Int. j. adv. multidisc. res. stud. 2023; 3(4):943-947

Received: 27-06-2023
Accepted: 07-08-2023

International Journal of Advanced Multidisciplinary Research and Studies

ISSN: 2583-049X

Enhanced Binary Tree Growth Algorithm with Linear Discriminant Analysis for Leukemia Dataset Classification

[1] Suzan Muhsen Al-Saffar, [2] Omar S Qasim
[1, 2] Department of Mathematics, University of Mosul, Mosul, Iraq

Corresponding Author: **Suzan Muhsen Al-Saffar**

Abstract
In this study, a novel meta-heuristic method named the Quasi-Opposition-Based Learning Tree Growth Algorithm (QOBL-TGA) is introduced to overcome the issues of slow convergence and local optima commonly associated with the original Tree Growth Algorithm (TGA). By combining the quasi-opposition (QOBL) and opposition-based learning (OBL) techniques, the QOBL-TGA approach enhances the global search capabilities of TGA. To assess its effectiveness, the QOBL-TGA is applied to the leukemia dataset using the Linear Discriminant Analysis (LDA) classifier. The results demonstrate that the QOBL-TGA approach surpasses the original TGA and other existing algorithms, exhibiting faster convergence and superior optimization performance. In summary, the QOBL-TGA method presents a valuable solution for addressing optimization challenges in high-dimensional problems.

Keywords: Feature Selection, Classification, Quasi Opposition-Based Learning, Tree Growth Algorithm, Leukemia

1. Introduction

Feature selection (FS) is a crucial strategy in developing an accurate educational system, which relies on the specific circumstances at hand. It is recognized as one of the most significant approaches for achieving this goal [1]. As an optimization problem in a high-dimensional space, feature selection can be viewed as a combinatorial task, as an exhaustive search would not yield the optimal subset of features. By employing this subset, we create a distinct set from the default dataset, which facilitates achieving maximal separation in the feature space. The fitness function measures the effectiveness of a classifier or another criterion while considering the computational cost of extracting value. It is often regarded as the ideal balance between classification accuracy and efficiency [2].

The utilization of Quasi Opposition Based Learning (QOBL), a widely recognized technique, has enhanced algorithms rooted in natural processes. Our study demonstrates that quasi-opposite points exhibit a higher probability of proximity to the solution compared to opposing points [3].

The Tree Growth Algorithm (TGA) is a versatile optimization method known for its remarkable characteristics, enabling it to address a wide range of problems. One of TGA's key features is the promotion of cooperation among four tree groups, striking a balance between exploration and exploitation [4]. Unlike its focus on discrete problems like feature selection, TGA prioritizes issues stemming from continuous development. This approach enhances the precision of existing tree growth models and subset data, paving the way for further discussions. Wang's algorithm (TGA) metaheuristics have also been employed to improve the performance of the Binary Tree Growth Algorithm (BTGA) in addressing the feature selection problem. By implementing the proposed QOBL-TGA algorithm, we achieved a significant performance boost for BTGA in this context [5].

The key outcome of this research is an enhanced TGA algorithm that incorporates the QOBL approach to generate an optimal initial ensemble. The utilization of OBL has also yielded improved solutions in another domain. When compared to the standard procedure, the proposed approach outperforms in terms of performance metrics such as accuracy and the size of the selected subset of features. Swarm algorithms and feature selection have garnered significant attention from researchers in recent years due to their broad range of potential applications. For instance, in 2018 [5], Jingwei *et al*. introduced two novel feature selection approaches based on thermo gravimetric analysis (TGA) for categorizing EMG signals. In 2019, Ivana *et al*. proposed a modification to the Tree Growth Algorithm, incorporating dynamic adjustments to search criteria for exploration and exploitation [6]. Furthermore, Changkang *et al*. proposed a new and improved TGA (iTGA) to address the challenges of feature selection and parameter tuning, and a comprehensive analysis revealed its effectiveness [7].

The structure of the article is as follows: Section II introduces the theoretical aspects. Section III describes the proposed algorithm. Experimental results and discussions are provided in Section IV. Finally, Section V presents the conclusion.

2. The Feature Selection (FS)

Within the BTGA framework, the initial step involves discretizing the dataset from a continuous space, followed by the selection of features using a meta-heuristic procedure. One approach that can be employed in this context is to uncover connections between the dataset's features. The objective of these algorithms and approaches is to enhance classification performance by minimizing the presence of unwanted features in the dataset [8].

A. Quasi Opposition-Based Learning (QOBL)

Opposition-based learning (OBL), an inventive concept in computational intelligence introduced by Tizhoosh [9], has been employed to accelerate the convergence rate of diverse optimization techniques. By simultaneously considering the current population and its opposing population, OBL aims to discover improved candidate solutions. This approach can be utilized by any population-based optimization method to expedite the convergence process. Researchers have found that alternative candidate solutions are more likely to be closer to the overall best solution [10].

In OBL, the concepts of the opposing number and opposite point are as follows:

Opposite number: When viewed from the hub of the search area, it acts as a mirror image of the solution. This can be written as a mathematical expression:

$$x^o = a+b-x \quad (1)$$

Where (a, b) represents the beginning and end of the search space.

Opposite point: IP (x_1, x_2,…., x_d) is a d-dimensional search space point, and its inverse point OP(x_1^o, x_2^o,…,x_i^o, x_d^o) may be defined as follows:

$$x_i^o = a_i + b_i - x_i \; ; x_i \epsilon \; [a,b]; \qquad i = 1,2,\dots,d \quad (2)$$

Quasi-opposition-based learning (QOBL), which was developed by many researchers [11], shown that a quasi-opposite point has a higher probability of being close to the solution than an opposite point. The following definitions apply to the quasi-opposite number and point used in QOBL:

Quasi-opposite number: It is the number that lies between the center (c) of the search area and the opposite number. A quasi-opposite number x^{qo} can be written mathematically as:

$$x^{qo} = rand\left(\frac{a+b}{2}, a+b-x\right) \quad (3)$$

Quasi-opposite point: The quasi opposite point QOP $\left(x_1^{qo}, x_2^{qo}, \dots\dots, x_i^{qo}, \dots . x_d^{qo}\right)$ for d-dimensional search space is given by:

$$x_i^{qo} = rand\left(\frac{a_i+b_i}{2}, a_i+b_i-x_i\right) \quad (4)$$

B. Tree Growth Algorithm (TGA)

The Tree Growth Algorithm (TGA) is an innovative meta-heuristic optimization algorithm inspired by nature. It was developed by Cheraghalipour and colleagues in 2018. Within this algorithm, the jungle is divided into four distinct groups of trees [12]. The first group represents the best trees, which experience gradual growth. The second group consists of light trees that compete with each other, with each tree situated next to two neighboring trees. The third group, known as the "remove or replace" group, suggests cutting down the worst trees and replacing them with new ones. The final group is the reproduction group, where new plants are propagated from the best trees. The TGA algorithm is described in detail [13].

Initially, a population of trees is randomly generated during the first stage. The fitness of each tree is subsequently evaluated, and the population is sorted in ascending order based on their fitness values. The top N trees are then selected to form the first tree group. The equation below illustrates the composition of the newly formed tree group:

$$x_i^{t+1} = \frac{x_i^t}{\emptyset} + r x_i^t \quad (4)$$

In this equation, t stands for number of repetitions, r for a random number between [0, 1], x_i for the population's tree at order r, and rp for the tree's power reduction rate. If the newly produced tree has a higher fitness value, the existing tree will be replaced. The current tree is preserved for the following generation in all other circumstances

$$D_i = \left(\sum_i^{N_1+N_2}\left(X_{N_2}^t - X_i^t\right)^2\right)^{\frac{1}{2}} \quad (5)$$

Where x_i stands for the current tree in the population, and x_{N2} stands for the ith tree, keep in mind that the distance has a formula $^{X_{N_2}=X_i}$ where N_2= i. The current tree then faces competition from nearby trees for light. The linear combination of the two closest trees can be found by using the following formula:

$$\gamma = \lambda T_1 + (1-\lambda)T_2 \quad (6)$$

The second group of trees undergoes an update in their positions, taking into account the proximity of trees T_1 and T_2, with the variable λ governing the extent of their influence. The following equation represents the update process for the trees in the second group [14]:

$$X_{N2}^{t+1} = X_{N2}^t + \alpha\gamma \quad (7)$$

The variable α denotes the range of angles, ranging from 0 to 1. In the third group, the N_3 trees with the lowest fitness values are removed, and new trees (new solutions) are introduced in their place. The determination of N_3 is performed using the following calculation method:

$$N_3 = N - N_1 - N_2 \quad (8)$$

Both the total number of trees in the first group N_1 and the total number of trees in the second group N_2 are equivalent to the population size N.

Within the final group, a set of new trees N_4 is generated, centered around the top-performing trees, utilizing the mask operator from the first tree group. It is important to note that

the combined values of N_1 and N_2 must not exceed the value of N_4.

Subsequently, the population is expanded by including the newly generated N_4 trees. The combined population is then sorted based on their fitness values. This process is repeated iteratively until the final criterion is satisfied. Ultimately, through this iterative procedure, the best tree in the population is identified as the optimal solution [15].

C. Binary Tree Growth Algorithm (BTGA)

The literature suggests that employing a transfer function is a highly effective method for transforming a continuous optimizer into a binary optimizer. The transfer function translates the probability value based on the tree's position, with higher probabilities indicating a higher likelihood of selecting a particular feature. In the context of feature selection in BTGA, the transfer function is defined as follows, as mentioned in:

$$S(X_{id}^t) = \frac{1}{1 + e^{-X_{id}^t}} \quad (9)$$

The search space's dth dimension is denoted as X. Through the utilization of transfer functions, the position is converted into a probability value ranging from 0 to 1. Subsequently, the tree's position is adjusted based on this probability value, following a specific procedure [5].

$$X_{id}^{t+1} = \begin{cases} 1, if\ \delta < \delta^t \\ 0, otherwise \end{cases} \quad (10)$$

δ is any number in the range of 0 and 1. The fourth tree group employs a mask operation, as was discussed in TGA. How the mask functions is seen in the BTGA picture

New Tree, S_t	1	0	0	1	0
Mask operator	0	1	1	0	1
Random best tree, X_K	1	0	1	0	1
New Tree after mask operator	1	0	1	1	1

Fig 1: An example of mask operation

The procedures for TGA and BTGA exhibit general similarities. Initially, a population of trees (solutions) is randomly generated. The fitness values of each tree are evaluated, and the trees are sorted in ascending order accordingly. Notably, the largest tree on earth is also selected. Equation (4) dictates the creation of N_1 new trees (trial trees) by the first group (1). To convert the new trees in both the first and second groups into binary form, Equation (9) is applied to them. If a new tree possesses a higher fitness value, it replaces the current tree; otherwise, the current tree is retained for the subsequent generation. Equation (2) is employed to identify the two closest neighbors of each tree from the second group. Following that, Equations (4) and (5) govern the update of the positions of the N2 trees (6). In the third group, N_3 trees are removed, and new trees are planted in their place. The fourth group is formed by generating N_4 new trees using the mask operation centered around the top trees in the first group. Subsequently, these newly generated N_4 trees are incorporated into the population. Afterward, the top N trees are retained for the subsequent iteration once the merged population has been ranked [16].

The process continues to repeat until the termination condition is satisfied. Eventually, the best overall tree is identified as the optimal feature subset. Algorithm1 presents a detailed description of the complete TGA execution process.

Algorithm 1: Binary Tree Growth Algorithm (BTGA)

1. Start
2. Set parameter $N_1, N_2, N_4, \theta, \lambda$ & initialize population
3. Evaluate fitness of trees & sort it from best to worst
4. For t=1 to max iteration, T
5. for i=1 to N_1
6. Generate trial tree, X_i^{trial} using Eq.(4)
7. convert the X into binary
8. if F(X_i) is better then F(X_i^{old})then
9. $X_i^{new} = X_i$ else $X_i^{new} = X_i^{old}$
10. end if
11. next i
12. for i= N_1 +1 to N_1+N_2
13. Find two nearest trees with shortest evaluation.
14. Generate new tree, X_i^{new} using Eq.(7)
15. compute the probalbility using Eq. (9)
16. Convert the X_i^{new} into binary form with Eq.(10)
17. Evaluate fitness, F(X_i^{new})
18. next i
19. for i= N_1+N_2+1 to N
20. Discard worst tree and evaluate fitness, F(X_i^{new})
21. for i=1 to N_4, creat new tree, S_i randomly
22. next i
23. Perform mask operator between X_k and S_i.
24. Add new population S^{new} into current population
25. Rank the population & select the best N trees.
26. End

3. The Proposed Algorithm (QOBL-TGA)

The authors of the article propose the utilization of a quasi-oppositional teaching learning-based tree growth optimization algorithm (QOBL-TGA) by incorporating the concept of quasi-opposition (QOBL) into the original TGA. This section focuses on implementing QOBL_TGA to address the feature selection problem, where the presence of a "1" bit in feature identification indicates a selected feature, while a "0" bit signifies an unselected feature. Quasi-opposition is generally employed in two areas to enhance algorithm performance. the first use is to use it in population initialization. Second, it can be incorporated into the algorithm produce dynamic jumps during position updating, which can improve the algorithm's ability to explore and keep it from succumbing to local optimization. The QOBL TGA algorithm has the benefit of ensuring more space for potential solutions through the concept of QOBL, and this favorably reflects the algorithm's performance by offering a set of suggested solutions that are put to the test by the fitness function in each iteration. Notably, the proposed QOBL-TGA algorithm distinguishes itself from other algorithms by testing two distinct solution sets (X and Y) in

each iteration, saving time and reducing computational complexity. In our proposed approach, the classifier LDA was employed. The experimental results demonstrated the effectiveness of the proposed method compared to traditional approaches in terms of classification error rate and the number of selected features.

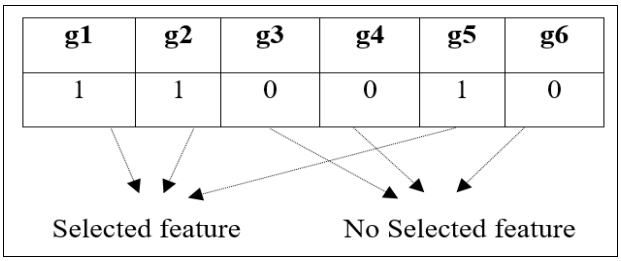

Fig 2: Explains the feature selection mechanism in OBL-BGTA

In QOBL-TGA, in order to obtain the classification accuracy, the classifier LDA is used, since it is used in the fitness function, which is illustrated in the following form [17].

$$Fitness\ Fun = w_1 * AC + (1 - w_1) * \left(\frac{F_0}{F_s - F_0}\right)$$
$$, w_1 \in [0,1] \quad (11)$$

AC denotes classification accuracy, F_O denotes the chosen feature, F_s denotes dataset features, and w_1 is the random parameter matching to the AC weight.

The following is a list of the key steps in the suggested approach:

Step 1: Create the initial population (P) at random using all of the individuals.

Step 2: Using Eq. (4), produce a quasi-opposite population (QOP).

Step 3: All the members of the current population and the quasi-opposite population set should have their fitness values evaluated.

Step 4: From the population (P) and quasi-opposite population (QOP), choose np (population size) of the fittest people as the starting population (P).

Step 5: The new set of randomly generated solutions is used to replace the impractical ones.

Step 6: Based on a jumping rate (i.e., jumping probability), after generating new populations by TGA algorithm, the opposite population is calculated. Quasi opposite population generation using jumping rate may be described as below:

Step 7: Choose np fittest individuals from current population and the quasi-oppositional population.

Select the best features from the Leukemia dataset by using BTGA, where bit "1" indicates a feature that has been selected, while a bit "0" a feature that has not been selected

Step 8: The computation is stopped if the halting requirement is met, and the results are presented; otherwise, move on to Step 5.

4. Experimental Results and Discussion

To assess the effectiveness of the proposed method, we will employ the leukemia dataset [17], which consists of 76 cases and includes a total of 7129 characteristics. Both the BTGA and QOBL_BTGA algorithms will be applied, and the classifier LDA will be used as it is widely recognized for its accuracy. All datasets used in the tables are binary and were obtained from the UCI repository. The leukemia dataset was divided into two sets: the training set, comprising 80% of the data, and the test set, comprising the remaining 20%.

Table 1: Comparing the classification accuracy of leukemia training data in the LDA classifier

Methods	Training data			Features
	ACC %	SE %	SD %	
QOBL_BTGA	92.59	0.9286	0	3493.6
BTGA	92.59	0.9286	0	3524
Original	92.59	0.9286	0	7129

Table 2: Comparing the classification accuracy of leukemia testing data in the LDA classifier

Methods	Testing data			Features
	ACC %	SE %	SP %	
QOBL_BTGA	81.81	0.6000	0	3493.6
BTGA	72.72	0.7500	0	3524
Original	72.72	0.7500	0	7129

The proposed QOBL-BTGA algorithm was compared with BTGA in terms of the number of proposed features and classification accuracy by the LDA classifier, through the tables (I and II).

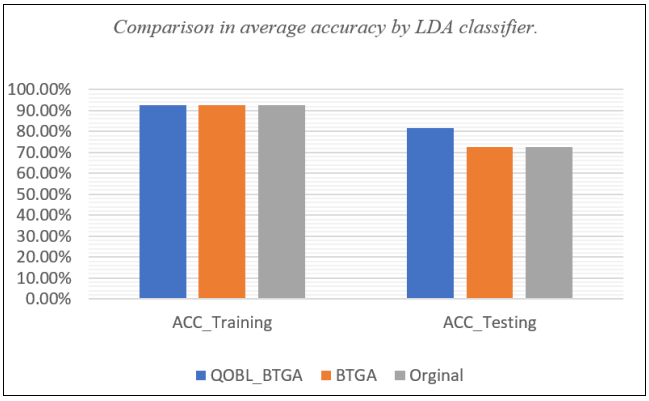

Fig 1: Comparison of average accuracy results in the LDA classifier

We note from the results obtained that the of the Leukemia dataset through the proposed method QOBL BTGA have given the fewest features compared to other methods. The proposed method QOBL-TGA uses the Quasi Opposition-Based Learning technique to prepare the initial population in a global search space, and this reflects the ability of the proposed algorithm to find the best possible solutions. In addition, the precision of the outcomes achieved by using the proposed strategy is noted. QOBL-TGA has outperformed the accuracy of the results of the standard conventional method, in which the concept of feature selection is not used in the standard method. For example, in the table data II, result precision of the proposed algorithm QOBL-TGA was 81.81 %, and the accuracy of the results of the BTGA algorithm was 72.72 %, while the accuracy of the results of the original method was 72.72 %. We also take note of the fact that the total number of features in the dataset is 7129, while the number of the features selected using the BTGA is 3524, and the number of features that were selected using the proposed algorithm QOBL-TGA is

3493.6, which is the lowest compared to the rest of the methods, and this shows the importance of the proposed algorithm in classifying datasets.

5. Conclusion and Discussion

In conclusion, the proposed Quasi-Opposition-Based Learning Tree Growth Algorithm (QOBL-TGA) has exhibited notable enhancements over the original TGA algorithm. Through of the QOBL technique, the QOBL-BTGA method has effectively addressed concerns related to slow convergence and local optima in the TGA algorithm. This has resulted in improved convergence speed and optimization performance. The experiments conducted on the leukemia dataset using the LDA classifier have demonstrated the superiority of the proposed QOBL-TGA algorithm in terms of performance metrics. The statistical results strongly suggest that the proposed method has the potential to tackle a wide range of classification problems. Therefore, the QOBL-BTGA approach can be a valuable tool for resolving optimization issues in high-dimensional problems, and can be further applied to other real-world problems in various domains.

6. References

1. Al Tawil A, Sabri KE. A feature selection algorithm for intrusion detection system based on moth flame optimization, in 2021 International Conference on Information Technology (ICIT), 2021, 377-381.
2. Zhang Y, Liu R, Wang X, Chen H, Li C. Boosted binary Harris hawks optimizer and feature selection, Engineering with Computers. 2021; 37:3741-3770.
3. Tripathy S, Debnath MK, Kar SK. Optimal design of PI/PD dual mode controller based on quasi opposition-based learning for power system frequency control, e-Prime-Advances in Electrical Engineering, Electronics and Energy. 2023; 4:p100135.
4. Cheraghalipour A, Hajiaghaei-Keshteli M, Paydar MM. Tree Growth Algorithm (TGA): A novel approach for solving optimization problems, Engineering Applications of Artificial Intelligence. 2018; 72:393-414.
5. Too J, Abdullah AR, Mohd Saad N, Mohd Ali N. Feature selection based on binary tree growth algorithm for the classification of myoelectric signals, Machines. 2018; 6:p65.
6. Strumberger I, Tuba E, Zivkovic M, Bacanin N, Beko M, Tuba M. Dynamic search tree growth algorithm for global optimization, in Technological Innovation for Industry and Service Systems: 10th IFIP WG 5.5/SOCOLNET Advanced Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2019, Costa de Caparica, Portugal, May 8-10, 2019, Proceedings 10, 2019, 143-153.
7. Zhong C, Chen Y, Peng J. Feature selection based on a novel improved tree growth algorithm, International Journal of Computational Intelligence Systems. 2020; 13:247-258.
8. Karimi F, Dowlatshahi MB, Hashemi A. SemiACO: A semi-supervised feature selection based on ant colony optimization, Expert Systems with Applications. 2023; 214:p119130.
9. Tizhoosh HR. Opposition-based learning: A new scheme for machine intelligence, In International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06), 2005, 695-701.
10. Joshi SK. Chaos embedded opposition-based learning for gravitational search algorithm, Applied Intelligence. 2023; 53:5567-5586.
11. Si T, Patra DK, Mondal S, Mukherjee P. Segmentation of breast lesion in DCE-MRI by multi-level thresholding using sine cosine algorithm with quasi opposition-based learning, Pattern Analysis and Applications. 2023; 26:201-216.
12. Balasubbareddy M, Dwivedi D. Optimal power flow solution for multi-fuel system using tree growth algorithm, Pramana Res J. 2019; 9:186-196.
13. Strumberger I, Tuba E, Bacanin N, Jovanovic R, Tuba M. Convolutional neural network architecture design by the tree growth algorithm framework, in 2019 International Joint Conference on Neural Networks (IJCNN), 2019, 1-8.
14. Cheraghalipour A, Paydar MM, Hajiaghaei-Keshteli M. Designing and solving a bi-level model for rice supply chain using the evolutionary algorithms Computers and Electronics in Agriculture. 2019; 162:651-668.
15. Kamel S, Jurado F, Sultan H, Menesy A. Tree growth algorithm for parameter identification of proton exchange membrane fuel cell models, 2020.
16. Commenges D, Alkhassim C, Gottardo R, Hejblum B, Thiébaut R. Cytometree: A binary tree algorithm for automatic gating in cytometry analysis, Cytometry Part A. 2018; 93:1132-1140.
17. Qasim OS, Algamal ZY. A gray wolf algorithm for feature and parameter selection of support vector classification, International Journal of Computing Science and Mathematics. 2021; 13:93-102.