Single Phase Multi-Group Teaching Learning Algorithm for Single Objective Real-Parameter Numerical Optimization (CEC2016)

Remya Kommadath
Department of Chemical Engineering
Indian Institute of Technology
Guwahati, India

Chinta Sivadurgaprasad
Department of Chemical Engineering
Indian Institute of Technology
Guwahati, India

Prakash Kotecha
Department of Chemical Engineering
Indian Institute of Technology
Guwahati, India
Email: pkotecha@iitg.ernet.in

***Abstract*— In this work, the performance of a variant of Teaching Learning Based Optimization algorithm, Single Phase Multi–Group Teaching Learning Optimization is evaluated on the basis of its performance on single objective real-parameter numerical optimization problems. These problems have been provided as part of one of the competition in IEEE Congress on Evolutionary Computation 2016. It was observed that the performance of the algorithm is competitive to other optimization techniques and satisfactory for many problems.**

Keywords—Single Phase Multi-Group Teaching Learning Optimization; CEC 2014; single objective optimization; stochastic optimization

I. Introduction

Evolutionary optimization techniques have become increasingly popular and have found applications in diverse fields. As most of these techniques do not require any analytical information of the functions, these are well suited for solving optimization problems which cannot be conveniently modelled in terms of traditional equalities and inequalities. Hence these techniques are the primary choice for a large number of problems in which the cost function is determined as a result of complex simulations or physical experiments. Though these techniques do not provide any guarantee on the determination of global optimal solutions even for well-structured problems, these are capable of handling a wide variety of complex problems such as multimodal, and multi objective optimization problems. Additionally many of these techniques provide a wide range of solutions which can help in informed decision making. However unlike mathematical programming techniques, these techniques need to be appropriately modified to solve complex combinatorial optimization problems. Another drawback of many of these techniques is that they require a large computational time. However this has been largely overcome with the significant advancements in computing technology including parallel computing.

A large number of evolutionary algorithms are proposed every year which claim to mimic some kind of natural phenomenon [1-3]. As these techniques are relatively simple, a large number of variants are also developed in quick succession. However very often conflicting claims are reported in literature [4, 5] on the performance of the techniques and its variants. In many instances, the performance of a new technique is demonstrated on problems which are no longer considered as difficult optimization problems and are selectively compared against techniques which perform poorly or are on par with the proposed technique [6]. Over the past few years, there have been regular competitions in prestigious conferences which provide a common platform to evaluate the performance of various algorithms. However many of these competitions require that the competing technique be novel and thus do not necessarily enable competitive performance evaluation of recently proposed techniques. Nevertheless, these competitions enable a fair comparison of the variants of the proposed algorithms particularly since the best ranking algorithms are required to share their codes. In most cases, the variants provide a reasonable idea on the performance of the original technique.

In CEC2016 there are multiple competitions on various aspects of optimization. One of the competition aims to evaluate the performance of single objective optimization algorithms on four different kinds of optimization problems viz., (i) bound constrained single objective computationally expensive numerical optimization, (ii) learning-based real-parameter single objective optimization, (iii) continuous multimodal optimization, and (iv) single objective real parameter numerical optimization. However it is not mandatory for an optimization technique to participate in all the four parts as the algorithms are ranked for a particular type of problem. Unlike previous years, all the problems in the above four competitions have been proposed earlier and are to be used for CEC2016. Teaching Learning Based Optimization algorithm [6] has been recently proposed and has been attaining a lot of interest from the scientific community. For the CEC2016 competitions, we have proposed a variant of TLBO which has been demonstrated on computationally expensive optimization problems as well as the single objective real parameter numerical optimization. In particular, this work focusses on the latter category but the variant is the same that has been demonstrated for the computationally expensive optimization problems in CEC2016 [7].

978-1-5090-0623-6/16/$31.00 ©2016 IEEE

The article is organized as follows: In the succeeding section, we provide a description of the variant. It is followed by a section on the experimental settings employed in this work which is succeeded by the results and discussion in Section IV. Subsequently we conclude the article by summarizing the developments in this work.

II. SINGLE PHASE MULTI-GROUP TEACHING LEARNING OPTIMIZATION

TLBO is a stochastic population based algorithm that is inspired from the teaching learning process. In the terminology of TLBO, the population is referred as *class* and the population members are referred as *students* while the values of the decision variables of the optimization problems correspond to the *marks* obtained by a student in a particular subject. TLBO consists of two phases, namely, the teacher phase and the learner phase, which each member of the population undergoes in every iteration. From the perspective of optimization, each of these two phases generates a potential new member based on mathematical equations which are assumed to model the knowledge gained by the student from the teacher and classmates. If the new member has a better fitness function, it replaces the existing member else it is discarded. This process of the members undergoing the teacher and student phase is repeated till the termination criteria is not satisfied. Subsequent to TLBO, the authors of TLBO proposed Improved TLBO [8] which incorporated features such as (i) learning through groups, (ii) adaptive teaching factor, (iii) tutorial learning, and (iv) self-motivated learning. However we have realized that many of the operations employed in ITLBO can either be simplified or eliminated which in our experiments have led to the discovery of better solutions while lowering the computational burden.

The Single Multi-Group Teaching Learning Optimization, has only two user defined parameters namely the population size and the number of groups. Similar to other evolutionary algorithms, a population is created within the bounds of the decision variables and the fitness of the population is evaluated. Subsequent to this, the population is randomly divided into a specified number of groups such that all groups contain equal number of students (except one group). Each group subsequently undergoes the students phase and the learners phase. In a group, each member has equal probability to undergo the teacher phase or the learner phase but does not undergo both the teacher and learner phase. If the student of a group undergoes the teacher phase, a potential new member is generated using any one of the following equations.

$$Xnew_i = X_i + \left(X_{best} - T_{f_i}\bar{X}_m\right)r_i + r_i'\left(X_i - X_p\right) \ if \ f\left(X_i\right) < f\left(X_p\right) \tag{1}$$

$$Xnew_i = X_i + \left(X_{best} - T_{f_i}\bar{X}_m\right)r_i + r_i'\left(X_p - X_i\right) if \ f\left(X_p\right) \geq f\left(X_i\right) \tag{2}$$

In the above equations $Xnew_i$ indicates the potential new solution whereas X_i indicates the student i who is undergoing the teacher phase, X_{best} indicates best member in the group, $\bar{X}_m$ indicates the mean of the decision variables of all the members in the group, T_{f_i} is the teaching factor which is the ratio of fitness value of the current member undergoing the teacher phase to the best fitness value in the group, r_i and r_i' indicate random numbers between 0 and 1, while X_p indicates the decision variables of the partner assigned to the member i. It should be noted that the number of random numbers required in this phase (for every member) is twice the number of decision variables. If the member undergoes the learner phase, the new population member is generated using any one of the two equations given below.

$$Xnew_i = X_i + r_i\left(X_i - X_p\right) + \left(X_{best_i} - E_g X_i\right)r_i' \ if \ f\left(X_i\right) < f\left(X_p\right) \tag{3}$$

$$Xnew_i = X_i + r_i\left(X_p - X_i\right) + \left(X_{best_i} - E_g X_i\right)r_i' if \ f\left(X_p\right) \geq f\left(X_i\right) \tag{4}$$

In the above equation, E_g is the exploration factor which is randomly assigned a value of 1 or 2. It should be noted that though the equations in teacher and learner phase are similar to a large extent, the learner phase does not utilize the fitness function of either the current member undergoing the learner phase or the best member in the group. Moreover the learner phase does not utilize the mean of the decision variables in the group. Subsequent to the generation of the new member, the following equations are used to bind it within the domain of the decision variables.

$$\begin{aligned} Xnew_i &= \max\left(Xnew_i, L\right) \\ Xnew_i &= \min\left(Xnew_i, U\right) \end{aligned} \tag{5}$$

In the above equation, L and U denote the lower and upper bounds of the decision variables respectively. If the fitness of the new bounded member is better than the member i, which was used to generate it, it replaces the member i in the group. If the fitness is not better, the newly generated solution is completely discarded. Thus each member of every group undergoes the teacher or the learners phase. At the end of this, the members of all the groups are combined and the population proceeds to the next iteration by randomly dividing the solutions into the specified number of groups. This process is repeated till the termination criteria are satisfied. The best solution in the population at the end of the solution procedure is the best solution determined by the algorithm. It should be noted that this variant does neither employ any elite mechanism nor does it require the identification and removal of the duplicate solutions. Since any new solution can replace an existing solution only if its fitness is better, the algorithm inherently has a monotonic convergence without employing any external archive. A detailed flow sheet of the algorithm is given in Fig. 1. and a detailed pseudo-code can be obtained from the article that has been submitted for the competition on the computationally expensive problems [7]. It should be noted that this variant is different from ITLBO in terms of (i) grouping of students, (ii) identification of group teachers, (iii) number of phases employed, (iv) elitism & duplicates, and (v) bounding of solutions. Detailed information on these differences are provided in [7].

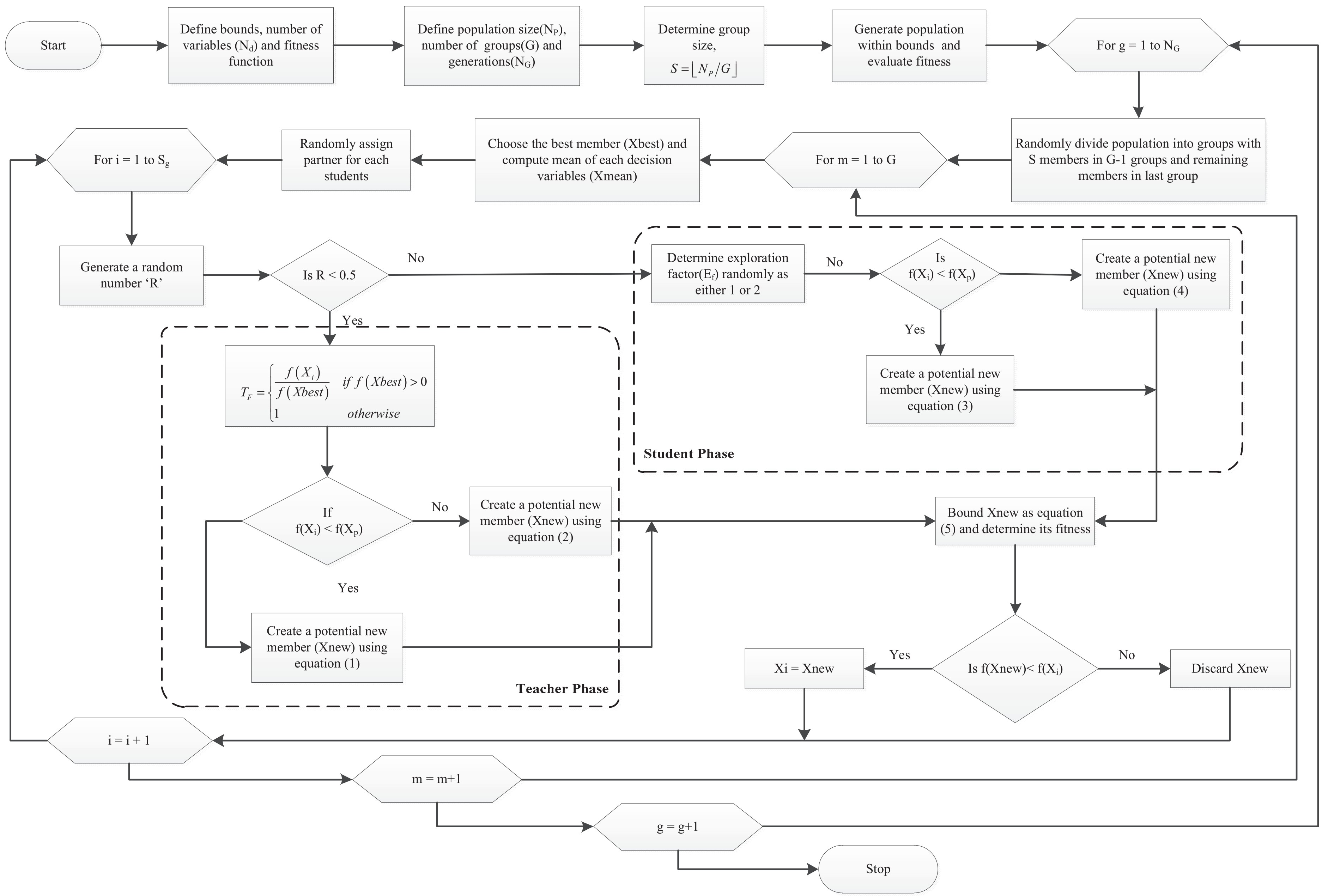

Fig. 1. **Flowchart of Single Phase multi-Group Teaching Learning Optimization algorithm.**

III. EXPERIMENTAL SETTINGS

The test suite provided by the organizers can be used with various platforms and consists of 30 problems which can be divided into four categories. A list of the functions with its optima is given in [9] and has been reproduced in Table I.

TABLE I. SUMMARY OF THE TEST PROBLEMS

Tag	Functions	F_i^*
F1	Rotated High Conditioned Elliptic Function	100
F2	Rotated Bent Cigar Function	200
F3	Rotated Discus Function	300
F4	Shifted and Rotated Rosenbrock's Function	400
F5	Shifted and Rotated Ackley's Function	500
F6	Shifted and Rotated Weierstrass Function	600
F7	Shifted and Rotated Griewank's Function	700
F8	Shifted Rastrigin's Function	800
F9	Shifted and Rotated Rastrigin's Function	900
F10	Shifted Schwefel's Function	1000
F11	Shifted and Rotated Schwefel's Function	1100
F12	Shifted and Rotated Katsuura Function	1200
F13	Shifted and Rotated HappyCat Function	1300
F14	Shifted and Rotated HGBat Function	1400
F15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500
F16	Shifted and Rotated Expanded Scaffer's F6 Function	1600
F17	Hybrid Function 1 (Three functions)	1700
F18	Hybrid Function 2 (Three functions)	1800
F19	Hybrid Function 3 (Four functions)	1900
F20	Hybrid Function 4 (Four functions)	2000
F21	Hybrid Function 5 (Five functions)	2100
F22	Hybrid Function 6 (Five functions)	2200
F23	Composition function 1 (Five functions)	2300
F24	Composition function 2 (Three functions)	2400
F25	Composition function 3 (Three functions)	2500
F26	Composition function 4 (Five functions)	2600
F27	Composition function 5 (Five functions)	2700
F28	Composition function 6 (Five functions)	2800
F29	Composition function 7 (Three functions)	2900
F30	Composition function 8 (Three functions)	3000

There are three unimodal functions (F1-F3), thirteen multimodal functions (F4-F16), six hybrid functions (F17-F22), and eight composition functions (F23-F30). These functions are scalable but are required to be tested for four dimensions (D=10, D=30, D=50 and D=100) for the competition. The search region of all the variable is $[-100, 100]^D$. The maximum number of permissible function evaluations is limited to 10,000D. We have employed a population size of 100 and 25 groups which limits the number of iterations for the 10D, 30D, 50D and 100D problems to 999, 2999, 4999, 9999 respectively as the algorithm requires each population member to be evaluated once in every generation. In view of the stochastic nature of the algorithm, the competition requires 51 independent runs to be performed for each problem. The 51 independent runs for each instance (combination of problem and dimension) was realized by varying the seed of the *twister* algorithm, in MATLAB 2015b, from 1 to 51 so as to enable independent verification of the results reported in this work. All the computations are performed on a PC running Windows 7 with an Intel i7 (3.4 GHz) processor and 20 GB RAM

IV. RESULTS AND DISCUSSION

As per the requirements of the competition, the performance of the algorithm is reported for the 10D, 30D, 50D and 100D problems.

A. Error values obtained

The difference between the fitness value obtained at the end of the maximum functional evaluations and the global optimum for each function is termed as error. The best, worst, mean, median and standard deviation of the error values corresponding to the best values in 51 runs of each problem for the all the four dimensions are given in Table II

From the mean of the errors reported in Table II, it can be observed that among the 30 problems, 10 problems have errors of similar magnitude in all dimensions. The error for all problems increases with an increase in the dimension except for F3, F7, F14, F18 and F23. The proposed variant is able to determine a solution close to the global optima in F5, F6, F7, F12, F13, F14, F15 and F16 in all dimensions. On considering the four categories of the test suite, SPMGTLO is able to perform well on multimodal functions except in F10 and F11. It can be observed that the algorithm exhibits unsatisfactory performance for the unimodal functions. In the composition functions, the algorithm is able to determine a solution close to the global optima only in 10D of F27. Among the eight composition functions, an error of magnitude in 102 is observed in almost all instances except the higher dimensional instances of F27 to F30. In the case of hybrid functions, SPMGTLO is able to obtain a solution close to global optima in all dimensions except 100D of F19. Unlike the previous competitions, a procedure to determine the score of the algorithm has not been reported in CEC2014 [9]. It can be concluded from the standard deviation values that the objective function value in all the 51 instances except 10D of F23 and F25 are similar.

B. Algorithm complexity

The technical report on CEC2014 [9] provides a procedure to determine the computational complexity of single objective optimization algorithms. Unlike the procedure employed in CEC2013, the computational complexity is to be separately determined for 10D and 30D problems in order to demonstrate the relationship between the algorithm complexity and the problem dimension. The results of algorithmic complexity for the proposed variant are tabulated in Table III.

TABLE II. ERROR VALUES FOR 10D, 30D, 50D AND 100D FUNCTIONS

Tag	10 D					30 D					50 D					100 D				
	Best	Worst	Mean	Median	Std	Best	Worst	Mean	Median	Std	Best	Worst	Mean	Median	Std	Best	Worst	Mean	Median	Std
F1	2.9E+3	3.1E+5	7.5E+4	6.1E+4	6.9E+4	1.0E+5	6.2E+5	2.3E+5	2.0E+5	1.1E+5	4.3E+5	1.9E+6	1.0E+6	1.0E+6	2.8E+5	4.3E+6	7.6E+6	5.7E+6	5.6E+6	7.7E+5
F2	4.0E-1	3.5E+3	8.7E+2	4.4E+2	9.9E+2	7.1E-1	2.5E+3	9.9E+2	8.3E+2	6.1E+2	3.7E+0	2.2E+4	5.2E+3	3.8E+3	4.8E+3	1.3E+3	4.2E+4	1.6E+4	1.5E+4	9.3E+3
F3	4.6E+2	4.2E+3	1.6E+3	1.5E+3	6.4E+2	4.6E+2	4.0E+3	1.6E+3	1.4E+3	7.0E+2	2.9E+3	1.8E+4	8.1E+3	7.4E+3	2.7E+3	4.8E+3	1.3E+4	8.1E+3	7.9E+3	1.6E+3
F4	3.2E-6	3.5E+1	2.4E+1	3.5E+1	1.6E+1	1.3E-1	1.4E+2	7.7E+1	7.6E+1	2.8E+1	1.2E-1	2.0E+2	1.1E+2	1.0E+2	4.0E+1	1.1E+2	3.8E+2	2.9E+2	2.9E+2	4.8E+1
F5	0.0E+0	2.0E+1	1.9E+1	2.0E+1	5.6E+0	2.1E+1	2.1E+1	2.1E+1	2.1E+1	5.1E-2	2.1E+1	2.1E+1	2.1E+1	2.1E+1	3.4E-2	2.1E+1	2.1E+1	2.1E+1	2.1E+1	2.0E-2
F6	0.0E+0	2.6E-1	1.5E-2	0.0E+0	5.3E-2	5.2E+0	1.2E+1	8.4E+0	8.3E+0	1.6E+0	1.9E+1	3.1E+1	2.4E+1	2.4E+1	2.5E+0	7.5E+1	9.2E+1	8.4E+1	8.5E+1	4.2E+0
F7	0.0E+0	6.9E-2	2.7E-2	2.5E-2	1.7E-2	0.0E+0	5.9E-2	1.6E-2	1.2E-2	1.6E-2	0.0E+0	2.6E-1	1.7E-2	9.9E-3	3.7E-2	0.0E+0	6.6E-2	1.2E-2	1.3E-8	1.9E-2
F8	9.9E-1	9.9E+0	4.5E+0	4.0E+0	2.1E+0	2.2E+1	7.1E+1	4.3E+1	4.3E+1	1.1E+1	8.3E+1	1.6E+2	1.2E+2	1.2E+2	1.8E+1	3.2E+2	4.4E+2	3.7E+2	3.7E+2	3.1E+1
F9	9.9E-1	9.0E+0	4.8E+0	5.0E+0	1.9E+0	3.1E+1	8.4E+1	4.7E+1	4.8E+1	1.1E+1	9.6E+1	1.6E+2	1.3E+2	1.3E+2	1.5E+1	3.1E+2	5.0E+2	4.0E+2	4.0E+2	3.5E+1
F10	4.2E+1	4.2E+2	1.3E+2	7.9E+1	9.6E+1	2.6E+2	2.9E+3	1.4E+3	1.3E+3	7.0E+2	1.4E+3	5.4E+3	3.5E+3	3.6E+3	9.8E+2	7.4E+3	1.4E+4	1.0E+4	1.1E+4	1.5E+3
F11	3.5E+0	1.2E+3	3.8E+2	3.4E+2	3.4E+2	1.8E+3	6.9E+3	5.9E+3	6.3E+3	1.0E+3	4.6E+3	1.3E+4	1.2E+4	1.2E+4	1.7E+3	7.5E+3	2.7E+4	1.3E+4	1.1E+4	4.5E+3
F12	3.7E-1	1.5E+0	1.1E+0	1.1E+0	2.1E-1	1.4E+0	2.9E+0	2.4E+0	2.5E+0	2.8E-1	2.7E+0	3.9E+0	3.3E+0	3.4E+0	2.8E-1	3.6E+0	4.4E+0	4.0E+0	4.1E+0	1.9E-1
F13	6.5E-2	1.6E-1	1.1E-1	1.1E-1	2.3E-2	2.1E-1	4.6E-1	3.1E-1	3.1E-1	5.6E-2	3.7E-1	6.7E-1	5.3E-1	5.1E-1	6.9E-2	4.4E-1	7.7E-1	5.6E-1	5.5E-1	5.6E-2
F14	1.7E-1	4.2E-1	2.9E-1	3.0E-1	6.0E-2	1.7E-1	3.2E-1	2.4E-1	2.4E-1	3.6E-2	2.6E-1	4.1E-1	3.1E-1	3.1E-1	3.2E-2	2.6E-1	3.8E-1	3.1E-1	3.1E-1	3.0E-2
F15	4.6E-1	1.9E+0	1.2E+0	1.2E+0	2.8E-1	2.8E+0	1.5E+1	7.0E+0	6.3E+0	2.8E+0	1.0E+1	4.0E+1	2.1E+1	2.1E+1	5.9E+0	4.7E+1	1.8E+2	8.0E+1	7.7E+1	2.3E+1
F16	1.2E+0	2.7E+0	2.0E+0	2.0E+0	3.8E-1	1.0E+1	1.2E+1	1.1E+1	1.1E+1	4.2E-1	1.9E+1	2.2E+1	2.0E+1	2.0E+1	5.0E-1	4.3E+1	4.6E+1	4.5E+1	4.5E+1	4.9E-1
F17	8.2E+2	7.9E+3	2.4E+3	2.1E+3	1.3E+3	2.5E+4	6.0E+5	1.9E+5	1.7E+5	1.3E+5	3.5E+4	5.9E+5	2.3E+5	1.7E+5	1.4E+5	3.5E+5	7.9E+5	5.4E+5	5.1E+5	1.1E+5
F18	2.9E+2	7.3E+3	3.2E+3	3.2E+3	1.9E+3	4.1E+1	3.4E+3	6.7E+2	4.1E+2	6.6E+2	1.6E+2	3.7E+3	1.5E+3	1.2E+3	1.2E+3	2.8E+2	6.4E+3	1.6E+3	1.3E+3	1.3E+3
F19	3.8E-2	1.6E+0	6.8E-1	7.0E-1	4.3E-1	3.5E+0	9.1E+0	5.6E+0	5.5E+0	1.2E+0	8.9E+0	8.1E+1	2.1E+1	1.6E+1	1.6E+1	2.9E+1	1.7E+2	1.0E+2	1.0E+2	2.7E+1
F20	8.9E+1	1.1E+3	4.1E+2	3.5E+2	2.5E+2	1.5E+3	9.7E+3	3.7E+3	3.3E+3	1.7E+3	1.2E+3	7.2E+3	3.9E+3	3.6E+3	1.5E+3	3.5E+3	1.2E+4	6.3E+3	6.0E+3	1.7E+3
F21	2.0E+2	8.9E+2	4.7E+2	4.4E+2	1.7E+2	1.1E+4	2.5E+5	8.8E+4	7.8E+4	5.2E+4	3.7E+4	6.6E+5	2.2E+5	2.1E+5	1.2E+5	1.7E+5	8.3E+5	3.3E+5	3.2E+5	1.1E+5
F22	2.2E-1	2.7E+1	1.4E+1	2.1E+1	9.8E+0	2.3E+1	3.0E+2	1.6E+2	1.6E+2	6.9E+1	3.1E+1	1.2E+3	5.1E+2	5.1E+2	3.0E+2	7.0E+2	2.6E+3	1.5E+3	1.4E+3	4.1E+2
F23	3.3E+2	3.3E+2	3.3E+2	3.3E+2	0.0E+0	3.2E+2	3.2E+2	3.2E+2	3.2E+2	0.0E+0	2.0E+2	3.4E+2	3.4E+2	3.4E+2	2.8E+1	2.0E+2	2.0E+2	2.0E+2	2.0E+2	0.0E+0
F24	1.0E+2	1.2E+2	1.1E+2	1.1E+2	4.2E+0	2.0E+2	2.0E+2	2.0E+2	2.0E+2	8.3E-4	2.0E+2	2.0E+2	2.0E+2	2.0E+2	5.3E-4	2.0E+2	2.0E+2	2.0E+2	2.0E+2	2.5E-4
F25	1.1E+2	2.0E+2	1.5E+2	1.3E+2	2.9E+1	2.0E+2	2.0E+2	2.0E+2	2.0E+2	0.0E+0	2.0E+2	2.0E+2	2.0E+2	2.0E+2	0.0E+0	2.0E+2	2.0E+2	2.0E+2	2.0E+2	0.0E+0
F26	1.0E+2	1.0E+2	1.0E+2	1.0E+2	2.2E-2	1.0E+2	2.0E+2	1.0E+2	1.0E+2	1.4E+1	1.0E+2	2.0E+2	1.4E+2	1.0E+2	4.9E+1	1.0E+2	2.0E+2	2.0E+2	2.0E+2	2.0E+1
F27	1.3E+0	3.3E+2	9.4E+1	2.3E+0	1.4E+2	4.0E+2	6.0E+2	5.0E+2	5.1E+2	5.1E+1	8.1E+2	1.2E+3	9.6E+2	9.4E+2	7.8E+1	1.9E+3	2.5E+3	2.2E+3	2.3E+3	1.5E+2
F28	3.7E+2	4.0E+2	3.8E+2	3.8E+2	5.9E+0	8.0E+2	1.4E+3	9.5E+2	9.2E+2	9.3E+1	1.2E+3	2.9E+3	1.6E+3	1.4E+3	4.1E+2	2.6E+3	6.1E+3	4.1E+3	4.1E+3	8.9E+2
F29	3.0E+2	1.1E+3	6.2E+2	6.3E+2	1.3E+2	1.1E+3	1.1E+7	1.8E+6	2.5E+3	3.6E+6	2.7E+3	6.0E+7	3.7E+7	3.9E+7	1.5E+7	2.0E+3	2.3E+8	1.0E+8	1.1E+8	4.0E+7
F30	4.6E+2	8.4E+2	5.1E+2	4.9E+2	5.7E+1	1.1E+3	5.1E+3	3.2E+3	3.4E+3	9.9E+2	8.9E+3	1.9E+4	1.2E+4	1.1E+4	2.4E+3	8.1E+3	1.3E+4	1.0E+4	1.0E+4	9.7E+2

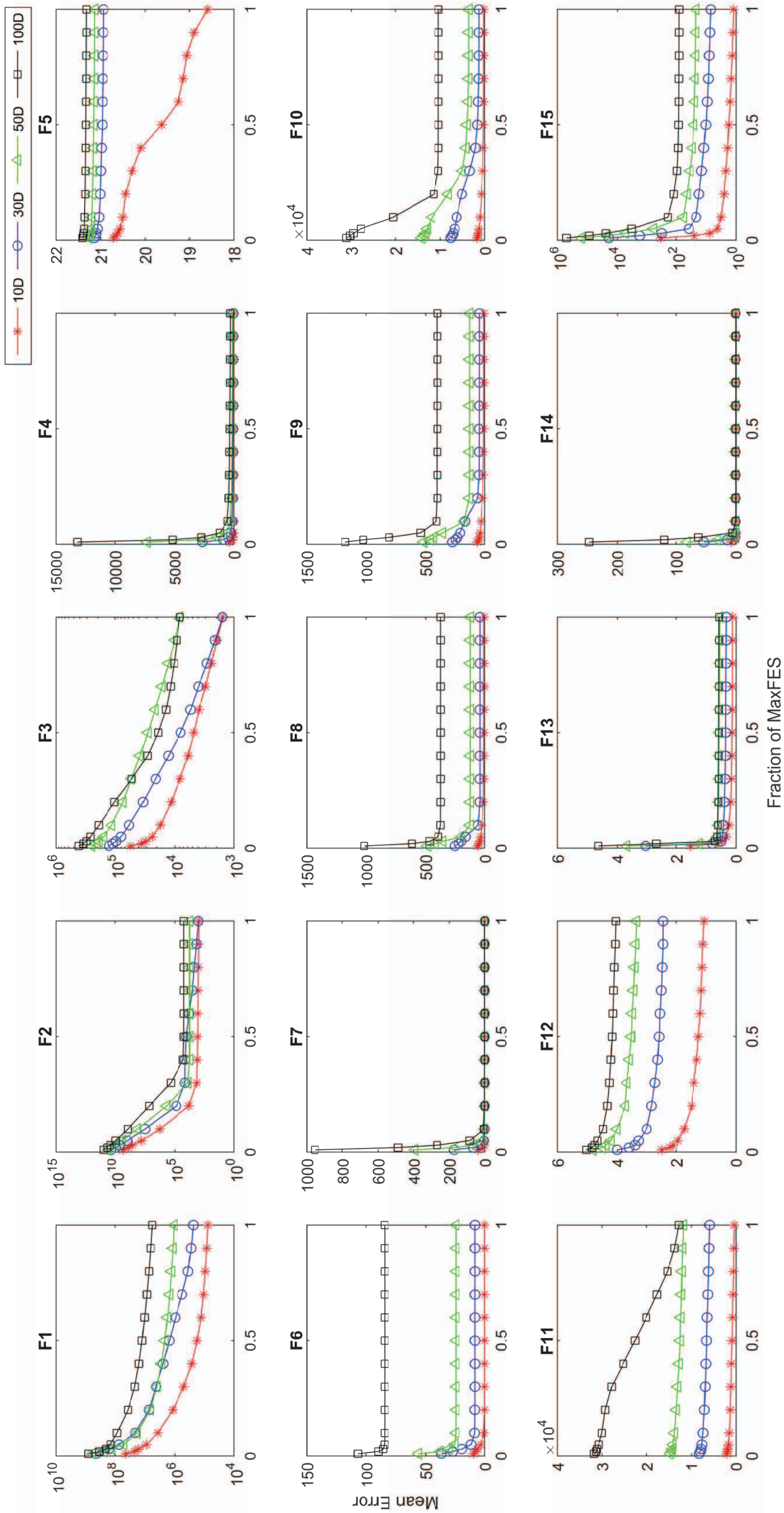

Fig. 2. Convergence curves for 10*D* and 30*D* cases of functions F1 to F15

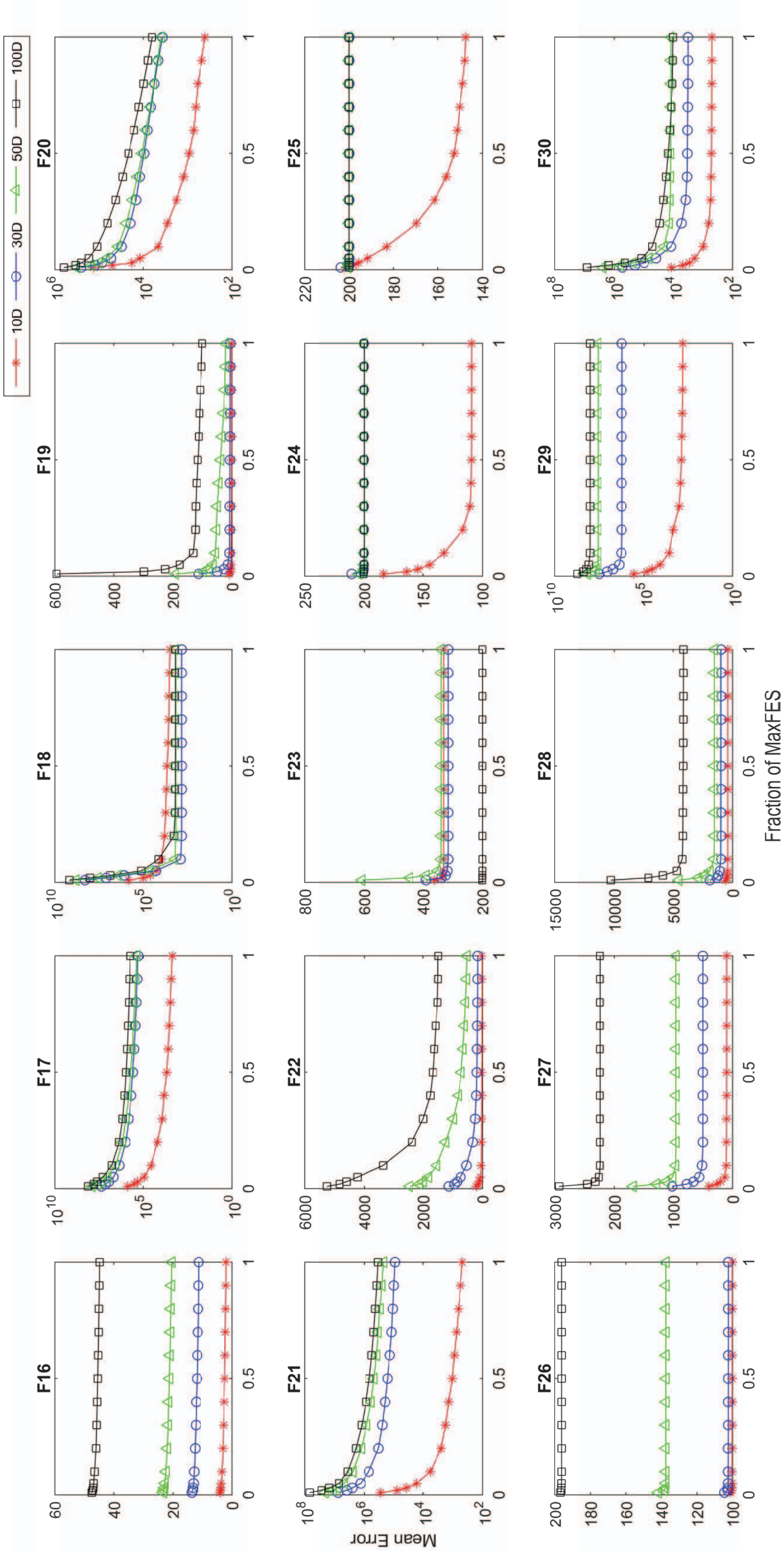

Fig. 3. Convergence curves for 10*D* and 30*D* cases of functions F16 to F30.

TABLE III. ALGORITHM COMPLEXITY

Dimension	T0 (sec.)	T1 (sec.)	$\overline{T2}$ (sec.)	($\overline{T2}$ -T1)/T0
10		0.98	3.84	26.93
30	0.11	1.12	12.18	104.31
50		1.38	21.19	186.84
100		2.61	52.43	469.71

T0 corresponds to the time required to solve a test program which consists of floating point operations, given in the CEC2014 report, for 100,000 times. T1 corresponds to the time required to evaluate the F18 function 200,000 times. The average time required by the algorithm to solve F18 with a maximum functional evaluations limit of 200,000 over five runs is represented as T2. It should be noted that values of T1 and T2 are to be evaluated for all the four different dimensions. The algorithm complexity is given by (T2-T1)/T0. It can be observed from the values in Table IV that the time required by the algorithm for solving 30D of F18 is around four times higher than that of 10D. This implies that as the dimension of the problems increases the algorithm complexity of the SPMGTLO also increases which can be attributed to the increased operations required in Equation (1) to Equation (4).

C. Convergence Curve

In view of the multiple runs required for each dimension of the problem, the convergence curve is plotted between the fraction of functional evaluations that have been utilized and the mean of the error (of 51 runs). The convergence plots corresponding to the functions F1 to F15 are shown in Fig. 1 whereas the convergence curves for the remaining 15 functions are shown in Fig. 2. Due to large variation in the magnitude of mean error, the plots for few problems are in semi-log axis. It has been observed that for most of the functions, there is no significant change in the mean error for a large number of the functional evaluations. However for all dimensions of few functions, such as F1, F3, F12, F20, F21 and except the 30D and 50D of F11 and 100D of F15 the algorithm has not converged to a solution even after utilizing the maximum number of permissible function evaluations. In many instances, though the algorithm has converged, it has not obtained the global optima. It is evident from the convergence curves that SPMGTLO is able to perform better in lower dimension than in higher dimensions except in F18 and F23. In problem F7 and F14 a quick convergence to a similar objective function value for all the dimensions is observed.

V. CONCLUSIONS

In this work, the performance of Single Phase Multi-Group Teaching Learning Optimization is evaluated on the 30 problems of CEC2016. The problems have been evaluated with 10, 30, 50 and 100 decision variable and it was observed that the performance of the algorithm is satisfactory for many of the multimodal functions. However, the performance on the complex hybrid and composition functions remained challenging. Many algorithms which have performed exceptionally well on similar optimization problems have a hybrid nature and thus future work can involve hybridizing the proposed variant of TLBO with other optimization techniques.

REFERENCE

1. Mirjalili, S., *Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm.* Knowledge-Based Systems, 2015. **89**: p. 228-249.
2. Patel, V.K. and V.J. Savsani, *Heat transfer search (HTS): a novel optimization algorithm.* Information Sciences, 2015. **324**: p. 217-246.
3. Li, M.D., et al., *A novel nature-inspired algorithm for optimization: Virus colony search.* Advances in Engineering Software, 2016. **92**: p. 65-88.
4. Črepinšek, M., S.-H. Liu, and L. Mernik, *A note on teaching–learning-based optimization algorithm.* Information Sciences, 2012. **212**: p. 79-93.
5. Mernik, M., et al., *On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation.* Information Sciences, 2015. **291**: p. 115-127.
6. Rao, R.V., V.J. Savsani, and D.P. Vakharia, *Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems.* Information Sciences, 2012. **183**(1): p. 1-15.
7. Remya, K., C. Sivadurgaprasad, and P. Kotecha, *Single Phase Multi-Group Teaching Learning Algorithm for Computationally Expensive Numerical Optimization*, in *IEEE World Congress on Computational Intelligence (Submitted)* 2016, IEEE CEC: Canada, , https://goo.gl/OTlz16.
8. Rao, R.V. and V. Patel, *An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems.* Scientia Iranica, 2013. **20**(3): p. 710-720.
9. Liang, J.J., B.Y. Qu, and P.N. Suganthan, *Problem Definitions and Evaluati on Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization* 2013, Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University.