/or finding a single solution that satisfies the subjective preferences of a decision-maker.

Two important approaches of solving the multiobjective optimization problems are considered in this book and these are (1) a priori approach and (2) a posteriori approach. In the a priori approach, the preferences of the decision-maker are asked and the best solution according to the given preferences is found. The preferences of the decision-maker are in the form of weights assigned to the objective functions. The weights may be assigned through any method like direct assignment, eigenvector method (Rao 2007), empty method, minimal information method, etc. Once the weights are decided by the decision-maker, the multiple objectives are combined into a scalar objective via the weight vector. However, if the objective functions are simply weighted and added to produce a single fitness, the function with the largest range would dominate the evolution. A poor input value for the objective with the larger range makes the overall value much worse than a poor value for the objective with smaller range. To avoid this, all objective functions are normalized to have same range. For example, if $f_1(x)$ and $f_2(x)$ are the two objective functions to me minimized, then the combined objective function can be written as,

$$\min f(x) = \left\{ w_1 \left[\left(\frac{f_{1(x)}}{f_1^*} \right) \right] + w_2 \left[\left(\frac{f_{2(x)}}{f_2^*} \right) \right] \right\} \tag{2.12}$$

where, $f(x)$ is the combined objective function and f_1^* is the minimum value of the objective function $f_1(x)$ when solved it independently without considering $f_2(x)$ (i.e., solving the multiobjective problem as a single objective problem and considering only $f_1(x)$ and ignoring $f_2(x)$). And f_2^* is the minimum value of the objective function $f_2(x)$ when solved it independently without considering $f_1(x)$ (i.e. solving the multiobjective problem as a single objective problem considering only $f_2(x)$ and ignoring $f_1(x)$). w_1 and w_2 are the weights assigned by the decision-maker to the objective functions $f_1(x)$ and $f_2(x)$ respectively. Suppose $f_1(x)$ and $f_2(x)$ are not of the same type (i.e., minimization or maximization) but one is a minimization function (say $f_1(x)$) and the other is a maximization function (say $f_2(x)$). In that case, the Eq. (2.12) is written as Eq. (2.13) and f_2^* is the maximum value of the objective function $f_2(x)$ when solved it independently without considering $f_1(x)$.

$$\min\ f(x) = \left\{ w_1 \left[\left(\frac{f_{1(x)}}{f_1^*} \right) \right] - w_2 \left[\left(\frac{f_{2(x)}}{f_2^*} \right) \right] \right\} \tag{2.13}$$

In general, the combined objective function can include any number of objectives and the summation of all weights is equal to 1.

A posteriori approach aims to generate all the Pareto optimal solutions or a representative set of Pareto optimal solutions and the decision-maker chooses the best one among them. Evolutionary algorithms are popular approaches for generating the Pareto optimal solutions to a multiobjective optimization problem. Currently, most evolutionary multiobjective optimization algorithms apply Pareto-based ranking schemes. Evolutionary algorithms such as the non-dominated sorting genetic algorithm-II (NSGA-II) and strength Pareto evolutionary algorithm 2 (SPEA-2) have become standard approaches. The main advantage of evolutionary algorithms, when applied to solve multiobjective optimization problems, is the fact that they typically generate sets of solutions, allowing computation of an approximation of the entire Pareto front. The main disadvantage of evolutionary algorithms is their low speed and the Pareto optimality of the solutions cannot be guaranteed. It is only known that none of the generated solutions dominates the others. In this book, a new approach for generating the Pareto optimal solutions to the multiobjective optimization problems is described and it is named as "non-dominated sorting teaching-learning-based optimization algorithm (NSTLBO)."

The NSTLBO algorithm is an extension of the TLBO algorithm. The NSTLBO algorithm is a posteriori approach for solving multiobjective optimization problems and maintains a diverse set of solutions. The NSTLBO algorithm consists of teacher phase and learner phase similar to the TLBO algorithm. However, in order to handle multiple objectives effectively and efficiently, the NSTLBO algorithm is incorporated with non-dominated sorting approach and crowding distance computation mechanism proposed by Deb (2001). The teacher phase and learner phase ensure good exploration and exploitation of the search space while non-dominated sorting approach makes certain that the selection process is always towards the good solutions and the population is pushed towards the Pareto front in each iteration. The crowding distance assignment mechanism assures the selection of teacher from a sparse region of the search space thus averting any chance of 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 not a simple task. 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.

At the beginning an initial population is randomly generated with P 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. Once the teacher is selected the mean of the learners is calculated and the learners are updated based on the teacher phase of the TLBO algorithm, i.e., according to Eqs. (2.1)–(2.3).

After the teacher phase the updated learners (new learners) are recombined with the initial population to obtain a set of $2P$ 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, P number of best learners are selected. These learners are further updated according to the learner phase of the TLBO algorithm.

In the learner phase, a learner interacts with another randomly chosen learner to enhance his or her knowledge in the subject. The learner with a higher rank is regarded as superior among the two learners. If both the learners hold the same rank then the learner with greater crowding distance value is regarded as superior to the other. Then learners are updated based on Eqs. (2.4) and (2.5) for minimization problems or Eqs. (2.6) and (2.7) for maximization problems.

After the end of 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, P number of best learners are selected and these learners are directly updated based on the teacher phase in the next iteration.

2.5.1 Non-dominated Sorting of the Population

In this approach the population is sorted into several ranks based on the dominance concept as follows: a solution $x^{(1)}$ is said to dominate other solution $x^{(2)}$ if and only if solution $x^{(1)}$ is no worse than solution $x^{(2)}$ with respect to all the objectives or the solution $x^{(1)}$ is strictly better than solution $x^{(2)}$ in at least one objective. If any of the two conditions are violated then solution $x^{(1)}$ does not dominate solution $x^{(2)}$. Among a set of solutions P, the non-dominated set of solutions n are those that are not dominated by any member of the set P (Deb 2001).

To determine the non-dominated set of solutions from the given population set P, each solution i in P is compared with every other solution in P. If solution i is not dominated by any other solution in P then it is a member of the non-dominated set n. The non-dominated solutions identified after the first sorting are ranked as 1 and are deleted from the set P. The remaining members of set P are sorted in the similar manner and the non-dominated set of solutions identified after second sorting are ranked as 2 and are deleted from the set P. This procedure is repeated until entire population is sorted or the set P becomes an empty set.

2.5.2 Crowding Distance Computation

The crowding distance computation requires sorting the population according to each objective function value in ascending order of magnitude. Thereafter, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. This calculation is continued with the other objective functions. The overall crowding distance value is calculated as the sum of individual distance values corresponding to each objective. Each objective function is normalized before calculating the crowding distance.

2.5.3 Constraint Handling

In order to effectively handle the constraints a constrained dominance concept (Deb 2001) 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.

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

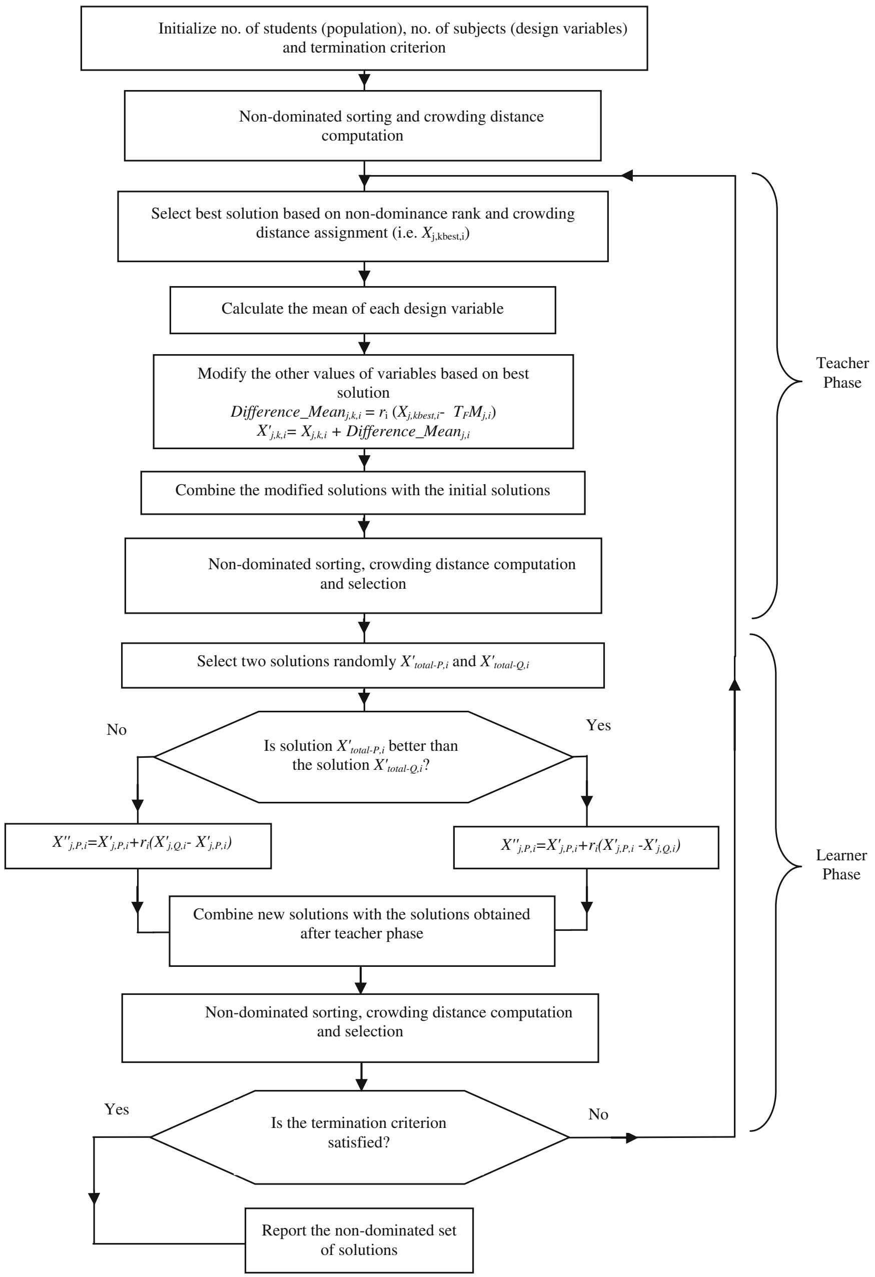
Initialize no. of students (population), no. of subjects (design variables) and termination criterion
Non-dominated sorting and crowding distance computation
Select best solution based on non-dominance rank and crowding distance assignment (i.e. $X_{\mathrm{j,kbest,i}}$)
Calculate the mean of each design variable
Modify the other values of variables based on best solution
$Difference\_Mean_{j,k,i} = r_{\mathrm{i}} \, (X_{j,kbest,i} - T_F M_{j,i})$
$X'_{j,k,i} = X_{j,k,i} + Difference\_Mean_{j,i}$
Combine the modified solutions with the initial solutions
Non-dominated sorting, crowding distance computation and selection
Select two solutions randomly $X'_{total\text{-}P,i}$ and $X'_{total\text{-}Q,i}$
Is solution $X'_{total\text{-}P,i}$ better than the solution $X'_{total\text{-}Q,i}$?
No
Yes
$X''_{j,P,i} = X'_{j,P,i} + r_i(X'_{j,Q,i} - X'_{j,P,i})$
$X''_{j,P,i} = X'_{j,P,i} + r_i(X'_{j,P,i} - X'_{j,Q,i})$
Combine new solutions with the solutions obtained after teacher phase
Non-dominated sorting, crowding distance computation and selection
Is the termination criterion satisfied?
Yes
No
Report the non-dominated set of solutions
Teacher Phase
Learner Phase

Fig. 2.3 Flowchart of NSTLBO algorithm

This constrained domination approach ensures better non-domination rank to the feasible solutions as compared to the infeasible solutions. The flowchart of NSTLBO algorithm is shown in Fig. 2.3.

2.6 Demonstration of the Working of NSTLBO Algorithm on a Bi-objective Constrained Optimization Problem

Let us consider the example of a bi-objective optimization problem of cutting parameters in turning process. Yang and Natarajan (2010) used differential evolution and non-dominated sorting genetic algorithm-II approachesfor solving the problem. The same problem is considered here to demonstrate the working of the NSTLBO algorithm. The problem has two objectives of minimizing the tool wear (T_w) and maximizing the metal removal rate (M_r). The objective functions, constraints, and the ranges of the cutting parameters are given below.

Objective functions:

$$\text{Minimize } (T_w) = 0.33349\, v^{0.1480} f^{0.4912} d^{0.2898} \tag{2.14}$$

$$\text{Maximize } (M_r) = 1000\, v f d \tag{2.15}$$

Constraints:

Temperature constraint:

$$88.5168\, v^{0.3156} f^{0.2856}\, d^{0.2250} \le 500 \tag{2.16}$$

Surface roughness constraint:

$$18.5167\, v^{-0.0757} f^{0.7593}\, d^{0.1912} \le 2 \tag{2.17}$$

Parameter bounds:

$$\text{Cutting speed (m/min)}: 42 \le v \le 201 \tag{2.18}$$

$$\text{Feed rate (mm/rev)}: 0.05 \le f \le 0.33 \tag{2.19}$$

$$\text{Depth of cut (mm)}: 0.5 \le d \le 2.5 \tag{2.20}$$

v = speed (m/min); f = feed (mm/rev); d = depth of cut (mm); M_r = metal removal rate (mm^3/min); T_w = tool wear (mm); T = tool-workpiece interface temperature ($^\circ$C) and R_a = surface roughness (μm).

Now to demonstrate the NSTLBO algorithm, let us assume a population size of 5 (i.e. number of learners), three design variables v, f, and d (i.e., number of subjects) and one iteration as the termination criterion. The initial population is randomly generated within the ranges of the variables and the corresponding values of the objective functions are shown in Table 2.23. The mean values of v, f, and d are also shown.

$$(\mathrm{Z_T})_{\max} = 0.000;\ (\mathrm{Z_{Ra}})_{\max} = 4.597$$

Z' = overall constraint violation and it is given as (Yang and Natarajan 2010),

$$Z' = \frac{Z_T}{(Z_T)_{\max}} + \frac{Z_{\mathrm{Ra}}}{(Z_{\mathrm{Ra}})_{\max}} \tag{2.21}$$

In Table 2.23, the values under Z_T and Z_{Ra} represent the values by which these constraints are violated by the candidate solution and $(Z_T)_{\max}$ and $(Z_{\mathrm{Ra}})_{\max}$ represent the maximum values of violations of the constraints of tool-workpiece interface temperature and surface roughness so far in the entire iteration. For example, $(Z_{\mathrm{Ra}})_{\max} = 6.597 - 2 = 4.597$. The crowding distance CD is 0. Now the teacher tries to improve the mean result of the class. Assuming random numbers $r_1 = 0.91$ for v and $r_2 = 0.67$ for f and $r_3 = 0.25$ for d and $T_f = 1$, the

$$\text{difference\_mean}_v = 0.91 * (171.541 - 149.90) = 19.693$$

$$\text{difference\_mean}_f = 0.67 * (0.0941 - 0.2387) = -0.09688$$

$$\text{difference\_mean}_d = 0.25 * (1.811 - 1.761) = 0.0125$$

The value of difference_mean is added to the corresponding columns of the variables in Tables 2.23 and 2.24 shows the new values of v, f, and d and the corresponding values of the objective functions and the values of constraints.

Now, the initial solutions of Table 2.23 are combined with the solutions obtained in Table 2.24. The Table 2.25 shows the combined population. The ranks are assigned based on the procedure described in Sects. 2.5.1–2.5.3.

$$(Z_T)_{\max} = 0.000;\ (Z_{\mathrm{Ra}})_{\max} = 4.597$$

Now, the selection is done based on the non-dominance rank and the crowding distance. Table 2.26 shows the selected candidate solutions based on the non-dominance rank and the crowding distance. It may be noted that the value of the crowding distance is 0 in Table 2.25.

$$(Z_T)_{\max} = 0.000;\ (Z_{\mathrm{Ra}})_{\max} = 4.597$$

Now, the learner phase starts and any student can interact with any other student for knowledge transfer. This interaction can be done in a random manner. In this example, interactions between learners 1 and 5, 2 and 4, 3 and 1, 4 and 5, and 5 and

Table 2.23 Initial population

Sr. no.	v	f	d	M_r	T_w	T	R_a	Z_T	Z_{Ra}	Z'	Rank	CD
1	171.541	0.0941	1.811	29233.18	0.2657	261.265	2.335	0	0.335	0.073	1	–
2	186.021	0.3217	0.571	34170.33	0.3519	293.677	4.734	0	2.734	0.595	3	–
3	62.1909	0.318	2.198	43469.2	0.4398	280.527	6.597	0	4.597	1	5	–
4	187.226	0.1859	2.367	82384.18	0.4063	346.486	4.095	0	2.095	0.456	2	–
5	142.545	0.274	1.857	72529.46	0.4401	336.292	5.358	0	3.358	0.730	4	–
Mean	149.90	0.2387	1.761									

Table 2.24 New values of the variables, objective functions, constraints, and violations (teacher phase)

Sr. no.	v	f	d	M_r	T_w	T	R_a	Z_T	Z_{Ra}	Z'
1	191.234	0.05[a]	1.8235	17435.76	0.1983	226.057	1.435	0	0	0
2	201[a]	0.2248	0.5835	26365.33	0.3004	272.99	3.6	0	1.6	0.348
3	81.884	0.2211	2.2105	40020.11	0.3838	276.157	4.908	0	2.908	0.633
4	201[a]	0.08902	2.3795	42576.44	0.2865	287.473	2.331	0	0.331	0.072
5	162.238	0.1771	1.8695	53715.13	0.3628	309.727	3.814	0	1.814	0.395

[a] This value has crossed the given range of the variable and hence it is assigned the bound value

2 are considered. Table 2.27 shows the new values of v, f, and d for the learners after the interactions and considering random numbers $r_1 = 0.81$ for v and $r_2 = 0.79$ for f and $r_3 = 0.56$ for d.

$$(Z_T)_{\max} = 0.000;\ (Z_{\text{Ra}})_{\max} = 4.597$$

Now the solutions obtained in the teacher phase (i.e. Table 2.26) are combined with the solutions obtained in the learner phase (i.e. Table 2.27) and are shown in Table 2.28.

The crowding distance values are calculated as described in Sect. 2.5.2. However, for demonstration purpose, sample steps of calculations of the crowding distance are given below.

Step 1: Sort and rank the population of Table 2.28 based on the non-dominated sorting concept.

Step 2: Collect all rank 1 solutions.

Sr. no.	Objective functions		Rank
	M_r	T_W	
1	17435.76	0.1983	1
6	18069.9	0.1989	1
7	25125	0.2189	1
8	20200.25	0.2147	1

Step 3: Determine the minimum and maximum values of both the objective functions for the entire population from Table 2.28 and these are, $(M_r)_{\min} = 17435.76$; $(M_r)_{\max} = 53715.13$; $(T_w)_{\min} = 0.1983$; and $(T_w)_{\max} = 0.3628$.

Step 4: Consider only the first objective and sort all the values of the first objective function in the ascending order irrespective of the values of the second objective function.

Table 2.25 Combined population (teacher phase)

Sr. no.	v	f	d	M_r	T_w	T	R_a	Z_T	Z_{Ra}	Z'	Rank	CD
1	171.541	0.0941	1.811	29233.18	0.2657	261.265	2.335	0	0.335	0.073	3	–
2	186.021	0.3217	0.571	34170.33	0.3519	293.677	4.734	0	2.734	0.595	7	–
3	62.1909	0.318	2.198	43469.2	0.4398	280.527	6.597	0	4.597	1	10	–
4	187.226	0.1859	2.367	82384.18	0.4063	346.486	4.095	0	2.095	0.456	6	–
5	142.545	0.274	1.857	72529.46	0.4401	336.292	5.358	0	3.358	0.730	9	–
6	191.234	0.05[a]	1.8235	17435.76	0.1983	226.057	1.435	0	0	0	1	–
7	201[a]	0.2248	0.5835	26365.33	0.3004	272.99	3.6	0	1.6	0.348	4	–
8	81.884	0.2211	2.2105	40020.11	0.3838	276.157	4.908	0	2.908	0.633	8	–
9	201[a]	0.08902	2.3795	42576.44	0.2865	287.473	2.331	0	0.331	0.072	2	–
10	162.238	0.1771	1.8695	53715.13	0.3628	309.727	3.814	0	1.814	0.395	5	–

[a] This value has crossed the given range of the variable and hence it is assigned the bound value

Table 2.26 Candidate solutions based on non-dominance rank and crowding distance (teacher phase)

Sr. no.	v	f	d	M_r	T_w	T	R_a	Z_T	Z_{Ra}	Z'	Rank	CD
1	191.234	0.05[a]	1.8235	17435.76	0.1983	226.057	1.435	0	0	0	1	–
2	201[a]	0.08902	2.3795	42576.44	0.2865	287.473	2.331	0	0.331	0.072	2	–
3	171.541	0.0941	1.811	29233.18	0.2657	261.265	2.335	0	0.335	0.073	3	–
4	201[a]	0.2248	0.5835	26365.33	0.3004	272.99	3.6	0	1.6	0.348	4	–
5	162.238	0.1771	1.8695	53715.13	0.3628	309.727	3.814	0	1.814	0.395	5	–

[a] This value has crossed the given range of the variable and hence it is assigned the bound value

Table 2.27 New values of the variables, objective functions, constraints, and violations (learner phase)

Sr. no.	v	f	d	M_r	T_w	T	R_a	Z_T	Z_{Ra}	Z'	Interaction
1	201[a]	0.05[a]	1.798	18069.9	0.1989	228.912	1.426	0	0	0	1 and 5
2	201[a]	0.05[a]	2.5[a]	25125	0.2189	246.534	1.519	0	0	0	2 and 4
3	187.5	0.05926	1.818	20200.25	0.2147	235.665	1.634	0	0	0	3 and 1
4	201[a]	0.2625	0.5[a]	26381.25	0.31	275.604	3.932	0	1.932	0.4203	4 and 5
5	193.635	0.1075	2.155	44857.97	0.3037	293.22	2.647	0	0.647	0.1407	5 and 2

[a] This value has crossed the given range of the variable and hence it is assigned the bound value

Table 2.28 Combined population (learner phase)

Sr. no.	v	f	d	M_r	T_w	T	R_a	Z_T	Z_{Ra}	Z'	Rank	CD
1	191.234	0.05[a]	1.8235	17435.76	0.1983	226.057	1.435	0	0	0	1	∞
2	201[a]	0.08902	2.3795	42576.44	0.2865	287.473	2.331	0	0.331	0.072	2	–
3	171.541	0.0941	1.811	29233.18	0.2657	261.265	2.335	0	0.335	0.073	3	–
4	201[a]	0.2248	0.5835	26365.33	0.3004	272.99	3.6	0	1.6	0.348	5	–
5	162.238	0.1771	1.8695	53715.13	0.3628	309.727	3.814	0	1.814	0.395	6	–
6	201	0.05	1.798	18069.9	0.1989	228.912	1.426	0	0	0	1	0.1759
7	201	0.05	2.5	25125	0.2189	246.534	1.519	0	0	0	1	∞
8	187.5	0.05926	1.818	20200.25	0.2147	235.665	1.634	0	0	0	1	0.3160
9	201	0.2625	0.5	26381.25	0.31	275.604	3.932	0	1.932	0.4203	7	–
10	193.635	0.1075	2.155	44857.97	0.3037	293.22	2.647	0	0.647	0.1407	4	–

[a] This value has crossed the given range of the variable and hence it is assigned the bound value

Sr. no.	Objective functions		Rank
	M_r	T_W	
1	**17435.76**	0.1983	1
6	**18069.9**	0.1989	1
8	**20200.25**	0.2147	1
7	**25125**	0.2189	1

Step 4a: Assign crowding distance as infinity (∞) to the first and last solutions (i.e., the best and the worst solutions).

Sr. no.	Objective functions		Rank	Crowding distance
	M_r	T_W		
1	**17435.76**	0.1983	1	∞
6	**18069.9**	0.1989	1	
8	**20200.25**	0.2147	1	
7	**25125**	0.2189	1	∞

Step 4b: The crowding distances d_6 and d_8 are calculated as follows (please note that 6 is between 1 and 8; and 8 is between 6 and 7).

$$d_6^{(1)} = 0 + \frac{(M_r)_8 - (M_r)_1}{(M_r)_{\max} - (M_r)_{\min}} = 0 + \frac{20200.25 - 17435.76}{53715.13 - 17435.76} = 0.0762$$

$$d_8^{(1)} = 0 + \frac{(M_r)_7 - (M_r)_6}{(M_r)_{\max} - (M_r)_{\min}} = 0 + \frac{25125 - 18069.9}{53715.13 - 17435.76} = 0.19446$$

Step 5: Consider only the second objective and sort all the values of the second objective function in the ascending order irrespective of the values of the first objective function.

Sr. no.	Objective functions		Rank
	M_r	T_W	
1	17435.76	**0.1983**	1
6	18069.9	**0.1989**	1
8	20200.25	**0.2147**	1
7	25125	**0.2189**	1

Step 5a: Assign crowding distance as infinity (∞) to the first and last solutions (i.e. the best and the worst solutions).

Sr. no.	Objective functions		Rank	Crowding distance
	M_r	T$_W$		
1	17435.76	**0.1983**	1	∞
6	18069.9	**0.1989**	1	
8	20200.25	**0.2147**	1	
7	25125	**0.2189**	1	∞

Step 5b: The crowding distances d_6 and d_8 are calculated as,

$$d_6^{(2)} = d_6^{(1)} + \frac{(T_W)_8 - (T_W)_1}{(T_W)_{\max} - (T_W)_{\min}} = 0.0762 + \frac{0.2147 - 0.1983}{0.3628 - 0.1983} = 0.1759$$

$$d_8^{(2)} = d_8^{(1)} + \frac{(T_W)_7 - (T_W)_6}{(T_W)_{\max} - (T_W)_{\min}} = 0.19446 + \frac{0.2189 - 0.1989}{0.3628 - 0.1983} = 0.3160$$

Sr. no.	Objective functions		Rank	Crowding distance
	M_r	T_W		
1	17435.76	0.1983	1	∞
6	18069.9	0.1989	1	0.1759
8	20200.25	0.2147	1	0.3160
7	25125	0.2189	1	∞

Similar procedure can be repeated for calculating the crowding distances for the solutions with ranks of 2, 3, 4, 5, 6, and 7. However, in this example, there are only single solutions with the ranks of 2, 3, 4, etc. Hence, no crowding distance is assigned to them.

Now Table 2.29 shows the candidate solutions based on the non-dominance ranks and the crowding distances.

This completes the learner phase and an iteration of the NSTLBO algorithm.

The website for the TLBO algorithm is https://sites.google.com/site/tlborao. The readers may refer to the website for updates on TLBO algorithm.

The next chapter demonstrates the working of TLBO algorithm on complex composite benchmark functions.

Table 2.29 Candidate solutions based on non-dominance ranks and crowding distances (learner phase)

Sr. no.	v	f	d	M_r	T_w	T	R_a	Z_T	Z_{Ra}	Z'	Rank	CD
1	191.234	0.05[a]	1.8235	17435.76	0.1983	226.057	1.435	0	0	0	1	∞
6	201[a]	0.05	1.798	18069.9	0.1989	228.912	1.426	0	0	0	1	0.1759
8	187.5	0.05926	1.818	20200.25	0.2147	235.665	1.634	0	0	0	1	0.3160
7	201	0.05	2.5	25125	0.2189	246.534	1.519	0	0	0	1	∞
2	201[a]	0.08902	2.3795	42576.44	0.2865	287.473	2.331	0	0.331	0.072	2	–

[a] This value has crossed the given range of the variable and hence it is assigned the bound value

References

Deb, K., 2001. Multiobjective Optimization using Evolutionary Algorithms. New York: John Wiley.

Rao, R.V., 2007. Decision Making in the Manufacturing Environment using Graph Theory and Multiple Attribute Decision Making Methods. London: Springer-Verlag.

Rao, R.V., Patel, V., 2012. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. International Journal of Industrial Engineering Computations 3(4), 535–560.

Rao, R.V., Savsani, V.J., 2012. Mechanical Design Optimization using Advanced Optimization Techniques. London: Springer-Verlag.

Rao, R.V., Savsani, V.J., Vakharia, D.P., 2011. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer-Aided Design 43 (3), 303–315.

Rao, R.V., Savsani, V.J., Vakharia, D.P., 2012a. Teaching-learning-based optimization: A novel optimization method for continuous non-linear large scale problems. Information Sciences 183 (1), 1–15.

Rao, R.V., Savsani, V.J., Balic, J., 2012b. Teaching-learning-based optimization algorithm for unconstrained and constrained real parameter optimization problems. Engineering Optimization 44 (12), 1447–1462.

Yang, S.H., Natarajan, U., 2010. Multiobjective optimization of cutting parameters in turning process using differential evolution and non-dominated sorting genetic algorithm-II approaches. International Journal of Advanced Manufacturing Technology 49, 773–784.