# Novel multiobjective TLBO algorithms for the feature subset selection problem

Hakan Ezgi Kiziloz[a,*], Ayça Deniz[b], Tansel Dokeroglu[a], Ahmet Cosar[b]

[a] *Computer Engineering Department, Turkish Aeronautical Association University, Ankara, Turkey*
[b] *Computer Engineering Department, Middle East Technical University, Ankara, Turkey*

**Abstract**

Teaching Learning Based Optimization (TLBO) is a new metaheuristic that has been successfully applied to several intractable optimization problems in recent years. In this study, we propose a set of novel multiobjective TLBO algorithms combined with supervised machine learning techniques for the solution of Feature Subset Selection (FSS) in Binary Classification Problems (FSS-BCP). Selecting the minimum number of features while not compromising the accuracy of the results in FSS-BCP is a multiobjective optimization problem. We propose TLBO as a FSS mechanism and utilize its algorithm-specific parameterless concept that does not require any parameters to be tuned during the optimization. Most of the classical metaheuristics such as Genetic and Particle Swarm Optimization algorithms need additional efforts for tuning their parameters (crossover ratio, mutation ratio, velocity of particle, inertia weight, etc.), which may have an adverse influence on their performance. Comprehensive experiments are carried out on the well-known machine learning datasets of UCI Machine Learning Repository and significant improvements have been observed when the proposed multiobjective TLBO algorithms are compared with state-of-the-art NSGA-II, Particle Swarm Optimization, Tabu Search, Greedy Search, and Scatter Search algorithms.

*Keywords:* Teaching learning based optimization, Multiobjective feature selection, Supervised learning

## 1. Introduction

With the recent improvements in science and technology, huge amounts of data is being generated everyday. The size of data is larger than a human can process without help of an intelligent system [1]. This exploding growth of data makes researchers search for new methods to extract meaningful information. Effective decision-making requires high quality in information/knowledge [2]. However, it becomes harder to extract meaningful information as the amount of raw input data increases. If the raw input data is not preprocessed (e.g.

*Corresponding author.

*Email addresses:* hakanezgi@etu.edu.tr (Hakan Ezgi Kiziloz), ayca.deniz@metu.edu.tr (Ayça Deniz), tansel.dokeroglu@thk.edu.tr (Tansel Dokeroglu), cosar@ceng.metu.edu.tr (Ahmet Cosar)

filtering), it may have adverse effects and mislead the decision making processes. This creates a rapidly increasing demand for advanced data processing techniques such as data mining and machine learning.

Data mining identifies the existing patterns that might help predict future behaviours. In addition to data mining techniques, machine learning techniques are also widely used in modern decision making process. Data mining modifies data by filtering, formatting, etc., whereas machine learning techniques benefit from historical data to build a smart model [3]. Large amounts of data can be analyzed in a limited time by using machine learning techniques.

Researchers agree on the fact that preprocessing enables data mining tools to perform more effectively [4]. One of the most commonly applied data preprocessing techniques is Feature Subset Selection (FSS), which is the process of reducing the number of features by identifying irrelevant or redundant attributes of a dataset that do not affect or make no contribution to the solution of the problem [5]. However, in the meantime, we should minimize any loss of critical information. Machine learning algorithms will, naturally, execute faster when the amount they process is decreased by using FSS. The accuracy of the results may also improve in some cases [6]. As data grow massively, FSS becomes indispensable in order to be able to extract meaningful information. FSS algorithms are widely applied in various real-world problems such as text categorization and recommendation systems [7][8][9].

FSS is a multiobjective optimization process with two objectives, maximizing the accuracy of the results and minimizing the number of features. Therefore, there can be a set of solutions rather than a single one. The set of solutions serves both objectives and cannot dominate each other. For example, a solution may have an accuracy value of 0.85 with five features whereas another solution may have an accuracy value of 0.75 with three features. The first solution provides a better result for the first objective (higher accuracy) and the second one is better for the second objective (minimum number of features). Figure 1 presents an example of pareto- optimal set of solutions for FSS in Binary Classification Problems (FSS-BCP).

In this study, we propose a set of novel multiobjective TLBO algorithms for the FSS-BCP. TLBO has been recently introduced as a novel metaheuristic that has an algorithm-specific parameterless concept [10][11]. During the optimization process, TLBO does not require any parameters to be optimized. Population size, number of generations, elite size, etc. are the common control parameters that need to be tuned by all of the population based metaheuristics (including TLBO). In addition to these parameters, Particle Swarm Optimization (PSO) uses inertia weight, social and cognitive parameters, Genetic Algorithms use crossover and mutation rate, Artificial Bee Colony uses number of bees, Harmony Search uses harmony memory consideration rate, pitch adjusting rate, and the number of improvisations. The optimal tuning of these parameters is crucial for successful optimization, otherwise we might unnecessarily increase the computational effort or get stuck at local optimal solutions. On the other hand, TLBO requires only the common control parameters to be tuned. The TLBO algorithm resembles a classroom environment of a teacher and learners/students. The algorithm has two phases: Teacher phase and Learner phase. In the first phase, individuals in the classroom (population) are evaluated and the best one is selected as teacher. Then,

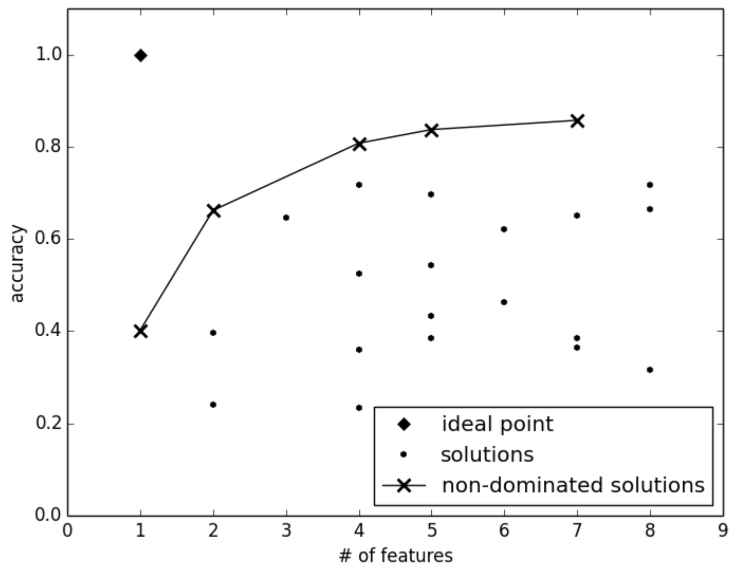


Figure 1: Non-dominated solutions fitting to a pareto curve for the multiobjective FSS problem.

each learner is trained by the selected teacher. In the second phase, learners interact with each other and train themselves. This iteration continues until the termination criteria is fulfilled.

Remarkable results have been reported about the performance of the TLBO in comparison with the other metaheuristics on many different constrained benchmark functions, constrained mechanical design problems and on continuous non-linear numerical optimization problems in terms of computational efficiency and also solution quality. Our proposed multiobjective TLBO algorithms use different selection mechanisms to construct the pareto-optimal set of solutions. Learners are trained by using recombination operators before they are given to a machine learning technique. The recombination operators do not require any parameter settings in accordance with the parameterless optimization concept of TLBO. There is also no need to select and apply an additional selection mechanism such as roulette wheel, tournament, or truncation.

Main contributions of our study are as follows. We introduce three novel multiobjective TLBO algorithms for FSS, which have different update mechanisms to find pareto-optimal set of solutions. To the best of our knowledge, the approaches we propose are implemented for the first time in FSS domain. We evaluate the proposed TLBO algorithms using three supervised machine learning techniques. Comprehensive experiments are carried out on the well-known machine learning datasets of UCI Machine Learning Repository and significant improvements are observed when the proposed algorithm is compared with state-of-the-art PSO, Tabu Search (TS), Greedy Search (GS), and Scatter Search (SS) based algorithm. Experiment results also show that the proposed TLBO algorithms obtain similar/better solutions when compared to NSGA-II based FSS algorithm.

The rest of the manuscript is organized as follows. Related studies about FSS and TLBO algorithm are given in Section 2. In Section 3, FSS-BCP is defined formally. In Section 4, proposed multiobjective TLBO algorithms and applied machine learning techniques (Logistic Regression, Support Vector Machines, and Extreme Learning Machine) are explained. Experimental environment and obtained results are presented in Section 5. Concluding remarks and future works are given in the last section.

## 2. Related Work

In this section, we give information about FSS and TLBO algorithms. FSS has been an ongoing research topic for many decades. Dash and Liu conduct a survey on FSS methods [12]. After giving a definition of FSS by discussing previous definitions of many other authors, the procedure of a typical FSS is explained. It is stated that when selecting a specific method for the problem, the guideline given in the paper is practical. A very recent survey conducted by Xue et al. [13] includes comprehensive evaluations on the FSS problem. They examine several evolutionary methods in literature by reviewing how and which analysis techniques are used and their number of objectives. The challenges and contributions of several FSS algorithms are presented. Moreover, it is stated that by reducing the number of dimensions, FSS improves the accuracy of classification.

Many different algorithms have been proposed to solve the FSS problem. Yang and Honavar [14] propose an algorithm that combines a genetic algorithm for finding a suitable subset with a neural network algorithm for classification, DistAI. The tests executed on benchmark datasets show that it improves the results obtained from DistAI by using all features (without subset selection). A state-of-the-art description of FSS problem is given by Inza et al. [15] and they present FSS by Estimation of Bayesian Network Algorithm. It is an evolutionary and randomized search algorithm that can be applicable when there is limited information about domain as it is derived from Estimation of Distribution Algorithm. Naive-Bayes and ID3 learning algorithms are used in experiments. As a result of the experiments, FSS does not affect accuracy significantly; however, it reduces CPU execution times dramatically. A genetic algorithm that optimizes the process of FSS and setting SVM parameters is proposed by Huang and Wang [16]. It is compared with the Grid Algorithm which is mostly applied for parameter searching. The experiments on 11 known real-world datasets present that this approach significantly affects the accuracy of classification in a favorable way.

Cervante et al. [17] combine PSO with two information metrics, Mutual Information and Entropy. Benefiting each measure, relevance and redundancy of the selected subsets are examined and they are used for fitness evaluation. For classification, they use Decision Trees. Experiments on benchmark datasets show that minimizing mutual information usually results in selecting a smaller feature subset; on the other hand, maximizing group entropy obtains higher accuracy. Unler and Murat [18] propose a PSO algorithm. In this study, features are selected according to two properties which are independent likelihood and predictive contribution to the feature subset that is already chosen. It is stated that they developed this algorithm for binary classification problems and they applied Logistic Regression as a machine learning technique. The evaluations of this algorithm presents that this adaptive feature selection algorithm performs better than TS and SS algorithms. Lopez et al. [19] propose a Parallel SS method for the FSS problem. In order to produce new feature subsets as solutions, they make use of greedy approach. The results show that the performance of this parallelized algorithm is better than Sequential SS. In order to solve the problem of feature selection for LR models, a TS method is proposed by Pacheco et al. [20]. The statistical comparisons with the classic ones support that the new method generates a better set of solutions than the other ones. However, more computation time is required.

Mlakar et al. [21] propose an efficient feature selection system that is applied to a Facial Expression Recognition (FER) system. The proposed system is based on a histogram of oriented gradient descriptor and difference feature vectors. The emotion feature selection is carried out by using a multi-objective differential evolution algorithm. Zhang et al. [22] present a multi-objective particle swarm optimization (PSO) algorithm for cost-based feature selection problems. In order to improve the exploration capability of the proposed algorithm, a probability-based encoding technology and an effective hybrid operator, together with the ideas of the crowding distance, the external archive, and the Pareto domination relationship, are implemented. Yong et al. [23] focus on tackling the feature selection problem with unreliable data. The problem is formulated as a multi-objective optimization one with objectives, the reliability and the classification accuracy. A novel effective multi-objective feature selection algorithm based on bare-bones particle swarm optimization is proposed by incorporating two new operators.

A multiobjective evolutionary algorithm is presented by Khan and Baig [24]. They apply NSGA-II, a multiobjective genetic algorithm, on four datasets obtained from UCI database. The results of the experiments show that NSGA-II is a promising algorithm for the FSS problem. They use ID3 as classifier and maximize both first class and second class accuracy values. A Multiobjective Differential Evolution is proposed by Sikdar et al. [25] for FSS and classifier altogether. Their objectives are adjusted as minimizing the number of features and maximizing the f-measure value. For the experiments, they use three biomedical datasets. Xue and Zhang [26] introduce multiobjective approach into PSO for the feature selection problem. In this recent study, they describe two PSO algorithms and make a comparison against two existing single objective PSO algorithms. They also compare their proposal algorithms against three existing multiobjective evolutionary algorithms. As a result of the experiments, the performance of first proposed algorithm is better than single objective methods and it obtains comparable results against multiobjective algorithms; whereas the other algorithm performs better than all mentioned algorithms.

TLBO is a recent optimization algorithm introduced by Rao et al. [10]. Later, TLBO is tested on different benchmark datasets in another study by Rao and Savsani [11]. Results present that it is more efficient than some other population based optimization algorithms. Another study by Rao and Patel [27] investigates the effects of population size and number of generations on the performance of the algorithm. They suggest that this algorithm can be easily applied on various optimization problems. Črepinšek et al. [28] use TLBO to solve the exact problems given in [10] and [11] and they state that those results are not reproducible.

Nayak and Rout [29] implement a type of multiobjective TLBO. For each objective, they create a matrix of solutions. Teachers are chosen according to the best solution in their related matrix of solutions and learners are taught only for maximization of that objective. Finally, they sort all solutions in all matrices and create a pool of optimal solutions. Similar to this approach, Xu et al. [30] present a multiobjective TLBO with a different teaching method. Instead of using a scalar function, they use crossover operator between solutions in both teaching and learning phases.

Dokeroglu [31] proposes a hybrid TLBO algorithm that merges TLBO and Robust TS. He runs the proposed algorithm both sequentially and parallel. Tests are executed on 126 instances of real-life Quadratic Assignment Problems and reported that 102 of them are solved optimally using the sequential algorithm, and 115 of them solved optimally by using the parallel TLBO algorithm. The performance of the TLBO algorithm is tested on combinatorial optimization problems, flow shop (FSSP) and job shop scheduling problems (JSSP) by Baykasoglu and Hamzadayi [32]. The performance of TLBO algorithm on these problems gives an idea about its possible performance for solving combinatorial optimization problems. Experimental results show that the TLBO algorithm has a considerable potential when compared to the best-known heuristic algorithms for scheduling problems. Niknam et al. [33] propose a new multiobjective optimization algorithm based on modified TLBO optimization algorithm in order to solve the optimal location of automatic voltage regulators in distribution systems at presence of distributed generators. The objective functions including energy generation costs, electrical energy losses and the voltage deviation are considered.

## 3. Feature Subset Selection Problem

FSS can be defined as a process of choosing a subset of features from a larger set of features. By reducing the number of features in a dataset, FSS can prevent complicated calculations, and hence, classifiers run much faster. There are many conceptually different definitions for FSS in the literature [12]. While some deal with reducing the size of selected subset, others care much about improving prediction accuracy. Essentially, FSS is constructing an effective subset that represents the dataset most informatively by eliminating irrelevant or redundant features. The main idea is finding the minimum number of features while keeping the classification accuracy (increasing it if possible). Since extracting the optimal feature subset is a challenging process and there is no exact polynomial time algorithm for solving it, FSS is known to be an NP-hard problem [34]. A typical FSS follows four steps [12]. In the first step, a search strategy selects candidate features and constitutes the subsets. These subsets are evaluated in the second step, and compared with each other. Third step, determines whether termination condition is fulfilled, or repeats first two steps, otherwise. The final step is to check whether optimal feature subset is found using apriori knowledge.

***Problem Definition:*** There are two main parts in our study; selecting the best feature subset and evaluating its performance. Since there are two objectives, FSS should be regarded as a multiobjective problem. Equation 1 gives a formal definition to find optimal solutions by satisfying both objectives.

$$
\begin{aligned}
& \min(f_1) \\
& \max(f_2) \\
\textit{subject to} & \\
& f_1 = |k| \\
& f_2 = accuracy(k) \quad where\ k \subseteq K
\end{aligned}
\tag{1}
$$

where $k$ is a subset of original dataset ($K$) which optimizes both objectives ($f_1$ and $f_2$). In the second part, quality of selected subset of features is evaluated by using a well-known performance metric, *Accuracy*, as given in Equation 2. To calculate *Accuracy*, correctly classified instances (true positives and true negatives) should be divided by all instances (true positives (TP), false positives (FT), false negatives (FN) and true negatives (TN)).

$$
Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2}
$$

## 4. Proposed Algorithms and Applied Machine Learning Techniques

In this section, we give information about the representation of the problem solution, operators (crossover and mutation), proposed multiobjective TLBO algorithms and applied machine learning techniques.

### *4.1. Problem Representation and TLBO Multiobjective Optimization Operators*

TLBO algorithm is implemented at the FSS phase of the proposed algorithms. TLBO algorithms start by randomly generating an initial population (set of students and the teacher). The population is the set of solutions. Every solution in the population (classroom) is called an individual or a chromosome (see Figure 2 for the structure of a chromosome). A feature gene of a chromosome is assumed to be selected if its value is 1, whereas the value 0 denotes

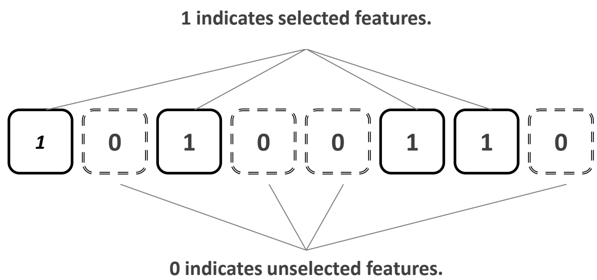


Figure 2: Chromosome structure of a solution for the FSS.

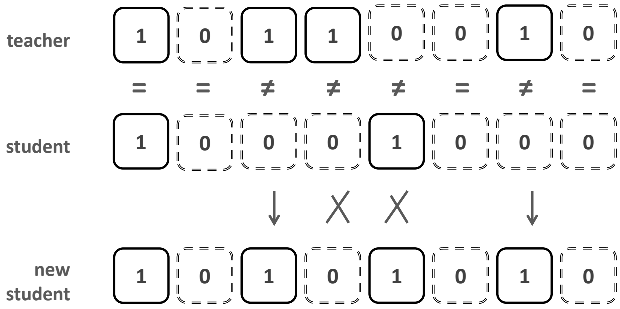


Figure 3: Crossover operator for the FSS

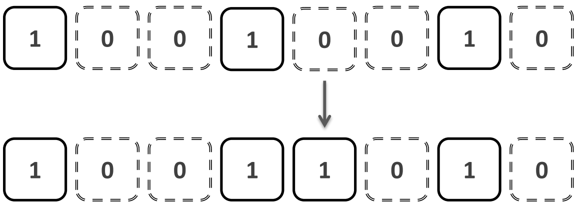


Figure 4: Mutation operator for the FSS

an unselected feature. In Figure 2, the dataset has eight features and the first, third, sixth and seventh features are selected for the solution of the problem.

TLBO algorithms run through iterations in which, the best individual in the population is defined as teacher and each remaining individual becomes a student. After selecting the teacher, TLBO works in two phases: teacher and learner phases. In teacher phase, the teacher shares its knowledge with every student and tries to improve their knowledge level. In the learner phase, students randomly interact with each other and try to improve their knowledge levels.

We used a special crossover operator called half uniform crossover and bit-flip mutation operators to generate new chromosomes in our proposed TLBO algorithms (see Figures 3 and 4). For the crossover operator, two parent chromosomes are required. Parent chromosomes may either be a teacher and a student, or two students. Crossover operator uses the information of both parent chromosomes. If a feature gene is the same in both parents, it is kept, whereas it randomly chooses a parent's gene for every different feature gene. One new chromosome is generated after this operation. Bit-flip mutation operates on a single chromosome and changes a single gene with respect to a probabilistic ratio. If the gene value is zero, then its value is updated as one, or vice versa.

**Algorithm 1:** MTLBO-ST Algorithm

Generate_population(*population*);<br>
Calculate_weighted_average_of_individuals (*population*);<br>
**for** *(*k*:=1 to* number_of_generations*)* **do**<br>
    $X_{teacher}$:= Best_individual (*population*);<br>
    /* Learning from Teacher */<br>
    **for** *(*i*:=1 to* number_of_individuals*)* **do**<br>
        $X_{new} := Crossover(X_{teacher}, X_i)$;<br>
        $X_{new} := Mutation(X_{new})$;<br>
        **if** *($X_{new}$ is better than $X_i$)* **then**<br>
            $X_i := X_{new}$;<br>
    /* Learning from Classmates */<br>
    **for** *(*i*:=1 to* number_of_individuals*)* **do**<br>
        $m$:=Select_random_individual_from (*population*);<br>
        $n$:=Select_random_individual_from (*population*); /* $n \neq m \neq teacher$*/<br>
        $X_{new} := Crossover(X_m, X_n)$;<br>
        $X_{new} := Mutation(X_{new})$;<br>
        **if** *($X_{new}$ is better than $X_m$)* **then**<br>
            $X_m := X_{new}$;<br>
        **if** *($X_{new}$ is better than $X_n$)* **then**<br>
            $X_n := X_{new}$;<br>
Show_the_pareto_optimal_set(*population*);

### *4.2. Proposed Multiobjective TLBO Algorithms*

In a multiobjective optimization process, finding the best solution or deciding whether the new individual (solution) has improved is not a straightforward process. An improvement in one objective may result in a massive decrement on the other objective. We implement three different approaches for solving this problem. The proposed algorithms are defined in the following subsections.

***Multiobjective TLBO with Scalar Transformation (MTLBO-ST)***

The first approach is suggested by Rao et al. [35]. In this approach, objective values are normalized and combined into a single scalar value. Therefore, the name of this approach is chosen as Multiobjective TLBO with Scalar Transformation (MTLBO-ST). The scalar value is used for determining better individuals and replacing them with worse individuals in the classroom (population). Later, the classical TLBO algorithm is executed (see Figure 5). Algorithm 1 presents the details of MTLBO-ST algorithm.

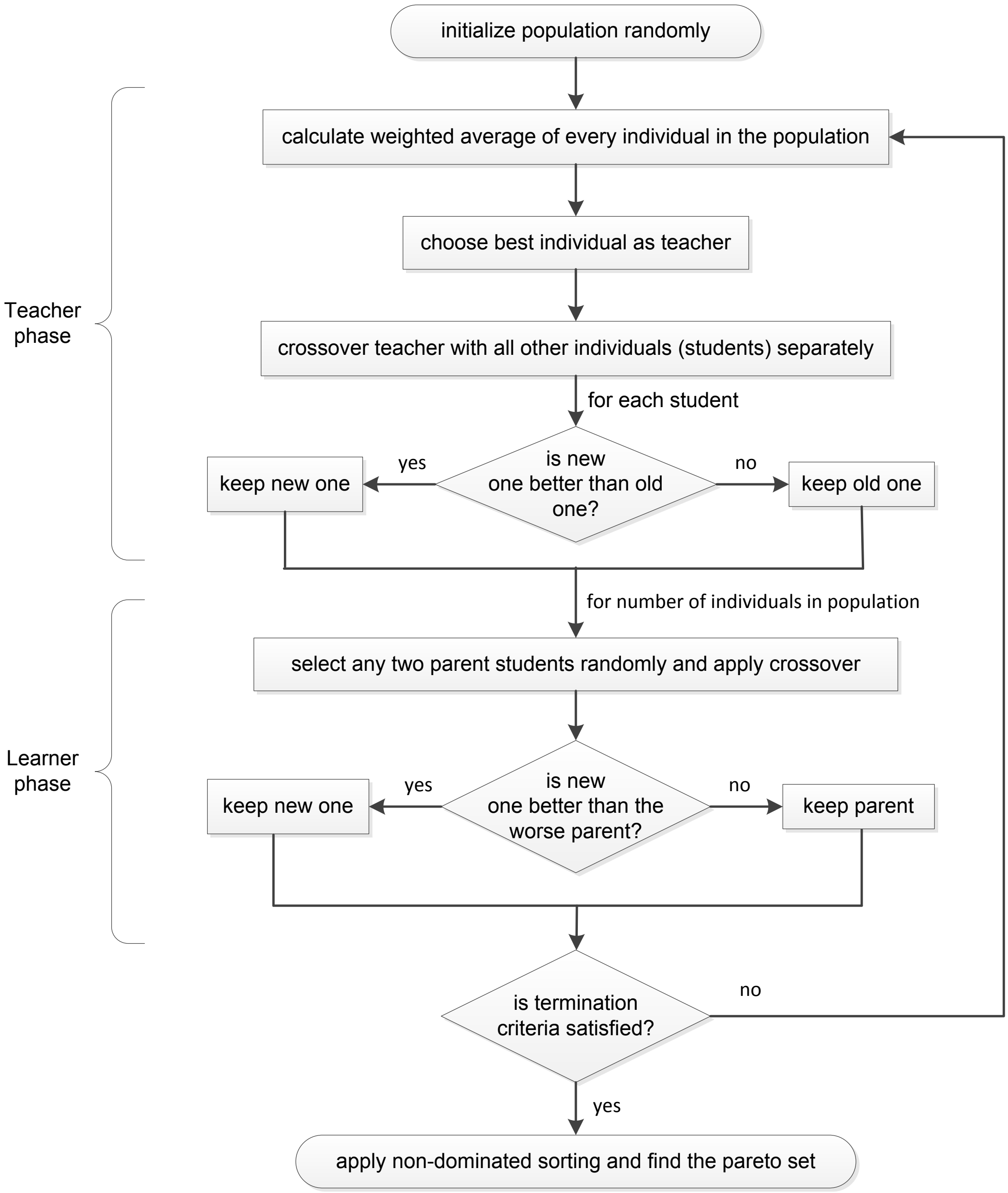


Figure 5: MTLBO with Scalar Transformation (MTLBO-ST).

245 ***Multiobjective TLBO with Non-Dominated Selection (MTLBO-NS)***

246 We use non-dominated sorting and selection in our second algorithm (see Figure 6). Thus, this algorithm is named as Multiobjective TLBO with Non-Dominated Selection (MTLBO-NS). In this approach, an individual is said to dominate another one if and only if at least one of its objectives is better than the other one's while keeping all other objectives same. If an individual is not dominated by any other individual, then it is said to be non-dominated.

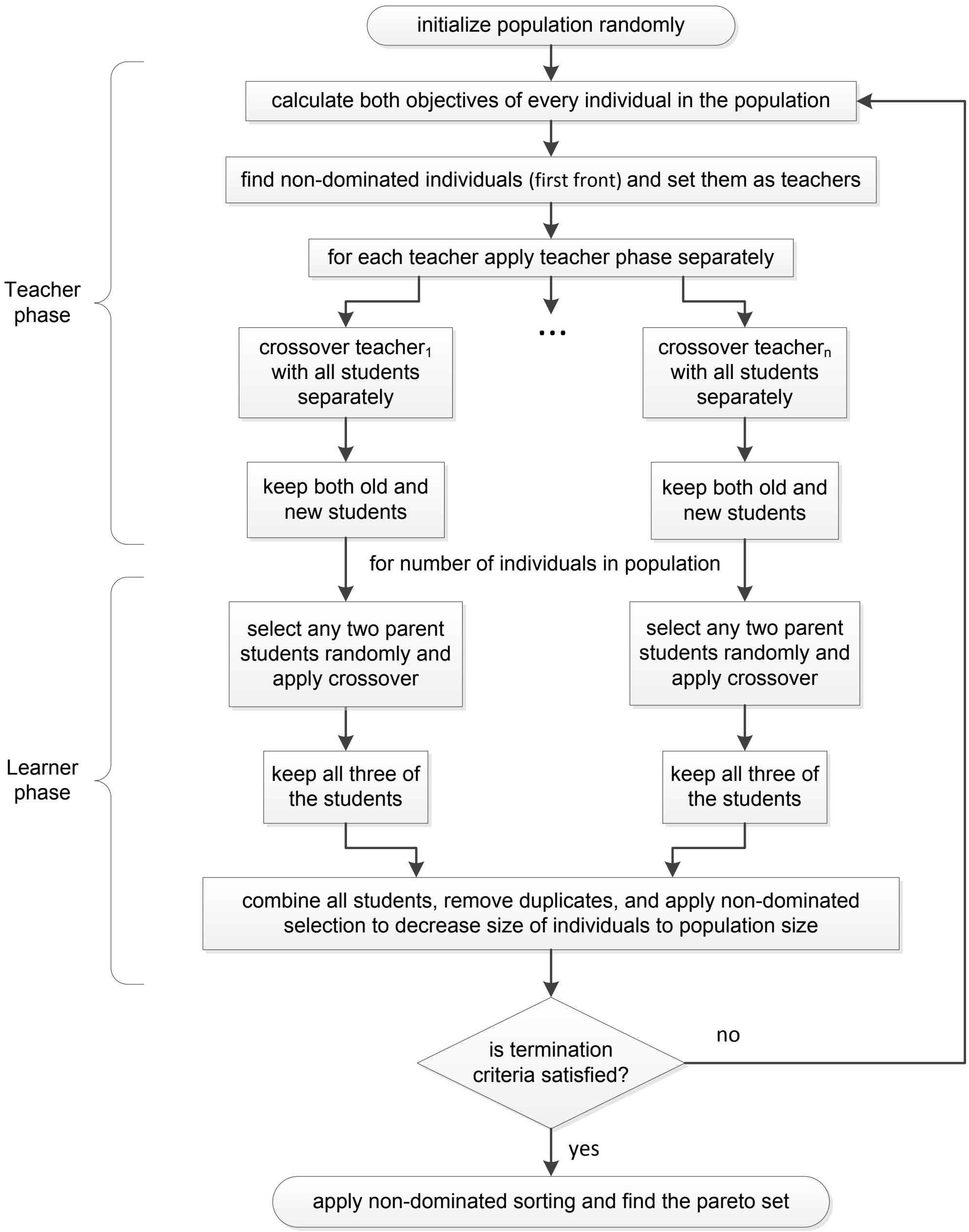


Figure 6: MTLBO with Non-Dominated Selection (MTLBO-NS).

All non-dominated individuals constitute the first front of the solution set. Individuals in the first front are selected as teachers. At the teacher and learner phases, all teachers teach all students discretely. In other terms, every teacher trains every student, but students which are taught by different teachers do not have the chance to interact with each other until the end of iteration. Distinct from regular TLBO, we do not compare students until the end of each iteration (before/after teaching/learning phases) and keep them in the possible population list. Finally, we combine all teachers and students into the same population, remove duplicates and use non-dominated selection algorithm to select the most promising chromosomes. For this purpose, we divide the possible population into fronts and starting from first front, select as many individuals as possible to fulfill the population size. Crowding distance value is used to select individuals in a front, if only a portion of the front is required in the new population.

***Multiobjective TLBO with Minimum Distance (MTLBO-MD)***

Our third approach, Multiobjective TLBO with Minimum Distance (MTLBO-MD), is a simplification of MTLBO-NS algorithm. In this approach, similar to MTLBO-NS, we find the chromosomes in the first front. However, we select the only one individual that is closest to the ideal point as teacher, rather than selecting all first front individuals. Thus, we expect a better performance when compared to MTLBO-NS in terms of computation time.

### *4.3. Applied Machine Learning Techniques*

Solutions obtained by TLBO are evaluated using three supervised machine learning techniques: Logistic Regression (LR), Support Vector Machines (SVM) and Extreme Learning Machine (ELM). LR is a well-known, easy and fast classifier. SVM is also popular as an effective classifier for binary classification. ELM, on the other hand, is a relatively new but promising classifier.

*Logistic Regression*: LR performs classification by estimating the occurrence probability of an event with respect to similarity of given data points. It uses Sigmoid Function (see Equation 3) in order to find probability of an event to occur. If event occurrence probability is greater than 0.5 then the event is predicted as 'occurred' otherwise it is predicted as 'not occurred'.

$$P(y = 1 \mid X, \theta) = \frac{1}{1 + e^{-\theta X}} \tag{3}$$

where $X$ is the given feature set, $\theta$ is the weights for all features, and $y$ is the probability result. Matlab function, *glmfit*, is used for LR classification in our experiments.

*Support Vector Machines*: SVM performs classification by constructing a separating line between given data points [36]. The closest data points to the separating line are called support vectors and the optimal separating line is constructed iteratively by maximizing the margin between the line and the support vectors of the classes. The idea comes from the intuition that the generalization error decreases as the margin increases. Matlab function, *fitcsvm*, is used for SVM classification in our experiments.

*Extreme Learning Machine*: ELM is a type of feedforward neural network with a single hidden layer. There are three layers in this model; input, hidden and output. Training data is given to the network by the input layer. Data is weighted and transferred by a function and passed to the hidden layer. Same transformation is done between the hidden layer and the output layer. Feedforward neural networks need iterative parameter tuning, whereas ELM does not require tuning. Therefore, learning time of ELM is much less when compared to the traditional feedforward neural networks since parameter tuning increases the learning time considerably. ELM library, developed by Huang et al. [37], is used for ELM classification in our experiments.

## 5. Experimental Setup and Results

In this section, experimental environment and problem instances are introduced and results of experiments are reported. Experiments are carried out on 13 datasets. 12 of them are obtained from a well-known machine learning data repository, University of California UCI Machine Learning Repository. Remaining dataset, Financial, is obtained from a study by Pacheco et al. [20]. All datasets are chosen or reduced to have two classes since the study is on binary classification. Reduction is applied by selecting the most occurred two classes in the dataset. Number of features in the datasets varies between 8 and 1558 and number of instances varies between 351 and $581,012$. Table 1 introduces these datasets. Experiments are carried on a computer with the following specifications: an Intel Core i7-6700 processor with a CPU clock rate of 3.40 GHz and 16 GB main memory. Java is utilized to implement FSS part of the algorithms. Matlab 2015a is utilized for the classification part of the algorithms.

In this study, a specialized random selection method is applied to generate training and test sets. For this purpose, 10 different training sets, and 10 test sets for each training set (100 test sets in total) are generated. First, proportions of each classes in the original dataset are calculated. Then, with regard to these proportions, training and test instances were randomly selected to meet the sizes given in Table 1. If an instance is in the training set, it is not included in any test set of that training set.

Population size and number of generations are two important parameters that must be decided before running TLBO. Higher values provide higher accuracy results but also they cause excessive computation time. Investigation of a new individual requires massive amount

Table 1: Specification of the datasets used in the experiments.

| **Dataset** | **Problem ID** | **Number of features** | **Actual number of classes** | **Number of instances** | **Size of each training set** | **Size of each test set** |
|---|---|---|---|---|---|---|
| Covertype | CT | 54 | 7 | $581,012$ | 600 | 200 |
| Mushrooms | MR | 22 | 2 | 8124 | 1300 | 200 |
| Spambase | SB | 57 | 2 | 4601 | 600 | 200 |
| Nursery | NU | 8 | 5 | $12,960$ | 400 | 200 |
| Connect-4 Opening | C4 | 42 | 3 | $67,557$ | 1200 | 200 |
| Waveform | WF | 40 | 3 | 5000 | 400 | 200 |
| Financial | FI | 93 | 2 | $17,108$ | 1000 | 200 |
| Pima Indian Diabetes | PM | 8 | 2 | 768 | 268 | 200 |
| Breast Cancer | BC | 9 | 2 | 699 | 199 | 100 |
| Ionosphere | IO | 34 | 2 | 351 | 101 | 50 |
| Wisconsin Breast Cancer | WBC | 30 | 2 | 569 | 169 | 80 |
| Musk | MU | 168 | 2 | 6598 | 400 | 200 |
| Internet Advertisements | NA | 1558 | 2 | 3279 | 400 | 200 |

of time. In order to improve the overall performance, we keep the objective values of investigated individuals in a hash map and do not reevaluate the same individual. Summing it up, it is important to decide the most promising values for these parameters. In our previous study [38], we ran extensive tests interchanging population size and number of generations between 10 and 100. The study shows that, increase in population size affects computation time worse than increase in number of generations; because as population size gets larger, number of diverse individuals in the population and hence number of evaluations increase. The ratio of number of evaluations decreases in each generation, since the probability of generating same individuals gets higher after each generation. As a result, we decide to choose population size as 40 and number of generations as 60, as similar to that study.

In order to see the effect of TLBO algorithm, initial, final and non-dominated solutions are presented in Figures 7, 8 and 9. Three datasets are selected to represent small, medium and large datasets according to their number of features (BC, MR and SB, respectively). In all these figures, initial population is randomly distributed, but the final population fits onto a pareto-like curve. Moreover, since we want to maximize accuracy and minimize the number of features, our ideal point can be represented as the point (1,1) and it can be seen from the results that, pareto-like curve converges to the ideal point. This is a process that individuals in the classroom improve through generations.

Accuracy results obtained for every dataset using each of the proposed algorithms and machine learning techniques are given in Table 2 in a multiobjective manner. Only non-dominated solutions in the final iteration are given in this table. Moreover, execution times of the algorithms and the number of unique evaluations are also presented at the bottom of each table.

Obtained results show that, MTLBO-ST tends to achieve single results like in a single objective optimization process, whereas non-dominated solutions of MTLBO-NS and MTLBO-MD fit to a pareto curve. On accuracy comparisons, MTLBO-NS could achieve higher values for the same number of features. On the other hand, MTLBO-ST dominates other two algorithms with its faster execution time. MTLBO-MD resembles MTLBO-NS in means of quality of solution set, and MTLBO-ST in means of execution time. As compared to MTLBO-NS, MTLBO-MD generates a similar solution set while keeping execution time considerably smaller for medium to large datasets. On the other hand, it requires longer execution time when compared to MTLBO-ST, but provides better solution sets. As a result, we can conclude that MTLBO-ST is a fast algorithm that provides single results with lower accuracy values, MTLBO-NS provides multiobjective solutions with higher accuracy values spending more amount of time and MTLBO-MD is an efficient algorithm that combines the good properties of the other two.

With respect to the comparison of machine learning techniques used in this study, there is no strict winner. All techniques achieve similar accuracy values with small deviations. On execution time comparisons, however, LR requires less execution time and dominates the other two techniques. ELM and SVM cannot dominate each other in terms of execution time. SVM executes faster in small datasets, but its time requirement massively increases as datasets get larger.

Table 3 presents classification results before and after FSS process is applied. For all

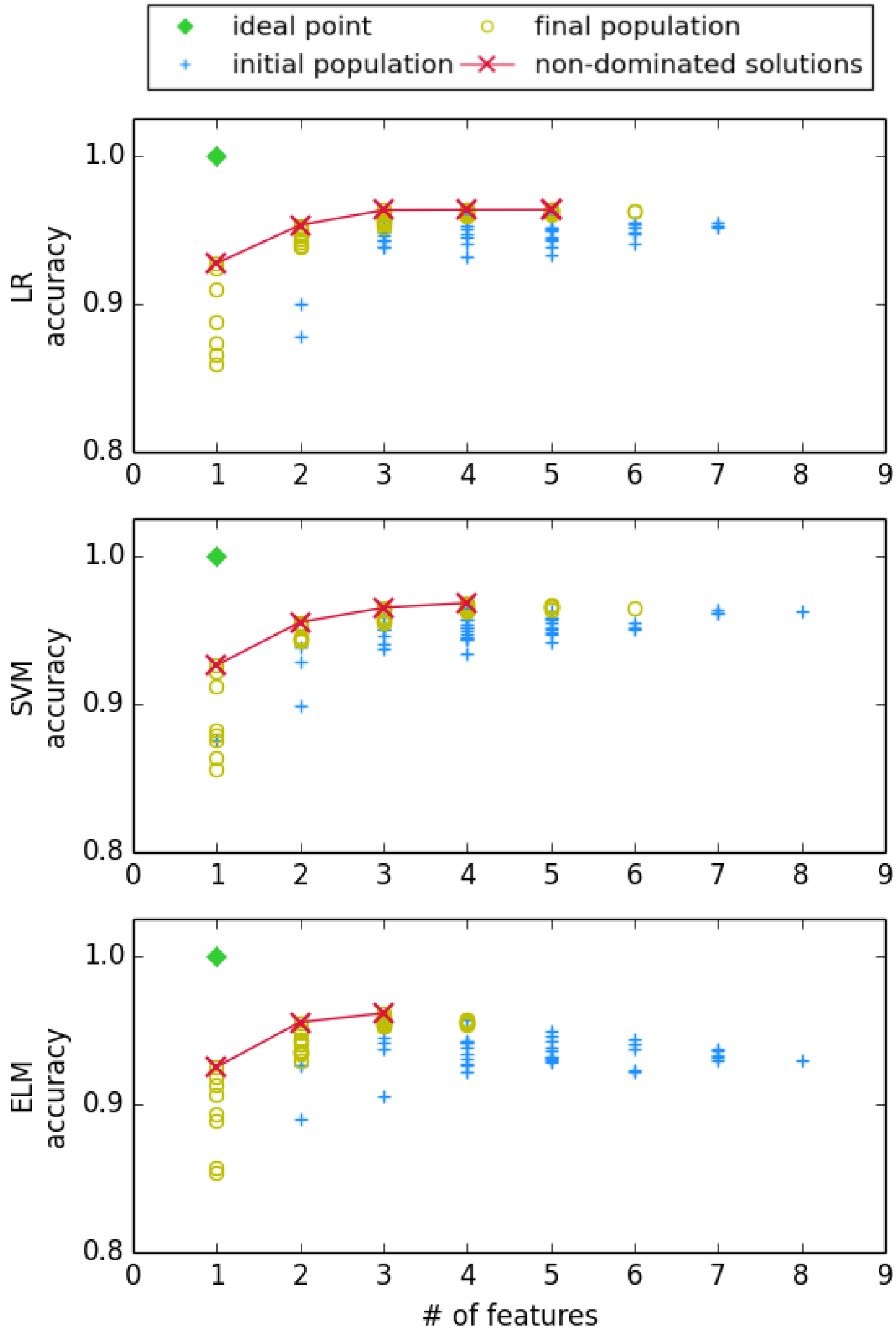


Figure 7: Distribution of TLBO-MD solutions on the BC dataset evaluated by LR, SVM, and ELM.

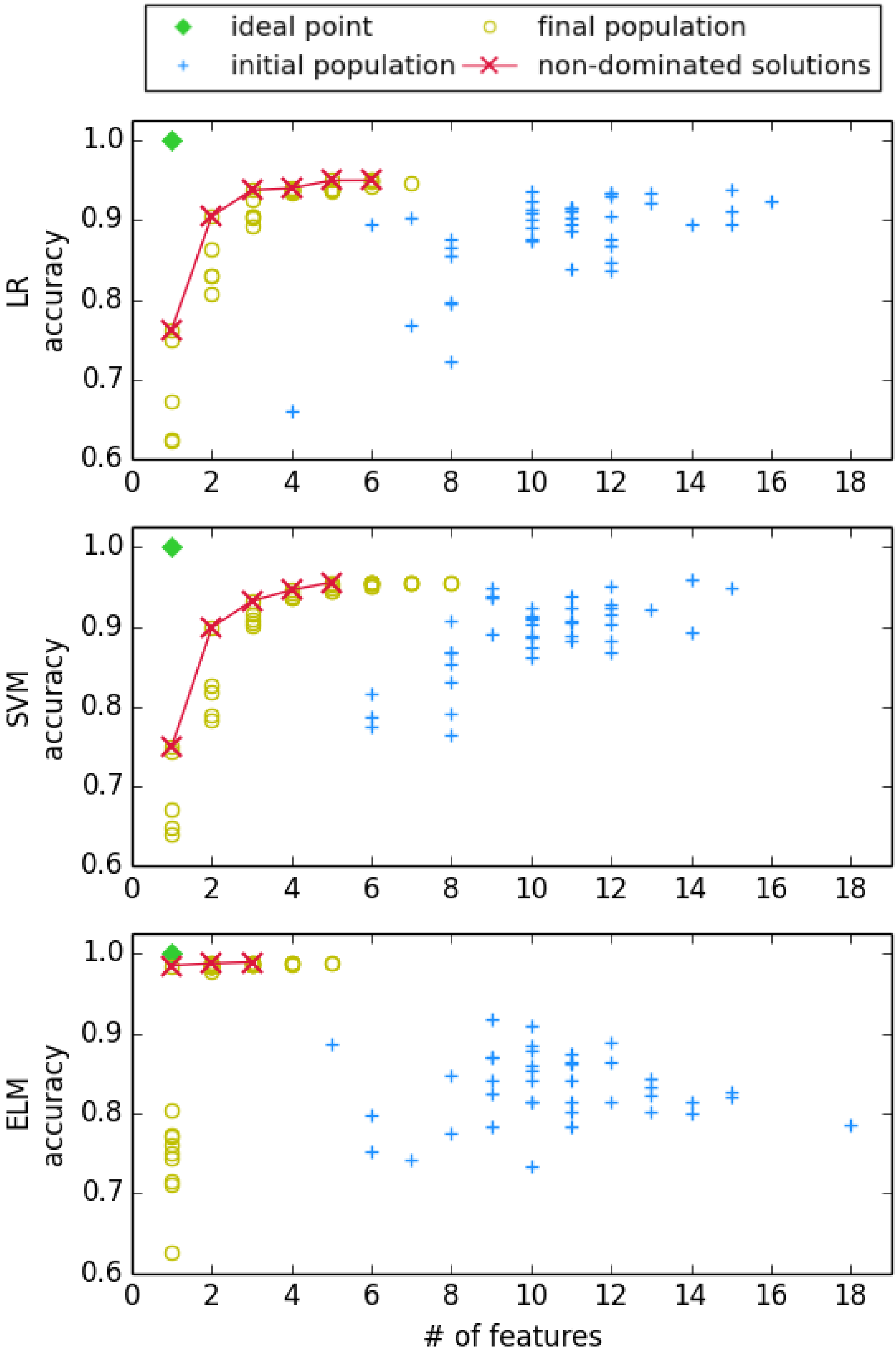


Figure 8: Distribution of TLBO-MD solutions on the MR dataset evaluated by LR, SVM, and ELM.

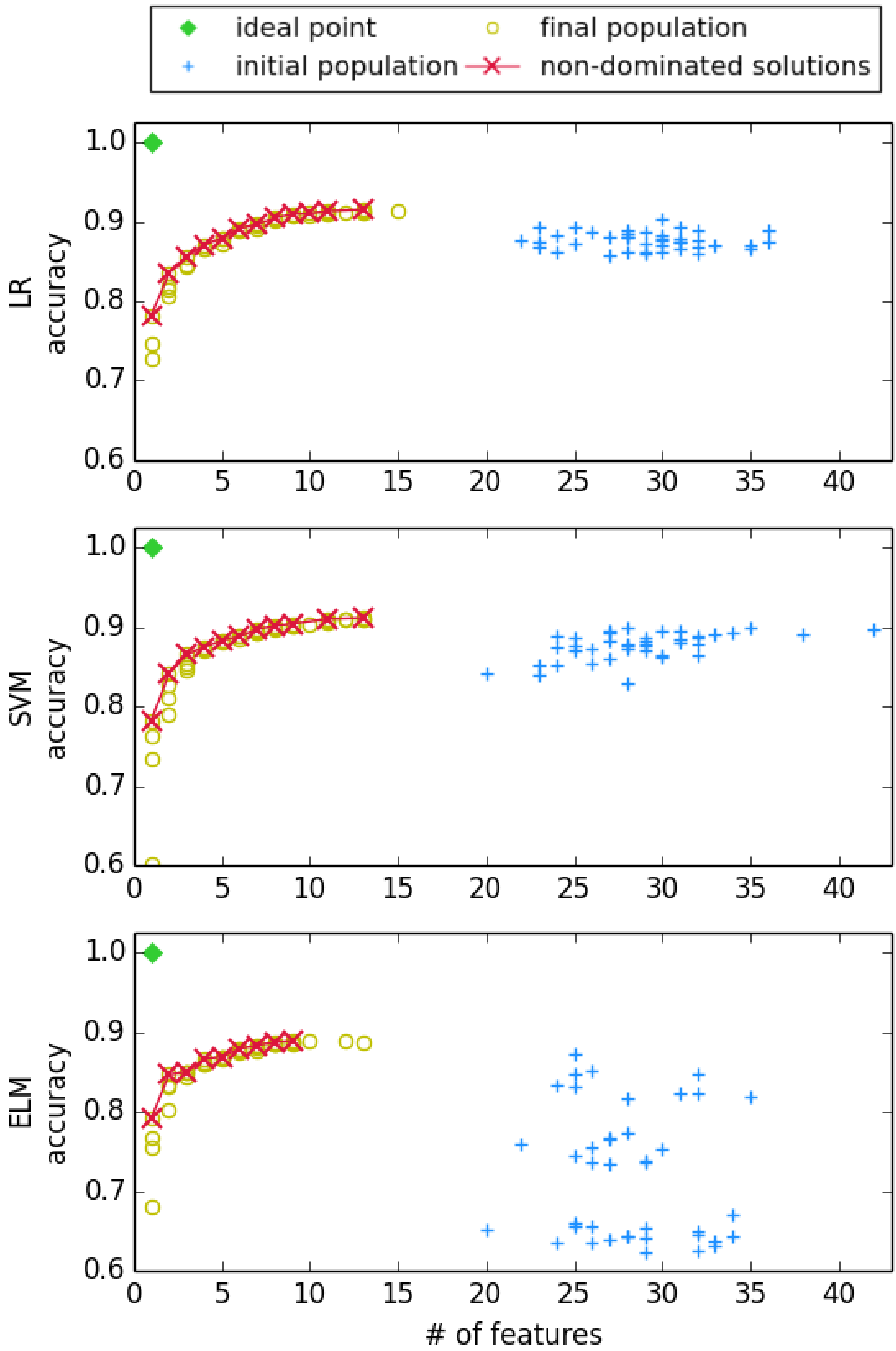


Figure 9: Distribution of TLBO-MD solutions on the SB dataset evaluated by LR, SVM, and ELM.

Table 2: Solution sets of all FSS algorithms evaluated by all machine learning techniques for all datasets.

(**bold values**: dominant solution, Time: in seconds, Eval: # of unique evaluations.)

(a) Solution sets of the CT dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | ST | NS | MD | ST | NS | MD | ST | NS | MD |
| 1 | 0.743 | - | 0.743 | 0.743 | 0.609 | 0.743 | **-** | 0.609 | 0.609 |
| 2 | - | 0.752 | **0.753** | 0.752 | 0.754 | 0.754 | 0.640 | 0.640 | 0.640 |
| 3 | - | 0.764 | 0.764 | - | **0.763** | 0.760 | - | **0.655** | 0.654 |
| 4 | - | 0.767 | 0.767 | - | 0.767 | 0.767 | - | **0.669** | 0.663 |
| 5 | - | 0.770 | 0.770 | - | 0.771 | 0.771 | - | 0.677 | 0.677 |
| 6 | - | 0.772 | 0.772 | - | **0.772** | 0.771 | - | 0.680 | **0.681** |
| 7 | - | 0.773 | 0.773 | - | 0.773 | 0.773 | - | **0.683** | 0.681 |
| 8 | - | **0.774** | 0.773 | - | **0.775** | 0.774 | - | **0.684** | 0.682 |
| 9 | - | 0.774 | 0.774 | - | **0.775** | 0.774 | - | **0.686** | 0.683 |
| 10 | - | 0.775 | - | - | **0.775** | 0.774 | - | - | - |
| 11 | - | 0.775 | - | - | 0.775 | - | - | 0.686 | - |
| 12 | - | 0.776 | - | - | 0.776 | - | - | - | - |
| 13 | - | 0.776 | - | - | - | - | - | - | - |
| 14 | - | 0.776 | - | - | - | - | - | - | - |
| 15 | - | 0.776 | - | - | - | - | - | - | - |
| 16 | - | 0.776 | - | - | - | - | - | - | - |
| 17 | - | 0.776 | - | - | - | - | - | - | - |
| Time | 192.2 | 6067.9 | 548.7 | 293.9 | 10943.4 | 1201.1 | 254.6 | 5556.6 | 983.4 |
| Eval | 1272 | 39192 | 4694 | 1253 | 33024 | 4756 | 1240 | 25103 | 4476 |

(b) Solution sets of the MR dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | ST | NS | MD | ST | NS | MD | ST | NS | MD |
| 1 | - | 0.763 | 0.763 | - | 0.750 | 0.750 | 0.985 | 0.985 | 0.985 |
| 2 | 0.897 | 0.905 | 0.905 | 0.867 | 0.899 | 0.899 | **-** | **0.989** | 0.988 |
| 3 | - | 0.937 | 0.937 | - | 0.932 | 0.932 | - | 0.990 | 0.990 |
| 4 | - | 0.940 | 0.940 | - | 0.946 | 0.946 | - | 0.992 | - |
| 5 | - | 0.949 | 0.949 | - | 0.956 | 0.956 | - | 0.992 | - |
| 6 | - | **0.952** | 0.950 | - | **-** | - | - | - | - |
| 7 | - | 0.953 | - | - | 0.956 | - | - | - | - |
| 8 | - | 0.954 | - | - | 0.958 | - | - | - | - |
| 9 | - | **-** | - | - | 0.960 | - | - | - | - |
| 11 | - | **-** | - | - | 0.960 | - | - | - | - |
| Time | 27.9 | 1114.1 | 158.8 | 237.7 | 5262.3 | 985.5 | 57.3 | 1393.6 | 302.7 |
| Eval | 501 | 6440 | 2416 | 495 | 13654 | 2584 | 278 | 4158 | 1352 |

(c) Solution sets of the SB dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | ST | NS | MD | ST | NS | MD | ST | NS | MD |
| 1 | - | - | 0.782 | **-** | - | 0.782 | **-** | 0.792 | 0.792 |
| 2 | - | - | 0.835 | - | - | 0.842 | **-** | 0.846 | **0.847** |
| 3 | - | 0.854 | **0.857** | 0.865 | - | 0.865 | 0.837 | **0.867** | 0.851 |
| 4 | 0.856 | 0.867 | **0.871** | - | 0.870 | **0.875** | 0.855 | **-** | **0.866** |
| 5 | - | **0.883** | 0.879 | - | 0.883 | 0.883 | - | **0.872** | 0.869 |
| 6 | - | 0.890 | **0.891** | - | **0.890** | 0.889 | - | 0.878 | **0.879** |
| 7 | - | **0.902** | 0.896 | - | 0.897 | 0.897 | - | 0.883 | **0.884** |
| 8 | - | **0.906** | 0.905 | - | 0.902 | 0.902 | - | **0.888** | 0.887 |
| 9 | - | 0.910 | 0.910 | - | **0.906** | 0.904 | - | **0.890** | 0.889 |
| 10 | - | **0.914** | 0.911 | - | 0.911 | - | - | 0.894 | - |
| 11 | - | **0.915** | 0.913 | - | **0.912** | 0.910 | - | 0.896 | - |
| 12 | - | 0.917 | - | - | 0.914 | - | - | 0.899 | - |
| 13 | - | **0.918** | 0.915 | - | **0.915** | 0.911 | - | 0.901 | - |
| 14 | - | 0.919 | - | - | 0.917 | - | - | 0.903 | - |
| 15 | - | 0.920 | - | - | 0.918 | - | - | - | - |
| 16 | - | 0.920 | - | - | 0.919 | - | - | - | - |
| 17 | - | 0.921 | - | - | 0.919 | - | - | - | - |
| 18 | - | 0.921 | - | - | 0.921 | - | - | - | - |
| 19 | - | 0.922 | - | - | 0.921 | - | - | - | - |
| 20 | - | 0.922 | - | - | 0.922 | - | - | - | - |
| 21 | - | - | - | - | 0.922 | - | - | - | - |
| Time | 164.2 | 7543.7 | 426.1 | 420.0 | 12161.1 | 1268.2 | 381.9 | 12331.9 | 992.0 |
| Eval | 1116 | 43918 | 5411 | 1551 | 47411 | 5447 | 1895 | 39155 | 5083 |

(d) Solution sets of the NU dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | ST | NS | MD | ST | NS | MD | ST | NS | MD |
| 1 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| Time | 6.7 | 13.7 | 12.5 | 10.5 | 27.5 | 24.2 | 15.5 | 75.9 | 40.7 |
| Eval | 98 | 195 | 196 | 82 | 207 | 186 | 79 | 232 | 192 |

(e) Solution sets of the C4 dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** |
| 1 | 0.729 | 0.729 | 0.729 | **0.730** | - | 0.729 | **0.731** | 0.730 | 0.730 |
| 2 | - | 0.746 | 0.746 | - | 0.737 | 0.737 | **-** | **0.746** | 0.744 |
| 3 | - | 0.755 | 0.755 | - | 0.746 | 0.746 | - | 0.753 | 0.753 |
| 4 | - | 0.764 | 0.764 | - | 0.757 | 0.757 | - | **0.763** | 0.762 |
| 5 | - | 0.772 | 0.772 | - | **0.764** | 0.758 | - | **0.768** | 0.765 |
| 6 | - | **0.778** | 0.777 | - | 0.772 | 0.772 | - | 0.776 | 0.776 |
| 7 | - | **0.785** | 0.784 | - | **0.781** | 0.780 | - | **0.781** | 0.778 |
| 8 | - | 0.791 | 0.791 | - | 0.787 | 0.787 | - | **0.787** | 0.783 |
| 9 | - | 0.796 | 0.796 | - | **0.795** | 0.793 | - | **0.792** | 0.789 |
| 10 | - | **0.802** | 0.797 | - | 0.800 | 0.800 | - | 0.797 | - |
| 11 | - | **0.806** | 0.799 | - | **0.805** | 0.802 | - | 0.798 | - |
| 12 | - | **0.811** | 0.799 | - | **0.811** | 0.802 | - | **0.801** | 0.792 |
| 13 | - | 0.815 | - | - | **0.814** | 0.804 | - | 0.804 | - |
| 14 | - | 0.818 | - | - | 0.818 | - | - | - | - |
| 15 | - | 0.821 | - | - | 0.821 | - | - | - | - |
| 16 | - | **0.824** | 0.805 | - | 0.824 | - | - | - | - |
| 17 | - | 0.827 | - | - | 0.827 | - | - | - | - |
| 18 | - | 0.828 | - | - | 0.830 | - | - | - | - |
| 19 | - | 0.829 | - | - | 0.831 | - | - | - | - |
| 20 | - | 0.830 | - | - | 0.832 | - | - | - | - |
| 21 | - | 0.830 | - | - | 0.834 | - | - | - | - |
| 22 | - | - | - | - | 0.834 | - | - | - | - |
| Time | 112.7 | 5638.5 | 431.6 | 427.9 | 52883.7 | 3454.8 | 178.8 | 9638.6 | 872.3 |
| Eval | 1315 | 39738 | 5218 | 972 | 41549 | 5043 | 862 | 28525 | 4322 |

(f) Solution sets of the WF dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** |
| 1 | - | **0.796** | 0.789 | - | 0.791 | **0.806** | - | 0.794 | **0.795** |
| 2 | 0.868 | 0.868 | 0.868 | 0.856 | 0.869 | 0.869 | **-** | **0.867** | 0.864 |
| 3 | - | 0.893 | 0.893 | 0.884 | 0.893 | 0.893 | 0.883 | 0.890 | **0.891** |
| 4 | - | 0.902 | 0.902 | - | 0.904 | 0.904 | - | **0.902** | 0.899 |
| 5 | - | 0.915 | 0.915 | - | 0.914 | 0.914 | - | **0.902** | 0.901 |
| 6 | - | 0.917 | 0.917 | - | 0.917 | 0.917 | - | 0.904 | **0.905** |
| 7 | - | 0.919 | 0.919 | - | 0.918 | 0.918 | - | 0.905 | - |
| 8 | - | 0.921 | 0.921 | - | 0.921 | 0.921 | - | 0.905 | - |
| 9 | - | 0.922 | 0.922 | - | **0.922** | 0.921 | - | - | - |
| 10 | - | 0.923 | 0.923 | - | 0.923 | - | - | - | - |
| 11 | - | 0.923 | - | - | 0.923 | - | - | - | - |
| 12 | - | 0.924 | - | - | - | - | - | - | - |
| 13 | - | - | - | - | 0.923 | - | - | - | - |
| 14 | - | - | - | - | 0.924 | - | - | - | - |
| Time | 21.5 | 751.8 | 88.8 | 278.7 | 3720.3 | 696.6 | 154.0 | 3820.2 | 582.7 |
| Eval | 896 | 22758 | 3817 | 1418 | 19679 | 3783 | 765 | 12495 | 2933 |

(g) Solution sets of the FI dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** |
| 1 | 0.966 | 0.966 | 0.966 | 0.966 | - | 0.966 | 0.966 | 0.966 | 0.966 |
| 2 | - | - | - | - | - | - | - | 0.966 | 0.966 |
| 3 | - | - | 0.967 | - | - | - | - | 0.966 | 0.966 |
| 4 | - | 0.967 | - | - | - | - | - | - | - |
| 5 | - | - | - | - | - | - | - | 0.967 | - |
| 8 | - | - | - | - | 0.966 | - | - | - | - |
| 9 | - | - | - | - | 0.966 | - | - | - | - |
| Time | 686.4 | 2172.2 | 702.8 | 3490.3 | 5382.9 | 4629.8 | 776.4 | 6904.6 | 1031.5 |
| Eval | 3339 | 11611 | 5332 | 2919 | 2014 | 5186 | 3650 | 36144 | 5334 |

(h) Solution sets of the PM dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** |
| 1 | 0.747 | 0.747 | 0.747 | 0.747 | 0.747 | 0.747 | **0.740** | 0.729 | 0.728 |
| 2 | - | 0.760 | 0.760 | - | 0.760 | 0.760 | - | 0.741 | - |
| 3 | - | 0.766 | 0.766 | - | 0.765 | 0.765 | - | - | - |
| 4 | - | 0.768 | 0.768 | - | 0.766 | 0.766 | - | - | - |
| 5 | - | 0.771 | 0.771 | - | 0.768 | 0.768 | - | - | - |
| 6 | - | - | - | - | 0.769 | 0.769 | - | - | - |
| 7 | - | 0.771 | - | - | - | - | - | - | - |
| Time | 2.9 | 5.3 | 4.9 | 15.9 | 41.4 | 38.3 | 19.9 | 40.5 | 41.7 |
| Eval | 123 | 249 | 223 | 96 | 251 | 231 | 102 | 219 | 209 |

(i) Solution sets of the BC dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** |
| 1 | 0.927 | 0.927 | 0.927 | 0.926 | 0.926 | 0.926 | 0.924 | 0.925 | 0.925 |
| 2 | - | 0.953 | 0.953 | - | 0.955 | 0.955 | - | **0.956** | 0.955 |
| 3 | - | 0.963 | 0.963 | - | 0.965 | 0.965 | - | **0.962** | 0.961 |
| 4 | - | 0.963 | 0.963 | - | 0.968 | 0.968 | - | - | - |
| 5 | - | 0.963 | 0.963 | - | - | - | - | - | - |
| Time | 2.9 | 8.8 | 7.2 | 13.2 | 52.2 | 45.5 | 21.0 | 61.8 | 47.4 |
| Eval | 148 | 456 | 352 | 119 | 464 | 389 | 134 | 387 | 301 |

(j) Solution sets of the IO dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | ST | NS | MD | ST | NS | MD | ST | NS | MD |
| 1 | - | 0.816 | 0.816 | - | 0.811 | 0.811 | - | **0.818** | 0.816 |
| 2 | - | 0.872 | 0.872 | 0.848 | 0.864 | 0.864 | **0.900** | 0.899 | 0.896 |
| 3 | 0.875 | 0.876 | 0.876 | - | 0.873 | 0.873 | - | - | - |
| 4 | - | **0.883** | 0.882 | - | 0.878 | 0.878 | - | - | - |
| 5 | - | **0.888** | 0.886 | - | **0.888** | 0.884 | - | - | - |
| 6 | - | **0.893** | 0.886 | - | **0.893** | 0.888 | - | - | - |
| 7 | - | **0.896** | 0.887 | - | 0.896 | - | - | - | - |
| 8 | - | **0.896** | 0.890 | - | - | - | - | - | - |
| 9 | - | **0.901** | - | - | 0.898 | - | - | - | - |
| 10 | - | **0.902** | - | - | 0.900 | - | - | - | - |
| 11 | - | **0.906** | - | - | 0.901 | - | - | - | - |
| Time | 25.4 | 1195.8 | 131.6 | 70.4 | 2413.1 | 326.1 | 125.4 | 731.4 | 322.6 |
| Eval | 645 | 20632 | 2988 | 595 | 20889 | 2813 | 908 | 5225 | 2314 |

(k) Solution sets of the WBC dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | ST | NS | MD | ST | NS | MD | ST | NS | MD |
| 1 | - | 0.920 | 0.920 | 0.921 | 0.919 | 0.921 | 0.906 | **0.917** | 0.915 |
| 2 | 0.958 | 0.961 | 0.961 | - | 0.960 | 0.960 | - | 0.947 | 0.947 |
| 3 | - | 0.971 | 0.971 | - | 0.970 | 0.970 | - | 0.954 | **0.955** |
| 4 | - | 0.975 | 0.975 | - | **0.975** | 0.974 | - | - | - |
| 5 | - | 0.975 | - | - | 0.976 | 0.976 | - | - | - |
| 6 | - | - | - | - | 0.978 | 0.978 | - | - | - |
| 7 | - | - | - | - | 0.978 | - | - | - | - |
| 8 | - | - | - | - | 0.979 | - | - | - | - |
| 10 | - | - | - | - | 0.979 | - | - | - | - |
| Time | 23.9 | 157.4 | 54.1 | 67.2 | 1413.4 | 335.3 | 97.4 | 1064.8 | 419.1 |
| Eval | 760 | 6587 | 2307 | 576 | 12204 | 2763 | 639 | 6997 | 2685 |

(l) Solution sets of the MU dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** |
| 3 | - | - | - | - | - | - | - | 0.858 | - |
| 4 | - | - | - | - | - | - | - | 0.869 | - |
| 5 | - | - | - | - | 0.844 | - | - | 0.889 | - |
| 6 | - | - | - | - | 0.881 | - | - | 0.892 | - |
| 7 | - | - | - | - | 0.901 | - | - | 0.894 | - |
| 8 | - | - | - | - | 0.906 | - | - | - | - |
| 9 | - | - | - | - | 0.913 | - | - | - | - |
| 10 | - | - | - | - | 0.919 | - | - | - | - |
| 11 | - | - | - | - | 0.923 | - | - | - | - |
| 12 | - | 0.891 | - | - | 0.925 | - | - | - | - |
| 13 | - | 0.901 | - | - | 0.927 | - | - | - | - |
| 14 | - | 0.906 | - | - | 0.929 | - | - | - | - |
| 15 | - | 0.910 | - | - | 0.929 | - | - | - | - |
| 16 | - | 0.913 | - | - | 0.930 | - | - | - | - |
| 17 | - | 0.916 | - | - | 0.930 | - | - | - | - |
| 18 | - | 0.918 | - | - | 0.932 | - | - | - | - |
| 19 | - | 0.919 | - | - | 0.932 | - | - | - | - |
| 20 | - | 0.920 | - | - | 0.933 | - | - | - | - |
| 21 | - | **0.921** | 0.907 | - | 0.934 | - | - | - | - |
| 22 | - | **0.921** | 0.910 | - | - | - | - | - | - |
| 23 | - | **0.922** | 0.912 | - | - | - | - | - | - |
| 24 | - | **0.922** | 0.914 | - | - | - | - | - | 0.849 |
| 25 | - | **0.922** | 0.916 | - | - | 0.897 | - | - | 0.860 |
| 26 | - | **0.923** | 0.917 | - | - | 0.904 | - | - | 0.864 |
| 27 | - | - | 0.918 | - | - | 0.908 | - | - | 0.866 |
| 28 | - | - | 0.919 | - | - | 0.911 | - | - | - |
| 29 | - | - | - | - | - | 0.915 | - | - | - |
| 30 | - | - | - | 0.907 | - | **0.917** | - | - | - |
| 31 | - | - | - | - | - | 0.918 | - | - | - |
| 32 | - | - | - | - | - | 0.920 | 0.843 | - | - |
| 33 | - | - | - | - | - | 0.921 | - | - | - |
| 34 | - | - | - | - | - | 0.921 | - | - | - |
| 35 | - | - | - | - | - | 0.922 | - | - | - |
| 36 | - | - | - | - | - | 0.922 | - | - | - |
| 43 | 0.883 | - | - | - | - | - | - | - | - |
| Time | 585.6 | 2410.8 | 931.2 | 715.6 | 16492.4 | 2494 | 687.8 | 7667.9 | 1725.8 |
| Eval | 828 | 16161 | 2399 | 1052 | 34855 | 4099 | 948 | 10828 | 2419 |

(m) Solution sets of the NA dataset.

| # of features | LR | | | SVM | | | ELM | | |
|---|---|---|---|---|---|---|---|---|---|
| | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** | **ST** | **NS** | **MD** |
| 246 | - | 0.998 | - | - | - | - | - | - | - |
| 247 | - | 0.998 | - | - | - | - | - | - | - |
| 391 | - | - | - | - | 0.999 | - | - | - | - |
| 479 | - | - | - | - | - | - | - | 0.999 | - |
| 515 | - | - | 0.997 | - | - | - | - | - | - |
| 516 | - | - | 0.997 | - | - | - | - | - | - |
| 517 | - | - | 0.998 | - | - | - | - | - | - |
| 520 | - | - | 0.998 | - | - | - | - | - | - |
| 521 | - | - | - | - | - | 0.999 | - | - | - |
| 522 | - | - | - | - | - | 0.999 | - | - | - |
| 532 | - | - | - | - | - | - | - | - | 0.999 |
| 573 | - | - | - | 0.998 | - | - | - | - | - |
| 593 | 0.997 | - | - | - | - | - | - | - | - |
| 619 | - | - | - | - | - | - | 0.998 | - | - |
| Time | 9920.8 | 62066 | 24001.3 | 3230.8 | 12778.4 | 6551.8 | 1733.5 | 4783.7 | 3276.2 |
| Eval | 1847 | 13790 | 4873 | 1693 | 7648 | 3570 | 1673 | 4568 | 3065 |

datasets, classification accuracy increases considerably and the number of features reduces after selecting the most valuable subset of features. Specifically, WBC dataset has a classification accuracy of 0.924 when all 30 features are included in classification process. After finding the most valuable subset of features by applying TLBO algorithm, new instances can be classified with an accuracy value of 0.975 by using only 4 features of the dataset. The results of the experiments show that applying multiobjective TLBO algorithm improves classification performance in terms of both objectives, accuracy and minimum number of features.

In order to verify the efficiency of the multiobjective TLBO algorithms, their results are compared with state-of-the-art NSGA-II, PSO, TS, GS, and SS based algorithms in Table 4.

Table 3: The effect of feature subset selection on classification performance.

| Dataset ID | Before FSS | | After FSS | |
|---|---|---|---|---|
| | **accuracy** | **# of features** | **accuracy** | **# of features** |
| CT | 0.761 | 54 | 0.774 | 9 |
| MR | 0.937 | 22 | 0.950 | 6 |
| SB | 0.893 | 57 | 0.915 | 13 |
| NU | 1.000 | 8 | 1.000 | 1 |
| C4 | 0.820 | 42 | 0.805 | 16 |
| WF | 0.893 | 40 | 0.923 | 10 |
| FI | 0.909 | 93 | 0.967 | 3 |
| PM | 0.762 | 8 | 0.771 | 5 |
| BC | 0.954 | 9 | 0.963 | 3 |
| IO | 0.812 | 34 | 0.890 | 8 |
| WBC | 0.924 | 30 | 0.975 | 4 |
| MU | 0.877 | 168 | 0.926 | 26 |
| NA | 0.993 | 1558 | 0.998 | 520 |

Table 4: Multiobjective comparison of the proposed algorithm with state-of-the-art algorithms.

| **Dataset ID** | **Proposed Alg.** | | **Deniz et al. [38]** | | **Unler et al. [18]** | | **Pacheco et al. [20]** | | | | | | **Lopez et al. [19]** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **MTLBO-MD** | | **NSGA-II** | | **PSO** | | **TS** | | **SFS** | | **SBS** | | **SSS-GC** | | **SSS-RGC** | | **PSS** | |
| | **Acc.** | **F. size** | **Acc.** | **F. size** | **Acc.** | **F. size** | **Acc.** | **F. size** | **Acc.** | **F. size** | **Acc.** | **F. size** | **Acc.** | **F. size** | **Acc.** | **F. size** | **Acc.** | **F. size** |
| CT | **0.770** | **5** | **0.770** | **5** | 0.770 | 7 | 0.755 | 7 | 0.764 | 5 | 0.761 | 7 | - | - | - | - | - | - |
| MR | 0.905 | 2 | 0.867 | 2 | 1.000 | 3 | 1.000 | 5 | 0.860 | 3 | 0.869 | 3 | - | - | - | - | - | - |
| SB | 0.905 | 8 | **0.906** | **8** | 0.902 | 8 | 0.900 | 8 | 0.879 | 8 | 0.876 | 8 | - | - | - | - | - | - |
| NU | **1.000** | **1** | **1.000** | **1** | 1.000 | 3 | 1.000 | 3 | 1.000 | 3 | 1.000 | 3 | - | - | - | - | - | - |
| C4 | 0.799 | 11 | 0.802 | 11 | 0.813 | 12 | 0.791 | 12 | 0.782 | 11 | 0.749 | 7 | - | - | - | - | - | - |
| WF | **0.915** | **5** | **0.915** | **5** | 0.906 | 7 | 0.903 | 7 | 0.899 | 5 | 0.899 | 5 | - | - | - | - | - | - |
| FI | **0.966** | **1** | **0.966** | **1** | 0.882 | 8 | 0.879 | 3 | 0.873 | 3 | 0.873 | 5 | - | - | - | - | - | - |
| PM | 0.768 | 4 | 0.768 | 4 | 0.774 | 6 | - | - | - | - | - | - | 0.679 | 4.1 | 0.677 | 4.0 | 0.681 | 4.2 |
| BC | **0.963** | **3** | **0.963** | **3** | 0.962 | 4 | - | - | - | - | - | - | 0.952 | 5.2 | 0.949 | 4.8 | 0.951 | 5.4 |
| IO | **0.882** | **4** | 0.878 | 4 | 0.862 | 4 | - | - | - | - | - | - | 0.878 | 6.1 | 0.871 | 5.7 | 0.874 | 3.9 |
| WBC | **0.975** | **4** | **0.975** | **4** | 0.963 | 7 | - | - | - | - | - | - | 0.947 | 6.8 | 0.936 | 5.5 | 0.937 | 6.0 |

In this table, bold results represent domination and underlined texts indicate non-dominated results. If two datasets find exact same solutions, both are marked equally. The results show that TLBO finds equivalent solutions with NSGA-II. They find the same exact solutions in 7 datasets, TLBO dominates in 2 datasets and is dominated in the remaining 2 datasets. TLBO, on the other hand, outperforms all other algorithms. TLBO dominates the PSO algorithm in 8 datasets, and generates solutions that are non-dominated for the remaining 3 datasets. We have the results of only 7 datasets when TS and GS based algorithms are used, and TLBO dominates in 6 of each and finds non-dominated solutions in only 1 of them. Similarly, only 4 of our datasets match with the datasets used in SS algorithms, and TLBO dominates in all of these datasets.

**Discussion**

Consequently, we can evaluate the proposed algorithms from different perspectives. These algorithms are robust because they provide stable and high quality accuracy results that do not change more than 1% at each run. These algorithms can be used for any classification problem in a multiobjective way. The multiobjective property is important because it makes these algorithms flexible. One of the objectives is to reduce the size of the problem by eliminating redundant and/or unrelated features which is very beneficial for big data applications. The proposed algorithms achieve high quality results with faster execution times. Crossover and mutation operators are carefully designed to generate diverse new candidate solutions and this is good for both the convergence speed and solution quality of the optimization process. In addition to having reasonable execution times, the algorithms are effective in producing good quality solutions. Crossovers and mutation operators always generate valid solutions. For the datasets that have more than 100 features the FSS problem becomes very hard, and it takes exponentially more time to analyze these datasets with too many features. The same problem is faces with each metaheuristics since the main purpose of the metaheuristic algorithms is dealing with exponentially increasing execution time problem for datasets with a large number of features. The proposed algorithms eliminate the parameter setting issues for the crossover and mutation operators, but the population size and the maximum number of generations parameters must still be carefully tuned for these algorithms. Increasing the number of generations may not always provide better results even though execution times will be increased significantly. As it is seen for the other population based algorithms such as PSO and genetic stagnation is always a critical problem that must be considered during optimization.

## 6. Conclusion

In this study, we propose three multiobjective TLBO algorithms (Multiobjective TLBO with Scalar Transformation (MTLBO-ST), Multiobjective TLBO with Non-dominated Selection (MTLBO-NS) and Multiobjective TLBO with Minimum Distance (MTLBO-MD)) for the FSS-BCP. MTLBO-ST is the fastest of these three algorithms, however, it provides small number of non-dominated solutions. MTLBO-NS examines an extensive search space and yields to a non-dominated solution set with more individuals and requires massive amount of time to execute. MTLBO-MD generates solution sets similar to MTLBO-NS in a considerably less amount of time, like MTLBO-ST. A more formal comparison of these proposed algorithms are given in Table 5. Three machine learning techniques, LR, SVM, and ELM, are used to evaluate the performance of the proposed multiobjective TLBO algorithms. Among these techniques, LR is more preferable due to its time efficiency, since all of them achieve similar accuracy results. Proposed best performing multiobjective algorithm, MTLBO-MD with LR, is compared with state-of-the-art algorithms, NSGA-II (genetic algorithm), Particle Swarm Optimization (PSO), Tabu Search (TS), Greedy Search (GS), and Scatter Search (SS). Results show that, our proposed algorithm achieves similar results with NSGA-II, while performing better than PSO, TS, GS, and SS algorithms.

A possible future work can be testing multiobjective TLBO algorithms on different datasets and comparing their results with some other state-of-the-art feature selection algorithms. Moreover, other machine learning techniques such as deep learning can be applied in classification phase of the algorithm. Finally, a more intelligent initial population method can be employed rather than randomization.

Table 5: Overall comparison of the proposed algorithms.

| | **MTLBO - ST** | **MTLBO - NS** | **MTLBO - MD** |
|---|---|---|---|
| **Teacher selection** | Teacher selection is handled by combining two fitness values into a scalar value and selecting the highest scalar value as teacher. | Every non-dominated individual is selected as teacher at each generation. All teachers teach their students separately, and eventually best students among all students are selected as the next generation. | Only the non-dominated solution that is closest to the ideal point (1,1) is selected as teacher. |
| **Execution time** | Executes fastest. | Executes slowest. | It has an average execution time, that is closer to MTLBO-ST than MTLBO-NS. |
| **Exploration** | Number of unique evaluations is small, and hence, its search space exploration is limited. | Number of unique evaluations is large, which means it explores the search space deepest. | Number of unique evaluations is medium. It explores the search space deeper than MTLBO-ST, but not as deep as MTLBO-NS. |
| **Feature selection performance** | It reduces number of selected features; however, it yields to a single solution and generally does not find a non-dominated solution set. | Reduces number of selected features while converging to a large non-dominated set. | Reduces number of selected features, and finds a medium sized non-dominated set. Its performance is better than MTLBO-ST, but not as good as MTLBO-NS. |
| **Accuracy performance** | Accuracy is lower than other two algorithms. | It generally finds same accuracy values with MTLBO-MD, but it finds better results on large datasets. | It finds same or close enough accuracy values with MTLBO-NS. |
| **Overall view** | MTLBO-ST provides single solution with a lower accuracy value, but in a small amount of time. It may be used when fast analysis is important. | MTLBO-NS provides a large non-dominated solution set with higher accuracy values; giving us a chance to choose optimal settings for a specific problem. On the other hand, its execution time is very high, especially for large datasets. | MTLBO-MD compromises both non-dominated set size and accuracy as compared to MTLBO-NS, but are both better than the MTLBO-ST algorithm. Its execution time is larger than MTLBO-ST, but smaller than MTLBO-NS. It may be the best option since it finds acceptable solutions in an acceptable amount of time. |

# References

[1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A. H. Byers, Big data: The next frontier for innovation, competition, and productivity .

[2] C. A. O'Reilly, Variations in decision makers' use of information sources: The impact of quality and accessibility of information, Academy of Management journal 25 (4) (1982) 756–771.

[3] E. Alpaydin, Introduction to Machine Learning, Adaptive Computation and Machine Learning, MIT Press, ISBN 9780262028189, URL `https://books.google.com.tr/books?id=NP5bBAAAQBAJ`, 2014.

[4] H. Liu, H. Motoda, Feature selection for knowledge discovery and data mining, vol. 454, Springer Science & Business Media, 2012.

[5] G. H. John, R. Kohavi, K. Pfleger, et al., Irrelevant features and the subset selection problem, in: Machine learning: proceedings of the eleventh international conference, 121–129, 1994.

[6] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, The Journal of Machine Learning Research 3 (2003) 1157–1182.

[7] Y. Yang, J. O. Pedersen, A comparative study on feature selection in text categorization, in: ICML, vol. 97, 412–420, 1997.

[8] I. Schwab, A. Kobsa, I. Koychev, Learning about users from observation, in: Adaptive user interfaces: Papers from the 2000 AAAI spring symposium, 102–106, 2000.

[9] B. Altay, T. Dokeroglu, A. Cosar, Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection, Soft Computing (2018) 1–15.

[10] R. V. Rao, V. J. Savsani, D. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Computer-Aided Design 43 (3) (2011) 303–315.

[11] R. V. Rao, V. Savsani, J. Balic, Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems, Engineering Optimization 44 (12) (2012) 1447–1462.

[12] M. Dash, H. Liu, Feature selection for classification, Intelligent data analysis 1 (3) (1997) 131–156.

[13] B. Xue, M. Zhang, W. Browne, X. Yao, A Survey on Evolutionary Computation Approaches to Feature Selection, IEEE Transactions on Evolutionary Computation 20 (4) (2016) 606–626, ISSN 1089-778X.

[14] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, in: Feature extraction, construction and selection, Springer, 117–136, 1998.

[15] I. Inza, P. Larrañaga, R. Etxeberria, B. Sierra, Feature subset selection by Bayesian network-based optimization, Artificial intelligence 123 (1) (2000) 157–184.

[16] C.-L. Huang, C.-J. Wang, A GA-based feature selection and parameters optimizationfor support vector machines, Expert Systems with applications 31 (2) (2006) 231–240.

[17] L. Cervante, B. Xue, M. Zhang, L. Shang, Binary particle swarm optimisation for feature selection: A filter based approach, in: Evolutionary Computation (CEC), 2012 IEEE Congress on, IEEE, 1–8, 2012.

[18] A. Unler, A. Murat, A discrete particle swarm optimization method for feature selection in binary classification problems, European Journal of Operational Research 206 (3) (2010) 528–539.

[19] F. G. López, M. G. Torres, B. M. Batista, J. A. M. Pérez, J. M. Moreno-Vega, Solving feature subset selection problem by a parallel scatter search, European Journal of Operational Research 169 (2) (2006) 477–489.

[20] J. Pacheco, S. Casado, L. Núñez, A variable selection method based on Tabu search for logistic regression models, European Journal of Operational Research 199 (2) (2009) 506–511.

[21] U. Mlakar, I. Fister, J. Brest, B. Potočnik, Multi-objective differential evolution for feature selection in facial expression recognition systems, Expert Systems with Applications 89 (2017) 129–137.

[22] Y. Zhang, D.-w. Gong, J. Cheng, Multi-objective particle swarm optimization approach for cost-based feature selection in classification, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 14 (1) (2017) 64–75.

[23] Z. Yong, G. Dun-wei, Z. Wan-qiu, Feature selection of unreliable data using an improved multi-objective PSO algorithm, Neurocomputing 171 (2016) 1281–1290.

[24] A. Khan, A. R. Baig, Multi-Objective Feature Subset Selection using Non-dominated Sorting Genetic Algorithm, Journal of applied research and technology 13 (1) (2015) 145–159.

[25] U. K. Sikdar, A. Ekbal, S. Saha, MODE: multiobjective differential evolution for feature selection and classifier ensemble, Soft Computing 19 (12) (2015) 3529–3549.
[26] B. Xue, M. Zhang, W. N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, Cybernetics, IEEE Transactions on 43 (6) (2013) 1656–1671.
[27] R. Rao, V. Patel, An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems, International Journal of Industrial Engineering Computations 3 (4) (2012) 535–560.
[28] M. Črepinšek, S.-H. Liu, L. Mernik, A note on teaching–learning-based optimization algorithm, Information Sciences 212 (2012) 79–93.
[29] M. Nayak, C. Nayak, P. Rout, Application of multi-objective teaching learning based optimization algorithm to optimal power flow problem, Procedia Technology 6 (2012) 255–264.
[30] Y. Xu, L. Wang, S.-y. Wang, M. Liu, An effective teaching–learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time, Neurocomputing 148 (2015) 260–268.
[31] T. Dokeroglu, Hybrid teaching–learning-based optimization algorithms for the Quadratic Assignment Problem, Computers & Industrial Engineering 85 (2015) 86–101.
[32] A. Baykasoğlu, A. Hamzadayi, S. Y. Köse, Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases, Information Sciences 276 (2014) 204–218.
[33] T. Niknam, R. Azizipanah-Abarghooee, M. R. Narimani, A new multi objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems, Engineering Applications of Artificial Intelligence 25 (8) (2012) 1577–1588.
[34] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, Knowledge and Data Engineering, IEEE Transactions on 17 (4) (2005) 491–502.
[35] R. V. Rao, G. Waghmare, A comparative study of a teaching–learning-based optimization algorithm on multi-objective unconstrained and constrained functions, Journal of King Saud University-Computer and Information Sciences 26 (3) (2014) 332–346.
[36] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273–297.
[37] G.-B. Huang, D. H. Wang, Y. Lan, Extreme learning machines: a survey, International Journal of Machine Learning and Cybernetics 2 (2) (2011) 107–122.
[38] A. Deniz, H. E. Kiziloz, T. Dokeroglu, A. Cosar, Robust multiobjective evolutionary feature subset selection algorithm for binary classification using machine learning techniques, Neurocomputing .