Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 2221762, 17 pages
https://doi.org/10.1155/2022/2221762

Research Article

Fuzzy Adaptive Teaching Learning-Based Optimization for Solving Unconstrained Numerical Optimization Problems

Fakhrud Din,[1] **Shah Khalid**,[1] **Muhammad Fayaz**,[2] **Jeonghwan Gwak**,[3,4,5,6] **Kamal Z. Zamli**,[7,8] **and Wali Khan Mashwani**[9]

[1]*Department of Computer Science & IT, University of Malakand, KPK, Pakistan*
[2]*Department of Computer Science, University of Central Asia, Naryn, Kyrgyzstan*
[3]*Department of Software, Korea National University of Transportation, Chungju 27469, Republic of Korea*
[4]*Department of Biomedical Engineering, Korea National University of Transportation, Chungju 27469, Republic of Korea*
[5]*Department of AI Robotics Engineering, Korea National University of Transportation, Chungju 27469, Republic of Korea*
[6]*Department of IT & Energy Convergence (BK21 FOUR), Korea National University of Transportation, Chungju 27469, Republic of Korea*
[7]*Faculty of Computing, Universiti Malaysia Pahang, 26600 Pekan, Pahang Darul Makmur, Malaysia*
[8]*Faculty of Science and Technology, Universitas Airlangga, C Campus JI. Dr. H. Soekamo, Mulyorejo, Surabaya 60115, Indonesia*
[9]*Institute of Numerical Sciences, Kohat University of Science & Technology, KPK, Pakistan*

Correspondence should be addressed to Muhammad Fayaz; muhammad.fayaz@ucentralasia.org and Wali Khan Mashwani; mashwanigr8@gmail.com

Received 9 August 2021; Revised 26 October 2021; Accepted 14 March 2022; Published 30 April 2022

Academic Editor: Ewa Rak

Teaching learning-based optimization is one of the widely accepted metaheuristic algorithms inspired by teaching and learning within classrooms. It has successfully addressed several real-world optimization problems, but it may still be trapped in local optima and may suffer from the problem of premature convergence in the case of solving some challenging optimization problems. To overcome these drawbacks and to achieve an appropriate percentage of exploitation and exploration, this study presents a new modified teaching learning-based optimization algorithm called the fuzzy adaptive teaching learning-based optimization algorithm. The proposed fuzzy adaptive teaching learning-based optimization algorithm uses three measures from the search space, namely, quality measure, diversification measure, and intensification measure. As the 50-50 probabilities for exploitation and exploration in the basic teaching learning-based optimization algorithm may be counterproductive, the Mamdani-type fuzzy inference system of the new algorithm takes these measures as a crisp inputs and generates selection as crisp output to choose either exploitation or exploration based on the current search requirement. This fuzzy-based adaptive selection helps to adequately balance global search or exploration and local search or exploitation operations during the search process as these operations are intrinsically dynamic. The performance of the fuzzy adaptive teaching learning-based optimization is evaluated against other metaheuristic algorithms including basic teaching learning-based optimization on 23 unconstrained global test functions. Moreover, adaptive teaching learning-based optimization is used to search for near-optimal values for the four parameters of the COCOMO II model, which are then tested for validity on a software project of NASA. Analysis and comparison of the obtained results indicate the efficiency and competitiveness of the proposed algorithm in addressing unconstrained continuous optimization tasks.

1. Introduction

Optimization is a process of searching and comparing acceptable solutions until finding the final best solution among the available solutions. The optimization process encompasses specific goals generally known as objective functions, a feasible search region with all valid solutions, and a search procedure as an optimization method [1]. A solution can be termed best or poor based on the objective function. The set of values for design variables in the objective function constitutes the search space. Finally, the optimization algorithm attempts to locate optimal solution(s) within the feasible search region.

Among other optimization methods, metaheuristic algorithms have emerged as promising optimization methods which solve different optimization problems by simulating a range of natural phenomena. Genetic algorithm (GA) [2], particle swarm optimization (PSO) [3], differential evaluation (DE) [4], and harmony search (HS) [5] are some widely adopted early metaheuristic algorithms. The success of metaheuristic algorithms lies in their problem and model-independent nature, as well as their flexibility and efficiency. Moreover, robustness to dynamic changes, broad applicability, hybridization with other approaches, and the ability to solve problems with no solutions are other essential characteristics of metaheuristic algorithms. As is evident from the "No Free-Lunch" theorem [6], two search algorithms have the same average performance when compared on a set of optimization problems. However, the statement does not imply that certain algorithms will not be able to produce better solutions for some objective functions. Hence, research on metaheuristic algorithms is very active and is continually extending the scientific literature with new and enhanced versions of the earlier metaheuristic algorithms. Recently, many strategies have been developed to improve the performance of the Mamdani fuzzy logic and have been deployed in many areas for different purposes. Similarly, different methods have been proposed to improve the performance of hierarchical Mamdani fuzzy inference systems and have been used for assessment and prediction purposes in various fields [7–9], such as underground risk assessment and energy consumption optimization in smart homes, along with different other techniques. Moreover, fuzzy systems are instrumental in areas where decision-making involves high uncertainties [10, 11].

Teaching learning-based optimization (TLBO) [12] is one of the recent metaheuristic algorithms proposed by Rao et al. for optimization of different hard problems. TLBO is inspired by human behavior (i.e., the teaching-learning process) to search for optimal solutions. As TLBO is a population-based algorithm, it employs a group of learners for optimization. Initially, the learners undergo the teaching phase, where their learning capabilities are improved through teaching by a teacher (the best learner). Afterward, the learners interact with each other in the learning phase for improvement of their knowledge. In TLBO, exploration or global search is simulated using the teaching phase, whereas exploitation or local search is achieved via the learning phase. Both these search operations are carried out sequentially in each iteration. This preplanned division of the searching process may be counter-productive and may result in a suboptimal solution. To address this issue, a new TLBO variant called adaptive TLBO (ATLBO) [13] has been introduced. ATLBO combines the Mamdani-type fuzzy inference system with the basic TLBO to adaptively select either exploitation or exploration based on current search requirements in each round of the search process. ATLBO successfully addressed discrete optimization problems, i.e., the t-way test suite generation problem [14–16]. Building on and complementing our earlier work, the main motivation in this work is to demonstrate the generality of ATLBO via its adoption for continuous optimization problems. To this end, the three measures of ATLBO, namely, quality measure, intensification measure, and diversification measure, are reformulated and the developed fuzzy rules are further tuned to avoid local optima and to balance exploration and exploitation. To investigate its performance on continuous optimization problems, ATLBO is evaluated on 23 different global test functions, and the results are compared against some preselected metaheuristic algorithms including TLBO and its other fuzzy variant named fuzzy adaptive TLBO (FATLBO) [17]. Moreover, ATLBO has successfully optimized the four coefficients, namely, A, B, C, and D, in the formulation of the COCOMO II model. As ATLBO can now best decide whether to go for global search or local search, it can not only balance the two searches but can also avoid trapping in local optima. Experimental results also demonstrate that ATLBO has outperformed all existing algorithms owing to its effective search procedure with an appropriate percentage of exploration and exploitation.

The contributions of this study are as follows:

(i) Design and development of ATLBO for optimization of continuous problems

(ii) Investigating the performance of ATLBO against some existing metaheuristic algorithms on test functions as benchmarks

(iii) Adopting ATLBO for optimization of parameters for the COCOMO II model

The paper layout is as follows: Section 2 presents an overview of basic TLBO along with its variants and their applications. Section 3 presents a detailed description of the ATLBO for solving numerical optimization problems. Section 4 discusses the software cost estimation model (COCOMO II). Section 5 elaborates on the validation of ATLBO based on experimental results. Section 6 of the paper concludes the presented work.

The notations with their corresponding descriptions are listed in Table 1.

2. Classical Teaching Learning-Based Optimization (TLBO) Algorithm

Teaching learning-based optimization (TLBO) [12] algorithm is a novel nature-inspired metaheuristic algorithm for unconstrained and constrained optimization problems. In TLBO, the entire optimization process is equated with the teaching and learning methodology inside a classroom. Students or learners are simulated as solutions, whereas their subjects are represented as dimensions of the solutions (Figure 1). The result of a learner is regarded as the objective function value. TLBO exhibits competitive performance owing to its promising characteristics such as no algorithm-specific parameters, ease of implementation, computationally lightweight, and effective ability to search for near-optimal solutions [18].

A learner learns either from a teacher or a peer learner to improve his knowledge. TLBO utilizes this concept for optimization. As a more knowledgeable person in a class, the

Table 1: Notations with descriptions.

Notation	Description
$\sum$	Summation
Π	Product
T_F	Teaching factor
X	Population of learners
X_i^t	i th learner at time t
X_{mean}	Mean of the population X
X'	Updated population
Q_m	Quality measure
I_m	Intensification measure
D_m	Diversification measure
Max_{int}	Maximum intensification
X_{cb}	Current best solution
X_{gb}	Global best solution
Max_{Div}	Maximum diversification
$A(Q_m)$	Trapezoidal membership function of quality measure
$A(I_m)$	Trapezoidal membership function of intensification measure
$A(D_m)$	Trapezoidal membership function of diversification measure
=	Not equal to
μ(A(x))	Membership function of output fuzzy set
EMi	i th cost driver
SFi	i th scale factor
REi	i th relative error

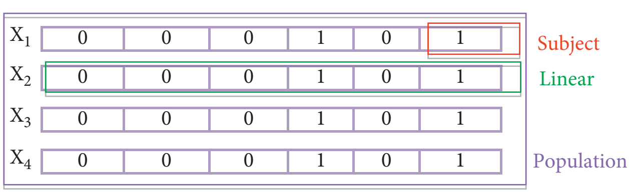

Figure 1: Concepts of TLBO for optimization.

teacher attempts to enhance the knowledge of his students called learners in TLBO. Similarly, a learner can learn from a peer whose knowledge level is better than their own. TLBO simulates these two steps of the learning process, one after the other, for addressing optimization problems.

As shown in Figure 1, each learner X_i within a population of X learners represents a potential solution to an optimization problem. Specifically, X_i is a vector of D where each vector element represents a subject enrolled by the student or delivered by the teacher.

TLBO divides searching into two main phases, namely, the teaching phase and the learning phase. To search optimal solutions, the population of learners is improved first by the teaching phase and then by the learning phase sequentially during each iteration. Exploration (i.e., global search operations) is the responsibility of the teacher phase. In this phase, the best individual X_i acts as the teacher X_{teacher} of all the learners. TLBO attempts to improve the knowledge level or position of every individual X_i via the best individual X_{teacher} in the population X. Mathematically, the definition of the teacher phase is given as follows:

$$X_i^{t+1} = X_i^t + r\left(X_{\text{teacher}} - T_F X_{\text{mean}}\right), \quad (1)$$

where X_i^{t+1} is the new updated X_i^t, r is assigned a value from [0, 1] randomly, and T_F is a teaching factor meant for emphasizing the quality of teaching. It is tested with various values, but TLBO is more successful when it is either 1 or 2. Finally, X_{mean} is the mean of the population X computed for the current iteration.

Exploitation (i.e., local search operation) is the responsibility of the learner phase in TLBO. Similar to the pear learning procedure in a typical class, this phase randomly selects a peer learner X_j^t and evaluates its position against the current learner X_i^t. The position of X_j^t is shifted towards X_i^t, if X_i^t has better quality than X_j^t (refer to equation (2)), otherwise X_i^t is shifted towards X_j^t (refer to equation (3)).

$$X_i^{t+1} = X_i^t + r\left(X_j^t - X_i^t\right). \quad (2)$$

$$X_i^{t+1} = X_i^t + r\left(X_i^t - X_j^t\right), \quad (3)$$

where X_i^{t+1} is the new form of X_i^t, X_j^t is the randomly selected peer, and r is randomly given value from [0, 1]. Figure 2 summarizes the classical TLBO algorithm.

2.1. Types of TLBO Variants

2.1. Types of TLBO Variants. Many TLBO variants have been proposed in the literature to search for more optimal solutions compared to the original TLBO algorithm. These variants are grouped into three categories, namely, modified-based, hybrid-based, and cooperative-based TLBO variants, as shown in Figure 3.

A fuzzy-based TLBO called the fuzzy adaptive teaching-learning-based optimization (FATLBO) algorithm by Cheng and Prayogo [17] addresses global numerical optimization. The modifications introduced in FATLBO include status monitor, fuzzy-based strategies, and remedial operator. A status monitor observes the performance of each phase. Fuzzy adaptive teaching-learning strategies are introduced for the selection of appropriate search operations. Finally, a remedial operator is included to avoid stagnation. With these modifications, FATLBO favors one phase more than the other, for example, the teacher phase than the learner phase or vice versa. Most recently, Lei et al. [19] proposed a teacher's teaching-learning-based optimization (TTLBO) algorithm for scheduling in a hybrid flow shop to minimize energy consumption. The learner phase is replaced with self-learning of teachers and a crossover operator for global search. In Niu et al. [20], MTLBO (modified TLBO) for global optimization is proposed. MTLBO divides the learners into two groups based on the mean results in both phases. The group of learners having the best mean results increases their knowledge by interaction among themselves, whereas the group of learners with average mean results increases their knowledge by learning from their teacher. MTLB has shown better solution quality as well as faster convergence. Wang et al. [21] proposed an improved TLBO (ITLBO) for constrained optimization problems that modify both the phases of TLBO. Most recently, Shukla et al. [22] modified TLBO for global optimization and gene selection using inertia weight and topological order to improve teacher and learner phases. Li et al. [23] introduced three

Input: Solutions learners: $X = X_1, X_2, \ldots, X_N$

Output: The best learner X_{best} from the improved learners: $X' = \{X_1', X_2', \ldots, X_N'\}$

Randomly generate population of learners X and evaluate their objective functions
While (criteria for stopping not reached)
    Select $X_{teacher}$ and calculate X_{mean}
    For (i=1 last learner)
        /* Teacher Phase…Global Search */
      $T_F = round\,(1 + r\,(0, 1))$
      $X_i^{t+1} = X_i^t + r\,(X_{teacher} - T_F X_{mean})$
      If ($f\,(X_i^{t+1})$ has better value than $f\,(X_i^t)$)
        $X_i^t = X_i^{t+1}$
    For (i=1 to last learner)
        /* Learner Phase…Local Search */
      Select one peer X_j^t randomly from X such that $i \neq j$
      If ($f\,(X_i^t)$ has better value than $f\,(X_j^t)$)
        $X_i^{t+1} = X_i^t + r\,(X_j^t - X_i^t)$
      Else
        $X_i^{t+1} = X_i^t + r\,(X_i^t - X_j^t)$
       If ($f\,(X_i^{t+1})$ has better value than $f\,(X_i^t)$)
        $X_i^t = X_i^{t+1}$
    Obtain the best result X_{best}

Figure 2: Classical TLBO algorithm.

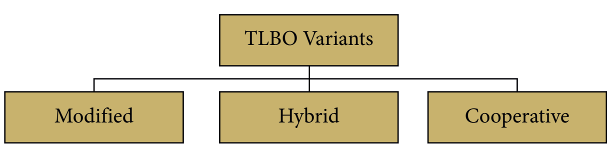

Figure 3: TLBO variants.

modifications in their reformative TLBO (RTLBO) for performance improvement of basic TLBO. The teaching phase is improved, while self-learning and mutation phases are proposed to have an appropriate ratio of exploitation and exploration in RTLBO. The teacher phase is divided into subpopulations to enhance diversity, whereas the learner phase is enhanced for promoting convergence by utilizing a ranking differential vector. All these variants are part of the adaptive or modified category of TLBO variants.

Hybrid-based variants of TLBO are discussed further in this study. Nenavath and Jatoth [24] recently proposed an optimization algorithm that comprises TLBO and a sine cosine algorithm (SCA) to solve different optimization problems including global test functions and visual tracking. Jiang and Zhou [25] attempted to combine TLBO with differential evolution (DE) for solving the short-term scheduling problem in a hydrothermal system. Tuo et al. [26] improved the search capability of harmony search by integrating it with TLBO (HSTL) to achieve a balance between population diversity and convergence speed and successfully adopted the work for solving general optimization problems with constraints. Deb et al. [27] proposed an ICSOTLBO that combines the chicken swarm optimization (CSO) algorithm with TLBO to exploit the strengths of both the algorithms. ICSOTLBO successfully solves synthetic as well as real-world optimization problems. In the case of these variants, researchers hybridized TLBO with other meta-heuristic algorithms.

The review of variants in the cooperative-based category is as follows: Biswas et al. [28] proposed a cooperative co-evolutionary TLBO with an improved exploration strategy to solve high-dimensional optimization problems. In a similar attempt, Satapathy and Naik [29] introduced co-operative TLBO (Co-TLBO) that exhibits cooperative behavior with the help of more than one population of learners. Zou et al. [30] proposed a new TLBO variant for optimization of global functions called hierarchical multi-swarm cooperative TLBO (HMCTLBO) that divides the population hierarchically into two levels to maintain the population diversity by improving the exploration characteristics of learners. All these variants divide optimization problems into k subproblems to be optimized concurrently before collecting the results.

Although, these variants attempted to enhance the solution diversity and convergence speed of original TLBO, they still trap at local optimal in case of more challenging optimization problems. Therefore, this research work attempts to further enhance the search ability of the original TLBO by proposing a modified type of variant that selects the appropriate phase, i.e., search operation in each iteration with the help of fuzzy logic.

2.2. TLBO for Software Effort Estimation. The role of software effort estimation in software engineering is essential for a successful software development project. More recently, TLBO and many other algorithms have been investigated for parameter optimization of different software effort estimation models including the COCOMO II model.

Khuat and Le [31] also implemented classical TLBO to generate optimal values for the four parameters of the COCOMO II model. With these optimized values, the model outperformed the original COCOMO II model by predicting better time and effort estimates. Ibrahim [32] adopted the bat algorithm to provide optimal cost estimation of a software project. Singh used an enhanced version of differential evolution for estimating software project costs. Khuat and Le [33] proposed a hybrid method that adopted artificial bee colony (ABC) and TLBO for parameter optimization of the COCOMO II model. Experimental results based on datasets from NASA software projects concluded that the algorithm offered better estimates than the competing COCOMO II model. Sehra et al. [34] proposed a model based on TLBO to optimize different parameters of the COCOMO model using a dataset from the software industry called interactive voice response (IVR). The proposed model estimated the effort more accurately than several existing models including the original COCOMO II model. Nandal and Sangwan [35] combined the bat algorithm with a gravitational search algorithm to predict project costs for software. Fadhil et al. [36] proposed two models for cost estimation. The first model employed the dolphin algorithm, whereas the second model hybridized the dolphin algorithm with the bat algorithm to optimize the two parameters, namely, A and B of the COCOMO II model. Recently, Rhmann et al. [37] adopted the firefly algorithm and genetic algorithm from the MetaheuristicsOpt *r* packages for cost prediction. Their study concluded that the two metaheuristic algorithms obtained optimal results when compared with the algorithms based on machine learning.

3. The Proposed Fuzzy Adaptive TLBO (ATLBO) Algorithm

Fuzzy logic has gained substantial acceptance in the industry as well as in academia owing to its numerous successful applications [38–40]. Integrating fuzzy inference systems with metaheuristic algorithms is appealing as they have successfully improved the performances of these algorithms in various ways. For instance, adaptive parameter tuning of metaheuristic algorithms using fuzzy inference systems appeared effective. Similarly, fuzzy inference systems have been used to hybridize metaheuristic algorithms for obtaining better results [41]. Finally, adaptive selection of appropriate search operations during the search with the help of fuzzy inference systems has recently been introduced in the literature.

The ATLBO's Mamdani fuzzy rule-based system allows it to adaptively select either local or global search and therefore enables it to search for more high-quality solutions. ATLBO uses three measures from the search space to keep track of how the search progresses while exploring the solution space. These measures are as follows: quality measure (Q_m), intensification measure (I_m), and diversification measure (D_m). As the name suggests, Q_m measures the quality of the current solution after each iteration. Equation (4) gives a formal definition of the normalized value of this measure.

$$Q_m = [\text{rangMin} + [\text{rangMax} - \text{rangMin}] \cdot r] \cdot r, \tag{4}$$

where r is randomly selected within the interval [0 1]. If the current solution is more optimal than previous solution, rangeMin = 50 and rangeMax = 100, otherwise rangeMin = 0 and rangeMax = 50.

I_m measure computes the position of the current best solution from global best. With this measure, ATLBO will be able to perform a local search as per search requirements. Equation (5) defines the intensification measure (I_m).

$$I_m = \left[\frac{\sqrt{\sum_{i=1}^{D}\left(X_{\text{cb},i} - X_{\text{gb},i}\right)^2}}{\text{Max}_{\text{int}}}\right], \tag{5}$$

where X_{cb} is the current best after completion of each iteration, X_{gb} is the global best solution, D is problem's dimension, and Max_{Int} is the maximum intensification which is $\text{Max}_{\text{int}} = n \cdot \sqrt{D.(\text{ub} - \text{lb})^2}$, ub and lb are problem's upper and lower bounds, respectively.

The diversification measure (D_m) computes the position of the best solution obtained after an iteration from the entire population. The ATLBO uses this measure to adequately explore the search space. The diversification measure is defined as follows:

$$D_m = \left[\frac{\sqrt{\sum_{i=1}^{n}\left[\sum_{i=1}^{D}\left(X_{\text{cb},i} - X_{\text{gb},i}\right)^2\right]}}{\text{Max}_{\text{Div}}}\right] \cdot 100, \tag{6}$$

where n is the number of students (i.e., population size), and Max_{Div} is maximum diversification which is defined as $\text{Max}_{\text{Div}} = n \cdot \sqrt{D.(\text{ub} - \text{lb})^{2)}}$.

The choice of selecting and using these three measures is novel for improving the performance of a metaheuristic algorithm. These measures capture all the necessary details which guide the algorithm to proceed its search in the right direction. Diversification and quality measures promote solution diversity, whereas the intensification measure facilitates convergence in ATLBO.

3.1. ATLBO's Mamdani Fuzzy Inference System. Fuzzy logic is an efficient alternative to traditional logic for solving decision-related problems [42, 43]. ATLBO integrates this logic to make the optimal decision of whether to apply global search or local search. To this end, ATLBO provides the three computed measures as crisp inputs to the Mamdani-type fuzzy inference system. The block diagram of the fuzzy inference system is shown in Figure 4. The system has three crisp inputs: Q_m; I_m; D_m, and one crisp output: Selection.

The fuzzy inference process begins with fuzzification that utilizes membership functions for translating the crisp

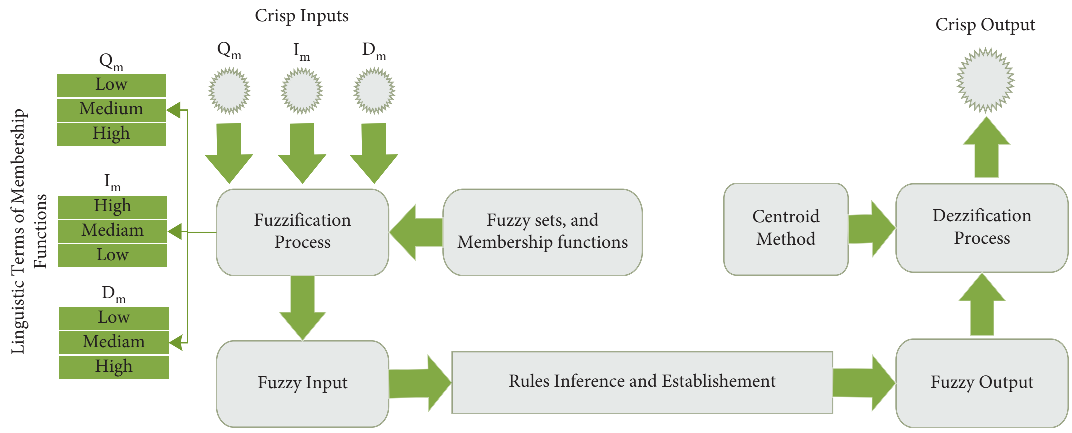

FIGURE 4: Structure of ATLB's Mamdani fuzzy inference system.

inputs into fuzzy sets. Each crisp variable has a universe of discourse that defines a range of acceptable values for it. The membership functions used for the fuzzy sets of the three input measures: Q_m, I_m, and D_m $A(Q_m)=\{$Low, Medium, High$\}$; $A(I_m)=\{$High, Medium, Low$\}$; $A(D_m)=\{$Low, Medium, High$\}$, respectively, are shown in Figure 5.

Similarly, the membership functions defined for the fuzzy sets of the single output Selection, $A(\text{Selection})=$ {Local_Search, Global_Search}, are shown in Figure 6. The range of 0 to 100 is defined as the universe of discourse for the input and output variables.

Owing to offering better performance than others [44], trapezoidal membership functions are adopted for all the linguistic terms of the input and output variables. Equation (7) represents the description of this function.

$$\text{Trapezoidal}\,(x;a,b,c,d)=\begin{cases}0 & x<0\\ \dfrac{x-a}{b-a} & a\le x\le b\\ 1 & b\le x\le c\,.\\ \dfrac{d-x}{d-c} & c\le x\le d\\ 0 & d\le x\end{cases} \tag{7}$$

where x is a crisp value within the universe of discourse, and a, b, c, and d are the four heads of trapezoid representing coordinates of the x. Linguistic variables, linguistic terms membership functions, and boundaries of membership functions have been given in Table 2.

Decision-making logic is the next step after finalizing membership functions for the inputs and outputs of the inferencing system. The Mamdani-type fuzzy inference system is the most common system for fuzzy decision making, where the IF-THEN rules have fuzzy prepositions in the antecedent as well as in the consequent parts [45]. Table 3 shows these four values for the membership functions of linguistic terms of each linguistic variable as shown in Figures 5 and 6. Fuzzy IF-THEN rule example is given as follows: IF "Q_m" IS Low THEN "Selection" IS "Global Search." A fuzzy rule base encompasses system-related information. Table 3 presents the five rules of the fuzzy system integrated with ATLBO.

The fuzzy inference system of ATLBO employs the max-min inference method. This method takes the minimum value of the antecedents in the case of the fuzzy AND operator, whereas the maximum value is considered when accumulating the antecedents.

Finally, the defuzzification step transforms the fuzzy conclusions made by the inferencing section of the fuzzy inference are transferred into the crisp output. As described earlier, Selection is the only output linguistic variable that is to be defuzzified. Eventually, the defuzzification process completed using the center of gravity (COG) generates results which dictate the actual selection of the appropriate search operation. COG is the most commonly adopted method in fuzzy systems owing to its accurate computation of results on the basis of weighted values of many output membership functions [46]. The result of defuzzification is assigned to the Selection crisp variable after the evaluation of the COG formula according to the following equation:

$$\text{Selection}=\text{COG}=\begin{cases}\dfrac{\int_x \mu(A(x))x\mathrm{d}x}{\int_x \mu(A(x))\mathrm{d}x}; & \text{if } x \text{ is continous}\\ \dfrac{\sum_x \mu(A(x))x}{\sum_x \sum_x \mu(A(x))}; & \text{if } x \text{ is discrete,}\end{cases} \tag{8}$$

where $\mu(A(x))$ denotes the membership function value of the output fuzzy set.

The fuzzy inference system of ATLBO could be implemented with other possible design choices such as triangular memberships and the number of linguistic terms. However, the proposed fuzzy inference system is easy to understand, functional, and sufficiently efficient owing to the adoption of the basic design choices.

It is evident from the overview of the original TLBO that both global search and local search operations get an equal

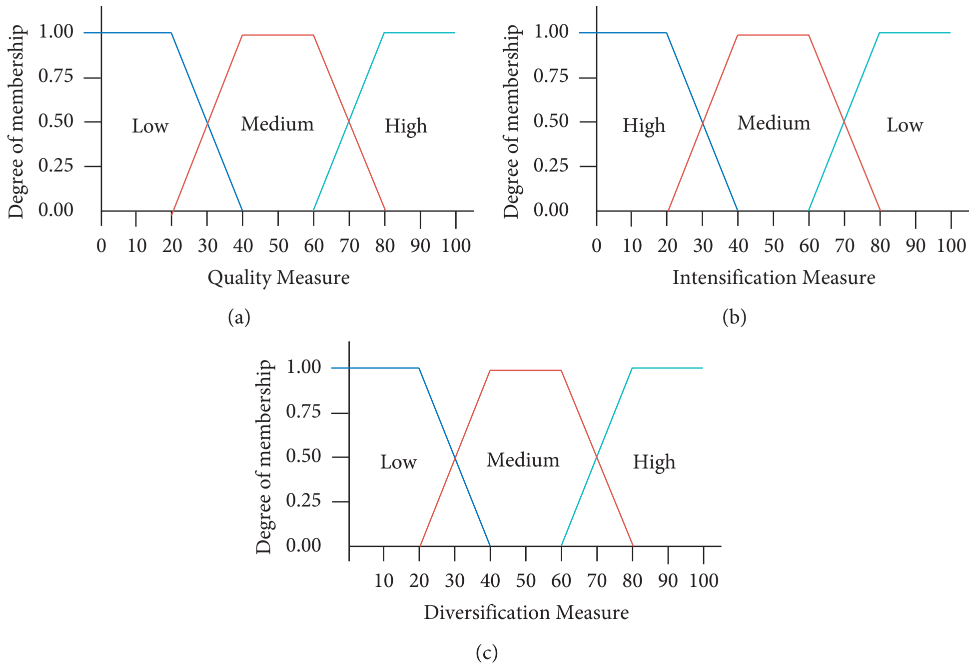

FIGURE 5: Membership functions for the three measures. (a) Quality measure. (b) Intensification measure. (c) Diversification measure.

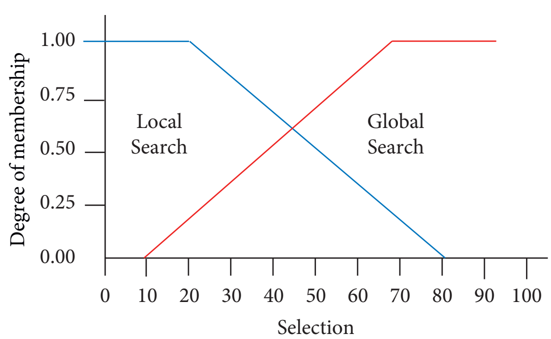

FIGURE 6: Selection's membership functions.

opportunity (50%) in each iteration during the search process. Therefore, when the defuzzification output of the fuzzy system assigned to the Selection parameter is greater than 50%, the proposed algorithm selects a global search or teacher phase. Otherwise, it selects the local search or learner phase. Figure 7 presents the fuzzy adaptive TLBO (ATLBO) based on the proposed Mamdani fuzzy inference system for addressing unconstrained global optimization problems.

3.2. Complexity. For time complexity of the proposed ATLBO, big-O is used as given here.

(i) The ATLBO initializes the population of learners in $O(\text{Population}_{\text{size}} \times \text{Problem}_{\text{dimension}})$.

(ii) The time required to select X_{teacher} and X_{mean} is $O(\text{Population}_{\text{size}})$.

(iii) The time required to update positions of learners in the teacher phase of ATLBO is $O(\text{Population}_{\text{size}} \times \text{Problem}_{\text{dimension}})$.

(iv) The time required to update positions of learners in the learner phase of the algorithm is $O(\text{Population}_{\text{size}} \times \text{Problem}_{\text{dimension}})$.

(v) $O(\text{Population}_{\text{size}})$ is the time required to evaluate objective function in the teacher phase of ATLBO.

(vi) $O(\text{Population}_{\text{size}})$ is the time required to evaluate objective function in the learner phase of ATLBO.

(vii) ATLBO takes $O(\text{Population}_{\text{size}} \times \text{Problem}_{\text{dimension}})$ as an additional time for computing the diversification measure.

(viii) Time $O(\text{Problem}_{\text{dimension}})$ is taken by ATLBO to compute the intensification measure.

In essence, the total time complexity of ATLBO is $O(\text{Population}_{\text{size}} \times \text{Problem}_{\text{dimension}} \times \text{Mat}_{\text{iterations}})$. Hence, both ATLBO and original TLBO have similar computational complexity. All other evaluations are considered constants including the evaluation of the 5 fuzzy rules.

Table 2: Details of the fuzzy input/output variables.

	Linguistic variables	Linguistic terms	Member functions	Corners of trapezoid
1	Quality measure	Low	$\mu_{\text{Low}}(A(Q_m))$	$a=b=0,\ c=20,\ d=40$
		Medium	$\mu_{\text{Medium}}(A(Q_m))$	$a=20,\ b=40,\ c=60,\ d=80$
		High	$\mu_{\text{High}}(A(Q_m))$	$a=60,\ b=80,\ c=d=100$
2	Intensification measure	High	$\mu_{\text{High}}(A(I_m))$	$a=b=0,\ c=20,\ d=40$
		Medium	$\mu_{\text{Medium}}(A(I_m))$	$a=20,\ b=40,\ c=60,\ d=80$
		Low	$\mu_{\text{Low}}(A(I_m))$	$a=60,\ b=80,\ c=d=100$
3	Diversification measure	Low	$\mu_{\text{Low}}(A(D_m))$	$a=b=-\infty,\ c=20,\ d=40$
		Medium	$\mu_{\text{Medium}}(A(D_m))$	$a=20,\ b=40,\ c=60,\ d=80$
		High	$\mu_{\text{High}}(A(D_m))$	$a=60,\ b=80,\ c=d=100$
4	Selection	Local search	$\mu_{\text{Local\_Search}}(A(\text{Selection}))$	$a=b=0,\ c=20,\ d=80$
		Global search	$\mu_{\text{global\_Search}}(A(\text{Selection}))$	$a=20,\ b=80,\ c=d=100$

Table 3: Fuzzy rules.

R#		Antecedent	Consequent
1	IF	$Q_m = \neg$ high	Selection = global search
2	IF	Q_m = medium	Selection = local search
3	IF	Q_m = high and D_m = high and $I_m = \neg$ high	Selection = local search
4	IF	Q_m = high and $D_m = \neg$ high and I_m = high	Selection = global search
5	IF	Q_m = high and D_m = high and I_m = high	Selection = local search

4. Overview of COCOMO II Model

The constructive cost model II (COCOMO II) [47] is an effective model developed by Barry Boehm for estimating future software project development cost, effort, and schedule. The input of the model is qualitative, whereas it produces quantitative output. Estimation of cost in the COCOMO II model is predicted using person-months (PMs) effort. The time spent by a person working on some part of the software development project in one month is equivalent to one person-month. The COCOMO II model utilizes the formulation shown in equation (9) to predict the software development effort [48].

$$\text{PM} = A.\text{Size}^{E} \prod_{i=1}^{17} \text{EM}_i, \tag{9}$$

where A represents a multiplicative constant and is equal to 2.94; project size computed as kilo line of code (KLOC) is represented by Size; EM_i represents a parameter from the set known as cost drivers which are listed in Table 4.

All these effort multipliers (EM) in postarchitecture (PA) adjust the nominal effort in the COCOMO II model. Each cost driver consists of six different rating levels, as shown in Table 4. Each rating level also known as the multiplier is assigned a value. These multipliers capture software development features that influence the effort necessary for the software project completion.

The exponent of *Size* (i.e., E) in equation (9) is computed by using equation (10). This exponent is basically an aggregated value of five scale factors (SFs) which excessively affect the effort or productivity of a software development project.

$$E = B + 0.01 \cdot \sum_{i=1}^{5} \text{SF}_i, \tag{10}$$

where B represents a multiplicative constant and is equal to 0.91. There are again six different rating levels for each *SF* with a predetermined weight, as shown in Table 5.

Besides effort, companies attempt to estimate the time (TDEV) for software development projects. TDEV is derived from the effort using the following equations:

$$\text{TDEV} = C \cdot \text{PM}^{F}, \tag{11}$$

$$E = D + 0.2 \cdot 0.01 \cdot \sum_{i=1}^{5} \text{SF}_i, \tag{12}$$

where $C = 3.67$ and $D = 0.28$, and these values have been obtained by utilizing 161 projects' actual schedule values from the COCOMO II database.

Prediction level PRED (x) and mean of magnitude of relative error (MMRE) are considered two accurate reference values in software effort estimation. The value of PRED (30) often determines the performance of COCOMO [49].The relative error (RE) is used for calculating PRED (30). RE represents the relative size and is given by the following equation:

$$\text{RE}_i = \frac{\text{estimate}_i - \text{actual}_i}{\text{actual}_i}. \tag{13}$$

Summation of the absolute values of RE_i for each individual project divided by T projects in the dataset and multiplied by 100 resultMMRE are as follows:

Input: Solutions as learners: $X = X_1, X_2, \ldots, X_N$

Output: The best learner X_{best} from the improved learners: $X' = \{X'_1, X'_2, \ldots, X'_N\}$

Randomly generate population of learners X and evaluate their objective functions
Define membership functions for Q_m, I_m, D_m
Fuzzy rules definition for the inference system
Set Selection=100
While (criteria for stopping not reach)
    If (Selection > 50)
    Select $X_{teacher}$ and calculate X_{mean}
        For (i=1 to last learner)
            /* Teacher Phase…Global Search */
            $T_F = round\ (1 + r\ (0, 1))$
            $X_i^{t+1} = X_i^t + r\ (X_{teacher} - T_F X_{mean})$
            If ($f\ (X_i^{t+1})$ has better value than $f\ (X_i^t)$)
                $X_i^t = X_i^{t+1}$
    Else
        For (i=1 to last learner)
            /* Learner Phase…Local Search */
            Select a learner X_j^t randomly such that $i \neq j$
            If ($f\ (X_i^t)$ is better than $f\ (X_j^t)$)
                $X_i^{t+1} = X_i^t + r\ (X_j^t - X_i^t)$
            Else
                $X_i^{t+1} = X_i^t + r\ (X_i^t - X_j^t)$
            If ($f\ (X_i^{t+1})$ has better value than $f\ (X_i^t)$)
                $X_i^t = X_i^{t+1}$
    Display the best result X_{best}
    Compute Q_m, I_m, and D_m
     Fuzzify based on the three measures
    Defuzzify and set Selection = crisp output
Display X_{best}

Figure 7: Fuzzy adaptive TLBO (ATLBO).

Table 4: Cost drivers in COCOMO II for the PA model.

Driver	Symbol	Very low	Low	Nominal	High	Very high	Extra high
RELY	EM1	0.82	0.92	1	1.1	1.26	—
DATA	EM2	—	0.9	1	1.14	1.28	—
CPLX	EM3	0.73	0.87	1	1.17	1.34	1.74
RUSE	EM4	—	0.95	1	1.07	1.15	1.24
DOCU	EM5	0.81	0.91	1	1.11	1.23	—
TIME	EM6	—	—	1	1.11	1.29	1.63
STOR	EM7	—	—	1	1.05	1.17	1.46
PVOL	EM8	—	0.87	1	1.15	1.3	—
ACAP	EM9	1.42	1.19	1	0.85	0.71	—
PCAP	EM10	1.34	1.15	1	0.88	0.76	—
PCON	EM11	1.29	1.12	1	0.9	0.81	—
APEX	EM12	1.22	1.1	1	0.88	0.81	—
PLEX	EM13	1.19	1.09	1	0.91	0.85	—
LTEX	EM14	1.2	1.09	1	0.91	0.84	—
TOOL	EM15	1.17	1.09	1	0.9	0.78	—
SITE	EM16	1.22	1.09	1	0.93	0.86	0.8
SCED	EM17	1.43	1.14	1	1	1	—

Table 5: Scale factors with their respective values for the COCOMO II model.

Scale factors	Six levels					
	Extra high	Very high	High	Nominal	Low	Very low
SF_1 = PREC	0	1.24	2.48	3.72	4.96	6.2
SF_2 = FLEX	0	1.01	2.03	3.04	4.05	5.07
SF_3 = RESL	0	1.41	2.83	4.24	5.65	7.07
SF_4 = TEAM	0	1.1	2.19	3.29	4.38	5.48
SF_5 = PMAT	0	1.56	3.12	4.68	6.24	0

$$\mathrm{MRE}_i = \left|\mathrm{RE}_i\right|. \tag{14}$$

$$\mathrm{MMRE} = \frac{100}{T}\cdot\sum_{i=1}^{T}\mathrm{MRE}_i. \tag{15}$$

The average percentage of the estimated values reported by PRED (N) were within $N\%$ of the real effort values of the projects from the adopted test dataset (equation (16)).

$$\mathrm{PRED}(N) = \frac{100}{T}\sum_{i=1}^{T}\begin{cases}1, & \text{if } \mathrm{MRE}_i \le \dfrac{N}{100},\\ 0, & \text{otherwise.}\end{cases} \tag{16}$$

In this study, ATLBO attempts to obtain optimal values for the parameters: A, B, C, and D of the COCOMO II model, which in turn will minimize MMRE and maximize PRED (N).

5. ATLBO Validation

This section presents two experiments conducted to validate the performance of the proposed ATLBO algorithm.

(i) Experiment 1 presents a wide range of test functions as optimization problems to be evaluated and provides a comparison of the obtained results for these functions by the proposed ATLBO with other swarm-based metaheuristic algorithms.

(ii) Experiment 2 adopts ATLBO for parameter optimization of the COCOMO II model. It is essential to search for optimal values for the four parameters of the COCOMO II model, namely A, B, C, and D, as the values of these parameters determine the minimal effort and cost for a software project.

5.1. Experiment 1. Here, the performance of ATLBO is compared on 23 different global test functions against TLBO, FATLBO, GA, DE, PSO, BA, and PBA. Out of these functions, functions f1–f11 (total 11) are two-dimensional, function *f12* is four-dimensional, function f13 is 10-dimensional, and functions f14–f23 (total 10) are 30-dimensional. The scalability of the proposed algorithm can be determined by using different dimensions. Moreover, 11 functions are unimodal, whereas 12 are multimodal. Unimodal functions have only one optimum known as the global optimal, and are presented to algorithms to test their exploitation efficiency. Multimodal functions can be used to evaluate the local optimal avoidance capability of a metaheuristic algorithm. Finally, 16 of the 23 functions are nonseparable, whereas seven are separable. Table 6 describes all the included test functions in detail.

For each of the functions listed in Table 6, experiments on all selected algorithms have been previously conducted by Cheng and Lien [50]. The maximum number of fitness function evaluations for ATLBO is set at 500,000. Results of less than $1E-12$ were considered 0. This study adopts these similar conditions for the purpose of maintaining consistency and to confirm the superior performance of the proposed metaheuristic algorithm. Parameter settings for all competing algorithms as well as ATLBO are shown in Table 7. By virtue of being *parameter-free*, TLBO, ATLBO, and FTLBO require only one parameter (i.e., population size). Here, like all other algorithms, the population size for the TLBO and its variants is $n = 50$.

Table 8 presents the results of ATLBO and other referenced metaheuristic algorithms. All these results except for ATLBO are adopted from Cheng and Lien [50] and Cheng and Prayogo [17]. The mean and standard deviation values reported against ATLBO were obtained after running ATLBO 30 times to maintain similarity with the earlier work. All the numbers in italic represent the best values in Table 8. ATLBO obtained the global optimum for 21 of the total 23 test functions. In the case of the most challenging function f20 (Rosenbrok), ATLBO outperformed all other algorithms by producing the overall best value. Similarly, ATLBO outperformed TLBO and FATLBO twice (i.e., for function f17 (Quartic) and function20 (Rosenbrok)). As far as total best solutions are concerned, ATLBO outperformed its competitors by obtaining 22 best results. TLBO and FATLBO obtained 21 best results. DE is third with 20 best results. PSO and PBA obtained 17 best results. BA is second last with 15 best results, whereas GA is last with only 9 best results. Table 8 also shows the statistical outcomes (S.O) which are recorded by employing the Wilcoxon rank-sum test with a 5% significance level. Here, the symbols +/≈/− indicate better, equal, and worse performance of the proposed algorithm against its competitor. These outcomes determine that the performance of ATLBO is comparable to existing state-of-the-art metaheuristic algorithms.

5.2. Experiment 2. The values of the parameters (A, B, C, and D) in the COCOMO II model are constants and have been less efficiently tuned based on the real effort and

Table 6: Details of global test functions.

No.	Function	Range	D	Type	Formulation	Min
1	Beale	[−4.5, 4.5]	2	UN	$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	0
2	Easom	[−100, 100]	2	UN	$f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	−1
3	Matyas	[−10, 10]	2	UN	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48 x_1 x_2$	0
4	Bohachevsky 1	[−100, 100]	2	MS	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	0
5	Booth	[−10, 10]	2	MS	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	0
6	Michalewicz 2	[0, π]	2	MS	$f(x) = -\sum_{i=1}^{D} \sin(x_i)(\sin(ix_i^2/\pi)^{20}$	−1.8013
7	Schaffer	[−100, 100]	2	MN	$f(x) = 0.5 + \sin^2\sqrt{x_1^2 + x_2^2} - 0.5/(1 + 0.001(x_1^2 + x_2^2))^2$	0
8	Six hump camel back	[−5, 5]	2	MN	$f(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	−1.03163
9	Bohachevsky 2	[−100, 100]	2	MN	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1).0.4\cos(4\pi x_2) + 0.3$	0
10	Bohachevsky 3	[−100, 100]	2	MN	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	0
11	Shubert	[−10, 10]	2	MN	$f(x) = (\sum_{i=1}^{5} i\cos(i+1)x_1 + i)(\sum_{i=1}^{5} i\cos(i+1)x_2 + i)$	−186.73
12	Colville	[−10, 10]	4	UN	$f(x) = 100(x_1^2 - x_2) + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_1 - 1)(x_4 - 1)$	0
13	Zakharov	[−5, 10]	10	UN	$f(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5ix_i)^2 + (\sum_{i=1}^{D} 0.5ix_i)^4$	0
14	Step	[−5.12, 5.12]	30	US	$f(x) = \sum_{i=1}^{D} (x_i + 0.5)^2$	0
15	Sphere	[100, 100]	30	US	$f(x) = \sum_{i=1}^{D} x_i^2$	0
16	SumSquares	[−10, 10]	30	US	$f(x) = \sum_{i=1}^{D} ix_i^2$	0
17	Quartic	[−1.28, 1.28]	30	US	$f(x) = \sum_{i=1}^{D} ix_i^4 + \text{Rand}$	0
18	Schwefel 2.22	[−10, 10]	30	UN	$f(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	0
19	Schwefel l 1.2	[−100, 100]	30	UN	$f(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	0
20	Rosenbrok	[−30, 30]	30	MN	$f_5(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	0
21	Dixon-price	[−10, 10]	30	UN	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_i - 1)^2$	0
22	Griewank	[−600, 600]	30	MN	$f(x) = 1/40000(\sum_{i=1}^{D} (x_i - 100)^2) - (\prod_{i=1}^{D} \cos(x_i - 100/\sqrt{i})) + 1$	0
23	Ackley	[−32, 32]	30	MN	$f(x) = -20\exp(-0.2\sqrt{1/n\sum_{i=1}^{D} x_i^2}) - \exp(1/n\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	0

Note. U: unimodal, D: dimensions, N: nonseparable, M: multimodal, and S: separable.

Table 7: Configuration parameters of the metaheuristic algorithms.

Algorithm	Parameters	Values
GA	Population; crossover rate mutation rate; generation gap	50; 0.8 0.01; 0.9
DE	Population; crossover rate scaling factor	50; 0.9 0.5
PSO	Population; inertia weight velocity limit	50; 0.9~0.7 $X_{\min}/10$~$X_{\max}/10$
BA	Population; no for elite bee no No for best bee; no for random bee Neighborhood size of elite bee Neighborhood of best bee	50; NP/2 NP/4; NP/4 2 1
PBA	Population; inertia weight Iteration for elite bee in PSO Iteration for best bee in PSO	50; 0.9~0.7 15 9
TLBO	Population	50
FATLBO	Population	50
ATLBO	Population	50

Table 8: Comparative results of ATLBO with FATLBO, TLBO, GA, DE, PSO, BA, and PBA.

Test function	GA			DE			PSO			BA			PBA			TLBO			FATLBO			ATLBO	
	Mean	Std	S.O	Mean	Std	S.O	Mean	Std	S.O	Mean	Std	S.O	Mean	Std	S.O	Mean	Std	S.O	Mean	Std	S.O	Mean	Std
f1	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	$1.88E-05$	$1.94E-05$	+	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f2	*−1*	*0*	≈	*−1*	*0*	≈	*−1*	*0*	≈	−0.99994	$4.50E-05$	+	*−1*	*0*	≈	*−1*	*0*	≈	−1	0	≈	−1	*0*
f3	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f4	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f5	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	0.00053	0.00074	+	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f6	*−1.8013*	*0*	≈	*−1.8013*	*0*	≈	−1.57287	0.11986	+	*−1.8013*	*0*	≈	*−1.8013*	*0*	≈	*−1.8013*	*0*	≈	−1.8013	0	≈	−1.8013	*0*
f7	0.00424	0.00736	+	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f8	*−1.03163*	*0*	≈	*−1.03163*	*0*	≈	*−1.03163*	*0*	≈	*−1.03163*	*0*	≈	*−1.03163*	*0*	≈	*−1.03163*	*0*	≈	−1.03163	0	≈	−1.03163	*0*
f9	0.06829	0.07822	+	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f10	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f11	*−186.73*	*0*	≈	*−186.73*	*0*	≈	*−186.73*	*0*	≈	*−186.73*	*0*	≈	*−186.73*	*0*	≈	*−186.73*	*0*	≈	−186.73	0	≈	−186.73	*0*
f12	0.01494	0.00736	+	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f13	0.01336	0.00453	+	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f14	$1.17E+03$	76.56145	+	*0*	*0*	≈	*0*	*0*	≈	5.12370	0.39209	+	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f15	$1.11E+03$	74.21447	+	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	*0*	*0*	≈	0	0	≈	0	*0*
f16	$1.48E+02$	12.40929	+	0	0	≈	0	0	≈	0	0	≈	0	0	≈	0	0	≈	0	0	≈	0	0
f17	0.18070	0.02712	+	0.00136	0.00042	≈	0.00116	0.00028	+	$1.72E-06$	$1.85E-06$	−	0.00678	0.00133	+	$7.49E-03$	$1.99E-03$	+	$3.52E-04$	$1.61E-04$	+	$7.19E-5$	$2.71E-5$
f18	11.0214	1.38686	+	0	0	≈	0	0	≈	0	0	≈	0	0	≈	$7.59E-10$	$7.10E-10$	+	0	0	≈	0	0
f19	$7.40E+03$	$1.14E+03$	+	0	0	≈	0	0	≈	0	0	≈	0	0	≈	0	0	≈	0	0	≈	0	0
f20	$1.96E+05$	$3.85E+04$	+	18.20394	5.03619	+	15.088617	24.170196	+	28.834	0.10597	+	4.2831	5.7877	+	$1.04E-07$	$2.95E-07$	+	$3.08E-07$	$4.84E-07$	+	$1.81E-8$	$4.21E-8$
f21	$1.22E+03$	$2.66E+02$	+	0.66667	$E-09$	+	0.66667	$E-08$	+	0.66667	$1.16E-09$	+	0.66667	$5.65E-10$	+	0.66667	0	+	0.66667	0	+	0.66667	0
f22	10.63346	1.16146	+	0.00148	0.00296	+	0.01739	0.02081	+	0	0	≈	0.00468	0.00672	+	0	0	≈	0	0	≈	0	0
f23	14.67178	0.17814	+	0	0	≈	0.16462	0.49387	+	0	0	≈	$3.12E-08$	$3.98E-08$	+	0	0	≈	0	0	≈	0	0

The entries of the table in italics represent the best values.

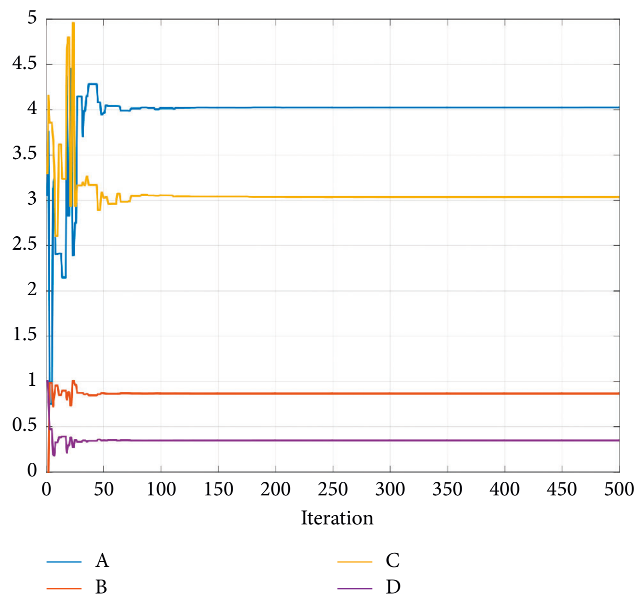

Figure 8: Convergence of the parameters.

Table 9: MRE values obtained by the three models.

Project ID	MRE of effort			MRE of time		
	TLBO	COCOMO II	ATLBO	TLBO	COCOMO II	ATLBO
3	0.0085	0.2008	0.0003	0.0722	0.0367	0.0813
13	0.3432	0.4597	0.3355	0.2285	0.2885	0.2193
15	0.1744	0.3000	0.1604	0.2029	0.2869	0.1952
16	0.0009	0.0774	0.0308	0.0010	0.2244	0.0016
22	0.1593	0.2403	0.1358	0.1449	0.0940	0.1436
23	0.0481	0.1760	0.0285	0.1121	0.0768	0.1148
28	0.1446	0.0172	0.1734	0.1588	0.0934	0.1555
29	0.1552	0.0310	0.1727	0.1197	0.0659	0.1219
31	0.0935	0.0485	0.1170	0.1450	0.0830	0.1437
32	0.0519	0.0806	0.0753	0.1449	0.0876	0.1432
34	0.1712	0.3212	0.1621	0.0805	0.0462	0.0882
35	0.0778	0.2327	0.0652	0.0893	0.0551	0.0956
36	0.2082	0.3675	0.2029	0.0622	0.1650	0.0553
37	0.0005	0.1716	0.0125	0.0468	0.1798	0.0421
39	0.1163	0.2862	0.1087	0.0610	0.1682	0.0543
40	0.2831	0.3993	0.2724	0.0315	0.1835	0.0284
44	0.0446	0.0675	0.0049	0.0362	0.2409	0.0292
47	0.2810	0.3131	0.2538	0.1502	0.3256	0.1494
56	0.2171	0.2449	0.1862	0.1656	0.1488	0.1570
58	0.5435	0.6716	0.5485	0.0006	0.0187	0.0175
61	0.3127	0.3772	0.2931	0.0488	0.2595	0.0514
69	0.0635	0.1184	0.0305	0.1592	0.1244	0.1541
70	0.0842	0.1357	0.0516	0.1304	0.1248	0.1284
72	0.1108	0.0246	0.1459	0.1359	0.1151	0.1338
73	0.0371	0.1184	0.0079	0.1305	0.1117	0.1292
76	0.0279	0.0262	0.0652	0.1865	0.1269	0.1783
77	0.4396	0.4210	0.5023	0.2201	0.1450	0.2071
93	0.1757	0.3679	0.1759	0.0835	0.1904	0.0787
MMRE	15.62%	22.49%	*15.43%*	11.25%	14.53%	*11.07%*

The number in italics represents the best values.

Table 10: PRED(30) values obtained using the three models.

Approach	Time (%)	Effort (%)
ATLBO	*100*	*89.29*
TLBO	*100*	85.71
COCOMO II	96.43	67.87

The number in italics represents the best values.

time necessary for new projects. Hence, project development activities can be predicted with low accuracy. In this work, ATLBO is adopted to optimize these coefficients by utilizing historical software projects with actual effort and time.

In the real practice of software development, estimation of the required effort and time for projects is common. An ideal estimation would offer 0% of MRE for both effort and time. Moreover, parameter effort MMRE and time MMRE are added together to constitute the fitness function (f) (equation (17)) for each learner in the population [31].

$$f = \mathrm{MMRE}(\text{effort}) + \mathrm{MMRE}(\text{time}). \tag{17}$$

Experimentation attempts to increase the accuracy of the multiplicative constants (A, B, C, and D) of the COCOMO II model by using ATLBO to obtain the best (i.e., near-to-actual) effort estimation. The "NASA 93" [51] dataset was used for conducting the experiments. The proposed algorithm was trained with 65 projects from this dataset to optimize the values of the four coefficients, whereas its performance was tested on the other 28 projects after optimization. The obtained results of ATLBO are evaluated against the results of TLBO reported in [31]. Unlike TLBO, ATLBO optimizes the coefficients for the COCOM II model with only 50 learners (i.e., population size) and 1000 iterations.

The optimized values for the parameters of the COCOMO II model obtained by using ATLBO are as follows: $A = 4.023$, $B = 0.866$, $C = 3.04$ and $D = 0.349$. The convergence of A, B, C, and D after each iteration is shown in Figure 8.

Table 9 presents the comparison of MRE time and effort values optimized by ATLBO with TLBO and the original model for the 28 projects from the test dataset. Considering these results, it can be stated that the ATLBO-based model improves the accuracy further by reducing the MRE compared to the TLBO and the original models. Moreover, MMRE values for both effort and time have also been reduced owing to the optimized parameters. Therefore, it is concluded that ATLBO for the optimization of the COCOMO II coefficients is effective. PRED (N) is another useful criterion for assessing the effectiveness of the proposed ATLBO-based model.

Table 10 presents the values of PRED (30) generated by ATLBO via equation (16). As compared to the other two models, ATLBO's results show high improvement in the accuracy of software cost estimation for both time and effort.

Despite its performance, ATLBO is not without limitations. Often, the developed fuzzy membership functions and rules are problem dependent. More precisely, there is a need to do tuning and calibration to ensure successful applications. On the positive note, tuning is desirable to ensure more control over the ATLBO's exploration and exploitation behavior depending on the problem at hand. In turn, better search performance can be obtained.

6. Conclusion

This paper presents a fuzzy-based variant of the teaching learning-based optimization (TLBO) algorithm called adaptive TLBO (ATLBO) for global optimization. ATLBO adaptively selects, based on the current search requirement, either the teacher phase (exploration) or learner phase (exploitation) by employing Mamdani-type fuzzy inference system. The high-quality solutions generated by ATLBO from the sample problems demonstrated its improved searching ability compared to other metaheuristic algorithms including the original TLBO and its variant (i.e., FATLBO). The ATLBO has generated optimum global results for 21 out of 23 benchmark test functions. Moreover, ATLBO was also adopted for the optimization of parameters in the COCOM II model. Results showed the superior performance of ATLBO with fewer learners and iterations than TLBO and the original model. The three measures, namely, quality measure, diversification measure, and intensification measure, utilized in ATLBO are simple to compute from search spaces. Therefore, it can be concluded that ATLBO, being efficient and easy to implement, is effective for addressing different numerical optimization problems.

Finally, the applicability of ATLBO with its powerful search ability will be investigated for other related optimization problems. Some of these problems include wireless sensor network localization and the generation of substitution boxes (S-boxes) in contemporary symmetric ciphers. Similarly, ATLBO will be used for solving search-based optimization problems in software engineering such as software effort estimation models and software redundancy reduction.

Data Availability

All the relevant data are available in this manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (MOE) (2021RIS-001(1345341783)).

References

[1] R. V. Rao, *Teaching Learning Based Optimization Algorithm and its Engineering Applications*, Springer, Berlin, Germany, 1 edition, 2016.

[2] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, WA, Australia, December 1995.

[4] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[5] Z. W. Zong Woo Geem, J. H. Joong Hoon Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *SIMULATION*, vol. 76, no. 2, pp. 60–68, 2001.

[6] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[7] M. Fayaz, Q. B. Pham, N. T. T. Linh et al., "A water supply pipeline risk analysis methodology based on diy and hierarchical fuzzy inference," *Symmetry*, vol. 12, no. 1, p. 44, 2020.

[8] M. Fayaz, I. Ullah, and D. Kim, "An optimized fuzzy logic control model based on a strategy for the learning of membership functions in an indoor environment," *Electronics*, vol. 8, no. 2, p. 132, 2019.

[9] S. Habib and M. Akram, "Decision-making system for washing machine using aifnn," *Mathematical Sciences Letters*, vol. 4, no. 3, p. 303, 2015.

[10] C. Jana, G. Muhiuddin, and M. Pal, "Some Dombi aggregation of Q -rung orthopair fuzzy numbers in multiple-attribute decision making," *International Journal of Intelligent Systems*, vol. 34, no. 12, pp. 3220–3240, 2019.

[11] C. Jana, G. Muhiuddin, and M. Pal, "Multiple-attribute decision making problems based on svtnh methods," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 9, pp. 3717–3733, 2020.

[12] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[13] K. Z. Zamli, F. Din, S. Baharom, and B. S. Ahmed, "Fuzzy adaptive teaching learning-based optimization strategy for the problem of generating mixed strength t-way test suites," *Engineering Applications of Artificial Intelligence*, vol. 59, pp. 35–50, 2017.

[14] F. Din, A. R. A. Alsewari, and K. Z. Zamli, "A parameter free choice function based hyper-heuristic strategy for pairwise test generation," in *Proceedings of the IEEE International Conference on Software Quality, Reliability and Security Companion Prague, Czech Republic*, pp. 85–91, Prague, Czech Republic, July 2017.

[15] F. Din and K. Z. Zamli, "Fuzzy adaptive teaching learning-based optimization strategy for pairwise testing," in *Proceedings of the Seventh IEEE International Conference on System Engineering and Technology*, pp. 17–22, Shah Alam, Malaysia, October 2017.

[16] F. Din and K. Z. Zamli, "Pairwise test suite generation using adaptive teaching learning-based optimization algorithm with remedial operator," in *Proceedings of the International Conference of Reliable Information and Communication Technology*, pp. 187–195, Kuala Lumpur, Malaysia, June 2018.

[17] M.-Y. Cheng and D. Prayogo, "A novel fuzzy adaptive teaching-learning-based optimization (FATLBO) for solving structural optimization problems," *Engineering with Computers*, vol. 33, no. 1, pp. 55–69, 2017.

[18] A. H. Gandomi and A. R. Kashani, "Construction cost minimization of shallow foundation using recent swarm intelligence techniques," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1099–1106, 2018.

[19] D. Lei, L. Gao, and Y. Zheng, "A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop," *IEEE Transactions on Engineering Management*, vol. 65, no. 2, pp. 330–340, 2018.

[20] P. Niu, Y. Ma, and S. Yan, "A modified teaching-learning-based optimization algorithm for numerical function optimization," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 6, pp. 1357–1371, 2019.

[21] B.-C. Wang, H.-X. Li, and Y. Feng, "An improved teaching-learning-based optimization for constrained evolutionary optimization," *Information Sciences*, vol. 456, pp. 131–144, 2018.

[22] A. K. Shukla, P. Singh, and M. Vardhan, "An adaptive inertia weight teaching-learning-based optimization algorithm and its applications," *Applied Mathematical Modelling*, vol. 77, pp. 309–326, 2020.

[23] Z. Li, X. Zhang, J. Qin, and J. He, "A reformative teaching–learning-based optimization algorithm for solving numerical and engineering design optimization problems," *Soft Computing*, vol. 24, 2020.

[24] H. Nenavath and R. K. Jatoth, "Hybrid SCA-TLBO: a novel optimization algorithm for global optimization and visual tracking," *Neural Computing & Applications*, vol. 31, no. 9, pp. 5497–5526, 2019.

[25] X. Jiang and J. Zhou, "Hybrid de-tlbo algorithm for solving short term hydro-thermal optimal scheduling with incommensurable objectives," in *Proceedings of the Control Conference (CCC)*, pp. 2474–2479, Xi'an, China, July 2013.

[26] S. Tuo, L. Yong, and T. Zhou, "An improved harmony search based on teaching-learning strategy for unconstrained optimization problems," *Mathematical Problems in Engineering*, vol. 2013, Article ID 413565, 29 pages, 2013.

[27] S. Deb, X.-Z. Gao, K. Tammi, K. Kalita, and P. Mahanta, "A new teaching-learning-based chicken swarm optimization algorithm," *Soft Computing*, vol. 24, no. 7, pp. 5313–5331, 2020.

[28] S. Biswas, S. Kundu, D. Bose, and S. Das, "Cooperative Co-evolutionary teaching-learning based algorithm with a modified exploration strategy for large scale global optimization," in *Proceedings of the International Conference on Swarm, Evolutionary, and Memetic Computing*, pp. 467–475, Maribor, Slovenia, July 2012.

[29] S. C. Satapathy and A. Naik, "Cooperative teaching-learning based optimisation for global function optimisation," *International Journal of Applied Research on Information Technology and Computing*, vol. 4, no. 1, pp. 1–17, 2013.

[30] F. Zou, D. Chen, R. Lu, and P. Wang, "Hierarchical multi-swarm cooperative teaching-learning-based optimization for global optimization," *Soft Computing*, vol. 21, no. 23, pp. 6983–7004, 2017.

[31] T. T. Khuat and M. H. Le, "A novel technique of optimization for the cocomo ii model parameters using teaching-learning-based optimization algorithm," *Journal of Telecommunications and Information Technology*, vol. 2016, 2016.

[32] M. Ibrahim, "A new model for software cost estimation using bat algorithm," *International Journal of Academic Research in Computer Engineering*, vol. 1, no. 1, pp. 53–60, 2016.

[33] T. T. Khuat and M. H. Le, "Applying teaching-learning to artificial bee colony for parameter optimization of software

effort estimation model," *Journal of Engineering Science & Technology*, vol. 12, no. 5, pp. 1178–1190, 2017.
[34] S. K. Sehra, Y. S. Brar, N. Kaur, and G. Kaur, "Optimization of cocomo parameters using tlbo algorithm," *International Journal of Computational Intelligence Research*, vol. 13, no. 4, pp. 525–535, 2017.
[35] D. Nandal and O. P. Sangwan, "Software cost estimation by optimizing cocomo model using hybrid batgsa algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 11, no. 4, pp. 250–263, 2018.
[36] A. A. Fadhil, R. G. H. Alsarraj, and A. M. Altaie, "Software cost estimation based on dolphin algorithm," *IEEE Access*, vol. 8, Article ID 75279, 2020.
[37] W. Rhmann, B. Pandey, and G. A. Ansari, "Software effort estimation using ensemble of hybrid search-based algorithms based on metaheuristic algorithms," *Innovations in Systems and Software Engineering*, pp. 1–11, 2021.
[38] S. Habib and M. Akram, "Neuro-fuzzy control for heater fans using anfis and nefcon," *Journal of Advanced Research in Scientific Computing*, vol. 6, no. 2, pp. 6–16, 2014.
[39] M. A. Butt and M. Akram, "A novel fuzzy decision-making system for cpu scheduling algorithm," *Neural Computing & Applications*, vol. 27, no. 7, pp. 1927–1939, 2016.
[40] S. Habib, M. Akram, and A. Ashraf, "Fuzzy climate decision support systems for tomatoes in high tunnels," *International Journal of Fuzzy Systems*, vol. 19, no. 3, pp. 751–775, 2017.
[41] K. Z. Zamli, F. Din, G. Kendall, and B. S. Ahmed, "An experimental study of hyper-heuristic selection and acceptance mechanism for combinatorial t-way test suite generation," *Information Sciences*, vol. 399, pp. 121–153, 2017.
[42] G. Shahzadi, G. Muhiuddin, M. Arif Butt, and A. Ashraf, "Hamacher interactive hybrid weighted averaging operators under fermatean fuzzy numbers," *Journal of Mathematics*, vol. 2021, Article ID 5556017, 17 pages, 2021.
[43] S. Habib and M. Akram, "Diagnostic methods and risk analysis based on fuzzy soft information," *International Journal of Biomathematics*, vol. 11, no. 8, Article ID 1850096, 2018.
[44] F. Camastra, A. Ciaramella, V. Giovannelli et al., "A fuzzy decision system for genetically modified plant environmental risk assessment using mamdani inference," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1710–1716, 2015.
[45] I. Iancu, "A mamdani-type fuzzy logic controller," in *Fuzzy Logic-Controls, Concepts, Theories and Applications*InTech, London, UK, 2012.
[46] C. P. Pappis and C. I. Siettos, "Fuzzy reasoning," *Search Methodologies*, Springer, Berlin, Germany, pp. 519–556, 2014.
[47] B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy, and R. Selby, "Cost models for future software life cycle processes: cocomo 2.0," *Annals of Software Engineering*, vol. 1, no. 1, pp. 57–94, 1995.
[48] C. Abts, B. Clark, S. Devnani-Chulani et al., *Cocomo Ii Model Definition Manual*, Rose-Hulman, Terre Haute, IN, USA, 1998.
[49] Z. Chen, T. Menzies, D. Port, and B. Boehm, "Feature subset selection can improve software cost estimation accuracy," *ACM SIGSOFT - Software Engineering Notes*, vol. 30, no. 4, pp. 1–6, 2005.
[50] M.-Y. Cheng and L.-C. Lien, "Hybrid artificial intelligence-based PBA for benchmark functions and facility layout design optimization," *Journal of Computing in Civil Engineering*, vol. 26, no. 5, pp. 612–624, 2012.
[51] T. Menzies, "The tera-promise repository for cocomo 93," 2015, http://promise.site.uottawa.ca/SERepository/datasets/cocomonasa_2.arff.