Applied Soft Computing 46 (2016) 115–127

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

On the convergence and origin bias of the Teaching-Learning-Based-Optimization algorithm

J.K. Pickard [a,*], J.A. Carretero [a], V.C. Bhavsar [b]

[a] Department of Mechanical Engineering, University of New Brunswick, Fredericton, New Brunswick, Canada
[b] Faculty of Computer Science, University of New Brunswick, Fredericton, New Brunswick, Canada

ARTICLE INFO

Article history:
Received 22 June 2015
Received in revised form 19 April 2016
Accepted 20 April 2016
Available online 27 April 2016

Keywords:
Teaching-Learning-Based-Optimization
Bias
Origin
Convergence
Evolutionary computing

ABSTRACT

Teaching-Learning-Based-Optimization (TLBO) is a population-based Evolutionary Algorithm which uses an analogy of the influence of a teacher on the output of learners in a class. TLBO has been reported to obtain very good results for many constrained and unconstrained benchmark functions and engineering problems. The choice for TLBO by many researchers is partially based on the study of TLBO's performance on standard benchmark functions. In this paper, we explore the performance on several of these benchmark functions, which reveals an inherent origin bias within the Teacher Phase of TLBO. This previously unexplored origin bias allows the TLBO algorithm to more easily solve benchmark functions with higher success rates when the objective function has its optimal solution as the origin. The performance on such problems must be studied to understand the performance effects of the origin bias. A geometric interpretation is applied to the Teaching and Learning Phases of TLBO. From this interpretation, the spatial convergence of the population is described, where it is shown that the origin bias is directly tied to spatial convergence of the population. The origin bias is then explored by examining the performance effect due to: the origin location within the objective function, and the rate of convergence. It is concluded that, although the algorithm is successful in many engineering problems, TLBO does indeed have an origin bias affecting the population convergence and success rates of objective functions with origin solutions. This paper aims to inform researchers using TLBO of the performance effects of the origin bias and the importance of discussing its effects when evaluating TLBO.

1. Introduction

A new technique has risen in the field of global optimization called Teaching-Learning-Based-Optimization (TLBO) [8,9]. The method is based on the philosophy of teaching and learning and mimics the influence of a teacher on the output of learners. Unlike many other global optimization algorithms, TLBO does not require user-defined algorithm-specific control parameters (*e.g.*, Genetic Algorithms use a mutation rate, crossover rate and a few others). Instead, these control parameters are probabilistically determined during runtime and therefore tuning is not required in order to improve performance. TLBO does however still require common control parameters such as population size and number of generations; thus, it has been referred to as an algorithm-specific parameter-less method [7]. Proper tuning of the common control parameters is still required to improve performance; however, the burden of tuning these control parameters is comparatively less, making TLBO a very desirable optimization method.

TLBO has been shown to attain comparatively good results on many constrained and unconstrained benchmark functions [1,9,16], multi-objective benchmark functions [11], and engineering problems [8,10,12,13]. It is shown to typically outperform many global optimization algorithms, solving the same benchmark problems with fewer function evaluations, better solution, and higher success rate [5–12]; thus TLBO has been shown to be an effective global optimization method. An interesting and insightful story unfolds around TLBO, as discussed below, which shows the repercussions that inexact experiment replications can have on the conclusion of an algorithm's performance, and alludes to the importance of being critical when evaluating an algorithm.

Rao et al. [9] originally conducted experiments on five sets of non-linear unconstrained benchmark functions to evaluate TLBO's comparative performance with other Evolutionary Algorithms

* Corresponding author.
E-mail addresses: joshua.pickard@unb.ca (J.K. Pickard), juan.carretero@unb.ca (J.A. Carretero), bhavsar@unb.ca (V.C. Bhavsar).

http://dx.doi.org/10.1016/j.asoc.2016.04.029
1568-4946/© 2016 Published by Elsevier B.V.

(EAs). The authors state that "*the results show better performance of* [the] *TLBO method over other nature-inspired optimization methods for the considered benchmark functions*". In the note by Črepinšek et al. [1], the authors reproduce these experiments using prototypical TLBO implementations from [8,9]. During their tests they reveal several issues and misconceptions that appeared in the original papers [8,9]. Most notably, that the five experiments conducted with TLBO in [9] were not performed using similar experimental settings for direct comparison to the results published using other EAs. In response to this note, Waghmare [16] repeated these experiments using the TLBO code from the appendix of [6] and argued that their work presents "*comparatively better results*" to that of [1]. Unfortunately, experiment 1 by Waghmare [16] is completed using a population size of 10, as stated in their Table 8, whereas Rao et al. [9] and Črepinšek et al. [1] (from their publicly available code) had used a population size of 20. Different results are to be expected, thus experiment 1 in [16] is somewhat irrelevant for comparison purposes. In addition to this, Črepinšek et al. [2] reiterate that duplicate removal was used for all unconstrained problems in [1], as was the case in [9], whereas [16] did not apply the technique. As well, Črepinšek et al. [2] reveal that the number of fitness evaluations using the publicly available code in [6] is approximated post-run and is not exact. Despite these differences, Črepinšek et al. [2] provide statistical evidence that the "*results are mostly insignificantly different, despite Waghmare's claim of significance*". During our replication of these experiments, we had much difficulty trying to reproduce the results of [16] using the same code obtained from [6]; although, we did modify the code to accommodate a maximum number of fitness evaluations, which may explain the difference in performance. In fact, our results were actually much closer to those in [1].

1.1. Motivation

Being critical of an algorithm is very important when deciding which EA to apply to a problem, as certain algorithms may perform much better on certain types of problems. As well, it can sometimes be difficult to find negative results in the literature, as most researchers tend to focus on benefits rather than drawbacks of a particular algorithm. For the intent of being critical, it is necessary to gain a deeper understanding of how an algorithm works. To better understand how TLBO works, a geometric interpretation is applied in this work in order to explain the impact of the *Teaching Phase* and *Learning Phase* on population convergence. Upon studying the convergence of TLBO, the geometric interpretation reveals a previously unexplored property of TLBO which introduces a bias allowing for quicker convergence and higher success rates when an objective function has an optimal solution at the origin, $\boldsymbol{x}^* = \mathbf{0}$ where $\boldsymbol{x} \in \mathbb{R}^d$ and d is the dimension of the problem.

This bias, while not debilitating (noting the vast amounts of literature which successfully apply TLBO to real engineering problems), is not accounted for in the conclusions of papers experimenting on the affected problems. In fact, the only previous mention of such a bias is made in the recent book [14], where the author briefly notes that TLBO may give an unfair advantage on problems whose solution is at the origin. Several researchers have chosen to apply TLBO to new problems following an initial test on standard benchmark functions, some affected by the origin bias, and noting that the results are exceptional, such as in [17], while others reference the original works in [9] stating that TLBO involves comparatively less computational effort, such as in [11].

This paper offers unbiased data-based evidence that conclusions for using TLBO are unjust without accounting for the origin bias within TLBO. The goal of this paper is to provide an understanding on the impact of the origin bias through geometric explanations, code review of [1,6], and experiments. The source of the origin bias is described by studying the convergence of TLBO, where the exploratory portion of the algorithm is shown to cause the bias. A detailed study considering the rate of convergence is presented which offers an explanation of the relationship between convergence and origin bias. As well, two modifications to the TLBO algorithm are provided which can improve the results by either removing or shifting the bias of the algorithm. These modifications achieve the expected results, whereby the performance on many problems is improved at the expense of additional function evaluations for problems with origin solutions, with respect to the original TLBO algorithm.

The rest of the paper is organized as follows. In Section 2 an overview of the basic TLBO algorithm with accepted improvements is presented. A geometric understanding applied to the *Teaching phase* and *Learning phase* is presented in Section 3. In Section 4 TLBO convergence is discussed and the origin bias is explored. Modifications are made to the TLBO algorithm in Section 5 to remove or shift the bias and experimental results are provided for a large set of benchmark functions. Finally, Section 6 concludes the paper.

2. Teaching-Learning-Based-Optimization

2.1. Analogy

TLBO uses an analogy which is based on the effect of the influence of a teacher on the output of learners in a class [9]. The job of a teacher is to teach the students such that each student learns and is therefore able to improve their grades. A good teacher is able to effectively improve the mean grade of the class. Rao et al. [8] describe this transfer of knowledge as the movement of a normally distributed set of grades. Outside of class, students also interact and are able to exchange knowledge. As a result of these interactions, the mean grade of the class is typically expected to improve.

The analogy of subjects and grades is applied to the understanding of TLBO. Students are enrolled in a number of different subjects, equal to the number of design variables, d, associated with the problem. For each subject, the student earns a score, x, which represents the value of a design variable. Depending on the combined scores that a student receives in all of their subjects, $\boldsymbol{x}$, they are assigned an overall grade describing how well they have performed in comparison to the other students. Grades are computed by evaluating the objective function using student scores for each subject, *i.e.*, $f(\boldsymbol{x})$. Depending on whether the objective function is being minimized or maximized, lower grades or higher grades represent better solutions respectively.

The basic TLBO algorithm uses two distinct phases to perturb the population towards better solutions. Subject scores are initialized, such that the scores represent a uniformly distributed set of data within the search space. Each student is evaluated to determine their respective grade. A teacher is selected as the student with the best grade and is then used to teach the remaining students. This is accomplished in the *Teacher Phase*. Students are then randomly paired and attempt to exchange knowledge with one another. This is accomplished in the *Learner Phase*. These phases are expanded on in the following subsections.

2.2. Teacher Phase

In the Teacher Phase, a teacher must be selected for each generation. This is accomplished by locating the best performing student (*i.e.*, the student with the lowest grade for a minimization problem). The teacher is then used to "teach" the remaining students in an attempt to improve the mean grade. The following equation