

Optimization of Design Problems Using TLBO and mTLBO Algorithms



A. S. Pendse and Avinash Kamble

Abstract Every design problem requires optimization. There are many traditional and advanced optimization techniques available. Teaching-learning-based optimization algorithm has been proved beneficial in many engineering applications. This algorithm is free from any algorithm-specific parameters and can be adapted to all types of design problems. However, there are some limitations like convergence to local optimal solution, large computational time and slow convergence rate for complex functions. Some modifications were introduced to overcome these drawbacks in mTLBO algorithm. This paper gives better idea of both algorithms by applying them to optimize standard engineering design problems. Their performance was compared with differential evolution algorithm and some hybrid approaches of this algorithm. Best solution was obtained for all the testing problems using mTLBO algorithm. Also the solutions were obtained using less number of evaluations than other algorithms. Therefore, from present work, it can be inferred that mTLBO algorithm gives global optimum solution and requires less computational efforts.

Keywords Optimization · Design problems · TLBO · mTLBO · Differential evolution

1 Introduction

Optimization is the selection of parameters within range to get the best solution possible while satisfying all constraints. It is used in various fields like engineering, medicine, economics, etc. There are many techniques available for optimization. Linear programming, nonlinear programming, quadratic programming, dynamic programming, geometric programming, generalized reduced gradient method, etc. are some of the traditional optimization techniques. These algorithms are commonly

A. S. Pendse (✉) · A. Kamble
School of Mechanical and Civil Engineering, MIT Academy of Engineering, Alandi, Pune
412105, India
e-mail: aspense@mitaoe.ac.in

used for solving simple optimization problems. But, whenever we fail to solve any problem with traditional techniques, we find new ways to solve such problems. Many researchers have developed algorithms based on natural phenomenon. Genetic algorithm (GA) is inspired by Charles Darwin's theory of natural evolution [1]. This algorithm uses the survival of the fittest criterion. Evolution strategies (ESs) also use principles of natural evolution [2]. Some other popular optimization algorithms are simulated annealing (SA), ant colony optimization (ACO), differential evolution (DE), cultural algorithm (CA), particle swarm optimization (PSO), evolutionary programming (EP), etc. [3].

Teaching-learning-based optimization (TLBO) algorithm is based on a relationship between teacher and students. Sample population considered for optimization can be considered as a group of students. Various design variables considered for evaluation are similar to different subjects taught in a class. Teacher in TLBO is analogous to the best solution so far. It works in two phases: teacher phase and learner phase. In mTLBO algorithm, modification was suggested in the learner phase.

In this paper, TLBO algorithm is discussed along with mTLBO. This paper is organized as follows: Sect. 2 gives summary of material and method used. Section 3 provides basic idea of TLBO algorithm. Section 4 explains the modification made in TLBO algorithm. Section 5 states the standard design problems. In Sect. 6, results are compared and discussed. In Sect. 7, paper is concluded.

2 Materials and Methods

Many evolutionary algorithms have been proved good for solving optimization problems. PSO was used to optimize various types of heat exchangers [4]. Artificial bee colony (ABC) algorithm was used to optimize mechanical draft counter flow wet-cooling tower [5]. The performance of these algorithms greatly depends upon their own algorithm-specific parameters. For example, PSO requires tuning of inertia weight and cognitive and social parameters, and GA requires tuning of mutation probability, selection operator, cross-over probability, etc. Also there are other issues like computation time and cost, convergence rate, population or swarm size, etc. which are of much concern. There were several modifications and variants made to overcome these problems.

TLBO algorithm was proposed to overcome parameter dependency of other algorithms [6]. Main advantage of this algorithm is that it is independent of any algorithm-specific parameters. It was successfully used for optimization of various engineering applications like casting process, flat plate solar air heater Stirling engine, etc. Recently, TLBO algorithm was used for thermo-economic analysis and optimization of a solar micro-CCHP [7]. Also TLBO along with DE was used for optimization of critical parameters of PEM fuel cell [8]. However, there are some drawbacks of TLBO algorithm like convergence to local optimal solution, slow convergence rate and large computational time for complex functions. Therefore, some modifications in the original algorithm were suggested in mTLBO algorithm [9]. Along with basic

teaching-learning process, concept of tutorials was incorporated in the algorithm by adding one term in learner phase.

A hybrid approach based on differential evolution and tissue P systems (DETSPS) was used to solve some standard design problems [10]. Modified differential evolution (MDE) algorithm and two-hybrid approaches, local search hybrid (MA-MDE) and harmony search hybrid (MDE-IHS) were also used by a researcher to optimize these problems [11]. TLBO or mTLBO algorithms have not been used to optimize these problems. In this paper, both TLBO and mTLBO algorithms are applied to these problems. The results are compared with DETSPS, MDE, MA-MDE and MDE-IHS algorithms.

3 TLBO

Following are the five steps in working of TLBO algorithm.

3.1 Problem Statement or Function Definition

Mathematical modeling of given problem is required for its optimization. Population size (N) is determined according to complexity of problem, number of design variables (D), constraints (G), etc. Population size and number of design variables are analogous to number of students and number of subjects, respectively.

3.2 Initialization

Population of i rows and j columns is generated randomly using following equation:

$$X_{(i, j)} = X_j^{\min} + \left(X_j^{\max} - X_j^{\min} \right) * \text{rand} \quad (1)$$

where rand represents a uniformly distributed random variable within the range (0, 1), and X^{\min} and X^{\max} represent minimum and maximum value for j th parameter, respectively. Initial solution is obtained by using these values in objective function and constraints.

3.3 Teacher Phase

Mean of each subject is calculated. The learner which gives the least objective function value (for minimization problem) is considered as the teacher for respective iteration. In this phase, mean of learners is shifted toward their teacher. Difference mean is calculated using following equation:

$$\text{Diff_mean_X}_{(i,j)} = r \left(\text{Teacher_X}_j - \text{mean_X}_j \right) \quad (2)$$

This difference mean is added to respective $X_{(i,j)}$, and new function values are obtained. Initial solution and newly obtained solution are compared, and the best function values are selected. This modified solution is the teacher phase solution.

3.4 Learner Phase

In this phase, the learners interact with each other. The process of mutual random interactions tends to improve their knowledge. For a given learner X_i , another learner X_r is randomly selected ($i \neq r$). The i th parameter of the matrix X_{new} in the learner phase is given as

$$\begin{aligned} X_{\text{new}}^g(i) &= X^g(i) + \text{rand} * \left(X^g(i) - X^g(r) \right) \text{ if } X^g(i) < X^g(r) \\ &\text{or} \\ X_{\text{new}}^g(i) &= X^g(i) + \text{rand} * \left(X^g(r) - X^g(i) \right) \text{ if } X^g(i) < X^g(r) \end{aligned} \quad (3)$$

New function values are obtained, and newly obtained solution is compared with teacher phase solution, and the best function values are selected. This modified solution is the learner phase solution.

3.5 Algorithm Termination

After some iteration, final solution is obtained, and algorithm is terminated.

4 mTLBO Algorithm

In this algorithm, teacher phase remains same. Only learner phase is modified. Concept of tutorials given by teacher to students is used to improve the final solution. TLBO may not always give global optimum solution. It sometimes provides local optimum solution. To obtain global solution, an extra term was added to learner phase equation mentioned in Sect. (3.4).

$$\begin{aligned}
 X_{new}^g(i) &= X^g(i) + \text{rand} * \left(X^g(i) - X^g(r) \right) + 0.5 * (1 + \text{rand}) \\
 &* \left(X^g_{\text{Teacher}} - X^g(i) \right) \text{ if } X^g(i) < X^g(r) \\
 \text{or} & \\
 X_{new}^g(i) &= X^g(i) + \text{rand} * \left(X^g(r) - X^g(i) \right) + 0.5 * (1 + \text{rand}) \\
 &* \left(X^g_{\text{Teacher}} - X^g(i) \right) \text{ if } X^g(r) < X^g(i)
 \end{aligned}
 \tag{4}$$

where X^g_{Teacher} is the minimum objective function value after teacher phase.

5 Optimization Problems

Six design problems are chosen for optimization [12]. They contain continuous and binary type variables. Details of the problems are given in Appendix.

6 Results and Discussions

For both TLBO and mTLBO algorithms, population size of 50 is selected in all problems. Mean solution for each problem was obtained after extensive number of runs. Numbers of evaluations were calculated for each run, and average numbers of evaluations are taken into consideration. Standard deviation (SD) was also calculated among the population in all problems. Mean SD was calculated for all problems. Results are obtained for TLBO and mTLBO algorithms as shown in Table 1.

Table 1 Results of TLBO and mTLBO

Problem		TLBO	mTLBO
1	<i>x</i>	[10.8044, 2.0247]	[12.5, 0]
	<i>y</i>	[0.5409, 0]	[0.625, 0]
	<i>f</i>	85.3531	84.6875
2	<i>x</i>	[1.118, 1.3104]	[1.118, 1.3104]
	<i>y</i>	[0, 1, 1]	[0, 1, 1]
	<i>f</i>	7.6672	7.6672
3	<i>x</i>	[0.2, 0.8, 1.908]	[0.246, 1.031, 2.016]
	<i>y</i>	[1, 1, 0, 1]	[0.556, 0.758, 0.392, 0.952]
	<i>f</i>	4.5793	3.1962
4	<i>x</i>	[0.5]	[0.5]
	<i>y</i>	[1]	[1]
	<i>f</i>	2.0000	2.0000
5	<i>x</i>	[1.3749]	[1.37485]
	<i>y</i>	[1]	[1]
	<i>f</i>	2.1246	2.1245
6	<i>x</i>	[0.59, -1.471]	[0.586, -1.471]
	<i>y</i>	[0.4281]	[0.428]
	<i>F</i>	0.5408	0.5374

From Table 1, it can be observed that mTLBO gives the least value of function in all problems, but the difference in solution of both TLBO and mTLBO is small. Table 2 gives the comparison of results obtained by various algorithms.

From Table 2, various observations can be made. Smallest function value is obtained by using mTLBO algorithm in all problems. For problems 2 and 4, TLBO also gives the least function value. Also mTLBO solves the problems in minimum number of evaluations. Standard deviation (SD) seems to be the least in case of DETPS except in problem 05 where MA-MDE has the minimum SD.

Table 2 Statistical result of mean solutions, standard deviations and numbers of evaluations

Problem		DETPS	MDE	MA-MDE	MDE-IHS	TLBO	m-TLBO
1	Mean	87.500012	89.879034	88.230145	87.497550	85.353100	84.845900
	SD	0.000002	2.768746	1.899683	0.002118	0.006850	0.003150
	Evaluations	14360	7777	4436	5359	3215	2000
2	Mean	7.931112	7.918619	7.883841	7.848896	7.667200	7.667200
	SD	0.000000	0.047891	0.098982	0.121909	0.006460	0.024130
	Evaluations	100000	96718	93524	83442	8653	5088
3	Mean	4.579593	4.661414	4.579595	4.579599	4.579300	3.1962
	SD	0.000000	0.311365	0.000003	0.000005	0.001500	0.000210
	Evaluations	7254	7688	13023	14518	6797	3586
4	Mean	2.000000	2.009348	2.000000	2.000001	2.000000	2.000000
	SD	0.000000	0.043579	0.000000	0.000000	0.000219	0.000001
	Evaluations	1720	1075	1430	3297	1000	1035
5	Mean	2.182346	2.167894	2.124574	2.124604	2.124600	2.124500
	SD	0.149789	0.132196	0.000071	0.000076	0.128315	0.099315
	Evaluations	1726	827	653	1409	584	320
6	Mean	1.076700	1.124453	1.099805	1.094994	0.540800	0.537400
	SD	0.000391	0.075163	0.055618	0.052898	0.015325	0.008472
	Evaluations	5309	30986	25766	22146	2700	2500

7 Conclusions

Following conclusions are drawn from the results obtained:

- i. Modified teaching-learning-based optimization algorithm (mTLBO) gives the best solution in all problems satisfying all constraints.
- ii. Compared with other algorithms, mTLBO has faster convergence rate.
- iii. Teaching-learning-based optimization (TLBO) algorithm gives better results than all algorithms except mTLBO.
- iv. Both TLBO and mTLBO give better solution in different types of problems and hence can be used in many engineering applications.

Appendix

Problem 01

$$\min f = 7.5y_1 + 6.4x_1 + 5.5y_2 + 6.0x_2$$

Subject to:

$$0.8x_1 + 0.67x_2 = 10$$

$$x_1 - 20y_1 \leq 0$$

$$x_2 - 20y_1 \leq 0$$

$$x_1, x_2 \in [0, 20]$$

$$y_1, y_2 \in \{0, 1\}$$

Problem 02

$$\min f = 2x_1 + 3x_2 + 1.5y_1 + 2y_2 - 0.5y_3$$

Subject to:

$$x_1^2 + y_1 = 1.25$$

$$x_2^{1.5} + 1.5y_2 = 3$$

$$x_1 + y_1 \leq 1.6$$

$$1.333x_2 + y_2 \leq 3$$

$$-y_1 - y_2 + y_3 \leq 0$$

$$x_1, x_2 \in [0, 2]$$

$$y_1, y_2, y_3 \in \{0, 1\}$$

Problem 03

$$\begin{aligned} \min f = & (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (y_1 - 1)^2 \\ & + (y_2 - 2)^2 + (y_3 - 1)^2 - \log_{10}(y_4 + 1) \end{aligned}$$

Subject to:

$$x_1 + x_2 + x_3 + y_1 + y_2 + y_3 = 5$$

$$x_1^2 + x_2^2 + x_3^2 + y_3^2 = 5.5$$

$$x_1 + y_1 \leq 1.2$$

$$x_2 + y_2 \leq 1.8$$

$$x_3 + y_3 \leq 2.5$$

$$x_1 + y_4 \leq 1.2$$

$$x_2^2 + y_2^2 \leq 1.64$$

$$x_3^2 + y_3^2 \leq 4.25$$

$$x_3^2 + y_2^2 \leq 4.64$$

$$x_1 \in [0, 1.2]$$

$$x_2 \in [0, 1.281]$$

$$x_3 \in [0, 2.062]$$

$$y_1, y_2, y_3, y_4 \in \{0, 1\}$$

Problem 04

$$\min f = 2x + y$$

Subject to:

$$1.25 - x^2 - y \leq 0$$

$$x + y \leq 1.6$$

$$x \in [0, 1.6]$$

$$y \in \{0, 1\}$$

Problem 05

$$\min f = -y + 2x - \ln(x/2)$$

Subject to:

$$-x - \ln(x/2) + y \leq 0$$

$$x \in [0.5, 1.4]$$

$$y \in \{0, 1\}$$

Problem 06

$$\min f = -0.7y + 5(x_1 - 0.5)^2 + 0.8$$

Subject to:

$$-\exp(x_1 - 0.2) - x_2 = 0$$

$$x_2 + 1.1y \leq -1$$

$$x_1 - 1.2y_1 \leq 0.2$$

$$x_1 \in [0.2, 1]$$

$$x_2 \in [-2.22554, -1]$$

$$y \in \{0, 1\}$$

References

1. Holland JH (1975) An introductory analysis with applications to biology, control, and artificial intelligence. In JH Holland (ed) *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, p 183
2. Bäck T (1991) A survey of evolution strategies. In: 4th international conference on genetic algorithms. Morgan Kaufmann, San Diego, pp 2–9
3. Pratihari DK (2012) *Computational optimization and applications*. Narosa Publishing House Pvt. Ltd, New Delhi
4. Rao RV, Patel VK (2011) Optimization of mechanical draft counter flow wet-cooling tower. *Energy Convers Manage* 52(7):2611–2622
5. Rao RV, Patel VK (2010) Thermodynamic optimization of cross flow plate-fin heat exchanger using a particle swarm optimization algorithm. *Int J Thermal Sci* 49(9):1712–1721
6. Rao RV, Sivasani VJ, Vakharia DP (2011) Teaching-learning based optimization: a novel method for constrained mechanical design optimization problems. *Comput-Aided Design* 43:303–315
7. Azizimehr, B, Assareh E, Moltames R (2020) Thermoeconomic analysis and optimization of a solar micro CCHP by using TLBO algorithm for domestic application. In: *Energy sources, part a: recovery, utilization, and environmental effects* 1747–1761
8. Guo C, Lu J, Tian Z, Guo W, Darvishan A (2019) Optimization of critical parameters of PEM fuel cell using TLBO-DE based on Elman neural network. *Energy Convers Manage* 149–158
9. Satapathy S, Naik A (2013) A modified teaching-learning-based optimization (mTLBO) for global search. *Recent Paten Comput Sci* 13(6):60–72
10. Zhang H, Cheng J, Gheorghe M, Meng Q (2013) A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems. *Appl Soft Comput* 13(3):1528–1542
11. Liao T (2010) Two hybrid differential evolution algorithms for engineering design optimization. *Appl Soft Comput* 10(4):1188–1199
12. Mathur M, Karale S, Priye S, Jayaraman V, Kulkarni B (2000) Ant colony approach to continuous function optimization. *Ind Eng Chem Res* 39(10):3814–3822