



# Design of an improved teacher-learner based optimization model using particle swarm technique

## *Hybrid 3-stage TLBO-PSO (H3PT) model for system optimization*

Amreen Khan<sup>1</sup>, Dr. M. M. Raghuwanshi<sup>2</sup>, Dr. Snehlata .S. Dongre<sup>3</sup>

<sup>1</sup>Ph. D Scholar, <sup>2</sup> Professor, Supervisor, <sup>3</sup>HoD

<sup>1</sup>G.H. Raison College of Engineering, Nagpur

<sup>2</sup>Department of Computer Engineering, GHRCEM, Pune, Supervisor at G.H. Raison College of Engineering, Nagpur.

<sup>3</sup>Department of Computer Science & Engineering, G.H. Raison College of Engineering, Nagpur.

**Abstract:** Teacher-learner based optimization (TLBO) belongs to a category of swarm optimization models which are inspired by the academic learning process. In this model, each solution is categorized into either a teacher particle or a learner particle depending upon its closeness with the optimum value. Learner particles update their positions w.r.t. the information learnt from teacher particles; which assists in faster convergence. But the performance of this model is limited by the teacher particle update performance; which affects both convergence time and error performance of the model. In order to reduce the dependency of learner particle on teacher particles; this text proposes a novel 3-stage particle swarm optimization (PSO)-based TLBO model. The model is able to initialize particle positions using TLBO, optimize them using PSO, and then update these positions via TLBO's final learning. After each iteration results of the updated teacher and learner particles are provided to PSO for Global optimization. Due to this, overall efficiency of the hybrid 3-stage PSO-TLBO (H3PT) model is higher when compared with original TLBO and PSO performance. This efficiency is compared in terms of output error and convergence delay; and it is observed that the proposed model is 8% effective in terms of convergence delay; and 5% effective in terms of error performance.

**Index Terms** - TLBO, PSO, convergence, error, hybrid.

## I. INTRODUCTION

Cloud backup, conjointly Swarm optimization belongs to a category of algorithms wherein each solution to a given problem is represented in terms of particles. Each of these particles are updated with the help of stochastic mathematical identities in order to bring the particle closer to a solution. This process is repeated for multiple iterations; and at the end of the final iteration a close to optimum solution is found. This optimization is possible due to randomized solution searching with the help of iterative learning mechanism. In order to perform these optimizations a wide variety of algorithms are proposed by researchers; these include particle swarm optimization (PSO), artificial bee colony (ABC) optimization, ant colony optimization (ACO), Teacher-Learner based Optimization (TLBO), etc. These algorithms can work with good efficiency in terms of convergence delay (delay needed to find a solution), and output error (difference between expected and obtained solution). But the efficiency of these algorithms can be improved by combining the characteristics of one optimization model with other. A wide variety of such hybrid algorithms [1] have been proposed by researchers in the past, these hybrid algorithms have proven to reduce the convergence delay, or output error or both, depending upon the type of algorithms combined. An example of such a hybrid algorithm that combines PSO with ACO can be observed from Fig. 1, wherein travelling salesman problem (TSP) is solved.

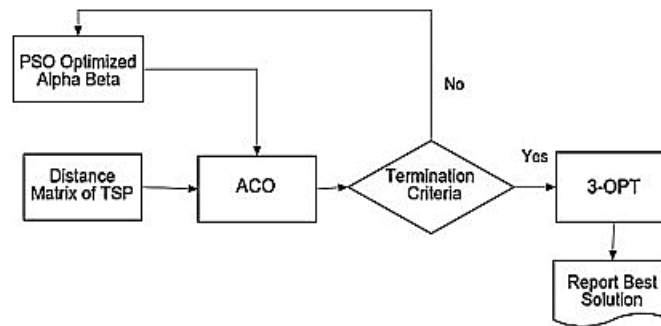


Figure 1. Hybrid combination of ACO and PSO

The model initially uses PSO to update alpha and beta parameters of ACO, and then uses these updated values for finding out a solution. If the solution is not valid, then PSO again updates these values; and re-evaluates the solution. This process continues till a particular optimization level (3-OPT) is obtained. Due to a combination of these models' overall delay in finding the best solution is reduced, thereby speeding up the system. A survey of such hybrid algorithms that are targeted for TLBO-based optimization and their performance is mentioned in the next section. This is followed by design of the hybrid TLBO PSO model, which is aimed at reducing convergence delay and output error when compared with individual TLBO and PSO algorithms. This performance is compared on different standard benchmark functions; which are designed to test efficacy of the underlying optimization models. Finally, this text concludes with some interesting observations about the hybrid TLBO-PSO model; and recommends methods to improve it.

## II. LITERATURE SURVEY

Teacher-Learner based Optimization (TLBO) algorithm is inspired from the academic learning process; wherein teachers are responsible for training the learners to get better results. In the TLBO model there are 2 phases of learning; in phase 1, teachers train the learners; while in phase 2, learners learn amongst themselves. It is observed that combination of these learning phases with other optimization models yields in better results; for instance, the work in [2] combines properties of Genetic Algorithm's mutation phase with TLBO in order to generate an improved reformative TLBO (RTLBO) algorithm. The RTLBO algorithm works in 3 phases, which are; an improved teaching phase, self-learning phase & a Genetic Algorithm (GA) mutation phase. The algorithm uses differential evolution (DE) for updating current particle position for both teacher and learner particles using the following equation,

$$X_k^{new} = X_{init} + F * (X_{prev} - X_{current}) \dots (1)$$

Where,  $X_k^{new}$  is the new position,  $X_{init}$  is the initial position, 'F' is scaling factor, while  $X_{prev}$  and  $X_{current}$  are previous and current particle positions. Due to this new location update step, the solution is able to converge quickly, and is able to demonstrate 5% improvement in terms of speed when compared with original TLBO model. A similar model is proposed in [3], wherein discrete TLBO (DTLBO) is defined. This model utilizes a discrete mean value difference (MVD) between current and previous particle positions in order to obtain the new solution. Due to inclusion of MVD in the algorithm; there is a reduction in output error. As a result of which, the proposed DTLBO model is able to reduce error by 8% when compared with Genetic Algorithm (GA), and 3% when compared with PSO models. Due to low error performance, these models can be applied to multiple signal processing applications; for instance, the work in [4] proposes the use of TLBO for image steganography. Here, the TLBO model performs efficient edge detection, due to which mean squared error (MSE), peak signal to noise ratio (PSNR), and structural similarity (SSIM) are improved, when compared with Canny and Sobel edge detection models.

Another hybrid optimization model that combines Jaya algorithm with TLBO for solving Economic Dispatch Problem under Nonconvex & Multiarea criterion is described in [5]. In this model, both Jaya and TLBO are executed separately; and a thresholding device is used to evaluate which solution out of these is better. The better solution is used for future particle learning, thereby achieving solution fusion. Due to this solution fusion as observed in Fig. 2, overall delay is increased; but accuracy of obtaining the solution is improved. It is observed that the proposed model is 15% more efficient than individual Jaya & TLBO models; thereby making it more useful for real-time deployments. A similar model that combines solutions from 2 different optimization models for generating a new & optimized solution can be observed from [6], wherein chicken swarm optimization (CSO) is combined with TLBO.

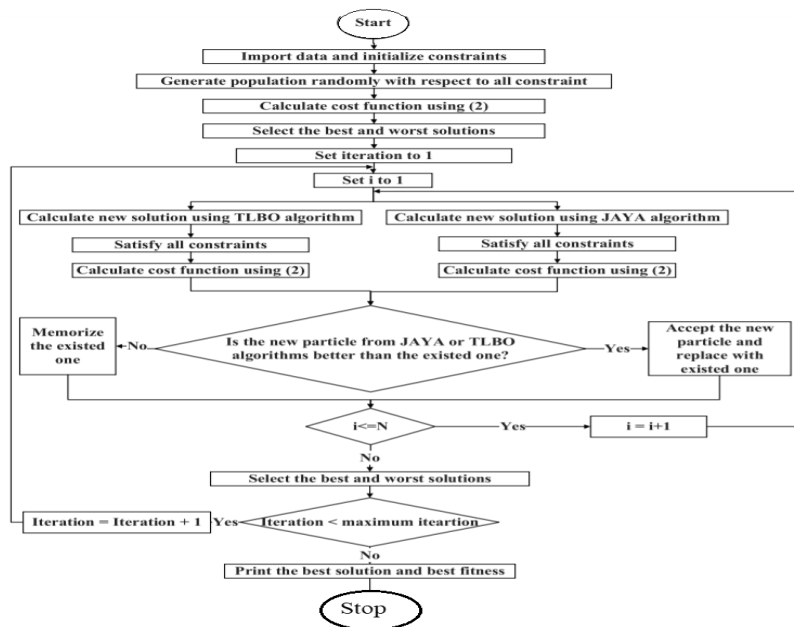


Figure 2. Fusion of Jaya with TLBO for improved accuracy [5]

Due to this combination of CSO with TLBO, accuracy improvement of 8% is achieved when compared with individual algorithms. A similar implementation can be observed from [7], wherein TLBO is modified using Lévy-Flight algorithm; which assists in self-study and reduces probability of the solution going into local minima. Moreover, this algorithm also hierarchically divides learner particles and teacher particles according to their learning ability. Particles with better learning capabilities are able to train other particles; thereby improving overall learning performance. The model outperforms TLBO algorithm in terms of overall accuracy by 8%, but has higher delay due to hierarchy division, and multiple learning criterion. This delay performance can be improved by optimizing internal steps of the TLBO model without increasing number of computations needed per particle update process. Such a model is defined in [8], wherein adaptive grasshopper optimization algorithm (GOA) is used for reducing probability of local trapping; and increased probability of global optimization. The modified GOA based TLBO's particle update process can be observed from equation 2, wherein upper bounds (UB) and lower bounds (LB) are considered while updating particle positions.

$$P_{new} = c * \left( \sum_{i=1}^N c * \frac{UB - LB}{2} * F(|P_{prev} - P_{current}|) * \frac{P_{prev} - P_{current}}{d_{pc}} \dots \right) \quad (2)$$

Where,  $P_{new}$  is the new position,  $P_{current}$  is current position,  $P_{prev}$  is previous position of the particle, while UB & LB are its upper and lower bounds; 'F' is activation function, 'c' is scaling constant, and  $d_{pc}$  is distance between current and previous particles. These particles are given to TLBO for further update; due to which overall error is reduced by 6% when compared with original TLBO model. A similar model is proposed in [9], wherein Quadratic Approximation (QA) is used for improvement of TLBO process. The model uses teacher refresh phase, which allows for better global & local search capabilities. It adds a new phase, namely random selection phase, wherein QA process is used for best case incremental location update as observed from Fig. 3. Due to which there is an improvement in overall accuracy by 5%, and reduction in convergence time by 9% when compared with the original TLBO model. These models can be applied to a variety of applications; a survey of these applications and their respective models can be observed from [10], from where it can be observed that both PSO and TLBO outperform other models for general purpose applicability. An instance of this application can be observed from [11], wherein TLBO is combined with support vector machine (SVM) for improving its recommendation performance. The SVM model is used for classification of stock index prices, and the classification accuracy is used as particle fitness value.

A solution is supposed to be optimum, only when its fitness values are high, which indicates that the selected features possess high classification accuracy. The work uses 10-day moving average, 20-day bias, Moving average convergence/ divergence, Stochastic indicators, Rate of change, Relative strength index, Commodity channel index, psychological line, Momentum, etc. for evaluation of stock values. It is observed that TLBO-SVM model outperforms original SVM model by 15%, and PSO-SVM model by 8% in terms of accuracy of stock prediction.

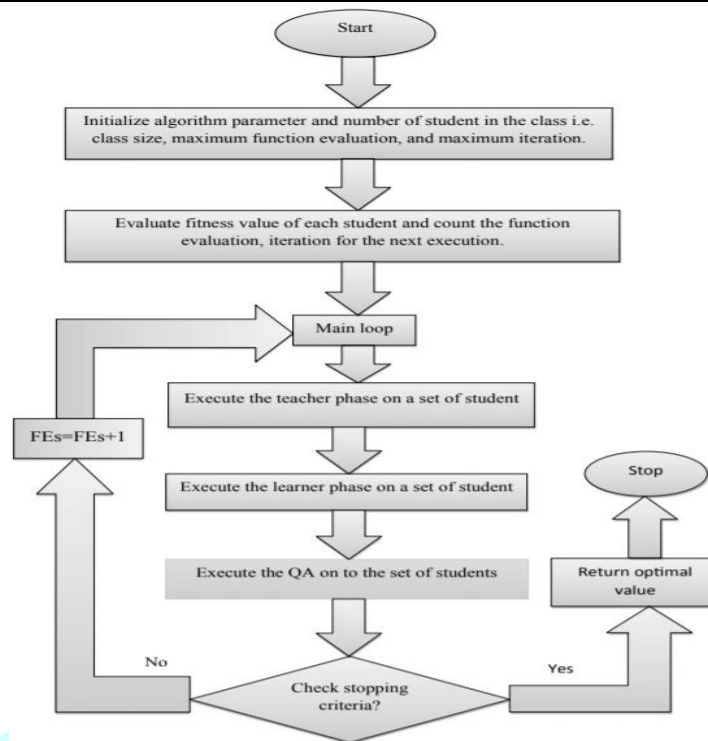


Figure 3. TLBO QA for improved accuracy and reduced convergence delay [9]

Similar application can be observed from [12], wherein protein sequence classification is done using hybrid and multi objective variants of TLBO. This variant is able to achieve a 12% performance improvement over simple TLBO thereby making it useful for real-time protein sequence classification. Another variation of TLBO in the form of master-slave learning can be observed from [13], here the masters (teacher particles) perform learning on high performance devices; while slaves (learner particles) learn on low performance devices. The master particles converge quickly, and train the slave particles for improving their convergence and error performance. Due to this segregation between performance intensive computations; these models are able to reduce convergence delay and improve overall accuracy of the TLBO process. Other variants of TLBO can be observed from [14], [15], and [16]; wherein Jaya TLBO, hybrid TLBO with particle filters, and hierarchical TLBO designs are showcased. Each of these designs have better error performance than individual optimization models. An application of such a hybrid model can be observed in [17], wherein placement & sizing of energy storage systems is done to improve their reliability. It is observed that hybrid TLBO models are able to improve overall reliability by 14% when compared with traditional non-TLBO linear optimization models. These models can also be used for route optimization as indicated in [18], wherein hybrid TLBO with variable neighborhood descent (VND) is used. This combination is able to optimize route length, and thereby reduce overall cost per route; which can be used for vehicle routing, node routing, etc. Thus, it is observed that hybrid TLBO models outperform their non-hybrid counterparts. Inspired by this observation, the next section proposes a hybrid TLBO-PSO model for improving accuracy and reducing convergence delay for various system optimization tasks. A description of these tasks is also mentioned in the consecutive section; which confirms the applicability of the proposed model for different types of applications.

### III. HYBRID 3-STAGE TLBO-PSO (H3PT) MODEL FOR SYSTEM OPTIMIZATION

Automation is vital below these ideas, the proposed 3-stage Hybrid TLBO-PSO model utilizes the teacher-learner phase in its first stage for initial particle placement (initial solution generation); and then uses PSO for cognitive and social learning; which is followed by learner-to-learner phase. This process is repeated for a large number of iterations (or until convergence) in order to obtain the final output. The working of proposed H3PT model can be described in the following 3 stages,

#### 3.1. Stage 1– Teacher to learner learning

Initially all particles are randomly placed, then based on these particle positions some particles are categorized as teachers; while others are categorized as learners. This selection is done on the basis of particle fitness error; the particles which have lowest error are selected as teachers, and other particles are selected as learners. Based on this selection, the following location update is performed.

- For each learner in 1 to N
  - Select a random teacher from the list of learned teachers which is different than this learner.
  - Let the learner position be  $P_i$  and teacher position be  $P_{rand}$ .
  - Update the learner positions using the following equation 3, and 4,

If  $F(P_{rand}) > F(P_i)$ , then

$$P_{new_i} = P_{rand} + r_1 * (P_i - P_{rand}) \dots (3)$$

$$\text{else, } P_{new_i} = P_{rand} + r_1 * (P_{rand} - P_i) \dots (4)$$

Where,  $r_1$  is a random number value for position update.

- The value of  $P_i$  is updated if,  $F(P_{new_i})$  is better than  $F(P_i)$
- Upon updating each particle, Global Best is evaluated, and error value is evaluated using equation 9.
- If this error is less than error threshold, then stop this process at the given iteration; and use these particle values as the best solution.

If the solution doesn't converge at this iteration, then go to the next phase for PSO-based update.

### 3.2 Stage 2 – Updated particle movements via PSO

In this phase, the system applies PSO-inspired particle movements for both teacher and learner particles. This phase works using the following sub-steps,

- Let the number of particles be 'N'
- Let the positions of each particle be  $P_1, P_2, P_3, \dots, P_N$
- Let the local best position of each particle be the current particle position itself ( $P_{best_i} = P_i$ ), while the global best position be position of the first particle ( $G_{best} = P_1$ )
- Let number of iterations be  $N_i$ , and minimum error needed be  $Err_{min}$ , and current error be  $E_c = Err_{min} + 1$ , and  $P_c = E_c$
- For each iteration in 1 to  $N_i$
- Calculate the velocity of each particle using the following equation 5,

$$V_{i_{new}} = w_i * V_{i_{old}} + w_c * r_1 * (P_{best_i} - P_i) + w_s * r_2 * (G_{best} - P_i) \dots (5)$$

Where,  $i = 1$  to  $N$ ,  $V_{i_{new}}$  and  $V_{i_{old}}$  are the particle's new and old velocities, while  $r_1$  and  $r_2$  are stochastic random numbers between the range (0, 1).

- Each particle position is updated using the following equation 6,
- $$P_i = P_i + V_{i_{new}} \dots (6)$$
- At the end of current iteration, update the values of  $P_{best_i}$  and  $G_{best}$  using the following equation 7 and 8,

$$P_{best_{i_{new}}} = F(P_i) \text{ if } F(P_i) > P_{best_i};$$

$$\text{else } P_{best_{i_{new}}} = P_{best_i} \dots (7)$$

$$G_{best} = \bigcup_{i=1}^N P_{best_{i_{new}}} \dots (8)$$

Where, 'F' is the fitness function; or equation of the task being solved.

- Also, evaluate iteration error using the following equation,

$$Err_i = |F(G_{best}) - V_{exp}| \dots (9)$$

Where,  $V_{exp}$  is expected value of solution from training set.

- If this error is less than error threshold, then stop this process at the given iteration; and use these particle values as the best solution.
- If the solution doesn't converge at this iteration, then go to the next phase for fine tuning particle positions.

### 3.3 Stage 3 – Learner to learner (LTL) learning

In this phase, the concept of learner-to-learner learning from TLBO is used. Here, each particle is termed as a learner, and each learner aims at learning from a randomly selected learner from the entire cohort. This LTL process is executed using the following steps,

- For each learner in 1 to N
- Select a random learner from the cohort which is different than this learner.
- Let these learner positions be  $P_i$  and  $P_{rand}$  for current and randomly selected learner respectively.
- Update the learner positions using the following equation 10, and 11,

*If  $F(P_{rand}) > F(P_i)$ , then*

$$P_{new_i} = P_{rand} + r_2 * (P_i - P_{rand}) \dots (10)$$

$$\text{else, } P_{new_i} = P_{rand} + r_2 * (P_{rand} - P_i) \dots (11)$$

Where,  $r_2$  is a random value for position update.

- The value of  $P_i$  is updated if,  $F(P_{new_i})$  is better than  $F(P_i)$
- Upon updating each particle, Global Best is evaluated, and error value is evaluated using equation 10.
- If this error is less than error threshold, then stop this process at the given iteration; and use these particle values as the best solution.
- If the solution doesn't converge at this iteration, then go to the next phase for fine tuning particle positions.

Repeat this process from stage 1, until either all iterations are completed; or the error has reduced below the given threshold. Due to combination of PSO inside the TLBO process, overall efficiency of the proposed model is improved when compared with original PSO & TLBO models. This comparison can be observed from the next section.

## IV. RESULT ANALYSIS AND COMPARISON

In order to evaluate the performance of proposed H3PT model, it was evaluated against 10 different benchmark functions. All these functions have a minimum optimum solution; therefore, the algorithm can be used in its original form as described in previous section. The minima benchmark functions used are, sphere (f1), separable ellipsoid with monotone transformation (f2), Rastrigin with monotone transformation separable condition (f3), skew Rastrigin-Bueche (f4), linear slope (f5), attractive sector function (f6), step-ellipsoid (f7), Rosenbrock (non-rotated) (f8), Rosenbrock (rotated) (f9), and, ellipsoid with monotone transformation (f10). For comparison, the following parameters were considered; and were evaluated for PSO, TLBO, and H3PT models.

- Convergence Error: This is minimum error obtained by the algorithm at the end of the final iteration.
- Convergence delay: Is the number of iterations needed by the algorithm for achieving this minimum error.

In order to compare the performance of PSO (P), TLBO (T), H3PT (H), these results are tabulated. The following table 1 indicates the results of convergence error for each of these algorithms,

Table 1. Error performance of different algorithms on benchmark functions

B	P	T	H
f1	0.0	0.0	0.0
f2	200	0.0	0.0
f3	0.0	0.0	0.0
f4	0.0	0.0	0.0
f5	0.0	0.0	0.0
f6	0.0	0.0	0.0
f7	0.0	0.0	0.0
f8	0.0	0.0	0.0
f9	0.0	0.0	0.0
f10	0.0	0.0	0.7

Similarly, the convergence delay of these algorithms is tabulated in table 2 as follows,

Table 2. Convergence performance of different algorithms on benchmark functions

B	P	T	H
f1	49	44	22
f2	50	50	50
f3	50	50	34
f4	50	50	39
f5	50	7	10
f6	50	50	50
f7	23	50	50
f8	50	50	49
f9	50	50	50
f10	50	50	50

The proposed model 10% lower convergence delay when compared with PSO; and 12% lower delay when compared with TLBO this can also be observed from Fig. 4.

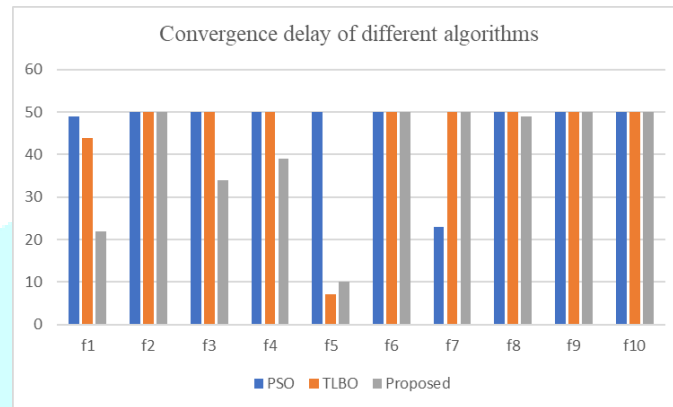


Figure 4. Convergence delay of different algorithms across various objective functions

Due to this low delay, this model is suited for high speed and low error applications. Moreover, this model can also be used for highly accurate classification, clustering, and other applications where the amount of data to be processed is large, and data processing has to be done at high data rates. Thus, it can be observed that the proposed H3PT model has better performance in terms of accuracy and convergence delay when compared with PSO and TLBO models. Thereby the proposed model can be used for high efficiency and high-speed applications for overall system optimization.

## V. CONCLUSION AND FUTURE WORK

It can be observed that the proposed model outperforms standard PSO and TLBO models in terms of convergence delay and accuracy values. This is due to the fact that it combines strengths of both PSO & TLBO models; by initially obtaining particle positions via TLBO's teacher learner phase, then optimizing these positions via PSO's cognitive and social learning; and finally using learner-to-learner phase the positions are corrected. This 3-stage fused combination allows the particles to move faster, and move towards the solution with minimum error. The algorithm converges faster due to addition of an extra PSO stage, but doesn't require more delay because the PSO stage increases particle step size; thereby making the particle converge to the solution quicker than existing models. In future, researchers can fuse a greater number of models like Elephant Hoarding Optimization (EHO), Whale Optimization (WO), etc. in order to further improve overall efficiency of TLBO and its counterparts in terms of accuracy and convergence delay.

## REFERENCES

- [1] Farivar, F., Shoorehdeli, M.A. & Manthouri, M. 2020. Improved teaching–learning based optimization algorithm using Lyapunov stability analysis. *J Ambient Intell Human Comput.* <https://doi.org/10.1007/s12652-020-02012-z>
- [2] Li, Z., Zhang, X., Qin, J. et al. 2020. A reformative teaching–learning-based optimization algorithm for solving numerical and engineering design optimization problems. *Soft Comput* 24, 15889–15906. <https://doi.org/10.1007/s00500-020-04918-4>
- [3] Masdari, M., Barshandeh, S. 2020. Discrete teaching–learning-based optimization algorithm for clustering in wireless sensor networks. *J Ambient Intell Human Comput* 11, 5459–5476. <https://doi.org/10.1007/s12652-020-01902-6>
- [4] Dhanasekaran, K, Anandan, P, Kumaratharan, N. 2020. A robust image steganography using teaching learning based optimization based edge detection model for smart cities. *Computational Intelligence*; 36: 1275– 1289. <https://doi.org/10.1111/coin.12348>
- [5] M. J. Mokarram, T. Niknam, J. Aghaei, M. Shafie-khah and J. P. S. Catalão, 2019. Hybrid Optimization Algorithm to Solve the Nonconvex Multiarea Economic Dispatch Problem, in *IEEE Systems Journal*, vol. 13, no. 3, pp. 3400-3409, doi: 10.1109/JSYST.2018.2889988.
- [6] S. Deb, K. Tammam, X. Gao, K. Kalita and P. Mahanta, 2020. A Hybrid Multi-Objective Chicken Swarm Optimization and Teaching Learning Based Algorithm for Charging Station Placement Problem, in *IEEE Access*, vol. 8, pp. 92573-92590, doi: 10.1109/ACCESS.2020.2994298.
- [7] Z. Cui, C. Li, W. Dai, L. Zhang and Y. Wu, 2020. A Hierarchical Teaching-Learning-Based Optimization Algorithm for Optimal Design of Hybrid Active Power Filter, in *IEEE Access*, vol. 8, pp. 143530-143544, 2020, doi: 10.1109/ACCESS.2020.2995716.
- [8] H. Zhang, Z. Gao, X. Ma, J. Zhang and J. Zhang, 2019. Hybridizing Teaching-Learning-Based Optimization With Adaptive Grasshopper Optimization Algorithm for Abrupt Motion Tracking, in *IEEE Access*, vol. 7, pp. 168575-168592, 2019, doi: 10.1109/ACCESS.2019.2954500.
- [9] Nama S., Saha A.K., Sharma S. (2020) A Hybrid TLBO Algorithm by Quadratic Approximation for Function Optimization and Its Application. In: Balas V., Kumar R., Srivastava R. (eds) *Recent Trends and Advances in Artificial Intelligence and Internet of Things*. Intelligent Systems Reference Library, vol 172. Springer, Cham. [https://doi.org/10.1007/978-3-030-32644-9\\_30](https://doi.org/10.1007/978-3-030-32644-9_30)
- [10] Parouha, RP, Verma, P. 2020. An innovative hybrid algorithm to solve nonconvex economic load dispatch problem with or without valve point effects. *Int Trans Electr Energ Syst.* <https://doi.org/10.1002/2050-7038.12682>
- [11] Das, S.P., Padhy, S. 2018. A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting. *Int. J. Mach. Learn. & Cyber.* 9, 97–111 (2018). <https://doi.org/10.1007/s13042-015-0359-0>
- [12] González-Álvarez, David & Vega-Rodríguez, Miguel A. & Rubio-Largo, Alvaro. 2015. Finding Patterns in Protein Sequences by Using a Hybrid Multiobjective Teaching Learning Based Optimization Algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics.* 12. 656-666. 10.1109/TCBB.2014.2369043.
- [13] Mane, Sandeep & Adamuthe, Amol & Omame, Rajshree. 2020. Master-Slave TLBO Algorithm for Constrained Global Optimization Problems. *ICST Transactions on Scalable Information Systems.* 10.4108/eai.26-5-2020.166292.
- [14] Mane, Sandeep & rama narasingarao, Manda. 2020. A chaotic-based improved many-objective Jaya algorithm for many-objective optimization problems. *International Journal of Industrial Engineering Computations.* 12. 49 - 62. 10.5267/j.ijiec.2020.10.001.
- [15] L. R. Rodrigues, D. B. P. Coelho and J. P. P. Gomes, 2020. A Hybrid TLBO-Particle Filter Algorithm Applied to Remaining Useful Life Prediction in the Presence of Multiple Degradation Factors, *IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1-8, doi: 10.1109/CEC48606.2020.9185898.
- [16] Z. Cui, C. Li, W. Dai, L. Zhang and Y. Wu, 2020. A Hierarchical Teaching-Learning-Based Optimization Algorithm for Optimal Design of Hybrid Active Power Filter, in *IEEE Access*, vol. 8, pp. 143530-143544, doi: 10.1109/ACCESS.2020.2995716.
- [17] Lata, P, Vadhera, S. 2020. TLBO-based approach to optimally place and sizing of energy storage system for reliability enhancement of radial distribution system. *Int Trans Electr Energ Syst.* 30:e12334. <https://doi.org/10.1002/2050-7038.12334>
- [18] Liu, Fuying & Liu, Chen & Zhao, Qi & He, Chenhao. 2021. A Hybrid Teaching-Learning-Based Optimization Algorithm for the Travel Route Optimization Problem alongside the Urban Railway Line. *Sustainability.* 13. 1408. 10.3390/su13031408.