Use of Teaching Learning based optimization algorithm for testing Benchmark functions sequentially

Ms. Shah Sujeeta R.¹
Computer Science and Engineering
Bharati Vidyapeeth College Of Engineering, Kolhapur, India

Prof. Takmare Sachin B.²
Computer Science and Engineering
Bharati Vidyapeeth College Of Engineering, Kolhapur, India

ABSTRACT—Inspiring by the phenomenon of teaching and learning introduced by teaching–learning based optimization algorithm, this paper presents testing of different benchmark functions sequentially by using TLBO algorithm. There are two types of objective functions constrained and unconstrained optimization. Constrained optimization is a type where the function to be optimized with respect to some variable having some constraints on that variable. Unconstrained optimization is a type where there is no restriction on the range of variables. The Teaching Learning Based Optimization (TLBO) algorithm is one of the important algorithms which are based on effect of teacher on outputs of learner class. The performance of algorithm on unconstrained and constrained benchmark functions is tested by running these algorithms sequentially.

Keywords—Central processing unit, Global optimization, Test functions, Teaching-learning-based optimization, constrained benchmark function and unconstrained benchmark functions.

1. INTRODUCTION

Engineering design optimization is a high reaching directing process towards certain objectives. The objectives, feasible solution space, and the searching method are important factors. Earlier analytical or numerical methods were used to solve design optimization problems. The analytical methods fail to solve complex design problem due to a large number of design variables and non-linear character.

TLBO is a population based method that uses a population of solutions to proceed to the global solution. For TLBO, the population is considered as a group of learners or a class of learners. In optimization algorithms, the population consists of different design variables. In TLBO, different design variables is analogous to different subjects offered to learners and the learners’ result is analogous to the ‘fitness’, as in other population based optimization techniques [2]. Actual research development challenge to develop a nature-based algorithm is its...
capacity to solve different optimization problems effectively and efficiently. It is assumed that the behavior of nature is always optimum in its performance.

Overview of solution is the performance of algorithm on unconstrained and constrained benchmark functions is tested by running these algorithms sequentially. The test of reliability, efficiency and validation of optimization algorithms is frequently carried out by using a chosen set of common standard benchmarks or test functions that is Ackley, Beale, Booth, Branin and Goldstein & Price Functions.

2. RELATED WORK

Domingo Ortiz-Boyer et al. in [1] describes crossover operator for evolutionary algorithm with real values that is based on the statistical theory of population distributions.


Momin Jami et al. in [3] give the literature survey for which test functions gets the optimum performance using the different algorithms.


Weishi Shao et al. [5] this paper presents a hybrid optimization algorithm based on teaching probabilistic learning mechanism(ADTPL) to solve No-Wait flow shop scheduling(NWFSSP) with minimization of makespan.

Atul B. patil [6] The main purpose of this paper is to overview of application solved during TLBO algorithm and findings of remarkable point to improve the solution.

KondreddyNageswara Reddy [7] the author tells TLBO is developed random population optimization technique for solving scheduling problems.

Shah S. R. [8] the author implements benchmark function by using Teaching-learning based optimization algorithm based on CUDA (Compute unified device architecture) and GPGPU (General purpose graphics processing unit).The author reviews the methodologies used for testing benchmark functions using Teaching-learning based optimization algorithm.

Shah S.R. [9] the author reviews the methodologies used for testing benchmark functions using Teaching-learning based optimization algorithm and different application where we can use TLBO algorithm.

3. SYSTEM ARCHITECTURE

3.1 System block diagram

![Proposed system block diagram](image)
In Init kernel step, the initialization of design vector and population size, then mean of each design vector has been calculated; Teacher phase is for improving the mean of each class. In student phase kernel, randomly picks up any two of the students and improves their results. Finally, Benchmark function is evaluated with new parameters up to global optima occurs; the design vector value of best fitness value is the best solution.

4. ALGORITHM

Teaching-Learning-Based Optimization Algorithm

Working of TLBO is as follows

1. Initialize the population size, the number of generations, the number of design variables.
2. Teacher phase:
   a. Calculate mean of each design variable.
   b. Identify the best solution (teacher).
   c. Calculate difference between existing and new mean as
      \[ \text{Diff } _\text{Mean}_i = r_i (M_{new} - TF.M_i) \]
      Where, TF is teaching factor selected random, either 1 or 2.
      \[ M_i \] is mean of the current iteration.
      d. Modify solution based on the best solution.
      \[ x_{new} = x_{old} + \text{Diff } _\text{Mean}_i \]
      Where \[ x_{old} \] is an older value of design variable and \[ x_{new} \] is modified value.
3. Learner phase:
   a. Select randomly two solutions \[ x_i \] and \[ x_j \].
      If \[ f(x_i) < f(x_j) \]
      \[ x_{new} = x_{old} + r_i (x_i - x_j) \]
      else
      \[ x_{new} = x_{old} + r_i (x_j - x_i) \]
4. If a termination criterion is satisfied, then stop.
5. Find the value of the solution.

The performance of TLBO will be tested with the parameters like population size, time efficiency, CPU utilization, memory utilization.

5. METHODOLOGY

5.1. Teaching-Learning-Based Optimization Algorithm:

TLBO is a population-based method. This method uses a population of solutions to proceed to the global solution. The population has been considered as a group of learners or a class of learners. In TLBO, different design variables will be similar to different subjects assigned to learners and the learners result is analogous to the fitness, as in other population-based optimization techniques. The TLBO process has been divided into two parts.
The first part consists of the “Teacher Phase” and the second part consists of the “Learner Phase”. The teacher is considered as the best solution obtained so far. The “Teacher Phase” means learning from the teacher and the “Learner Phase” means learning through the interaction between learners [2].

5.2. Benchmark Functions:

In general, unconstrained problems can be classified into two categories: test functions and real-world problems. Test functions are artificial problems, and can be used to evaluate the behavior of an algorithm in sometimes diverse and difficult situations. On the other hand, real-world problems originate from different fields such as physics, chemistry, engineering, mathematics, etc. These problems are hard to manipulate and may contain complicated algebraic or deferential expressions and may require a significant amount of data to compile. A collection of real-world unstrained optimization problems will be found [3]. There are some unconstrained functions like Ackley, Beale, Branin, Booth, Goldstein and G1-G5 are constrained functions.

Benchmark functions can be classified in terms of features like modality, basins, valleys, separability and dimensionality.

5.2.1. Modality:

The number of ambiguous peaks in the function landscape corresponds to the modality of a function. If algorithms encounter these peaks during a search process, there is a tendency that the algorithm may be trapped in one of the peaks. This will have a negative impact on the search process, as this can direct the search away from the true optimal solutions.

5.2.2. Basins:

A relatively steep decline surrounding a large area is called a basin. Optimisation algorithms can be easily attracted to such regions.

5.2.3. Valleys:

A valley occurs when a narrow area of little change is surrounded by regions of steep descent. As like the basins, minimisers are initially attracted to this region. The progress of a search process of an algorithm may be slowed down accordingly on the floor of the valley.

5.2.4. Separability:

The separability is a measure of difficulty of different benchmark functions. In general, separable functions are easy to solve, when compared with their inseparable counterpart, because each variable of a function is independent of the other variables. If all the parameters or variables are independent, then a sequence of n independent optimisation processes can be performed. As a result, each design variable or parameter can be optimised independently [3].

Following equation shows that standard unconstrained benchmark function [9]-

1. Ackley Function

□ Definition:
\[ f(x_0 \cdots x_n) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e \]

- **Search domain:** \(-15 \leq x_i \leq 30, i = 1, 2, \ldots, n.
- **The global minimum:** \(x^* = (0, \ldots, 0), f(x^*) = 0.\)

### 2. Beale Function

- **Definition:**
- **Search domain:** \(-4.5 \leq x_i \leq 4.5, i = 1, 2.\)
- **The global minimum:** \(x^* = (3, 0.5), f(x^*) = 0.\)

### 3. Booth Function

- **Definition:**
- **Search domain:** \(-10 \leq x_i \leq 10, i = 1, 2.\)
- **The global minimum:** \(x^* = (1, 3), f(x^*) = 0.\)

### 4. Branin Function

- **Definition:**
- **Search domain:** \(-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15.\)
- **The global minima:** \(x^* = (-\pi, 12.275), (\pi, 2.275), (9.42478, 2.475), f(x^*) = .397887.\)

### 5. Goldstein-Price Function

- **Definition:**
- **Search domain:** \(-2 \leq x_i \leq 2, i = 1, 2.\)
- **The global minima:** \(x^* = (0, 1), f(x^*) = 3.\)

Following equation shows that standard constrained benchmark function [9]-

### 1. G1 Function
Search Space: $0 \leq x_i \leq u_i$, $i = 1, 2, \ldots, n$,

$$u = (1, 1, \ldots, 1, 100, 100, 100, 1)$$

The global minima: $x^* = (1, 1, \ldots, 1, 3, 3, 3, 1)$, $f(x^*) = -15$

**2.G2 Function**

$$\begin{align*}
\max f(x) & = \left| \sum_{i=1}^{n} \cos^4(x_i) - 2 \prod_{i=1}^{n} \cos^2(x_i) \right|, \\
\text{s.t. } g_1(x) & = -\prod_{i=1}^{n} x_i + 0.75 \leq 0, \\
g_2(x) & = \sum_{i=1}^{n} x_i - 7.5n \leq 0.
\end{align*}$$

Search Space: $0 \leq x_i \leq 10$, $i = 1, 2, \ldots, n$.

Best known: at $n = 20$, $f(x^*) = 0.803619$.

**3.G3 Function**

$$\begin{align*}
\max f(x) & = (\sqrt{n})^n \prod_{i=1}^{n} x_i, \\
\text{s.t. } h_1(x) & = \sum_{i=1}^{n} x_i^2 - 1 = 0.
\end{align*}$$

Search Space: $0 \leq x_i \leq 1$, $i = 1, 2, \ldots, n$.

The global minima: $x^* = (1/n^{0.5}, \ldots, 1/n^{0.5}), f(x^*) = 1$

**4. G4 Function**

$$\begin{align*}
\min f(x) & = 5.3578547x_1^3 + 0.8356891x_1x_2x_3 + 37.293239x_1 - 40792.141, \\
\text{s.t. } g_1(x) & = u(x) - 92 \leq 0, \\
g_2(x) & = -u(x) \leq 0, \\
g_3(x) & = v(x) - 110 \leq 0, \\
g_4(x) & = -v(x) + 90 \leq 0, \\
g_5(x) & = w(x) - 25 \leq 0, \\
g_6(x) & = -w(x) + 20 \leq 0,
\end{align*}$$

where

$$\begin{align*}
u(x) & = 85.334407 + 0.0056858x_2x_5 + 0.00062632x_1x_4 - 0.00220532x_3x_5, \\
v(x) & = 80.51249 + 0.0071317x_2x_3 + 0.0029955x_1x_2 + 0.0021813x_3^2, \\
w(x) & = 9.30961 + 0.0047262x_3x_5 + 0.0012547x_2x_3 + 0.0019085x_3x_4.
\end{align*}$$
The global minima: $x^* = (78.33,29.995,45.36.7758)$, $f(x^*) = -30665.539$.

### 5. G5 Function

$$
\begin{align*}
\min_{\mathbf{x}} f(\mathbf{x}) &= 3x_1 + 10^{-6}x_1^3 + 2x_2 + \frac{2}{3} \times 10^{-6}x_2^3, \\
\text{s.t. } g_1(x) &= x_3 - x_4 - 0.55 \leq 0, \\
g_2(x) &= x_4 - x_3 - 0.55 \leq 0, \\
h_1(x) &= 1000 [\sin(-x_3 - 0.25) + \sin(-x_4 - 0.25)] + 894.8 - x_1 = 0, \\
h_2(x) &= 1000 [\sin(x_3 - 0.25) + \sin(x_3 - x_4 - 0.25)] + 894.8 - x_2 = 0, \\
h_3(x) &= 1000 [\sin(x_4 - 0.25) + \sin(x_4 - x_3 - 0.25)] + 1294.8 = 0.
\end{align*}
$$

Search Space: $l_i \leq x_i \leq u_i$, $i = 1, \ldots, 5$, $l=(78,33,27,27,27)$, $u=(102,45,45,45,45)$. 

Best known: $x^*=(679.9453,1026,0.118876,-0.3962336)$, $f(x^*) = 5126.4981$.

### 6. EXPERIMENTAL RESULTS

Following table shows that required time and minimum optimized value for particular unconstrained benchmark function and different number of generations, it is implemented sequentially.

<table>
<thead>
<tr>
<th>Sr.No</th>
<th>Function name</th>
<th>Number of Generations</th>
<th>Time</th>
<th>Optimized Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Ackley</td>
<td>10</td>
<td>2.028000 s</td>
<td>5.721760</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>5.772000 s</td>
<td>7.575433</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>24.554000 s</td>
<td>5.389569</td>
</tr>
<tr>
<td>2.</td>
<td>Beale</td>
<td>10</td>
<td>1.966000 s</td>
<td>1.123394</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>5.897000 s</td>
<td>0.127556</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>21.403000 s</td>
<td>0.254852</td>
</tr>
<tr>
<td>3.</td>
<td>Booth</td>
<td>10</td>
<td>1.778000 s</td>
<td>1.568115</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>6.022000 s</td>
<td>6.747962</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>21.731000 s</td>
<td>0.682581</td>
</tr>
<tr>
<td>4.</td>
<td>Branin</td>
<td>10</td>
<td>1.825000 s</td>
<td>3.759225</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>5.148000 s</td>
<td>0.518599</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>20.32600 s</td>
<td>0.401906</td>
</tr>
<tr>
<td>5.</td>
<td>Goldstein &amp;Price</td>
<td>10</td>
<td>2.543000 s</td>
<td>38.155398</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>10.99800 s</td>
<td>11.302861</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>48.78100 s</td>
<td>32.752505</td>
</tr>
</tbody>
</table>

TABLE1: Number of generations, time and optimized value for unconstrained benchmark functions sequentially.
Following table shows that required time and maximum optimized value for particular constrained benchmark function and different number of generations, it is implemented sequentially.

<table>
<thead>
<tr>
<th>S.N</th>
<th>Function name</th>
<th>Number of Generations</th>
<th>Time</th>
<th>Optimized Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>G1</td>
<td>10</td>
<td>2.714000 s</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>14.13400 s</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>67.11100 s</td>
<td>0.00000</td>
</tr>
<tr>
<td>2.</td>
<td>G2</td>
<td>10</td>
<td>4.383000 s</td>
<td>0.148637</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>28.76600 s</td>
<td>0.161281</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>139.3860 s</td>
<td>0.132467</td>
</tr>
<tr>
<td>3.</td>
<td>G3</td>
<td>10</td>
<td>3.167000 s</td>
<td>916.335079</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>19.0010 s</td>
<td>3708.995636</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>89.9030 s</td>
<td>1649.379845</td>
</tr>
<tr>
<td>4.</td>
<td>G4</td>
<td>10</td>
<td>2.636000 s</td>
<td>22500.6454</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>10.7000 s</td>
<td>23640.50460</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>39.5460 s</td>
<td>18846.15834</td>
</tr>
<tr>
<td>5.</td>
<td>G5</td>
<td>10</td>
<td>2.605000 s</td>
<td>7291.711331</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>9.610000 s</td>
<td>7561.392353</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>33.13400 s</td>
<td>7753.061306</td>
</tr>
</tbody>
</table>

TABLE 2: Number of generations, time and optimized value for constrained benchmark function sequentially.

7. CONCLUSION

By implementing Sequential Teaching–Learning-Based Optimization (TLBO) algorithm, we obtain global solutions with less computational effort and high consistency and to solve different unconstrained benchmark functions and different constrained benchmark functions and analyze its performance. The performance of TLBO is tested with the parameters like population size, time efficiency, CPU utilization, memory utilization.

The traditional approach is sequential with requiring more time to optimize benchmark functions or mechanical problems. So the future work is the implementation of parallel Teaching–Learning-Based Optimization (TLBO) algorithm for obtaining the global solutions for continuous non-linear functions with great computational effort than the sequential one.

8. ACKNOWLEDGMENTS

There have been many contributors for this to take shape and the authors are thankful to each of them. We specifically would like to thank Prof. Mrs. Mulla S. M. (Head of computer science and technology(Bharati Vidyapeeth’s College of Engineering, Kolhapur )) and Prof. Takmare S.B.
REFERENCES


