ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 9(34), DOI: 10.17485/ijst/2016/v9i34/95638, September 2016 ISSN (Online) : 0974-5645

A Multi Objective Teacher-Learning-Artificial Bee Colony (MOTLABC) Optimization for Software Requirements Selection

N. Ranjith[1*] and A. Marimuthu[2]

[1]Karpagam University, Coimbatore - 641021, Tamil Nadu, India; ranjithphd789@gmail.com
[2]Government Arts College, Coimbatore - 641018, Tamil Nadu, India; marimuthuphd789@gmail.com

Abstract

Background/Objectives: To select optimal software requirements by introducing Multi-objective Teacher-Learning-Artificial Bee Colony Optimization. **Methods/Statistical Analysis:** Teaching learning based optimization for the multi-objective software requirements selection has two objectives of minimized cost and maximum client satisfaction. Similarly the constraints namely interaction constraints and cost threshold constraints are considered. However, the efficiency of the software product development can be improved further when more efficient optimization techniques is used for the selection of software requirements along with consideration of more objectives and more constraints in larger real datasets. **Findings:** In this article, a hybrid optimization technique named Multi-Objective Teacher-Learning Artificial Bee Colony Optimization (MOTLABC) is proposed with set of multiple objectives and constraints. The objectives are minimum cost, maximum client satisfaction, minimum time consumption and maximum reliability. The constraints such as time threshold constraint, interaction constraints and cost threshold constraints are considered. The hybrid approach of MOTLABC with the above objectives improves the collection of set of needs for the development of the software. The Pareto optimal problem occurs in multi objective optimization solutions is resolved by the use of Pareto tournament function. **Improvements/Applications:** The experimental consequences prove that they obtained results perform improved than algorithms proposed in the literature.

Keywords: Artificial Bee Colony Optimization, Interaction Constraints, Software Requirements Selection, Teaching Learning Based Optimization

1. Introduction

Software product development is a difficult practice composing of computer programming, documenting, testing and bug fixing carried out by the software engineers in developing the applications and frameworks based on the client requirements. The main objective of some software concern is for providing greatest satisfying services to the clients with the consideration for the allowable resource limits. However when the number of requirements increases the task of balancing the budget and the satisfaction of clients becomes difficult. This creates a scenario where the software concerns focuses either on the budget of development on their side or the client satisfaction on the other side. In order to tackle these kinds of situations, the software requirements selection and optimization becomes more important and challenging processes for the software concerns. In requirements selection and optimization, the major assignment is to choose the optimal set of needs from the available large amount of candidates to maintain the production budget and also to satisfy the clients. A minor deviation in this process results in loss for the concern either by over-limiting budget or loss of clients due to dissatisfaction. Selecting the requirements with the minimization of implementing the expenditure and developing the client fulfillment are the two conflicting objectives and are prepared as a Multi-objective Next Release problem. The Teaching-learning

*Author for correspondence

based optimization has been developed for the multi-objective software requirement selection. It includes two constraints namely the interaction constraints and the cost threshold constraints. Teaching-learning based optimization enhances the efficiency of the software product development. However the efficiency can be further improved when the more efficient optimization algorithms like the swarm intelligent based optimizations are employed along with the consideration of more number of objectives and the more constraints for the software development.

Hence in this paper, a hybrid optimization approach called Multi-Objective Teacher Learning Artificial Bee Colony (MOTLABC) optimization is developed by combining the features of Teacher-Learning and Artificial Bee Colony. MOTLABC includes multiple objectives and constraints for the efficient requirement selection. The objectives namely minimum cost, maximum client satisfaction, minimum time consumption and maximum reliability are considered while the constraints includes the time threshold constraint, interaction constraint and cost threshold constraint are considered. The Pareto optimal problem is occurring due to the multiple objectives and the constraints are resolved by the use of a Pareto tournament function. Thus by using the multiple objectives and constraints, the proposed hybrid optimization approach of MOTLABC enhances the selection of requirements in the software product development.

In[1] proposed Teacher Learning Based Optimization (TLBO) method for non linear optimization problems. It defines how the influence of a teacher affects the output of the learners in the class. The TLBO defines the optimization problem, initialize the population size, calculate the mean of the population in teacher phase, learners increase their knowledge with the help of their mutual interactions, and it terminates the process if it reaches the criteria. The advantages are that it need not require any parameters to work, and its less computational effort and high consistency.

In[2] proposed stochastic 2m+1 PEM to solve optimization problem. The main aim of this method is to find the optimal generation output of units. In this study, IBA algorithm is used to exchange the information. It uses the 2m+1 PEM to classify the uncertainty in load demand and wind speed. The advantages of this method are effective utilization of energy resources and the reduction in emission.

In[3] defines SBSE as, an approach in Software Engineering which Search-Based Optimization (SBO) algorithms are used. This algorithm is used to identify the problems in the software engineering. It offers solution for the automated and semi automated problems which are applied throughout the software engineering lifecycle.

In[4] proposed a Modified Teaching–Learning-Based Optimization (MTLBO) algorithm. This algorithm modifies the teacher's and learner's phase of original TLBO algorithm. The author normalizes the objective functions by using the Fuzzy method. The optimal location is found by analyzing the solutions stored in the repository by using the Pareto method. This method increases the convergence velocity and accuracy of TLBO and decreases the cost and minimizes the losses. To improve the performance, additional objective functions are needed.

In[5] proposed an Improved Teaching–Learning-Based Optimization (ITLBO) for the energy management optimization. This algorithm uses different rules to generate new vectors of continuous and discrete variables. After the new vectors generation, the author uses Pareto-based approach, Fuzzy-based clustering and Niching technique to obtain better pareto-optimal solutions. This algorithm can reduce the cost and loss of the system.

In[6] proposed a modified teaching-learning algorithm where the author introduces a modified phase with the original TLA algorithm. Modified phase is used for obtaining the better optimal solutions, less computational time and improve meet in the robustness of a TLA algorithm.

In[7] proposed a Teaching–Learning-Based Optimization (TLBO) algorithm for the parameter optimization in the manufacturing industries. The author uses this algorithm to achieve an optimal parameter setting. By using this, the manufacturers can decrease the cost of the product and minimizes the loss rejections.

In[8] proposed an elitist teaching-learning-based optimization algorithm. The author implements this algorithm to find the best solutions by replacing the worst solutions based on the size of elite. This algorithm shows better performance on unconstrained optimization problems.

In[9] proposed a modified teaching–learning-based optimization algorithm. Modification is done by introducing more than one teacher for learners. Such modification in TLBO can speed up the searching process and maximizes the convergence rate. The author implements this algorithm on thermal system to customize the huge number of variables and objective functions.

In[10] proposed MQHDE technique to give the solution for MONRP. It has strengths of Quantum Computing, Differential Evolution and Genetic Algorithm. The MONRP features achieving high performance in the basis of convergence to Pareto-optimal front, good spread among the obtained Pareto-optimal front solutions. The advantages of this technique are fast convergence and obtaining more number of solutions. It is tested using Spread and Hyper Volume metrics.

In[11] introduces a method "Ant Colony System" to select the effective solution from the variety of solutions. It compares the Greedy Randomized Adaptive Search Procedure and Non-dominated Sorting Genetic Algorithm. The goal of this paper is to offer a set of solutions to the developers and stakeholders a set of possibilities satisfying several objectives of the Pareto front.

In[12] proposed a new multi objective stochastic framework based on the chance of constrained programming. This technique uses the jointly distributed random variables method, it calculates the meet of electrical and heat load requirement while maintain the cost below the specified value. The framework additionally uses the hybrid modified cuckoo search algorithm to extract the Pareto optimal surface for the minimum cost and maximize the customer satisfaction.

2. Multi-Objective Optimization Problem (MOP)

A common MOP is described as reducing (or exploiting) $F(x) = (f_1(x), f_2(x) \ldots f_k(x))$ focus to $gi(x) \leq 0$, $i = 1,2,\ldots.m$ and $h_j(x) = 0$, $j = 1,2,\ldots p$ and $x \in \Omega$. A MOP solution reduces (or exploits) the elements of a vector F(x) wherein x refers the n-dimensional decision variable vector $x = (x1, x2, \ldots, xn)$ from a few space Ω; and $g_i(x) \leq 0$ and $hj(x) = 0$ denote limitations that should be satisfied as reducing (or exploiting) F(x) and Ω includes the entire achievable x which can be utilized for satisfying an estimation of F(x).

Therefore, a MOP includes k objectives signified as k objective functions along with m inequality and p equality restrictions on the objective functions and n decision variables. The estimation function F defined as the mapping from the vector of decision variables to output vectors. The output vector which convinces (m + p) the restrictions is identified as a possible solution and the group of each possible solution composes the possible region.

2.1 Pareto Optimality

A solution $x \in \Omega$ is said to be Pareto optimal in order to Ω if and only if there is no $x' \in \Omega$ for which $v = F(x') = (f_1(x'), f_2(x'), \ldots, f_k(x'))$ controls $u = F(x) = (f_1(x), f_2(x), \ldots, f_k(x))$.

On the other hand, the solution x* is referred Pareto optimal, if there exists no other possible solution x, which would reduces a number of criterion without causing a instantaneous raise in at least one added criterion.

2.2 Pareto Dominance

A vector u = (u1,u2,....uk) is said to dominate the other vector v = (v1,v2,...vk) if and only if u is moderately less than v, i.e., $\forall\, i \in \{1,2,\ldots.k\}$, $ui \leq vi$ and $\exists\, i \in \{1,2,\ldots k\}: ui < vi$.

A solution is said to Pareto dominate the other, if the initial solution is not inferior to the subsequent solution in every objectives, and there is at least one objective where it is improved.

2.3 Pareto Optimal Set

For a given MOP, F(x), and the Pareto Optimal Set P* is denoted as follows:

$$P^* = \{x \in \Omega \mid \neg\, x' \in \Omega\ F(x') \preccurlyeq F(x)\}.$$

The Pareto optimal set consists all solutions that can satisfy the condition of Pareto dominance.

2.4 Pareto Front

For a given MOP,F(x), and Pareto Optimal Set, P*, the Pareto Front PF* is defined as

$$PF^* = \{u = F(x) \mid x \in P^*\}.$$

Pareto front is acquired while the Pareto optimal set is plotted on an objective space.

The idea of Pareto Optimality is integral to the hypothesis and explaining of MOPs.

In Multi–Objective Optimization (MOO), the two different goals are making progress towards the Pareto-optimal front and sustaining a different group of solutions in the front. Because both the goals are significant, a valuable MOO algorithm should accomplish both of them surrounded by the reasonable computational endeavor.

3. Multi-Objective Next Release Problem

This section explains the software necessities for selection procedure as a Multi-objective Next Release Problem as formulated by[13]

Given is a presented software package, there is a group C consisting of m consumers.

C = {c1, c2, c3... cm},

Whose necessities are to be measured for the next release of the product.

The group of necessities proposed by the consumers for the next release is defined by follows:

$$R = \{r1, r2, r3... r_n\} \quad (3)$$

Every consumer has a level of significance for the company based on factors such as reliability in instructions, payment conditions, and integrity etc. which be able to be exposed by a weight factor. The group of comparative weights related with every consumer cj ($1 \le j \le m$) is defined by follow:

$$Weight = \{w1, w2, w3,......,w_m\} \quad (4)$$

where wj ∊ [0,1].

For accomplishing every necessity, resources such as manpower, hardware and software tools are required, which may be converted by means of costs. The cost related with every necessity ri ($1 \le i \le n$) for its accomplishment is selected by

$$Cost = \{cost_1, cost_2, cost_3, ...cost_n\} \quad (5)$$

Since all the necessities are not regularly significant for the consumers, every consumer cj ($1 \le j \le m$) allocates a value for necessity r_i ($1 \le i \le n$), defined by value (r_i,c_j). The score of necessity r_i can be computed as

$$score_i = \sum_{j=1}^{m} w_j * value(r_i, c_j) \quad (6)$$

The decision vector x = {x_1, x_2, x_3, ...x_n } ∊{0,1} specifies the necessities which are to be incorporated in the next release of the product.

The time consumed can be designated as

$$Time = \{time_1, time_2, \ldots.., \; time_n\} \quad (7)$$

Reliability can be designated as the probability of case failures to the number of cases considered. It is measured in terms of mean time between the failures.

$$\begin{aligned} reliability_i &= \text{Mean Time Between Failure(MTBT)} \\ &= \text{Mean Time TO Failure(MTTT)} + \text{ Mean Time To Repair(MTTR)} \end{aligned}$$

The objectives to optimize minimum cost, maximum client satisfaction (score), minimum time consumption and maximum reliability can be formulated as

$$\text{minimize } f_1 = \sum_{i=1}^{n} cost_i * x_i \quad (8)$$

$$\text{maximize } f_2 = \sum_{i=1}^{n} score_i * x_i \quad (9)$$

$$\text{minimize } f_3 = \sum_{i=1}^{n} time_i * x_i \quad (10)$$

$$\text{maximize } f_4 = \sum_{i=1}^{n} reliability_i * x_i \quad (11)$$

The major goal in implementing the MONRP is for finding the group of necessities which are to be incorporated in the next release of the software product by reducing the cost and at the same time by improving the consumer fulfillment with less time consumption and highest reliability.

The problem constraints are needed to maintain the selection of optimal requirement selection. The time threshold constraint L_t, interaction constraints L_I and cost threshold constraints L_C are considered.

$$\sum_{i=1}^{n} time_i \le L_C \quad (12)$$

$$L_I = r_i \oplus r_j \quad (13)$$

$$\sum_{i=1}^{n} cost_i \le L_C \quad (14)$$

Where r_i and r_j are the requirements.

3.1 Teaching–Learning-based Optimization

Teacher-learning based optimization method[14]depends on manipulate of a teacher on the performance of students learning in a class. TLBO employs the set of learners as the population of solutions to determine the global solution. In TLBO the design variables are equivalent to special subjects of the learners while their results are equivalent to the fitness. The teacher is the most learned person the teacher is considered as the best solution initially in TLBO. TLBO performs in two phases of processing. The first phase is the 'Teacher phase' while the second phase is the 'Learner phase'. In teacher phase, learning is from the teacher and in learner phase, the learning is through interaction between the learners.

3.1.1 Initialization

N = number of learners in class.

D = number of subjects presented to the learners.

MAXIT = maximum number of permissible iterations.

Population X is randomly initialized by a search space enclosed by matrix of N rows and D columns.

The j^{th} parameter of the i^{th} learner is assigned values randomly using the Equation

$$x^0_{(i,j)} = x_j^{min} + rand \times (x_j^{max} - x_j^{min} \tag{15}$$

where rand refers a uniformly distributed random variable within the range (0, 1),x_j^{min} and x_j^{max} denotes the minimum and maximum value for j^{th} parameter.

The parameters of i^{th} learner for the generation g are

$$x^g_{(i,j)} = [x^g_{(i,1),} \; x^g_{(i,2),} \; x^g_{(i,3),\ldots.} \; x^g_{(i,j)} \ldots\ldots\ldots\ldots \; x^g_{(i,D)} \tag{16}$$

3.1.2 Teacher phase

The mean parameter M^g of each subject of the learners in the class at generation g is given as

$$m^g = [m_1^{\,g} \; m_2^{\,g} \; m_3^{\,g} \ldots\ldots\ldots \; m_j^{\,g} \; \ldots\ldots\ldots m_D^{\,g}] \tag{17}$$

The learner having least objective function value is measured as the teacher $x^g_{Teacher}$ for the particular iteration. The teacher processes the algorithm by means of changing the mean of learners towards the new teacher. In order to acquire a new group of enhanced learners, the random weighted differential vector is produced from the current mean and the preferred mean parameters which are added to the current population.

$$xnew^g_{(i)} = x^g_{(i)} + rand \times (x^g_{Teacher} - T_F M^g) \tag{18}$$

TF is the teaching factor which chooses the value of mean to be altered. Value of TF can be either 1 or 2. The value of TF is chosen randomly with the equivalent probability as,

$$TF = round\,[1 + rand(0,1)\{2-1\}] \tag{19}$$

TF is not a parameter of the TLBO algorithm. The value of TF is not known as an input in the algorithm and its value is randomly determined by the above equation. After performing the amount of experiments on the benchmark functions, it is finalized that the algorithm performs better when the value of TF is between 1 and 2. But the algorithm is established as it performs much better if the TF value is either 1 or 2. So, in order to simplify the algorithm, the teaching factor is recommended to take either 1 or 2 based on the rounding up constraints.

If $Xnew^g_{(i)}$ is established to be advanced learner than $x^g_{(i)}$ in generation g, and it substitutes the inferior learner $x^g_{(i)}$ in the matrix.

3.1.3 Learner phase

The interaction among the learners takes place in the learners phase. The mutual interaction process enhances the learner's knowledge. The random interaction among learners advances the knowledge. For a given learner $x^g_{(i)}$ another learner $x^g_{(r)}$ is randomly chosen (i ≠ r). The i^{th} parameter of the matrix x_{new} in the learner phase is expressed as

$$Xnew^g_{(i)} = \begin{cases} X^g_{(i)} + rand \times \left(X^g_{(i)} - X^g_{(r)*} \right) iff \left(X^g_{(i)} \right) < f(X^g_{(r)}) \\ X^g_{(i)} + rand \times \left(X^g_{(r)} - X^g_{(i)} \right) otherwise \end{cases} \tag{20}$$

3.2 Multi Objective Teacher-Learning-Artificial Bee Colony Optimization

Teaching Learning Based Optimization (TLBO) and Artificial Bee Colony (ABC) algorithm is the population based on modern method of optimization utilized for solving the different complex engineering and real time applications. To acquire the best solution for the complex problem, it desires extra time and consequences in performance degradation. The hybridization of both the algorithms will provide solutions for a complex problem quickly.

3.2.1 MOTLABC Algorithm

Input

Population size S, Maximum iterations K

Create a random initial population of learners x_i i€1 s,

For each k

For each s

Update learners through teachers

Select m particle randomly apply ABC algorithm

Set $p_t = x_i^{(t+1)}$ iff $(x_i^{(t+1)}) < f(p_t)$

Set g =arg min $f(p_t)$

Check termination condition

End s

End k

The convergence of OTLBO is guaranteed because of the elitism preservation strategy. A learner moves only if the movement will lower the objective function.

3.2.2 Artificial Bee Colony Algorithm for Requirement Selection

ABC approach[15] is a Meta heuristic population based method. Solution for the optimization problem is denoted by every test case. The quality of each selected requirement is computed by the fitness rate of a problem. The work proposes by using the ABC algorithm to produce an optimize requirement selection and it will include all feasible independent paths with the requirement specification. The algorithm steps as follows

Initialization Phase

Population size S, Maximum iterations K

DO AGAIN

Employed Bees Phase

Onlooker Bees Phase

Scout Bees Phase

Memorize the best solution achieved so far

UNTIL (Cycle = Maximum Cycle Number or a Maximum

CPU time)

- In initialization phase, the population of food (Solutions) is initialized by artificial scout bees and control parameters are set.
- 2. Investigate for an executable state and calculate the testnode.
- 3. Initialize the current path as cycle=1
- 4. Do again
- 5. Construct initial food sites randomly according to

Solution using:

Xij= Xjmin + rand (Xjmax-Xjmin)

Rand (0, 1)

- 6. Requirements are searched within the range of initialized boundary values.
- 7. Greedy selection process is applied for selecting requirement.
- 8. Calculate the fitness value for selected requirements.
- 9. Requirement with maximum fitness rate is selected by onlookers' bee and leaves the rest.
- 10. Similar process is continued till specific requirement with maximum fitness rate obtained.
- 11. Return the selected requirement to main algorithm.
- 12. New Test case selection by scout bee in next iteration.

4. Experimental Results

In this section, the performance evaluation of the proposed MOTLABC is performed by describing the experimental methodology, the datasets and comparing the performance results of MOTLABC with MO-TLBO method. The performances are evaluated with the interms of hyper-volume (HV) indicator, spread indicator and Number of non-dominated solutions (NDS).

4.1 Experimental Methodology

Each experiment is carried out with 100 independent runs as the optimization is a stochastic algorithm. The average results of the 100 runs are given in the following sections. The results generated by MOTLABC are compared with MOTLBO.

4.2 Dataset Description

The effectiveness of the MOTLABC algorithm is tested by using the two real time datasets. The datasets are constrained by using the four different improvement endeavor boundaries 30%, 50%, 70% and 100% in order to evaluate the proposed hybrid approach at four different instances in the dataset. The performance is also evaluated without the effort limit. The first dataset is taken from[16]. And it involves 20 necessities and 5 consumers with the 10 necessity interactions. The second dataset was proposed by[11]. It contains 100 necessities and 5 clients with the 44 necessity interactions.

The datasets includes the improvement endeavor related to every necessity, the level of priority assigned to every necessity for every consumer, and the interactions constraints. The priority level for each necessity takes the values from 1 to 5 depending on the level of significance. These values can be understood as follows: 1. Not significant necessity, 2. Insignificant necessity, 3. Significant necessity, 4. Very significant necessity, and 5. Exceedingly significant necessity. Every necessity has a related improvement cost endeavor which is predictable in terms of a score between 1 and 10. Finally, the group of suggestion and combination interactions between the necessities is also considered. Additionally every consumer has a comparative significance in the decision making which can be determined by the interaction constraints.

The normalized points used for the datasets are shown in the Table 1.

4.3 Results and Discussion

The results of the MOTLABC performance evaluated in four different instances of effort limits are compared with

MOTLBO in terms of HV indicator, spread indicator and number of NDS.

Table 1. Datasets main properties and HV reference points

Dataset 1	20 necessities, 5 consumers, 10 interactions restraints $r_{min}(cost, satisfaction) = (0,0)$ $r_{max}(cost, satisfaction) = (85,893)$
Dataset 2	100 necessities, 5 consumers, 44 interactions restraints $r_{min}(cost, satisfaction) = (0,0)$ $r_{max}(cost, satisfaction) = (1037,2656)$

4.3.1 Hyper-Volume (HV) Results

The results of MOTLABC are compared with MOTLBO in terms of HV. The results are shown in average HV and the standard deviation of 100 independent runs for the 2 datasets with the four instances of effort boundaries and without the limit. Therefore a total of 8 instances have been analyzed and the results are obtained. For a better optimization, the HV indicator must be higher as possible in order to obtain better quality results. Table 2 and Table 3 show the comparison in terms of Average HV and standard deviation for two datasets.

Table 2. Average HV and standard deviation of the results for the 4 instances of dataset 1

Dataset 1 Effort boundary	**MOTLBO Mean ± Std. Deviation**	**MOTLABC Mean ± Std. Deviation**
30%	42.981% ± 1.10e-5	54.892% ± 1.11e-5
50%	56.219% ± 2.61e-4	62.567% ± 2.12e-4
70%	62.837% ± 3.24e-4	68.769% ± 2.81e-4
Without effort limit	66.562% ± 1.14e-3	74.897% ± 1.21e-3

Table 3. Average HV and standard deviation of the results for the 4 instances of dataset 2

Dataset 2 Effort boundary	**MOTLBO Mean ± Std. Deviation**	**MOTLABC Mean ± Std. Deviation**
30%	43.182% ± 1.10e-2	55.541% ± 1.1e-2
50%	53.122% ± 2.61e-2	61.234% ± 2.21e-2
70%	59.992% ± 3.24e-3	67.879% ± 2.76e-3
Without effort limit	64.126% ± 1.14e-3	72.654% ± 1.03e-3

The obtained results show that the MOTLABC presents very low dispersions for the all endeavor boundary tested and hence it can be proved that the entire significant enhancement is achieved by using MOTLABC. It also shows that the MO-TLBO is capable to discover the search space better than the other approaches published, and therefore, the solutions provided for the necessities selection problem will be of better quality.

4.3.2 Spread Results

The spread indicator results are shown in Table 4 and 5. For obtaining the better performance, the spread indicator must be low.

Table 4. Average spread and standard deviation of the results for the 4 instances of dataset 1

Dataset 1 Effort boundary	MOTLBO Mean ± Std. Deviation	MOTLABC Mean ± Std. Deviation
30%	0.52 ± 0.01	0.48 ± 0.009
50%	0.46 ± 0.01	0.44 ± 0.01
70%	0.40 ±0.02	0.37 ±0.01
Without effort limit	0.38 ±0.04	0.35 ±0.02

Table 5. Average spread and standard deviation of the results for the 4 instances of dataset 2

Dataset 2 Effort boundary	**MOTLBO Mean ± Std. Deviation**	**MOTLABC Mean ± Std. Deviation**
30%	0.45 ± 0.02	0.40 ± 0.01
50%	0.41 ± 0.02	0.38 ± 0.01
70%	0.36 ±0.02	0.34 ±0.02
Without effort limit	0.34 ±0.03	0.31 ±0.02

The results of the proposed MOTLABC show that it provides better results than the MOTLBO in all the cases with the reduced standard deviation, thus providing distribution of solutions in all effort boundaries. It also shows that the MOTLABC provides the group of optimal solutions which present more variety than the results acquired by MOTLBO.

4.3.3 Number of NDS

The average number of NDS is compared in Tables 6 and 7. In multi-objective optimization, the more optimal solutions found, the better for the human expert when the selection of the final solution has to be performed. Hence the number of non-dominated solutions has to be compared to find its efficiency. For better results, the number of NDS must be higher.

Table 6. Average number of NDS and standard deviation of the results for the 4 instances of dataset 1

Dataset 1 Effort boundary	MOTLBO Mean ± Std. Deviation	MOTLABC Mean ± Std. Deviation
30%	15 ± 0.00	17.8 ± 0.00
50%	24.16 ± 0.18	28.76 ± 0.12
70%	32.95 ± 1.12	36.21 ± 1.00
Without effort limit	41.85 ± 1.36	48.98 ± 1.13

Table 7. Average number of NDS and standard deviation of the results for the 4 instances of dataset 2

Dataset 2 Effort boundary	MOTLBO Mean ± Std. Deviation	MOTLABC Mean ± Std. Deviation
30%	129 ± 7.57	144.25 ± 6.23
50%	136.13 ± 7.60	154.78 ± 6.57
70%	144.85 ± 7.93	160.12 ± 6.89
Without effort limit	152.55 ± 7.90	169.09 ± 7.10

The results in the tables show that MOTLABC obtains a higher number of non-dominated solutions in every case. The results show that MOTLABC provides more number of optimal set of solutions.

5. Conclusion

In this article, the hybrid approach of Multi-Objective Teacher-Learning Artificial Bee Colony (MOTLABC) optimization is proposed to resolve the software requirements selection problem. The proposed hybrid approach also includes a Pareto tournament function to resolve the Pareto optimal problem. MOTLABC includes multiple objectives and constraints such that the addition of more objectives enhances the selection process of optimal solutions. The comparison of the experimental results of MOTLABC with MOTLBO shows that the optimal set of requirements can be selected efficiently using the proposed hybrid approach of MOTLABC in all the cases. The experiments conducted on the two real time datasets proves that the MOTLABC has better performance in terms of HV indicator, spread indicator and number of NDS.

6. References

1. Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Information Sciences. 2012; 183(1):1–15.
2. Azizipanah-Abarghooee R, Niknam T. A new improved bat algorithm for fuzzy interactive multi-objective economic/emission dispatch with load and wind power uncertainty. Proceedings of the 10th International FLINS Conference; Istanbul, Turkey. 2012. p. 388.
3. Harman M, Mansouri SA, Zhang Y. Search-based software engineering: Trends, techniques and applications. ACM Computing Surveys (CSUR). 2012; 45(1):11.
4. Niknam T, Azizipanah-Abarghooee R, Narimani MR. A new multi objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems. Engineering Applications of Artificial Intelligence. 2012; 25(8):1577–88.
5. Niknam T, Azizipanah-Abarghooee R, Narimani MR. An efficient scenario-based stochastic programming framework for multi-objective optimal micro-grid operation. Applied Energy. 2012; 99:455–70.
6. Niknam T, Azizipanah-Abarghooee R, Aghaei J. A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch. IEEE Transactions on Power Systems. 2013; 28(2):749–63.
7. Rao RV, Kalyankar VD. Parameter optimization of modern machining processes using teaching–learning-based optimization algorithm. Engineering Applications of Artificial Intelligence. 2013; 26(1):524–31.
8. Rao RV, Patel V. Multi-objective optimization of two stage thermoelectric cooler using a modified teaching–learning-based optimization algorithm. Engineering Applications of Artificial Intelligence. 2013; 26(1):430–45.
9. Rao RV, Patel V. Comparative performance of an elitist teaching-learning-based optimization algorithm for solving unconstrained optimization problems. International Journal of Industrial Engineering Computations. 2013; 4(1):29–50.
10. Kumari A. Charan K. Gupta MPS. Software requirements optimization using multi-objective quantum-inspired hybrid differential evolution. EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II. Springer Berlin Heidelberg; 2013. p. 107–20.

11. del Sagrado J, del Aguila IM, Orellana FJ. Multi-objective ant colony optimization for requirements selection. Empirical Software Engineering. 2015; 20(3):577–610.
12. Azizipanah-Abarghooee R, Niknam T, Bina MA, Zare M. Coordination of combined heat and power-thermal-wind-photovoltaic units in economic load dispatch using chance-constrained and jointly distributed random variables methods. Energy. 2015; 79:50–67.
13. Zhang Y, Harman M, Mansouri AS. The multi-objective next release problem. GECCO '07: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation; ACM, New York. 2007. p. 1129–36.
14. Padma S, Latha K. Multiple-loop PI controller design for TITO system using teaching learning based optimization. Indian Journal of Science and Technology. 2016; 9(12). DOI: 10.17485/ijst/2016/v9i12/89939.
15. Saranya R, Balamurugan K, Karuppasamy M. Artificial bee colony algorithm based congestion management in restructured power system. Indian Journal of Science and Technology. 2015; 8(S7):171–8.
16. Des Greer, Ruhe G. Software release planning: An evolutionary and iterative approach. Information and Software Technology. 2004; 46(4):243–53.