MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

APPLICATION OF METAHEURISTICS FOR THE FEATURE SELECTION PROBLEM

GÜLŞAH KALAYCI

MASTER THESIS

Department of Computer Engineering

Thesis Supervisor

Assoc. Prof. Ali Fuat ALKAYA

ISTANBUL, 2019

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

APPLICATION OF METAHEURISTICS FOR THE FEATURE SELECTION PROBLEM

GÜLŞAH KALAYCI

(524115022)

MASTER THESIS

Department of Computer Engineering

Thesis Supervisor

Doç. Dr. Ali Fuat ALKAYA

ISTANBUL, 2019

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN PURE AND APPLIED SCIENCES

Gülşah KALAYCI, a Master of Science student of Marmara University Institute for Graduate Studies in Pure and Applied Sciences, defended her thesis entitled "Application of Metaheuristics for the Feature Selection Problem", on June 26, 2019 and has been found to be satisfactory by the jury members.

Jury Members

Assoc. Prof Ali Fuat ALKAYA (Advisor)
Marmara University ....(SIGN) ....

Asst. Prof. Fatma CORUT ERGİN (Jury Member)
Marmara University..(SIGN) ....

Asst. Prof. Gonca GÜRSUN (Jury Member)
Özyeğin University..(SIGN) ....

APPROVAL

Marmara University Institute for Graduate Studies in Pure and Applied Sciences Executive Committee approves that Gülşah KALAYCI be granted the degree of Master of Science in Computer Engineering at Marmara University on June 26, 2019. (Resolution no: 2019/13-02

Director of the Institute
Prof. Dr. BÜLENT EKİCİ

TABLE OF CONTENTS

ABSTRACT

Application of Metaheuristics for the Feature Selection Problem

Feature selection aims to gain a minimal feature subset in a problem domain while conserving the accuracy of the original data. Feature selection is a process for making more efficient data analysis by selecting more relevant features for the related problem solving. Feature selection increases prediction probability of algorithms by decreasing the dimensionality, eliminating irrelevant features. In this study, four computational intelligence techniques are implemented and compared on the well-known data instances taken from University of California, Irvine (UCI) database. This is the first time that Migrating Birds Optimization (MBO) is used for the feature selection problem. Specifically, the exploited algorithms are (i) MBO, which is a recently proposed but successful technique, (ii) particle swarm optimization, which has originated from the simulation of behavior of biological organisms, (iii) simulated annealing, which is a well-known and frequently used as a benchmark algorithm and (iv) differential evolution. In our filter-based approach, we also implemented the inconsistency based subset evaluator to evaluate the performance of a given feature subset. Performance comparison is done with k-nearest neighbor, as the classifier where all features are used in the benchmark. Results show that the MBO algorithm presents the best performance in terms of number of winning cases.

ÖZET

Metasezgisellerin Özellik Seçimi Problemi İçin Uygulanması

Özellik seçimi, bir problemde veri doğruluğunu korurken, asgari özellik kümesini elde etmeyi amaçlar. Özellik seçimi, problem çözümüyle ilgili özellikleri seçerek daha verimli bir veri analizi yapabilmek için ön süreçtir. Özellik seçimi, alakasız özellikleri eleyip kolon sayısını azaltarak algoritmaların tahmin yeteneğini arttırır. Bu çalışmada, 4 algoritma gerçeklendi ve özellik seçimi literatürde sıklıkla kullanılan UCI veritabanındaki veriler üzerinde koşturularak karşılaştırıldı. Bu çalışma, göçmen kuşlar optimizasyon algoritmasının özellik seçimi problemine ilk uygulanışı olma niteliğini taşımaktadır. Kullanılan algoritmalar şu şekilde sıralanabilir; son zamanlarda gelişen ancak kuvvetli bir algoritma olan göçmen kuşlar optimizasyonu, biyolojik organizmaların davranışlarından esinlenen parçacık sürü algoritması, demire şekil verilmesinden esinlenerek gelişirilen benzetilmiş tavlama algoritması ve diferansiyel gelişim algoritması. Seçilen özellik kümesinin doğruluğunu ölçmek için tutarlılık tabanlı küme değerlendirici kullanıldı. Algoritmaların performansı en yakın k komşu algoritması kullanılarak karşılaştırıldı. Sonuçlar, üstün çıkma sayısı bazında göçmen kuşlar algoritmasının kazandığını gösterdi.

ABBREVIATIONS

10-FCV : 10-Fold Cross Validation
ABC : Artificial Bee Colony
CART : Classification and Regression Tree
CFS : Correlation-based FS
CSF : Cosine Similarity-based Filter Feature Selection Technique
CXP : Crossover Rate
DE : Differential Evolution
EC : Evolutionary Computation
ELM : Extreme Learning Machine
FRFS : Fuzzy-Rough FS
FS : Feature Selection
GA : Genetic Algorithm
GSO : Gram-Schmidt Orthogonalization
HC : Hill Climbing
k-NN : k Nearest Neighbour
MA : Memetic Algorithm
MBO : Migrating Birds Optimization
MIC : Maximal Information Coefficient
mp : mutation rate
NB : Naive Bayes
nob : number of birds
nof : number of flapping
nog : number of generations
non : number of neighbors
nos : number of solutions
olf : overlap factor
OMICFS : Orthogonal MIC Feature Selection
PCFS : Probabilistic Consistency-based FS
PSO : Particle Swarm Optimization
r : radius
SA : Simulated Annealing
SMO : Sequential Minimum Optimization

SRKNN **:** Sequential Random k Nearest Neighbor

SVM **:** Support Vector Machine

SWR **:** Stepwise Regression Algorithm

T **:** Initial Temperature

TS **:** Tabu Search

UCI **:** University of California, Irvine

WEKA **:** Waikato Environment for Knowledge Analysis

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In changing world, data has become very important. Several analysis is made on past (or instant) data to expect and shape the future. After human being understood the importance of data, every possibly important data from every device is collected. When analysing these data in real-world applications, it is not easy to decide which data is really important. Eliminating unnecessary features from a feature set, provides time consuming and ease of applying a mining model. Feature selection is a very important preprocessing step for some applications like data mining, machine learning and pattern recognition (Figure 1.1). It does not only decrease computational expense with reducing column number but also increases the accuracy of the learning model with eliminating redundant or irrelevant columns [1,2].

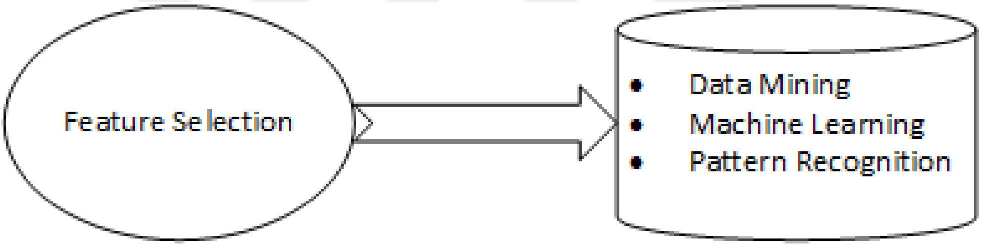

Figure 1.1 Feature Selection is a preprocessing method

There are some methods to evaluate candidate subsets and decide performance of the selected features. These methods are being categorized in three classes which are filter, wrapper and hybrid methods (Figure 1.2). Some filter methods rank the features based on a certain criteria like chi-square analysis [5], rough set and fuzzy-rough set-based dependency [6,7], information gain, and symmetrical uncertainty [8], group-based fuzzy-rough FS (FRFS) [6], probabilistic consistency-based FS (PCFS) [9] and correlation-based FS (CFS) [10]. On the other hand, wrapper-based methods [11] contain learning algorithms. Learning algorithms generally provide better solution, but it takes more time to execute when compared with the filter-based algorithms. Hybrid methods [12] are combinations of filter and wrapper methods.

Feature selection problem is in the group of NP-hard problems. Building a prediction model with studying all possible subsets is not a preferred way due to its computational cost. Also, checking out the contribution of every predictor individually does not give correct information as there are correlations between predictors. Because of these reasons, there is no polynomial time exact method for this problem. Sometimes, alternative approaches, based on approximate solutions, are preferable. Regarding the NP-Hard problems, there are two classes of approximate methods proved to be successive for supplying powerful and flexible search strategies to generate well-quality solutions in acceptable run times: heuristics and metaheuristics. Heuristics or metaheuristics generally utilize random processes in the discovery of the search space to deal with the combinatorial explosion caused by the use of exact methods. In addition to be randomized, meta-heuristics are often repetitive, so the same search process is iterated till closing the solution. Metaheuristics produce good approximate solutions when applied to real-world or larger size problems [2,4].

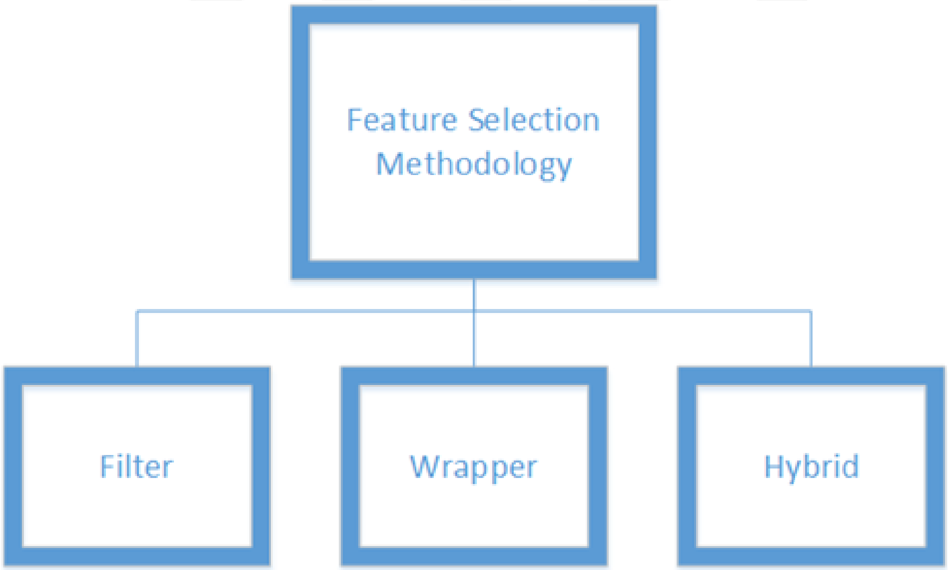

Figure 1.2 Classes of feature selection methodology

A feature selection example is simulated in the Figure 1.3. At the beginning, there are 13 columns to analyze. Feature selection operation is processing over these 13 columns and 8 of them are selected according to their accuracy rate. At the beginning, analysis accuracy of 13 columns is 85 percent but in the end, accuracy is increased. It must be noted that, feature selection not always has to increase accuracy because it reduces the column number and this benefit is already enough to use this preprocessing step. Generally, feature selection increases the accuracy with eliminating redundant columns.

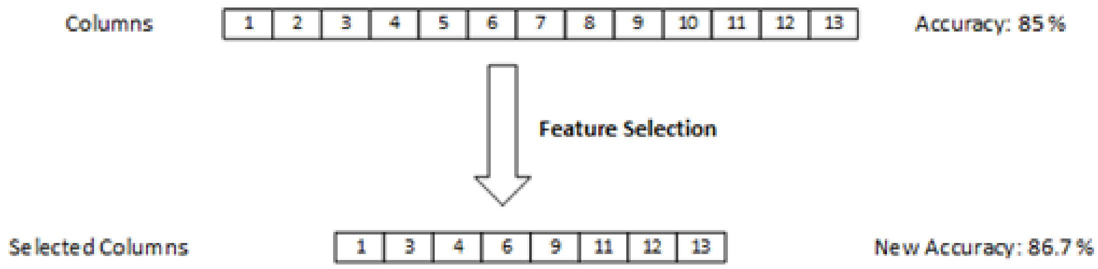

Figure 1.3 An example of feature selection

In this paper, we tackle the feature selection problem by exploiting the recently proposed and well-known metaheuristics by using the filter approach. The metaheuristics that we selected are migrating birds optimization (MBO), particle swarm optimization (PSO) , differential evolution (DE) and simulated annealing (SA). These algorithms are implemented from scratch by designing novel operators. As the subset evaluator, we preferred to use the consistency based method. The learner algorithm preferred in this study is the k nearest neighbor (k-NN) algorithm. To our best, it is the first time that the migrating birds optimization algorithm is exploited for solving the feature selection problem.

This thesis provides contribution by applying MBO to the feature selection problem for the first time and comparison of MBO, PSO, SA, DE algorithms with our framework and our parameters. A paper named “Exploitation and Comparison of Computational Intelligence Techniques on the Feature Selection Problem” which summarizes this study is accepted by INFUS 2019 Conference [13]. This framework is ready to be enhanced with consistency-based subset evaluator, MBO, PSO, SA, DE algorithms for searching new solutions and k-NN algorithm as classifier.

The rest is organized as follows: in section 2, the literature survey related to metaheuristics used in this study is given. Section 3 gives the implementation details of the implemented algorithms. The details about experiments and discussion about the results are presented in Section 4. Finally, section 5 concludes the study with possible future work.

2. LITERATURE SURVEY

Literature is expanding day by day for feature selection problem. In this section, various feature selection problem solving methodology are presented. Previous filter, wrapper and hybrid approaches used in feature selection problem are mentioned here. Also, some search algorithms which are used for exploring new solutions in the solution space are given with some example work. Four algorithms which are used in our framework are investigated in detailed.

2.1. Filter Approach

In filter methods, there is no learning algorithm to evaluate the quality of the solution. The quality and the performance of the columns are measured by looking at the relationship between the columns. Some examples from literature is given below.

An unsupervised feature selection is proposed by Solorio-Fernández with filter approach. One importance of their work is building this model for mixed data which includes both numerical and non-numerical data. They compared their work to the popular unsupervised filter methods and stated that their study has great performance against them [14].

Lyu et al. proposed Orthogonal MIC Feature Selection (OMICFS) based on the Maximal Information Coefficient (MIC) and Gram-Schmidt Orthogonalization (GSO). It is an improved approach under the max-relevance and min-redundancy criterion for biomedical data. Their work is compared with other filter feature selection methods and they showed that their method is more efficient at computational time and accuracy [15].

Adlakha and Chhikara used four filter approaches (Chi Square Attribute Evaluation, Relief, CFS and Gain Ratio) and five different classifiers (Naïve Bayes, Multilayer Perceptron, Sequential Minimum Optimization (SMO), J48 and Random Forests) for Steganalysis. They compared these filter and classifier combinations and showed their results. They emphasized the importance of the feature selection for steganalysis problem. They also concluded that CFS as a subset evaluator and SMO as a classifier is not a good combination [16].

Dubey and Saxena proposed a filter method called Cosine Similarity-based Filter feature selection Technique (CSF) for High-Dimensional Datasets. They used multi-classifier system which consists of k-NN, Classification and Regression Tree (CART), Naive Bayes (NB) and Support Vector Machine (SVM). This method is applied to four high-dimensional binary class datasets and the results are compared three known methods which are Sequential Random k Nearest Neighbor (SRKNN), Random Forest with backward elimination and Hamming Distanced Particle Swarm Optimization. They summarized that their method is performs at least the other methods [17].

We also used filter approach because its good performance in less time. For our all search algorithms which are MBO, PSO, SA, DE consistency-based subset evaluator is used to evaluate the candidate solutions' performances.

2.2. Wrapper Approach

In wrapper methods, there is a learning algorithm to evaluate the quality of the solution. The quality and the performance of the columns are measured by a learning algorithm. This means wrapper approach can give a better performance than the filter methods but wrapper approach is very expensive for computation because its learning mechanism. Some examples from literature is given below.

Sun et al. made a study based on lightning attachment procedure optimization and support vector machine for feature selection. They used KDD Cup 99 dataset. They compared their wrapper approach with classical PSO and GA. Their experiment showed that their proposed approach gained a better effiency and accuracy against the compared algorithms [18].

Chhikara and Kumari made a wrapper feature selection study that becomes from an improved version of PSO using neural networks. They used their approach and compared their results with the dataset without any feature selection. They got 7% improvement with the new version against the complete feature set. So, they showed the benefit of feature selection for steganalysis [19].

Yu et al. proposed a wrapper feature selection combining genetic algorithm and multiple logistic regression. After this study, they claimed that their proposed approach provides dimension reduction and adresses the redundancy problem [20].

Xue et al. proposed a novel wrapper method using extreme learning machine (ELM). They used GA and ELM together for improving GA to search for the optimum subsets in the huge feature space. They showed that their proposed mechanism is very effective [21].

2.3. Hybrid Approach

Hybrid approaches are the frameworks which uses both filter and wrapper approaches' power. Using filter method and wrapper method together forms the hybrid approach. Some examples of hybrid approach for feature selection from literature is given below.

Zheng et al made a hybrid methodology for predicting diabetes mellitus which becomes a common health problem for recent years. This approach uses information gain feature selection approach in the filter phase and support vector machine with sequential search method in the wrapper phase. To understand the efficiency, they compared their study with some other known methods like gain ratio, Pearson correlation and ReliefF. They noted that their approach outperformed against the other methods [22].

Alenizi & Mansour proposed a new intelligent hybrid feature selection method using information gain and genetic algorithm. They tested their approach against the other versions which uses information gain method alone. They used k-NN for classification. The comparison was made using ten datasets from UCI repository. They claimed their method is superior than the single information gain method [23].

Li et al. proposed a hybrid method using an improved F-score and extreme learning machine. They used five fold cross-validation on four datasets which is taken from UCI machine learning repository [24].

2.4. Search Algorithms

In this sub-section, ignoring being filter, wrapper or hybrid approach and focusing just on search algorithms, previous works are mentioned to have an idea about which search algorithms are used for the feature selection problem.

Genetic algorithm (GA) is applied to feature selection problem by Raymer et al. They used k-NN for classification. They made a comparison of their approach and some known feature selection and extraction techniques, namely sequential floating forward feature selection, and linear discriminant analysis [25].

Memetic algorithm (MA) is applied to feature selection problem by Yang et al. They used ReliefF and k-NN. According to their tests, they claimed that their work finds better accuracy than the existing methods [26].

Yu and Yuan, used *branch and bound* algorithm for feature selection. They improved the classical branch and bound algorithm and compared their version with the original branch and bound. They analysed the results and claimed that their version gives better output than the traditional version [27].

Deraeve and Alexander used novel *neural networks* for solving feature selection problem. They used cross-validation during feature selection phase. They compared their study to alternative known methods to see efficieny and they claimed that they provides increasing the accuracy, decreasing the computational cost with their approach [28].

Tabu search (TS) is also used for selecting features. They compared tabu search to other known methods like sequential methods, branch and bound method, genetic algorithm and sequential forward (backward) floating search methods on some synthetic data. They found tabu search as promising [29].

Diao and Shen adapted *harmony search* for feature selection problem. They used real valued datasets from UCI repository. They used 10 fold cross-validation to validate data. They compared their algorithm to GA, PSO and hill climbing (HC) [30].

Dara et al compared some heuristic search algorithms like GA, PSO, and gravitational search algorithm for feature selection problem. They used 10 fold cross-validation. [31]

Migrating birds optimization, particle swarm optimization, simulated annealing and differential evolution algorithms are selected for this study because of their strength. In the next sub-sections, the metaheuristics used in this study are explained briefly with their literature survey on the feature selection problem.

2.4.1. Migrating birds optimization

Migrating Birds Optimization algorithm is a search method that starts to search off from its neighbors. The algorithm begins with a set of predetermined solutions similar to birds in a V formation (Figure 2.1). The first solution which stands for the leader bird, is begining point and it proceeds to the lines until the tails. Each solution is improved by using its neighbor solutions. If the best neighbor solution helps to improve, this solution is taken instead of the current solution. Here, the previous solution makes benefits to the next solutions in a repetition mechanism. A solution evaluates its neighbors and its previous best neighbors for choosing the best of them instead of the current solution. Firstly, all solutions in the flock are developed by this neighbor method, then one of the second solutions in the V form takes the first place and another loop starts. The algorithm runs until it reaches to the stopping criteria. In this structure, the most energy spending bird is the leader bird. The other birds in the flock, can get benefit from the flight of the leader bird. The main aim of the MBO is discovering more areas of the solution space simultaneously with a number of parallel solutions. Every time, one solution (the leader bird) is explored in more detail. The main advantage in MBO is its parallel processing and mutual benefit mechanism among solutions (birds) [32].

Even though MBO is a recently proposed algorithm, it has been applied to many problems by the researchers. Migrating birds algorithm is applied to cell formation problem by Almonacid et al.. They compared two versions of MBO on this problem [33].

Zhang et al. used MBO on assembly line balancing problem. Their enhanced MBO algorithm is compared to the other versions of MBO and other algorithms which is included in the literature like PSO, TS, SA, GA, artificial bee colony and immune algorithms on U-shaped assembly worker assignment and balancing problem. They concluded their proposed algorithm outperformed better than the other all algorithms [34].

Duman and Elikucuk used MBO for solving credit card fraud detection problem with a little modification. The new version has a good performance according to the test results [35].

Niroomand et al. Applied MBO to closed loop layout problem. They modified MBO algorithm with using GA operators. The original MBO and SA algorithms are used for testing the proposed method. Their work is better than the other methods they compared according the tests [36].

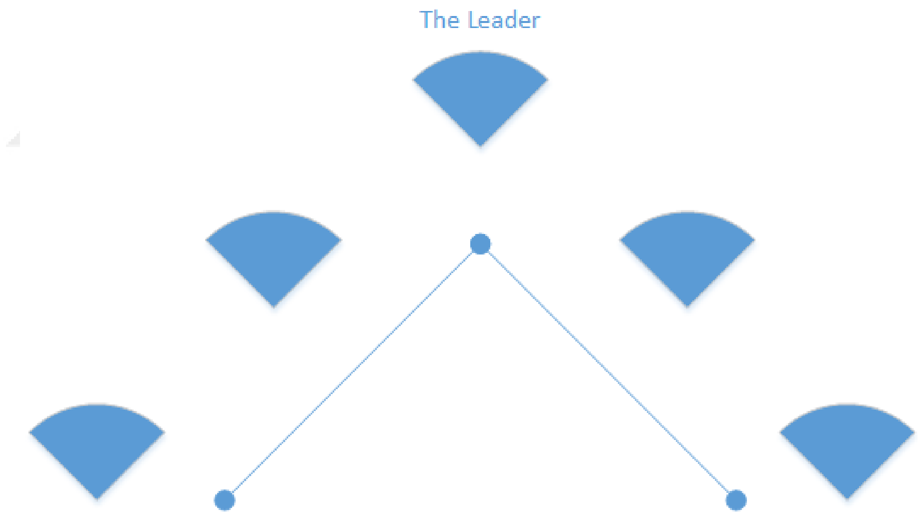

Figure 2.1 V formation of MBO

There is no study about application of the MBO algorithm for the feature selection problem in the literature.

2.4.2. Particle swarm optimization

The particle swarm concept is originated from the simulation of behavior of biological organisms. The starting point was to simulate visually the nuanced but unpredictable motion of bird flocking. The particle swarm optimization algorithm imitates the flying birds and their information exchange ways to solve optimization problems. Potential solution imitates a flying particle with a certain velocity. Velocity of the particle is tuned by itself according to the flying experience and other birds' experiences in the swarm (Figure 2.2). PSO has a variety of application area with hard combinatorial optimization problems. PSO is very useful for feature selection problem because

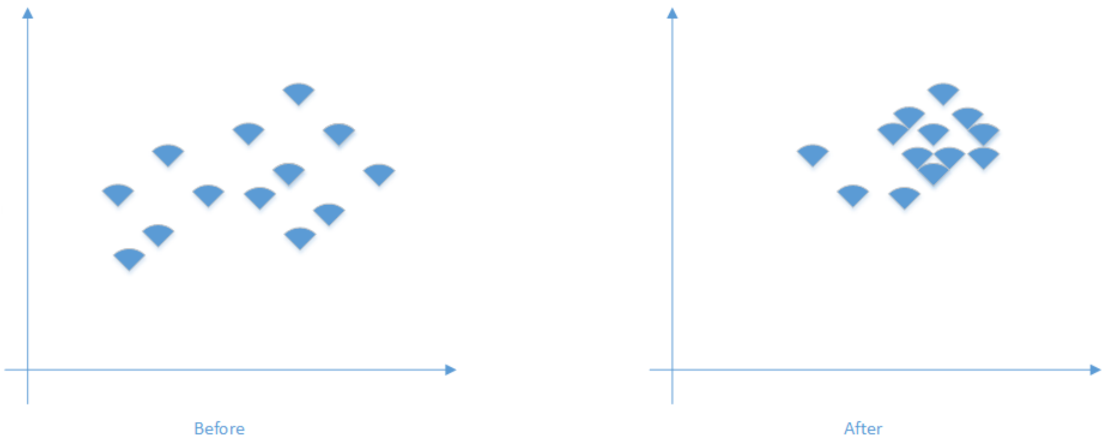

Figure 2.2 Before and after positions of birds in PSO

particle swarms has an opportunity to explore the best feature combinations while flying within the problem space. Also, the computational performance is evaluated with training algorithm on several datasets. It is understood that PSO has powerful search capability in the problem domain and has a capability of exploring almost optimal solutions in a reasonable computational time [37,38].

Liu et al. used an improved version of PSO for the feature selection problem and compared this new version to the traditional PSO, GA and grid search algorithm. They also used SVM with F-score method. They used 10 benchmark datasets from UCI machine learning and StatLog databases [39].

Chuang et al. contributed by improving binary PSO using catfish effect to contribute to the feature selection methodology. They used k-NN as classifier and leave one out cross validation method to compare the quality of the solutions. They found their method effective according to test results [2].

Djellali et al., compared enhanced PSO and genetic algorithm on feature selection problem. They used UCI data repository to carry the tests on. They concluded that their enhanced PSO is superior to their enhanced GA [40].

2.4.3. Simulated annealing

Simulated annealing is a stochastic metaheuristic to find an approximate solution to the global optimum of a defined function. This method takes the advantage of the annealing technique in metallurgy which heats and cools metals over and over for reaching the best quality (Figure 2.3).

Annealing in the metallurgy provides benefits like increasing the size of crystals and decreasing the faults. SA can be applied to feature selection problem as tuning up the root algorithm to ensure writing a record down for the current best solution. While other population-based algorithms in the feature selection manage selecting and improving multiple possible solutions, SA based feature selection method works on just one feature subset to search the most appropriate solution for the problem. This single solution is used to reach the thermal equilibrium. Once it becomes in a good state according to pre-given improvement criteria, SA adjusts the temperature parameter with an assumed cooling rate. Managing only one subset provides high efficiency clearly with decreasing the computational time but it also causes the algorithm not to find the best solution when the initial settings are not adjusted very well [41,42].

When we look at into literature there are some work which is done previously about feature selection problem. Meiri and Zahavi applied simulated annealing approach to feature selection problem in marketing applications. They also compared SA performance to the stepwise regression (SWR) algorithm [2].

Keerthana et al. used SA on an efficient cancer prediction mechanism. They combined SA and SVM algorithms to make a strong method. They compared their hybrid work with linear regression classifier. They found their work more efficient and accurate after this comparison [43].

Wan, made a comparison of simulated annealing and genetic algorithms on feature selection problem. They used NB on half of the datasets and k-NN on the other half of the datasets as classifier to see that their method's performance on different classifiers. This work showed that simulated annealing is more efficient than the genetic algorithm [44].

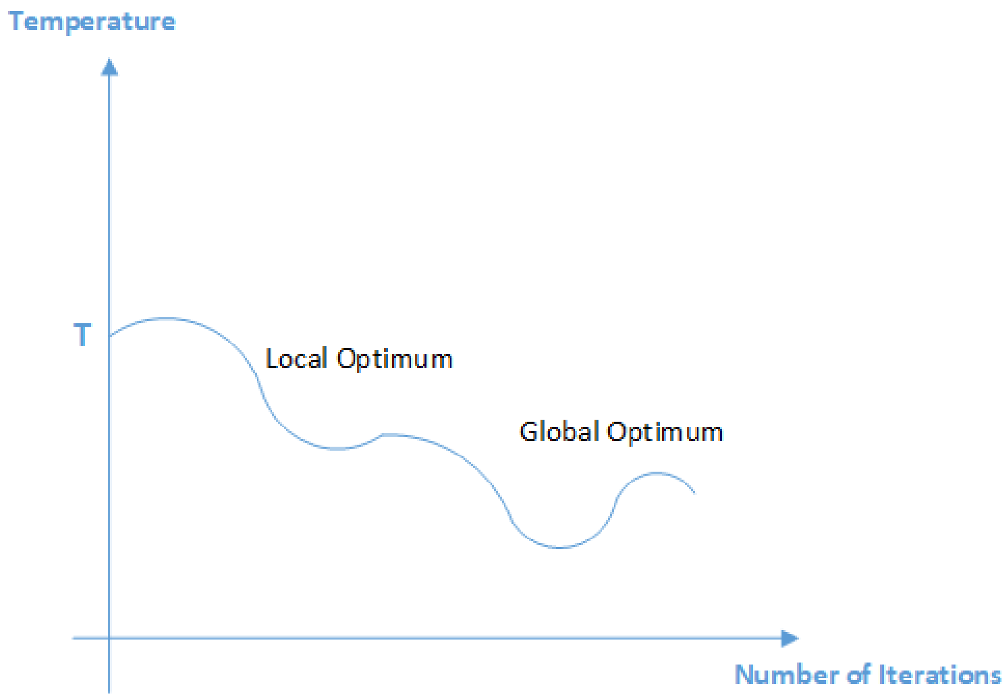

Figure 2.3 SA decreases the temperature starting with an initial value

2.4.4. Differential evolution

Differential evolution algorithm is an evolutionary computation (EC) technique that looks for the optimal solution for a problem by iteratively trying to enhance a possible solution with regards to a given measure of quality. DE is a stochastic direct search method and can be used for multi-dimensional real-valued optimization problems. DE is easy to implement with a few predefined parameters compared to other EC techniques and less expensive for the computation. DE tends to find global optimum solutions with escaping from local optimum. DE begins searching with an initial candidate solution which is generated randomly. After the initialization step, mutation (shown in Figure 2.4) and crossover operations which makes DE evolutionary, are applied to generate a better population [45,46].

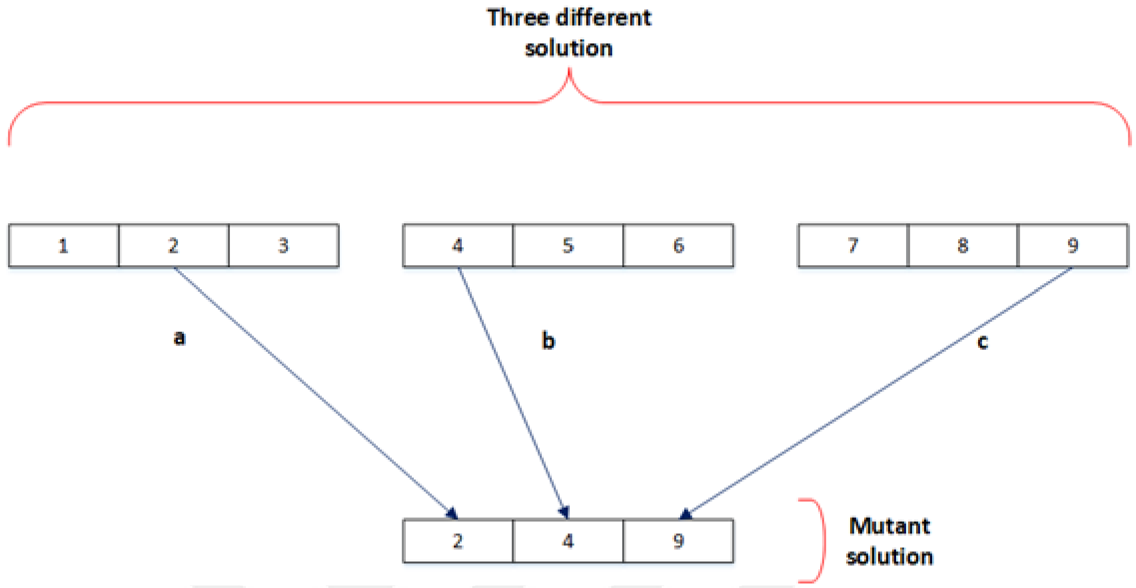

Figure 2.4 Mutation operation for DE

Hancer et al., used DE for feature selection with an improved version. They proposed a new filter based method based on mutual information and ReliefF. They used k-NN as classifier and tested their approach on 8 datasets from UCI machine learning repository [47].

Zorarpacı and Özel made a hybrid approach using DE and artificial bee colony (ABC) algorithms for feature selection. They used 15 datasets from UCI machine learning repository. They compared their hybrid method to original ABC and DE for making an evaluation about the performance of the new approach. They used 3 different subset evaluator which are information gain, chi-square, and correlation feature selection when making comparison. After some tests, they said that their approach is efficient and improves runtime performance [48].

Vivekanandan et al. used a modified differential evolution algorithm in feature selection problem and showed its effectiveness for prediction of heart disease. Datasets are taken from UCI machine learning repository [49].

Bhadra and Bandyopadhyay implemented an improved version of DE. This version takes into account three criterion which are average standard deviation of the selected features, average dissimilarity of the selected features and the average similarity of non-selected features. They mad some tests on 8 datasets from UCI repository. They concluded that their proposed algorithm is very effective according to their tests [50].

3. MATERIAL AND METHOD

In this section, implementation methodology, general structure, datasets, subset evaluator, search algorithms and classifier information of this work is given in detail. The panoramic visualisation of the study is drawn with giving the project framework.

In this study, first of all, the accuracies of datasets are calculated before any operation. These calculations are being used to compare the results to see any improvements. After that, these datasets are given four metaheuristic algorithms which were MBO, PSO, SA and DE. The best combination of features for every algorithm, is decided with using a subset evaluator. After this elimination phase, the accuracy of the new feature set is calculated and compared to the old ones.

The framework designed for comparing the algorithms on the feature selection problem is given in Figure 3.1.

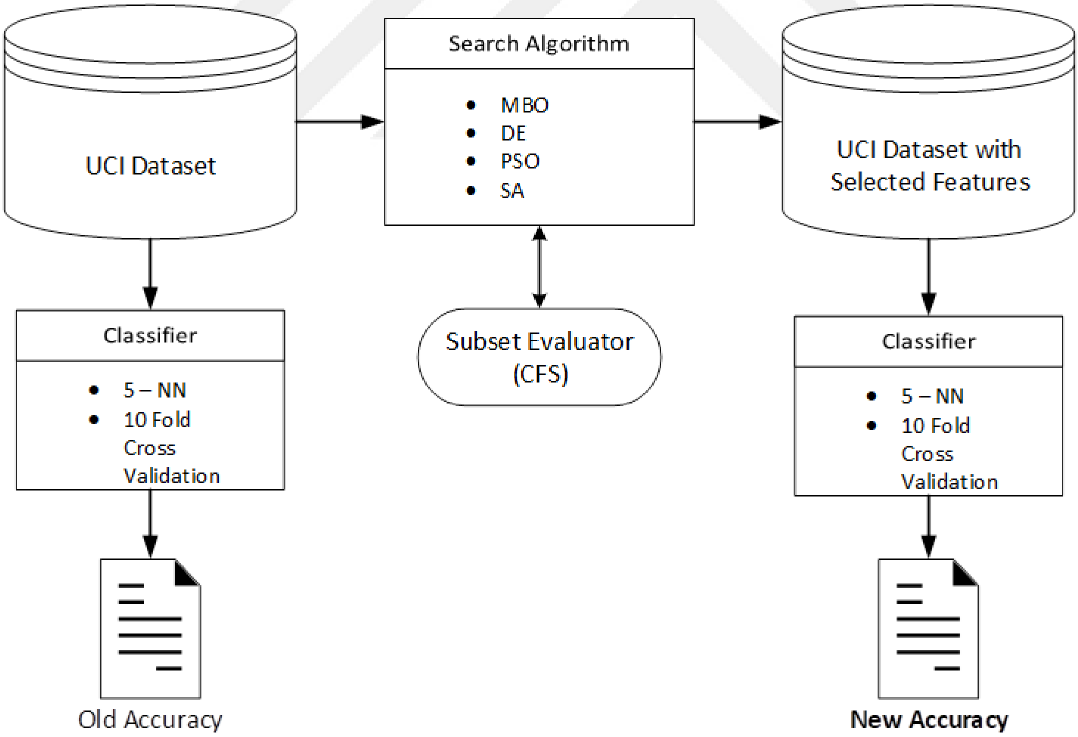

Figure 3.1 The framework designed for comparing the algorithms

3.1. Implementation Materials and Design

In this study, Java[51] is preferred as software development language. Eclipse is the IDE that is used development environment [52]. An object-oriented mechanism is designed for this project. Polymorphism properties of Java are used and the classes which has same qualifications are generated from the same super classes.

3.1.1. Solution Design

Every solution is thought as a bit string with the length of features in given dataset. Every bit represents the features given in order. Bits can be 0 or 1. 1 bit means the feature, the bit represented, is included; 0 bit means the feature, the bit represented, is not included. An example representation is shown below (Figure 3.2).

Given Dataset:

alcohol	malic acid	ash	alcalinity of ash	magnesium	total phenols	flavanoids
14.24	1.71	2.43	15.6	127	2.8	3.06
13.28	1.36	2.67	16.8	100	2.65	3.05
13.45	2.48	2.43	15.47	110	2.5	2.76
14.01	1.55	2.5	18.6	125	2.77	2.5

Solution Representation:

alcohol	malic acid	ash	alc. of ash	magnesium	tot. phenols	flavanoids
1	0	0	1	1	1	0

1: included
0: not included

Figure 3.2 Solution representation example from wine dataset

When calculating, feature existance is calculated as continous value to keep diversity. After all selection is done, the values are converted to binary values as selecting all values equal or bigger than 0.5 as 1 and all values less than 0.5 as 0. In all search algorithms, solution is constructed as continous and rounded to binary values.

3.1.2. Class Design

Every logic component is designed as a class. Subset evaluator, classifier, solution, population, metaheuristic, instance classes are created. MBO, PSO, SA and DE algorithms are designed as different classes which extend the metaheuristic class.

The object-oriented design is shown as a class diagram in Figure 3.3

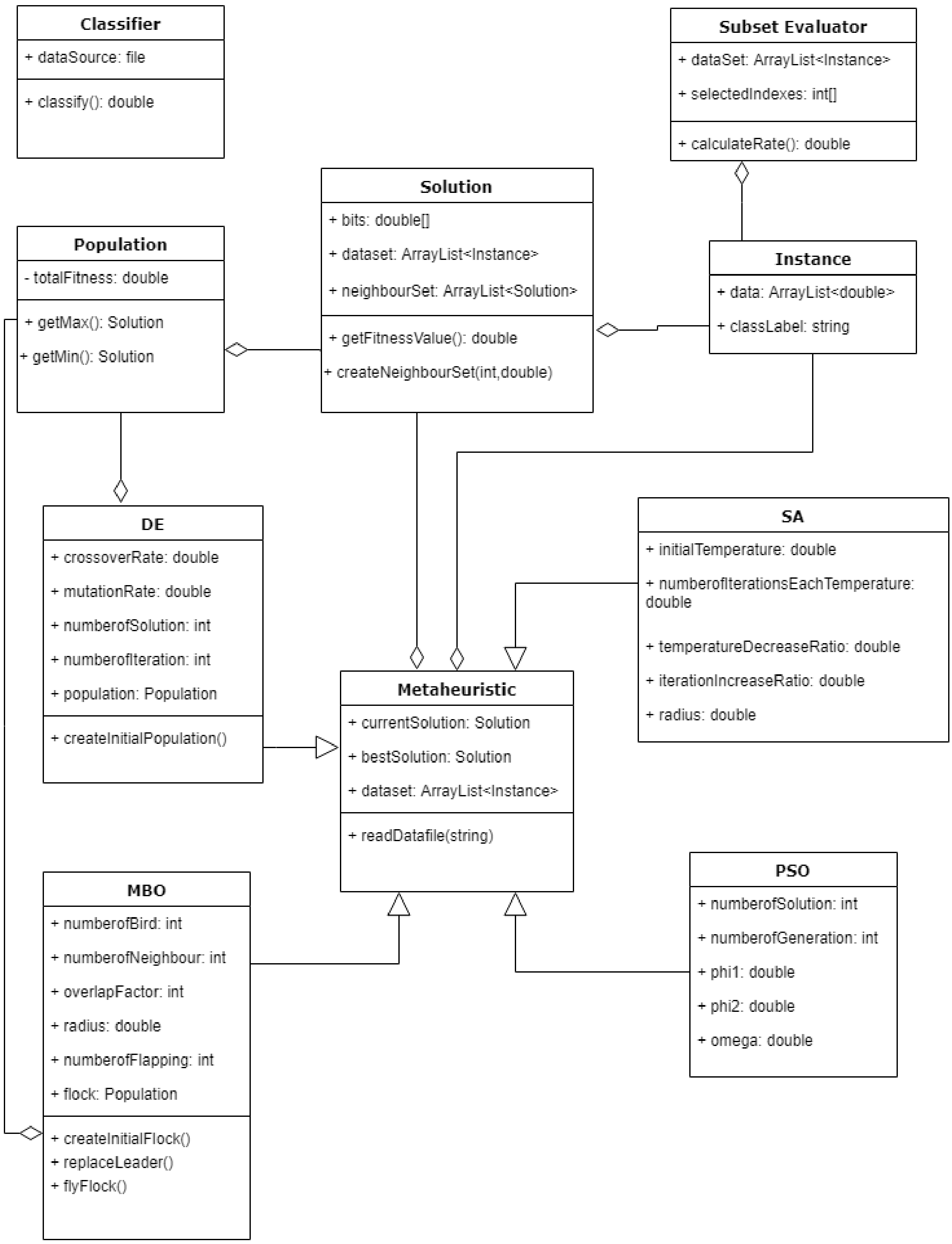

Figure 3.3 Class Diagram of the framework

3.2. Dataset

Proper UCI machine learning datasets [53] are selected for testing. UCI datasets are commonly used in the feature selection literature like [30,39,40,42,47,49]. Attributes of the datasets are listed in Table 3.1. To be able to make a comparison, categorical class labeled datasets are chosen.

In these datasets;

- if missing values exist, missing values are eliminated.
- class labels are converted to string format
- dataset files are converted to csv format

Table 3.1 UCI Datasets

Dataset	Number of Features	Number of Classes	Number of Instances
abalone	8	29	4177
iris	4	3	150
lymphography	18	4	148
optdigits	64	10	5620
sonar	60	2	208
spect	22	2	267
waveform	21	3	5000
wine	13	3	178

3.3. Subset Evaluator

The subset evaluation step is not a part of the feature selection process itself. Subset evaluator is useful for deciding if a column is needed or not and it checks the validity of the selected combination of the features.

A fitness function evaluates the fitness of a subset which is generated by some other procedures and this fitness value is compared with the found previous best. If it is much better than the previous one, then it will be replaced the previous best subset. An optimal subset is changable according to the evaluation function. In this study, we used consistency measure as subset evaluator.

Consistency-based feature selection

Consistency measure is identified by inconsistency rate over the given feature set. Inconsistency can be defined as, there must be at least two instances which have the same features but a different class label. For example, (0,1,a) and (0,1,b) instances have the same features but one of them has a class label named “a” and the other one has a class label named “b”.

The inconsistency count is the number of times a pattern appears in the data minus the largest number among different class labels. For example, for a subset S, a pattern p appears in np instances which np count consists of c1 instances has class label 1, c2 instances has class label 2, c3 instances has class label 3 where c1+c2+c3= np. If c3 is the largest number among all the three class labels, the inconsistency count is calculated as (n-c3). Also note that, the sum of all np’s over different patterns p is the total number of instances (P) in the dataset. The inconsistency rate of a feature subset S is the sum of all the inconsistency counts over all patterns of the feature subset that appears in the data divided by P [9].

3.4. Classifier and Data Validation

Classifier is used to test the accuracy. It is used before and after application of feature selection. This**,** provides the option of comparing the new combination of columns accuracy to the old one.

3.4.1. k-nearest neighbour

In this study, k-nearest neighbour algorithm is used for classification. The k-nearest neighbour is a popular nonparametric method which stores all cases and classifies new objects according to similarity measure. This algorithm is mentioned in 1951 by Fix and Hodges [56]. Classifiers do not need any model for k-nearest neighbors and purely based on the minimum distance between samples. k-nearest neighbour algorithm has started to be preferred since 1970 in many subjects like statistical estimation, pattern recognition, classification and interpretation, function learning etc. k-NN is a simple method and it is easy to implement. [1,57]. Pseudo code of k-NN [41] is given in Figure 3.4.

```
for all the unknown samples UnSample(i)
    for all the known samples Sample(j)
        compute the distance between UnSample(i) and Sample(j)
    end for
find the k smallest distances
locate the corresponding samples Sample(j1)...Sample(jk)
assign UnSample(i) to the class which appears more frequently
end for
```

Figure 3.4 Pseudo code of k-NN

There are 2 basic step in k-NN algorithm:

1. Find the k training instances which are closest to unknown instance
2. Pick the most commonly occurring classification for these k instances

Figure 3.5 simulates decision mechanism of the k-NN for k = 1 and k = 4. New items like with the question mark labeled, are categorized according to their nearest items' classes [58].

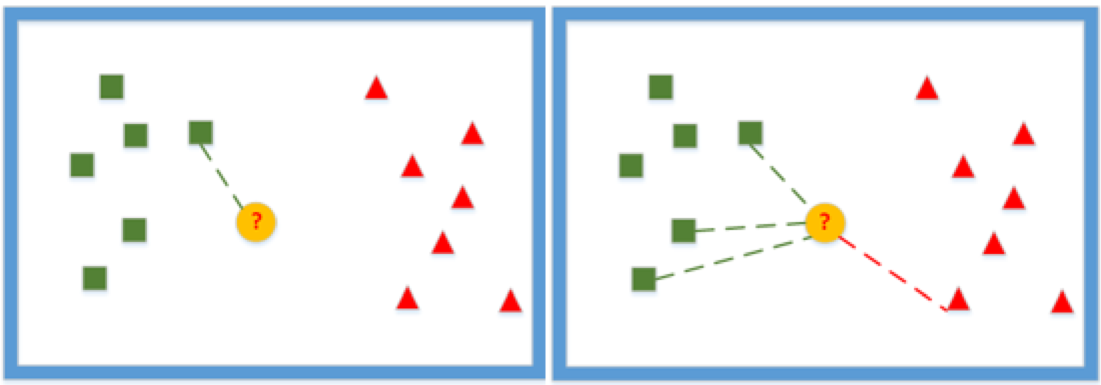

Figure 3.5 1-NN (left) and 4–NN (right) class assignment

3.4.2. 10-fold cross validation

10-fold cross validation (10-fcv) is a technique for fair data validation. In this method, given dataset is partitioned into 10 part. 9 of partitioned dataset are used for one training fold. The other subset is reserved as the testing data. This, provides comparison of the built classifiers with the same unseen data. This process is repeated 10 (fold number) times. 10 fold cross validation has the advantage of using every data as both training and testing. An extra advantage of this technique is its decreasing bias/variance problems [8].

3.5. Search Algorithms

Search algorithms are the techniques to reproduce new combinations of features. Four search algorithms are used in this study which are migrating birds optimization, particle swarm optimization, simulated annealing, differential evolution algorithms. In the next sub-sections, these four search algorithms are explained in detail with how they are applied to the feature selection problem.

3.5.1. Migrating birds optimization

MBO has some predetermined parameters like number of birds (*nob*), number of neighbors (*non*), number of flapping (*nof*), overlap factor (*olf*) and radius (*r*). *Nob* parameter determines how many birds fly in the flock, non determines how many neigbour a bird will have, *nof* specifies how much a flock will fly, *olf* takes place determining the neighbour set and radius parameter determines how far away a neigbour take place from a bird.

In this problem, an initial flock is created first. This flock is genereated from the given dataset randomly. A leader bird is selected among the other birds. The following birds are placed according to radius and olf parameters randomly. Every bird flaps and flock flies together. Every bird (solution) looks its neighbours for a better solution. If any neighbour solution provides any progress for the flock, it becomes the new current solution. After a while, when it accessed the number of flapping, the leader bird moves to the end and one of the following birds become the leader.

The pseudo code of MBO is represented by Figure 3.6. *noi* shows the number of iterations, *m* shows number of flapping, *x* shows the number of solutions to be shared with the next solution, *non* shows the number of neighbour solutions to be considered.

In MBO, a neighbour of a solution is created as swapping bits of solution randomly and a *radius* parameter is used to adjust the magnitude of change.

```
1. Generate n initial solutions in a random manner and place them on an
hypothetical V formation arbitrarily
2.  i = 0
3.  while(i<noi)
4.          for( j=0; j<m ; j++)
5.                      Try to improve the leading solution by generating
                        and evaluating k neighbours of it
6.                      i = i + non
7.                      for each solution in the flock (except leader)
8.                                Try to improve solution by evaluating (non-x)
                                            neighbours of it and x unused
                                  best neighbours from the solution
                        in the front
9.                                i = i + (non-x)
10.                     end for
11.         end for
12.  Move the leader solution to the end and forward one of the
            solutions following it to the leader position
13.  end while
14.  return the best solution in the flock
```

Figure 3.6 Pseudo code of MBO

3.5.2. Particle swarm optimization

The parameters of PSO are *phi1*, *phi2*, *omega*, number of generations (*nog*) and number of solutions (*nos*). *phi1*, *phi2* and *omega* parameters are for changing velocity of particles. Pseudo code of PSO is given in Figure 3.7. They are used as coefficients to change velocities (Eq. (1)). *nog* parameter is for determining how many generation is produced and *nos* parameter determines solution number of a population.

$$V=V*omega+phi1*random()*(Pb-C)+phi2*random()*(Gb-C) \qquad (1)$$

Where *V* is the velocity of the particle, *Pb* is the personal best solution, *Gb* is the global best solution and *C* is the current solution.

```
Inputs: phi1
phi2
omega
Number of generations (nog)
Number of solutions (nos)

1. Create initial population
2. create velocity for each solution
3. while (iteration < iteration number
4.          change velocity of each solution (with phi1, phi2,
                      omega)
5.          change position of each solution
6.          if ( fitness(ns) < fitness(bs))
7.          bs = ns
8.          add bs to the next generation
11. end while
```

Figure 3.7 Pseudo code of PSO

In appliance of this problem to PSO, first of all an initial population is created. Each solution is given a velocity. Best solution of first generation is saved into a variable.

After this initialization part, the velocity and position of each solution are changed one by one. If the new position gives better performance than the previous one, this new version is saved as the best solution. This process lasts until the stopping criteria is met and the best solution found until the run finished, is given as the best output.

3.5.3. Simulated annealing

SA algorithm has some predetermined parameters like *initial temp* (T), *number of iterations in each temperature* (R), *temperature decrease ratio* (a), *increase ratio of R* (b) and *radius* (r). T is the starting temperature, temperature decreases as much as "a" parameter. It is a predefined value that how many iterations will be done in every temperature (R). R increases in ratio of "b" parameter. "r" parameter determines how far away a neigbour take place from a solution. The pseudo code of SA is given in Figure 3.8

```
Inputs: initial temp (T)
Number of iterations in each temperature (R)
Temperature decrease ratio (a)
Increase ratio of R (b)
radius (r)
Number of iterations (noi)

1. Create initial population
2. while (T>0 )
3.        for every solution in population
4.                if (fitness(ns) - fitness(cs))
5.                cs = ns
6.                else if (randomization appears)
7.                cs = ns
8.                if (fitness(ns)<fitness(bs))
9.                bs = ns
10.       end for
11.       T=T/a
12. end while
```

Figure 3.8 Pseudo code of SA

Simulated annealing is an algorithm which is easy to implement. Current solution is determined initially. Neighbour solution can place according to *radius* parameter. It started with an initial temperature and the temperature decreases in every iteration. If fitness value of neighbor solution is better than the fitness value of current solution, neighbour solution becomes current solution. There is also possibility of changing the current solution with the neighbour with a random value. This facility helps the algorithm for skipping local optimum and looks for another places in the solution space. In our SA implementation, a neighbour of a solution is created as swapping bits of solution randomly and a *radius* parameter is used to adjust the magnitude of change.

3.5.4. Differential evolution

Differential evolution algorithm has some predetermined parameters like *crossover rate (cxp), F, number of generations (nog) and number of solutions (nos). cxp* parameter is one of the crossover possibility criteria for DE. *F* parameter is a coefficient used for mutation. *Nog* parameter defines how many generations will be produced and nos parameter specifies how many solutions will be in population.

This algorithm starts with an initial solution. Three different solutions are selected randomly. A mutant solution is created from these three solutions. Difference of two solution is multiplying with *F* parameter and the other one solution is added. Cross over operation which exists in evolutionary techniques is made. Cross over is made by mixing current solution and the mutant solution with a possibility (*cxp* parameter specifies this). The new mutant solution is compared to the current solution and if the fitness value of mutant solution is better than the fitness value of the current solution, the mutant solution takes place in the next generation. Here, crossover operation provides probability of skipping the local optimum and achieving the global optimum.

The pseudo code of DE is given in Figure 3.9.

```
Inputs: crossover rate (cxp)
Mutation rate (mp)
F
Number of generations (nog)
Number of solutions (nos)

1. Create initial population
2. for every solution in the population
3.        cs = ith solution
4.        randomly select 3 different solution from
                the population
5.        create a mutant solution
6.        make crossover to mutant solution
7.        compare mutant and cs
8.        if (mutant> cs)
9.        add mutant to next generation
10.       else
11.       add cs to next generation
12. end for
```

Figure 3.9 Pseudo code of DE

4. RESULTS AND DISCUSSION

4.1. Parameter Setting

For deciding which parameters to use for the search algorithms, some preliminary tests are conducted. Each parameter combination is applied 5 times and their average is computed. The parameter values which are tried for each algorithm are given inTable 4.1. These preliminary tests are conducted on two datasets and the number of solutions that algorithm can create while surfing in the solution space is limited to 20000. The best set of parameter values are given in Table 4.1 in bold.

Table 4.1. Values of parameters used in the fine tune experiments. (Bold ones are the best)

MBO		**PSO**		**SA**		**DE**	
Param.	Values	Param.	Values	Param.	Values	Param.	Values
nob	5, 11, **21**, 51	phi1	**0.1**, 0.5, 1,2	T	**100**, 1000	cxp	**0.1**,0.5,0.9
non	3, **5**, 7	phi2	0.1, **0.5**, 1,2	R	5, **20**	mp	**0.1**,0.5,0.9
nof	**5**, 10	omega	0.1, 0.5, **1**	a	1.1, **1.5**	F	**0.1**, 0.5, 1, 2
olf	**1**, 2, 3	nog	50, **100**, 200	b	1.1, **1.5**	nog	**50**, 100, 200
r	0.01, **0.02**,0.05	nos	**50**, 100, 200	r	0.01, 0.02, **0.05**	nos	**50**, 100, 200

4.2. Run Results

To compare these four search algorithms, common subset evaluator (consistency based) and classifier (5-NN) are used. They are all applied to public 8 datasets which are taken from UCI (Table 2) and each parameter combination is applied 10 times and their average is computed. k-Nearest Neighbor algorithm from Waikato Environment for Knowledge Analysis (WEKA) libraries used for classifier [54]. 5 nearest neighbors are used for classification as in [55]. 10 fcv is done for every classification step.

Next sub-sections shows the run results for 1K, 10K, 20K, 30K, 50K iterations.

4.2.1. Migrating birds optimization

Here, 1K, 10K, 20K, 30K, 50K iteration results of MBO algorithm are shown in Table 4.2-Table 4.6. In these tables, dataset names, feature number before any operations (old feature number), average feature number (average feature number), accuracy before any operations (old accuracy), max accuracy found after this method (max accuracy) is given.

Table 4.2 MBO Algorithm's 1K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,7	22,96	24,23
iris	4	3,9	95,33	96
lymphography	18	12,7	79,73	79,05
optdigits	64	33,1	98,7	96,87
sonar	60	35,5	84,62	86,06
spect	22	15,7	80,9	83,15
waveform	21	14,8	82,26	81,3
wine	13	7,2	95,51	98,88

Table 4.3 MBO Algorithm's 10K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,5	22,96	24,66
iris	4	3,8	95,33	96
lymphography	18	12,9	79,73	81,76
optdigits	64	33,2	98,7	97,44
sonar	60	36,3	84,62	85,58
spect	22	16,4	80,9	83,52
waveform	21	14,3	82,26	81,4
wine	13	8,2	95,51	97,19

Table 4.4 MBO Algorithm's 20K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,9	22,96	24,11
iris	4	3,9	95,33	95,33
lymphography	18	12,4	79,73	81,08
optdigits	64	30,4	98,7	97,14
sonar	60	33,1	84,62	85,58
spect	22	17	80,9	83,9
waveform	21	14,3	82,26	80,52
wine	13	7,1	95,51	96,07

Table 4.5 MBO Algorithm's 30K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,1	22,96	23,73
iris	4	4	95,33	95,33
lymphography	18	12,6	79,73	83,11
optdigits	64	31,1	98,7	97,4
sonar	60	35,9	84,62	84,62
spect	22	16,8	80,9	84,64
waveform	21	14	82,26	80,64
wine	13	7,7	95,51	97,19

Table 4.6 MBO Algorithm's 50K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,6	22,96	23,32
iris	4	4	95,33	95,33
lymphography	18	12,5	79,73	79,73
optdigits	64	32,3	98,7	97,56
sonar	60	36,3	84,62	83,65
spect	22	17,2	80,9	83,9
waveform	21	14,9	82,26	81,96
wine	13	8,4	95,51	97,19

When comparing the averages of maximum accuracies, the best results are found for 10K iterations. But when looking individually, for example, for iris dataset accuracy does not change between 1K and 10K or for lymphography dataset 30K iterations gives better result than 10K iterations. To generalize, we can say 10K iteration number is suitable with looking at the overall results.

4.2.2. Particle swarm optimization

Here, 1K, 10K, 20K, 30K, 50K iteration results of PSO algorithm are showed in Table 4.7--Table 4.11. In these tables, dataset names, feature number before any operations (old feature number), average feature number (average feature number), accuracy before any operations (old accuracy), max accuracy found after this method (max accuracy) is given.

Table 4.7 PSO Algorithm's 1K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	7	22,96	23,27
iris	4	4	95,33	95,33
lymphography	18	14,8	79,73	80,41
optdigits	64	31,2	98,7	97,37
sonar	60	57,5	84,62	85,58
spect	22	21,9	80,9	82,4
waveform	21	14,6	82,26	81,3
wine	13	7,4	95,51	98,31

Table 4.8 PSO Algorithm's 10K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,8	22,96	23,27
iris	4	4	95,33	95,33
lymphography	18	13,6	79,73	82,43
optdigits	64	33,2	98,7	97,1
sonar	60	57	84,62	86,54
spect	22	21,9	80,9	82,4
waveform	21	15,1	82,26	81,98
wine	13	8,8	95,51	96,63

Table 4.9 PSO Algorithm's 20K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,4	22,96	23,32
iris	4	4	95,33	95,33
lymphography	18	14,5	79,73	79,73
optdigits	64	34,5	98,7	97,72
sonar	60	57,4	84,62	85,58
spect	22	21,7	80,9	82,4
waveform	21	14,3	82,26	81,64
wine	13	7,9	95,51	98,31

Table 4.10 PSO Algorithm's 30K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	7	22,96	23,32
iris	4	4	95,33	95,33
lymphography	18	13,1	79,73	81,08
optdigits	64	30,9	98,7	97,15
sonar	60	57,8	84,62	85,58
spect	22	21,8	80,9	82,4
waveform	21	14,5	82,26	82,28
wine	13	8	95,51	97,75

Table 4.11 PSO Algorithm's 50K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,6	22,96	23,32
iris	4	4	95,33	95,33
lymphography	18	14,2	79,73	81,76
optdigits	64	27,9	98,7	97,72
sonar	60	56,4	84,62	86,06
spect	22	21,8	80,9	81,65
waveform	21	14,8	82,26	81,42
wine	13	7	95,51	98,31

When comparing the averages of maximum accuracies, the best results are found for 10K iterations. But when looking individually, for example, for iris dataset accuracy does not change between 1K and 10K or for wine dataset 20K iterations gives better result than 10K iterations. To generalize, we can say 10K iteration number is suitable with looking at the overall results.

4.2.3. Simulated annealing

Here, 1K, 10K, 20K, 30K, 50K iteration results of SA algorithm are shown in Table 4.12- Table 4.16. In these tables, dataset names, feature number before any operations (old feature number), average feature number (average feature number), accuracy before any operations (old accuracy), max accuracy found after this method (max accuracy) is given.

Table 4.12 SA Algorithm's 1K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	5,7	22,96	24,4
iris	4	3,5	95,33	96
lymphography	18	12,3	79,73	81,76
optdigits	64	33	98,7	97,12
sonar	60	35,1	84,62	85,58
spect	22	15,1	80,9	85,02
waveform	21	13,8	82,26	81,14
wine	13	7,6	95,51	97,75

Table 4.13 SA Algorithm's 10K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,8	22,96	23,27
iris	4	4	95,33	95,33
lymphography	18	12,8	79,73	79,05
optdigits	64	32,4	98,7	97,42
sonar	60	34,9	84,62	84,13
spect	22	17,8	80,9	84,64
waveform	21	13,8	82,26	81,68
wine	13	6,8	95,51	96,63

Table 4.14 SA Algorithm's 20K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,7	22,96	23,1
iris	4	4	95,33	95,33
lymphography	18	12,3	79,73	75,88
optdigits	64	33	98,7	94,39
sonar	60	37,3	84,62	82,36
spect	22	19,1	80,9	81,91
waveform	21	13,9	82,26	79,79
wine	13	7,3	95,51	94,21

Table 4.15 SA Algorithm's 30K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,4	22,96	23,32
iris	4	4	95,33	95,33
lymphography	18	12,7	79,73	81,08
optdigits	64	31,4	98,7	96,83
sonar	60	37,8	84,62	85,1
spect	22	18,9	80,9	83,52
waveform	21	14,5	82,26	81,24
wine	13	7,5	95,51	97,19

Table 4.16 SA Algorithm's 50K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,6	22,96	23,22
iris	4	4	95,33	95,33
lymphography	18	13	79,73	81,76
optdigits	64	33,8	98,7	96,8
sonar	60	38,3	84,62	87,02
spect	22	19,5	80,9	83,9
waveform	21	14,2	82,26	80,7
wine	13	7,4	95,51	97,75

When comparing the averages of maximum accuracies, the best results are found for 1K iterations. But when looking individually, for example, for optdigits dataset 10K iterations gives better result than 1K iterations. To generalize, we can say 1K iteration number is suitable with looking at the overall results.

4.2.4. Differential evolution

Here, 1K, 10K, 20K, 30K, 50K iteration results of DE algorithm are showed in Table 4.17 - Table 4.21. In these tables, dataset names, feature number before any operations (old feature number), average feature number (average feature number), accuracy before any operations (old accuracy), max accuracy found after this method (max accuracy) is given.

Table 4.17 DE Algorithm's 1K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,4	22,96	23,32
iris	4	4	95,33	95,33
lymphography	18	13,4	79,73	82,43
optdigits	64	31,5	98,7	97,38
sonar	60	38,4	84,62	86,06
spect	22	19,5	80,9	82,02
waveform	21	15,1	82,26	81,1
wine	13	8,6	95,51	97,19

Table 4.18 DE Algorithm's 10K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,3	22,96	23,32
iris	4	4	95,33	95,33
lymphography	18	15,1	79,73	81,08
optdigits	64	32	98,7	96,53
sonar	60	40,9	84,62	84,13
spect	22	21,8	80,9	82,4
waveform	21	18,3	82,26	81,66
wine	13	10,1	95,51	97,75

Table 4.19 DE Algorithm's 20K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,7	22,96	23,32
iris	4	4	95,33	95,33
lymphography	18	15,2	79,73	80,41
optdigits	64	35,4	98,7	98,36
sonar	60	39,5	84,62	85,1
spect	22	21,8	80,9	82,4
waveform	21	18	82,26	82,26
wine	13	10,2	95,51	97,19

Table 4.20 DE Algorithm's 30K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6,2	22,96	23,32
iris	4	4	95,33	95,33
lymphography	18	15,8	79,73	81,76
optdigits	64	34	98,7	97,97
sonar	60	39,8	84,62	85,1
spect	22	21,7	80,9	82,4
waveform	21	18,2	82,26	82,22
wine	13	9,7	95,51	98,31

Table 4.21 DE Algorithm's 50K iteration results (All values are in percentages)

Dataset	Old Feature Number	Average Feature Number	Old Accuracy	Max Accuracy
abalone	8	6	22,96	23,32
iris	4	4	95,33	95,33
lymphography	18	15,6	79,73	80,41
optdigits	64	30,6	98,7	97,38
sonar	60	40,9	84,62	85,58
spect	22	22	80,9	80,9
waveform	21	18,1	82,26	82,06
wine	13	9,7	95,51	97,19

When comparing the averages of maximum accuracies, the best results are found for 30K iterations. But when looking individually, for example, for spect dataset accuracy does not change between 20K and 30K or for optdigits dataset 20K iterations gives better result than 30K iterations. To generalize, we can say 30K iteration number is suitable with looking at the overall results.

4.3. Discussions

Next tables (Table 4.22-4.26) are summarized 1K, 10K, 20K, 30K, 50K iteration accuracy results for every algorithm. Here, bold ones are the best accuracy results can be found. It is understood that MBO is better than the other algorithms in terms of number of winning cases when looking at total cases.

Table 4.22 Algorithms' 1K iteration comparison

Dataset	Old Accuracy	MBO	PSO	SA	DE
abalone	22,96	24,23	23,27	**24,4**	23,32
iris	95,33	**96**	95,33	**96**	95,33
lymphography	79,73	79,05	80,41	81,76	**82,43**
optdigits	98,7	96,87	97,37	97,12	**97,38**
sonar	84,62	**86,06**	85,58	85,58	**86,06**
spect	80,9	83,15	82,4	**85,02**	82,02
waveform	82,26	**81,3**	**81,3**	81,14	81,1
wine	95,51	**98,88**	98,31	97,75	97,19

Table 4.23 Algorithms' 10K iteration comparison

Dataset	Old Accuracy	MBO	PSO	SA	DE
abalone	22,96	**24,66**	23,27	23,27	23,32
iris	95,33	**96**	95,33	95,33	95,33
lymphography	79,73	81,76	**82,43**	79,05	81,08
optdigits	98,7	**97,44**	97,1	97,42	96,53
sonar	84,62	85,58	**86,54**	84,13	84,13
spect	80,9	83,52	82,4	**84,64**	82,4
waveform	82,26	81,4	**81,98**	81,68	81,66
wine	95,51	97,19	96,63	96,63	**97,75**

Table 4.24 Algorithms' 20K iteration comparison

Dataset	Old Accuracy	MBO	PSO	SA	DE
abalone	22,96	**24,11**	23,32	23,32	23,32
iris	95,33	**95,33**	**95,33**	**95,33**	**95,33**
lymphography	79,73	**81,08**	79,73	**81,08**	80,41
optdigits	98,7	97,14	97,72	97,54	**98,36**
sonar	84,62	**85,58**	**85,58**	85,1	85,1
spect	80,9	83,9	82,4	**84,64**	82,4
waveform	82,26	80,52	81,64	81,2	**82,26**
wine	95,51	96,07	**98,31**	97,19	97,19

Table 4.25 Algorithms' 30K iteration comparison

Dataset	Old Accuracy	MBO	PSO	SA	DE
abalone	22,96	**23,73**	23,32	23,32	23,32
iris	95,33	**95,33**	**95,33**	**95,33**	**95,33**
lymphography	79,73	**83,11**	81,08	81,08	81,76
optdigits	98,7	97,4	97,15	96,83	**97,97**
sonar	84,62	84,62	**85,58**	85,1	85,1
spect	80,9	**84,64**	82,4	83,52	82,4
waveform	82,26	80,64	**82,28**	81,24	82,22
wine	95,51	97,19	97,75	97,19	**98,31**

Table 4.26 Algorithms' 50K iteration comparison

Dataset	Old Accuracy	MBO	PSO	SA	DE
abalone	22,96	**23,32**	**23,32**	23,22	**23,32**
iris	95,33	**95,33**	**95,33**	**95,33**	**95,33**
lymphography	79,73	79,73	**81,76**	**81,76**	80,41
optdigits	98,7	97,56	**97,72**	96,8	97,38
sonar	84,62	83,65	86,06	**87,02**	85,58
spect	80,9	**83,9**	81,65	**83,9**	80,9
waveform	82,26	81,96	81,42	80,7	**82,06**
wine	95,51	97,19	**98,31**	97,75	97,19

4.3.1. Effect of Number of Solutions

Next table shows number of winning cases for every algorithms in terms of solution number. Bold values are the best accuracy average for every algorithms. Actually, we can not say the accuracy decrease or increase for all algorithms when the iteration number increases. It is a problem and algorithm-spesific issue. There is no dramatically change in accuracy but some algorithms get a little more accuracy in some iteration numbers. **Figure 4.1** shows the comparison of results.

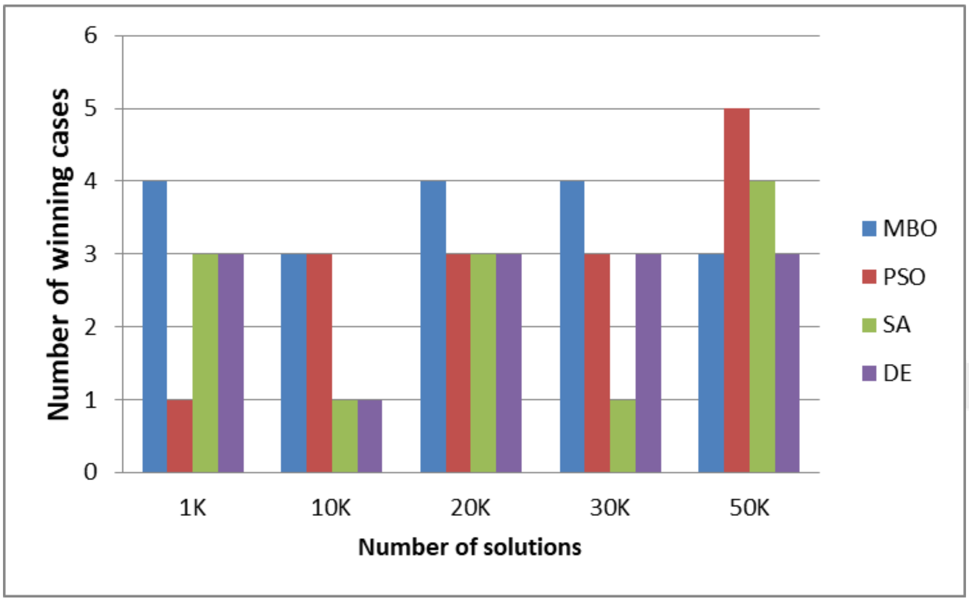

Figure 4.1 Comparison of number of winning cases

4.3.2. Effect of number of rows in datasets

Below, there is a list of datasets which is sorted from least row number to most row number and algorithm which founds best accuracy overall tests. It can be notable that when the row number increases, performance of PSO and DE algorithms increases.

Table 4.27 Algorithms find the maximum accuracy of datasets, in row number order

Dataset	Row Number	Best Found Accuracy Algorithm
lymphography	148	MBO
iris	150	MBO, SA
wine	178	MBO
sonar	208	SA
spect	267	SA
abalone	4177	MBO
waveform	5000	PSO
optdigits	5620	DE

4.3.3. Effect of number of features in datasets

Below, there is a list of datasets which is sorted from least column number to most column number and algorithm which founds best accuracy overall tests. It can be notable that when column number is less, MBO finds maximum accuracy value against the other algorithms.

Table 4.28 Algorithms find the maximum accuracy of datasets, in row number order

Dataset	Column Number	Best Found Accuracy Algorithm
iris	4	MBO,SA
abalone	8	MBO
wine	13	MBO
lymphography	18	MBO
waveform	21	PSO
spect	22	SA
sonar	60	SA
optdigits	64	DE

5. CONCLUSIONS

In this study, a filter based approach for feature selection problem is developed. MBO algorithm is applied to feature selection problem as a novel work and its performance is compared to other selected algorithms in the literature which are particle swarm optimization, simulated annealing and differential evolution. 8 public real-valued datasets are used for this purpose from UCI machine learning repository. Within the framework developed, consistency based selection evaluator is implemented and k-NN is used as the learner algorithm. The metaheuristic algorithms and their novel operators are implemented from scratch. Their parameters' best performing values are obtained after extended computational experiments and they are all compared under the same conditions. The framework is run for every search algorithms with the best parameter values found. Some tests are made to compare the algorithms in different number of solutions. Effect of number of solutions which is a predefined parameter, is discussed. Effect of row and column numbers of datasets are considered and compared. For every cases, algorithms are run 10 times and their average is calculated. When looking on the all datasets and all number of solutions for all algorithms, results show that MBO is better than the other algorithms in terms of number of winning cases.

MBO algorithm has some properties which distinguishes it from the other metaheuristic approaches. The properties are benefit mechanism between the solutions, more detailed exploration of promising areas and a number of solutions running in parallel. We believe that the superior performance of MBO algorithm is due to its aforementioned properties.

This study has the property of using MBO for feature selection problem and comparing MBO to some other algorithms on feature selection for the first time.

As a future work, exploitation of different evaluators and learner algorithms may be included in the framework. Some other search algorithms can be added into comparison. Different datasets which has some characteristic features like having much more or much less rows or columns can be added to exist ones.

REFERENCES

1. Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, *1*(1-4), 131-156.
2. Chuang, L. Y., Tsai, S. W., & Yang, C. H. (2011). Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Systems with Applications*, *38*(10), 12699-12707.
3. Meiri, R., & Zahavi, J. (2006). Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research*, *171*(3), 842-858.
4. Apipie, F. M. (2017). HYBRID METAHEURISTICS FOR OPTIMIZATION. *Young Economists Journal/Revista Tinerilor Economisti*, *14*(29).
5. Zheng, Z., Wu, X., & Srihari, R. (2004). Feature selection for text categorization on imbalanced data. *ACM Sigkdd Explorations Newsletter*, *6*(1), 80-89.
6. Jensen, R., & Shen, Q. (2007). Fuzzy-rough sets assisted attribute selection. *IEEE Transactions on fuzzy systems*, *15*(1), 73-89.
7. Jensen, R., & Shen, Q. (2008). *Computational intelligence and feature selection: rough and fuzzy approaches* (Vol. 8). John Wiley & Sons.
8. Kannan, S. S., & Ramaraj, N. (2010). A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm. *Knowledge-Based Systems*, *23*(6), 580-585.
9. Dash, M., & Liu, H. (2003). Consistency-based search in feature selection. *Artificial intelligence*, *151*(1-2), 155-176.
10. Hall, M. A. (1999). Correlation-based feature selection for machine learning.
11. Hsu, C. N., Huang, H. J., & Dietrich, S. (2002). The ANNIGMA-wrapper approach to fast feature selection for neural nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *32*(2), 207-212.
12. Zhu, Z., Ong, Y. S., & Dash, M. (2007). Wrapper–filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *37*(1), 70-76.
13. “İTÜ Intelligent and Fuzzy Systems Conference, 2019” [Online]. Available:

https://infus.itu.edu.tr/

14. Solorio-Fernández, S., Martínez-Trinidad, J. F., & Carrasco-Ochoa, J. A. (2017). A new unsupervised spectral feature selection method for mixed data: a filter approach. *Pattern Recognition*, *72*, 314-326.
15. Lyu, H., Wan, M., Han, J., Liu, R., & Wang, C. (2017). A filter feature selection method based on the Maximal Information Coefficient and Gram-Schmidt Orthogonalization for biomedical data mining. *Computers in biology and medicine*, *89*, 264-274.
16. Adlakha, A., & Chhikara, R. R. (2016, April). Comparative analysis of filter feature selection techniques with different classifiers for image steganalysis. In *2016 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 1122-1127). IEEE.
17. Dubey, V. K., & Saxena, A. K. (2016, October). Cosine similarity based filter technique for feature selection. In *2016 International Conference on Control, Computing, Communication and Materials (ICCCCM)* (pp. 1-6). IEEE.
18. Sun, S., Ye, Z., Yan, L., Su, J., & Wang, R. (2018, September). Wrapper Feature Selection Based on Lightning Attachment Procedure Optimization and Support Vector Machine for Intrusion Detection. In *2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)* (pp. 41-46). IEEE.
19. Kumari, M., & Chhikara, R. (2017, May). Blind image steganalysis using neural networks and wrapper feature selection. In *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 1065-1069). IEEE.
20. Yu, Y., Wang, B., Wang, Z., Wang, F., & Liu, L. (2017, October). Wrapper feature selection based multiple logistic regression model for determinants analysis of residential electricity consumption. In *2017 Asian Conference on Energy, Power and Transportation Electrification (ACEPT)* (pp. 1-8). IEEE.
21. Xue, X., Yao, M., & Wu, Z. (2018). A novel ensemble-based wrapper method for feature selection using extreme learning machine and genetic algorithm. *Knowledge and Information Systems*, *57*(2), 389-412.

22. Zheng, H., Park, H. W., Li, D., Park, K. H., & Ryu, K. H. (2018, November). A Hybrid Feature Selection Approach for Applying to Patients with Diabetes Mellitus: KNHANES 2013-2015. In *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)* (pp. 110-113). IEEE.
23. Alenizi, M. A., & Mansour, H. Y. A. (2018, October). A New Intelligent Hybrid Feature Selection Method. In *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)* (pp. 1-6). IEEE.
24. Li, W., Hou, Y., Wang, H., & Wei, J. (2018, August). ELM combined with hybrid feature selection for classification. In *2018 International Conference on Advanced Mechatronic Systems (ICAMechS)* (pp. 201-207). IEEE.
25. Raymer, M. L., Punch, W. F., Goodman, E. D., Kuhn, L. A., & Jain, A. K. (2000). Dimensionality reduction using genetic algorithms. *IEEE transactions on evolutionary computation*, *4*(2), 164-171.
26. Yang, C. S., Chuang, L. Y., Chen, Y. J., & Yang, C. H. (2008, November). Feature selection using memetic algorithms. In *2008 Third International Conference on Convergence and Hybrid Information Technology* (Vol. 1, pp. 416-423). IEEE.
27. Yu, B., & Yuan, B. (1993). A more efficient branch and bound algorithm for feature selection. *Pattern Recognition*, *26*(6), 883-889.
28. Deraeve, J., & Alexander, W. H. (2018). Fast, Accurate, and Stable Feature Selection Using Neural Networks. *Neuroinformatics*, *16*(2), 253-268.
29. Zhang, H., & Sun, G. (2002). Feature selection using tabu search method. *Pattern recognition*, *35*(3), 701-711.
30. Diao, R., & Shen, Q. (2012). Feature selection with harmony search. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*(6), 1509-1523.
31. Dara, S., Reddy, M. J., & Eluri, N. R. (2018, March). Evolutionary Computation based Feature Selection: A Survey. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 1541-1547). IEEE.
32. Duman, E., Uysal, M., & Alkaya, A. F. (2012). Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment

problem. *Information Sciences*, *217*, 65-77.

33. Almonacid, B., Soto, R., & Crawford, B. (2017, July). Comparing three simple ways of generating neighboring solutions when solving the cell formation problem using two versions of migrating birds optimization. In *2017 17th International Conference on Computational Science and Its Applications (ICCSA)* (pp. 1-9). IEEE.
34. Zhang, Z., Tang, Q., Han, D., & Li, Z. (2018). Enhanced migrating birds optimization algorithm for U-shaped assembly line balancing problems with workers assignment. *Neural Computing and Applications*, 1-15.
35. Duman, E., & Elikucuk, I. (2013, June). Solving credit card fraud detection problem by the new metaheuristics migrating birds optimization. In *International Work-Conference on Artificial Neural Networks* (pp. 62-71). Springer, Berlin, Heidelberg.
36. Niroomand, S., Hadi-Vencheh, A., Şahin, R., & Vizvári, B. (2015). Modified migrating birds optimization algorithm for closed loop layout with exact distances in flexible manufacturing systems. *Expert Systems with Applications*, *42*(19), 6586-6597.
37. Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization (PSO). In *Proc. IEEE International Conference on Neural Networks, Perth, Australia* (pp. 1942-1948).
38. Wang, X., Yang, J., Teng, X., Xia, W., & Jensen, R. (2007). Feature selection based on rough sets and particle swarm optimization. *Pattern recognition letters*, *28*(4), 459-471.
39. Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X., & Wang, S. (2011). An improved particle swarm optimization for feature selection. *Journal of Bionic Engineering*, *8*(2), 191-200.
40. Djellali, H., Guessoum, S., Ghoualmi-Zine, N., & Layachi, S. (2017, October). Fast correlation based filter combined with genetic algorithm and particle swarm on feature selection. In *2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)* (pp. 1-6). IEEE.
41. Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671-680.

42. Diao, R., & Shen, Q. (2015). Nature inspired feature selection meta-heuristics. *Artificial Intelligence Review*, *44*(3), 311-340.
43. Keerthana, R., Haripriya, C., Pabitha, P., & Moorthy, R. S. (2018, June). An Efficient Cancer Prediction Mechanism Using SA and SVM. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1140-1145). IEEE.
44. Wan, C. (2010, August). Efficiency comparison of simulated annealing algorithm and genetic algorithm in feature selection. In *2010 Sixth International Conference on Natural Computation*(Vol. 5, pp. 2330-2333). IEEE.
45. Xue, B., Zhang, M., Browne, W. N., & Yao, X. (2015). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, *20*(4), 606-626.
46. Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4), 341-359.
47. Hancer, E., Xue, B., & Zhang, M. (2017, November). A differential evolution based feature selection approach using an improved filter criterion. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1-8). IEEE.
48. Zorarpacı, E., & Özel, S. A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Systems with Applications*, *62*, 91-103.
49. Vivekanandan, T., & Iyengar, N. C. S. N. (2017). Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease. *Computers in biology and medicine*, *90*, 125-136.
50. Bhadra, T., & Bandyopadhyay, S. (2015). Unsupervised feature selection using an improved version of differential evolution. *Expert Systems with Applications*, *42*(8), 4042-4053.
51. “Java Homepage” [Online] Available: https://www.java.com/tr/
52. “Eclipse Homepage” [Online] Available: https://www.eclipse.org/
53. “UCI machine learning repository” [Online]. Available: http://archive.ics.uci.edu/ml

54. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11: 10–18.
55. Xue, B., Cervante, L., Shang, L., Browne, W. N., & Zhang, M. (2013). Multi-objective evolutionary algorithms for filter based feature selection in classification. *International Journal on Artificial Intelligence Tools*, *22*(04), 1350024.
56. Fix, E., & Hodges Jr, J. L. (1951). *Discriminatory analysis-nonparametric discrimination: consistency properties*. California Univ Berkeley.
57. Deekshatulu, B. L., & Chandra, P. (2013). Classification of heart disease using k-nearest neighbor and genetic algorithm. *Procedia Technology*, *10*, 85-94.
58. Imandoust, S. B., & Bolandraftar, M. (2013). Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications*, *3*(5), 605-610.

RESUME

Gülşah KALAYCI

Marmara University, Goztepe Campus
Engineering Faculty, Computer Science and Engineering Dept.
PK:34722, Kadikoy, Istanbul, Turkiye

e-mail: gulsahtarakci@gmail.com, gulsahtarakci@marun.edu.tr

EDUCATION

B.S., Computer Science and Engineering Marmara University, Faculty of Engineering

Istanbul, Turkey, (2011-2015) GPA:3.32

WORK EXPERIENCE

Software Engineer, Datawarehouse, T.A.O. Vakiflar Bankasi

Istanbul, Turkey, (2015,now)

FOREIGN LANGUAGE

English

RESEARCH INTERESTS

Feature Selection, Vehicular Network