applied sciences

MDPI

Article

Co-Operative Binary Bat Optimizer with Rough Set Reducts for Text Feature Selection

Aisha Adel, Nazlia Omar *, Salwani Abdullah and Adel Al-Shabi

Centre for Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia
* Correspondence: nazlia@ukm.edu.my

Abstract: The process of eliminating irrelevant, redundant and noisy features while trying to maintain less information loss is known as a feature selection problem. Given the vast amount of the textual data generated and shared on the internet such as news reports, articles, tweets and product reviews, the need for an effective text-feature selection method becomes increasingly important. Recently, stochastic optimization algorithms have been adopted to tackle this problem. However, the efficiency of these methods is decreased when tackling high-dimensional problems. This decrease could be attributed to premature convergence where the population diversity is not well maintained. As an innovative attempt, a cooperative Binary Bat Algorithm (BBA_{CO}) is proposed in this work to select the optimal text feature subset for classification purposes. The proposed BBA_{CO} uses a new mechanism to control the population's diversity during the optimization process and to improve the performance of BBA-based text-feature selection method. This is achieved by dividing the dimension of the problem into several parts and optimizing each of them in a separate sub-population. To evaluate the generality and capability of the proposed method, three classifiers and two standard benchmark datasets in English, two in Malay and one in Arabic were used. The results show that the proposed method steadily improves the classification performance in comparison with other well-known feature selection methods. The improvement is obtained for all of the English, Malay and Arabic datasets which indicates the generality of the proposed method in terms of the dataset language.

Keywords: multi-population; binary bat algorithm; cooperative; text feature selection; population diversity

Citation: Adel, A.; Omar, N.; Abdullah, S.; Al-Shabi, A. Co-Operative Binary Bat Optimizer with Rough Set Reducts for Text Feature Selection. *Appl. Sci.* **2022**, *12*, 11296. https://doi.org/10.3390/app122111296

Academic Editor: Giancarlo Mauri

Received: 29 September 2022
Accepted: 27 October 2022
Published: 7 November 2022

1. Introduction

Text classification is the process of automatic grouping of documents into some predefined categories. The idea of text classification is to assign one document to one class (i.e., category), based on its contents. It can provide conceptual views of document collection and has essential applications in the real world. For example, news stories are typically organized by subject categories (topics) or geographical codes; academic papers are often classified by technical domains and sub-domains; even patient reports in health-care organizations are often indexed from multiple aspects, using taxonomies of disease categories, types of surgical procedures, insurance reimbursement codes and so on.

Text Feature Selection (TFS) is an important part of text classification, and much research has been completed on various feature selection methods. A document usually contains hundreds or thousands of distinct words regarded as features. However, many of them may be noisy, less informative or redundant with respect to the class label. This may mislead the classifiers and degrade their performance in general [1,2]. Feature selection (FS) can be thought of as selecting the best words of a document that can help classify that document. Feature selection has been an active research area in pattern recognition, machine learning, statistics and data mining communities. The main idea of feature selection is to choose a subset of the original features by eliminating redundant ones and

those with little or no predictive information. Feature selection is an essential process, as it can make or break a classification engine [3]. Feature selection is considered an optimization problem [4–6] where the aim is to select the most representative features that give the highest prediction performance. The idea of TFS, in simple words, is to determine the importance of words using a defined measure that can keep informative words, and remove non-informative words, which can then help the text classification engine.

During the past few decades, many feature selection methods have been proposed. On one hand, some of those methods work by ranking features and filtering out the low-ranked ones. Although those methods are fast and independent from any classification algorithm, they ignore the dependencies between features which affect the quality of the selected feature set [7]. On the other hand, population-based meta-heuristic methods such as genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO) and Bat algorithm (BA) have attracted a lot of attention [5,7–18]. These methods try to gather better solutions by using knowledge from previous steps. Therefore, the focus on search strategies has shifted to meta-heuristic algorithms, which are well suited for searching among a large number of possibilities for solutions. Most of these methods utilizes a classification method to evaluate the feature set, resulting in higher classification accuracy. However, the main drawback of these methods is that they are dependent on the utilized classification algorithm, and this makes the resulted feature sets biased to the choice of classifier [19].

The Bat algorithm (BA) is a meta-heuristic method proposed by [20] and based on the fascinating capability of micro-bats to find their prey and discriminate different types of insects even in complete darkness. The algorithm is formulated to imitate the ability of bats in finding their prey. The main advantage of the BA is that it combines the benefits of population-based and single-based algorithms to improve the quality of convergence [21]. BA and its variants have been successfully applied to solve many problems such as optimization, classification, feature selection, image processing and scheduling [8,21–26]. For more details about the Bat algorithm and the binary version of it, the reader may refer to [20,27], respectively.

As mentioned above, BA was successfully applied to many application domains including FS. However, one of the limitations with many meta-heuristic algorithms, including BA, is their deficient performance with high-dimensional problems. This problem is most likely to appear as the search space is not effectively explored due to losing population diversity during the search process [28]. Many methods were proposed in the literature to control population diversity including cooperative algorithms [29,30]. However, for high-dimensional problems, co-evolutionary algorithms are preferred as they can divide the dimension of the solution into multiple parts, and optimize each part separately [31,32]. Moreover, as the text data are represented as a sequence of terms where each term is considered as one feature, this aggravates the problem of high dimensionality. The coevolutionary strategy was successfully employed in several evolutionary computations, such as job-shop scheduling [33], path-planning problem [34], supply chain-gap analysis [35], flow-shop scheduling problem [36], large-scale optimization [37], hierarchized Steiner tree problems [38] and sensor ontology meta-matching [39]. However, the majority of these applications are continuous problems. Applying this technique to a discrete problem such as text feature selection is still challenging and needs to be further studied.

In this paper, a cooperative coevolutionary BBA is proposed and evaluated as a TFS method, that provides the following contributions:

i. Controlling the population diversity during the search process using the multi-population BBA;
ii. Handling the high dimensionality of the feature space by using the divide and conquer strategy;
iii. Initializing a diverse population using the modified Latin Hypercube Sampling (LHS) initialization method;

iv. Better evaluation of the solutions using the adapted Rough Set (RS)-based fitness function that is independent of any classification method.

The rest of this paper is organized as follows: Section 2 provides a summary of the related work. Then, the details of the proposed algorithm are given in Section 3, followed by the experimental setup in Section 4. After that, the discussion and analysis of the experimental results are shown in Section 5. Finally, the work is concluded in Section 6 of this paper.

2. Related Work

The simplest definition of a coevolutionary algorithm is that it is an evolutionary algorithm (or a collection of evolutionary algorithms) in which the fitness of an individual depends on the relationship between that individual and other individuals [40]. Such a definition immediately imbues these algorithms with a variety of views differing from those of more traditional evolutionary algorithms. Therefore, the interaction between individuals of different populations is the key to the success of coevolutionary techniques.

In the literature, coevolution is often divided into two classes: cooperative and competitive, regarding the type of interaction employed. In cooperative coevolution, each population evolves individuals representing a component of the final solution. Thus, a full candidate solution is obtained by joining an individual chosen from each population. In this way, increases in a collaborative fitness value are shared among individuals of all the populations of the algorithm. In competitive coevolution, the individuals of each population compete with each other. This competition is usually represented by a decrease in the fitness value of an individual when the fitness value of its antagonist increases [41].

Additionally, coevolution is a research field that has recently started to grow. Some research efforts have been applied to tackle the question about how to select the members of each population that will be used to evaluate the fitness function. One way is to evaluate an individual against every single collaborator in the other population. Although it could be a better way to select the collaborators, it would consume a very high number of evaluations in the computation of the fitness function. To reduce this number, there are other options, such as the use of just a random individual or the use of the best individual from the previous generation [42].

The coevolutionary technique was successfully utilized in literature with different domains. In an early work, the authors of [43] presented the cooperative particle swarm optimizer and applied their method to several benchmark optimization problems. The authors of [44] proposed an approach based on coevolutionary particle swarm optimization to solve constrained optimization problems formulated as min–max problems. Another study [45] proposed a cooperative coevolution framework in order to optimize large scale non-separable problems. The authors of [46] adapted a competitive and cooperative coevolutionary approach for a multi-objective particle swarm optimization algorithm design, which appeared to solve complex optimization problems by explicitly modelling the coevolution of competing and cooperating species. In another work, the authors of [47] proposed a cooperative coevolving particle swarm optimization algorithm in an attempt to address the issue of scaling-up particle swarm optimization algorithms in solving large-scale optimization problems.

Later, the authors of [48] proposed a direction vector-based coevolutionary multi-objective optimization algorithm, that introduced the decomposition idea from multi-objective evolutionary algorithms to coevolutionary algorithms. The authors of [49] proposed an adaptive coevolutionary algorithm based on genotypic diversity measure. In another study [50] a coevolutionary improved multi-ant colony optimization algorithm was proposed for ship multi and branch-pipe route design. The author of [51] proposed a cooperative coevolutionary artificial bee colony algorithm that has two sub-swarms, with each addressing a sub-problem. The sub-problems were a charge scheduling problem in a hybrid flow-shop, and a cast scheduling problem in parallel machines.

Later, the authors of [52] proposed a multi-objective cooperative coevolutionary algorithm to optimize the reconstruction term, the sparsity term and the total variation regularization term, simultaneously, for Hyperspectral Sparse Unmixing. In [53] the authors proposed a parallel multi-objective cooperative coevolutionary variant of the Speed-constrained Multi-objective Particle Swarm Optimization algorithm. In [54], the authors proposed a two-layer distributed cooperative coevolution architecture with adaptive computing resource allocation for large-scale optimization. In another study, [55], the authors proposed an approach utilizing a Cooperative Co-evolutionary Differential Evolution algorithm to optimize high-dimensional ANNs.

In a recent study [56], the authors proposed a hybrid cooperative coevolution algorithm for the minimization of fuzzy makespan. In [57], the authors developed a cooperative coevolution algorithm for *seru* production with minimizing makespan by solving the *seru* formation and *seru* scheduling simultaneously. In another study [58] the authors proposed a multi-population coevolution-based multi-objective particle swarm optimization algorithm to realize the rapid search for the globally optimal solution to solve the problem of Weapon–Target Assignment. In addition to these studies, the authors of [59] proposed a cooperative coevolution hyper-heuristic framework to solve workflow scheduling problem with an objective of minimizing the completed time of workflow.

For feature selection problems, a few studies in the literature have utilized the cooperative coevolutionary algorithm. In two early works, refs. [60,61], the authors performed instance and feature selection by creating three populations in different sizes. The first population performed feature selection, while the second population performed instance selection and the third population was for both feature and instance selection. The authors of [62] presented a hybrid learning algorithm based on a cooperative coevolutionary algorithm (Co-CEA) with dual populations for designing the radial basis-function neural network (RBFNN) models with an explicit feature selection. In this algorithm, the first sub-population used binary encoding masks for feature selection, and the second sub-population tended to yield the optimal RBFNN structure.

Another study presented a cooperative coevolution framework to render the feature selection process embedded into the classification model construction within the genetic-based machine learning paradigm [42]. Their approach had two coevolving populations cooperate with each other regarding the fitness evaluation. The first population corresponded to the selected feature subsets and the second population was for rule sets of classifier. Later, the authors of [63] proposed an attribute equilibrium dominance reduction accelerator (DCCAEDR) based on the distributed coevolutionary cloud model. The framework of N-populations distributed coevolutionary MapReduce model is designed to divide the entire population into N sub-populations, sharing the rewards of different sub-populations' solutions under a MapReduce cloud mechanism. After that, a CCFS algorithm was proposed that divided vertically (on features) the dataset by random manner and utilized the fundamental concepts of cooperation coevolution in order to search the solution space via Binary Gravitational Search Algorithm (BGSA) [28]. Another study utilized a genetic algorithm for the coevolution of meteorological data for attribute reduction [64]. In this work, the evolutionary population was divided into two sub-populations; one for elite individuals to assist crossover operations to increase the convergence speed of the algorithm, and the other for balancing the population diversity in the evolutionary process by introducing a random population.

It is noticed that in most of the mentioned studies [42,60–62,64], the authors have attempted to solve the feature selection problem as a multi-objective problem by creating two or more populations, where each of them optimizes one objective. However, those methods are not applicable for single objective problems and they do not solve the high dimensionality of the feature space. In the work of Ding et al. [63], the focus was to distribute the optimization process on multiple machines in order to reduce the computational time. However, the requirement of their model, such as the hardware (e.g., multiple PC machines), the mechanism of distribution of the dataset, the means of communication

between different machines, and the way of forming the complete solution, was not always available. In the work of the authors of [28] the dimension of the full solution was divided into smaller subsets where each of them is optimized in a separate population. Although their method was effective with high dimensional feature selection problem, there were multiple aspects that needed further improvement. For example, the method might have a better parameter tuning in order to improve its performance. In addition, the solutions in the different sub-populations need to be combined with each other in each generation in order to be evaluated, which is computationally expensive and reduces the chance of each solution to be optimized separately from the other sub-populations.

3. The Proposed Algorithm

In this work, the cooperative and coevolution mechanisms are utilized with the binary bat algorithm. A combination of these two approaches (which we call BBA_{CO}) is proposed for the text feature selection problem. Figure 1 shows the main stages of the proposed method that are explained in the following subsections.

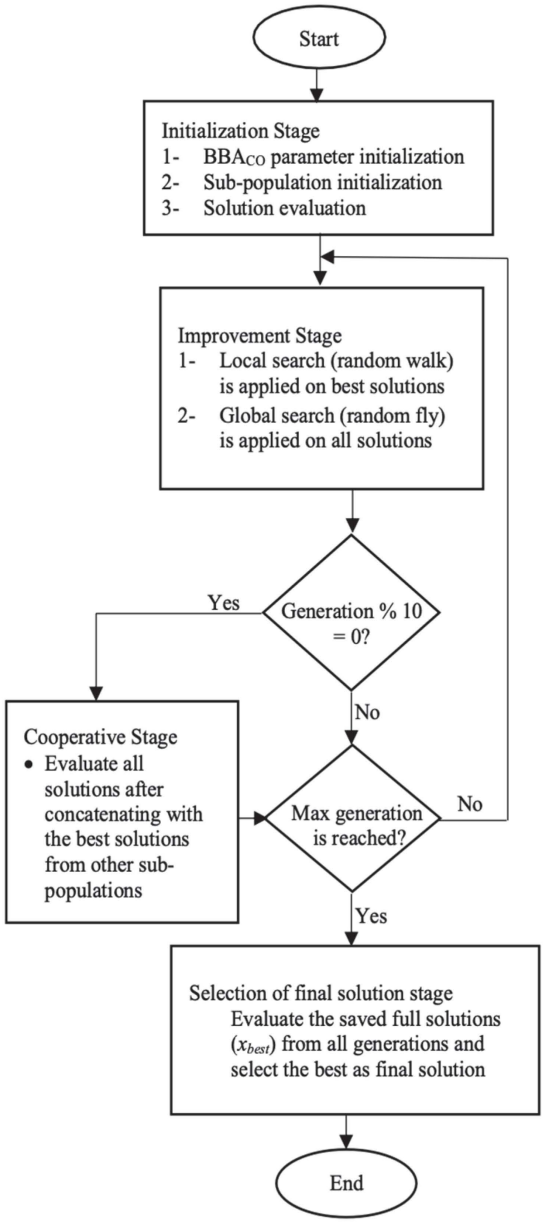

Figure 1. Flowchart of BBA_{CO}.

3.1. Initialization Stage

This stage contains three steps, i.e., (i) BBA_{CO} parameter initialization that is based on the parameter tuning of the Taguchi method, (ii) sub-population initialization using a modified LHS method and (iii) solutions' evaluation to choose the best candidate solution in each sub-population based on the dependency measurement using rough set theory (RST) [65]. Algorithm 1 shows the pseudocode for the initialization stage.

Algorithm 1. Pseudocode of the initialization stage.

Initialization

```
//Step 1: BBA_CO parameter initialization
Initialize SubPop-no, SubPop-size, Evaluate-fullSol-rate
//Step 2: Sub-population initialization
Size-PartialSolution = F/SubPop-no, where F total represents number of features
Remainder = F%SubPop-no
For each Sub-Population (i), where i represent the index from 0 to SubPop-no
    If i <= Reminder
        Size-PartialSolution(i) = Size-PartialSolution + 1
    Else
        Size-PartialSolution(i) = Size-PartialSolution
     For 1 to SubPop-size
        Generate initial solution using modified LHS method
        Initialize loudness (A), pulse rate (r), minimum frequency (F_min), velocity (v),
                maximum frequency (F_max);
//Step 3: Solution evaluation
Evaluate each solution in each sub-population using rough-set based objective function
Assign the best solution into x_best^i
Combine all x_best^i into x_best
Save x_best in memory
```

In Step 1, *SubPop-no* refers to the number of sub-populations, *SubPop-size* refers to the number of candidate solutions in each sub-population, and *Evaluate-fullSol-rate* refers to the number of generations reproduced before evaluating the full dimension solutions (referred to as *FullSolution*), are initialized. In this work, a Taguchi method [66] was used to identify the best values of the parameters for the BBA_{CO} algorithm. Three levels were considered for each factor as shown in Table 1. The BBA_{CO} algorithm runs three times for each factor at each level, and the average Signal-to-Noise (SN) ratio plot for each level of the factors is shown in Figure 2. The level with the maximum SN ratio is the optimum parameter determined by the Taguchi method. According to Figure 2, the optimum value for *SubPop-size* is set to 100, the *Evaluate-fullSol-rate* is set to 10, and the *SubPop-no* is set to 15 as shown in Table 1.

Table 1. The BBA_{CO} algorithm parameter levels for the Taguchi method.

Parameter	Definition	Level		
		1	**2**	**3**
SubPop-size	The sub-population size	50	100	150
SubPop-no	The number of sub-populations	10	15	20
Evaluate-fulSol-rate	The number of reproduced generations before evaluating the full dimension solutions (referred to as *FullSolution*)	10	20	30

Figure 2. Graphical results of the Taguchi method for BBA_{CO} algorithm.

In Step 2, the candidate solutions of each sub-population are initialized using a modified LHS method. Note that the candidate solution in each sub-population that contains subset of features is referred to as *PartialSolution*, while *FullSolution* refers to a candidate solution with a full dimension where its length is equal to the total number of features. The size of *PartialSolution* (*Size-PartialSolution*) is determined based on the number of features and the number of sub-populations as in the following equations:

$$Size\text{-}PartialSolution = F/SubPop\text{-}no$$

$$Remainder = F \,\%\, SubPop\text{-}no$$

where F is the total number of features, and *Remainder* is the number of sub-populations that will be assigned extra one feature. For example, if F is 20, and *SubPop-no* is 3, thus the *Size-PartialSolution* and *Remainder* are:

$$Size\text{-}PartialSolution = 20/3 = 6$$

$$Remainder = 20\%3 = 2$$

Based on the value of the *Remainder*, the size of two out of three sub-populations will contain one extra feature. Thus, seven features are yielded for sub-population #1 to #2 (i.e., *Size-PartialSolution* +1 = 6 + 1 = 7). The remaining one sub-population remains with six features. Figure 4 shows how the full dimension of features, *FullSolution* (i.e., 20 features) is divided and assigned into three sub-populations where the first two sub-populations consist seven features, and the third sub-population consists of six features. Note that the letters in Figure 3 represent the features where the arrows represent the movement of features to the sub-populations and # is number (#2 equals to *number 2*).

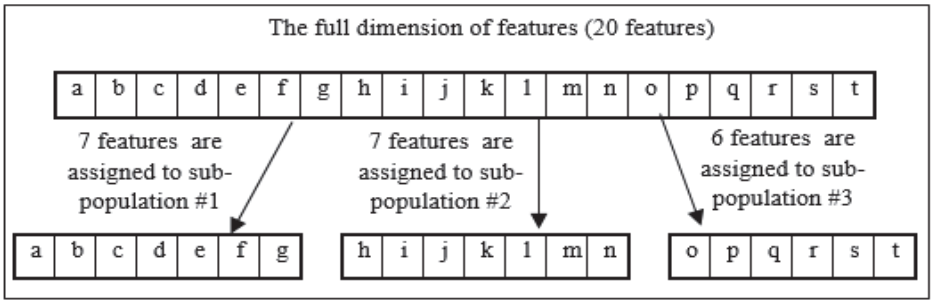

Figure 3. Subsets of features into sub-populations assignment.

In Step 3, each candidate solution in all sub-populations is evaluated based on the adapted dependency degree measure using rough set theory. For example, in Figure 4 the selected features are *a* and *d*. Thus, the quality of the *PartialSolution* is calculated based on the dependency degree between feature *a* and *d* using the adapted rough set theory (RST). In the adapted RST, the features are represented by their presence or absence in the document. In this way, the candidate solutions could be compared with the instances (i.e., documents) to define the lower and upper approximations. However, it was found that there is no instance that can have the same pattern of the solution in order to be added to a lower approximation due to the high dimensionality of the feature space. To handle this limitation, the similarity between each instance and the candidate solution is calculated using cosine similarity measure as in the following equation. The similarity threshold ($\delta = 0.70$) is defined based on the preliminary experiments:

$$cos(d_i, cs) = \frac{\sum_{k=1}^{n} d_i^k \cdot cs^k}{\sqrt{\sum_{k=1}^{n} \left(d_i^k\right)^2} \cdot \sqrt{\sum_{k=1}^{n} \left(cs^k\right)^2}}$$

where d_i is the document number i; cs is the candidate solution, $d_i^k.cs^k$ is the dot product between d_i and cs; k is the index of the term and n is the number of the selected terms in the candidate solution. After calculating cosine similarity, if its value is equal or greater than δ the document is added to the lower approximation, otherwise, it is not added.

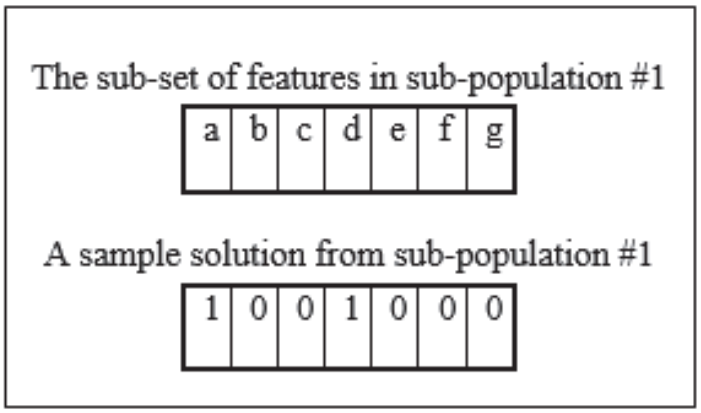

Figure 4. A sample *PartialSolution* from sub-population #1.

The best *PartialSolution* from each sub-population is assigned to x_{best}^i, and is concatenated as follows:

$$x_{best} = concatenate(x_{best}^1, x_{best}^2, x_{best}^3, \ldots, x_{best}^{subPop-no})$$

Modified LHS: Initial Population Generation

Latin Hypercube Sampling (LHS) is a statistical method developed by the author of [67] and used for sampling by ensuring that all portions of the continuous variable were sampled. LHS was used as an initialization method by the authors of [68], where the solutions were represented using real values as the problem was continuous. For initializing the candidate solutions of BBACO, the LHS method was modified to be applicable to the problems with binary representation. The modified LHS works as follows:

1. Divide the length of the solution into equal segments, where the length of the solution is equal to the number of features. The following equation is used to determine the number of segments in each solution:

$$sn = \frac{F}{n} random(1, m)$$

$$m = \begin{cases} \frac{F}{n} \; if \frac{F}{n} \leq \frac{n}{2} \\ \frac{n}{2} \; if \frac{F}{n} > \frac{n}{2} \end{cases}$$

where *sn* refers to the number of segments; *F* is the number of features and *n* is the number of solutions in the population. $\frac{F}{n}$ provides the number of segments that guarantees using each feature only one time in one solution. The parameter *m* is the upper band of the random number, and it ensures that the number of the selected features does not exceed half of the features. It should be noted that *m* is defined one time at the beginning of the initialization process. The reason behind using two different ways to calculate *m* (depending on the size of features and population) is to make the method more suitable with datasets of a different size;

2. Calculate the length of the segments (*sl*) for each solution as follows:

$$sl = \frac{F}{sn}$$

Then, one feature is selected randomly from each segment. The steps of the modified LHS are shown in Algorithm 2.

Algorithm 2. The modified LHS initialization method steps.

Modified LHS initialization method

1. *Calculate m, where m is the maximum number of the features that can be selected*
   For 1 to number of solutions
2. *Calculate the number of segments (sn)*
3. *Calculate the length of segment (sl)*
4. *Randomly choose one feature in each segment.*
5. *Check if this is the final solution:*
   If yes: go to solutions' evaluation.
   Otherwise: go to 2

3.2. Improvement Stage

This stage contains two steps, i.e., (i) local search (random walk) and (ii) global search (random fly). The local search is applied on each x^i_{best} (best *PartialSolution* of each sub-population i), while the global search is applied on all *PartialSolutions* in all sub-populations with the aim of reproduction. In step i, a local *PartialSolution* is generated based on the best *PartialSolution* in each sub-population if the condition of local search (i.e., $r_i > rand[0,1]$, where r_i is the pulse rate of the best *PartialSolution* in sub-population #*i*) is met. The pseudocode in Algorithm 3 shows the steps of local search.

Algorithm 3. Pseudocode of the local search.

Local search (random walk)

For each x^i_{best}
  If (ri > rand [0, 1])
    $x^i_{new} = x^i_{old} + \epsilon \overline{A_g}$
    $S(x^i_{new}) = \frac{1}{1+e^{-x_{new}}}$
    $x^i_{new} = \begin{cases} 1 & if\ S(x^i_{new}) > \sigma[0,1] \\ 0 & otherwise \end{cases}$

In Algorithm 3, r_i is the pulse rate of the best *PartialSolution* in sub-population #*i*, *rand*, ϵ, and σ are random numbers between 0 and 1, x^i_{new} is the generated *PartialSolution* near x^i_{best} (i.e., 15 local *PartialSolutions* are generated as the number of sub-populations is 15), $\overline{A_g}$ is the average loudness of all *PartialSolutions* at the generation g, and $S\left(x^i_{new}\right)$ is the sigmoid function that used to restrict the values of x^i_{new} into 0 or 1.

Step 2 generates new *PartialSolutions* for the next generation. First, the frequency of each *PartialSolution* in each sub-population is updated. Then, the *PartialSolution's* velocity is updated based on the new value of frequency; the best *PartialSolution* in the corresponding

sub-population, and the position (i.e., the *PartialSolution* itself). After that, the original velocity and the current position are used to generate a new position (i.e., *PartialSolution*). These operations are shown in the pseudocode of global search in Algorithm 4.

Algorithm 4. Pseudocode of the global search.

Global search (random fly)

For each PartialSolution(pSol) in each sub-population (i)

$f_i^{sol} = f_{min} + (f_{max} - f_{min})\beta[0,1]$
$v_i^g(pSol) = v_i^{g-1}(pSol) + \left(x_i^{g-1}(pSol) - x_{best}^i\right) f_i^{pSol}$
$x_i^g(pSol) = x_i^{g-1}(pSol) + v_i^g(pSol)$
$S\left(x_i^g(pSol)\right) = \frac{1}{1+e^{-x_i^g(pSol)}}$
$x_i^g(pSol) = \begin{cases} 1 & if\ S\left(x_i^g(pSol)\right) > \sigma \\ 0 & otherwise \end{cases}$
$if\left(A_i^{pSol} \left\langle random[0,1] \& fit\left(x_i^g(pSol)\right)\right\rangle fit\left(x_i^{g-1}(spSolol)\right)\right.$
AcceptthenewPartialSolution
$r_i^{g+1} = r_i^0[1 - exp(-\gamma g)]$
$A_i^{g+1} = \alpha A_i^g \quad where\ \alpha = \gamma = 0.9$

As shown in Algorithm 4, the frequency of each *PartialSolution* is updated as in the second line in the pseudocode where f_i^{pSol} is the new frequency of the *PartialSolution* in sub-population i, f_{min} is the minimum frequency, f_{max} is the maximum frequency and β is a random number between 0 and 1. Then, the velocity is updated as in the third line in the pseudocode, where $v_i^g(pSol)$ is the velocity of the *PartialSolution* in generation g and sub-population i, $v_i^{g-1}(pSol)$ is the velocity of the *PartialSolution* in the previous generation and x_{best}^i is the best *PartialSolution* in sub-population *i.* After that, a new position (i.e., *PartialSolution*) is generated as in the fourth and fifth lines in the pseudocode, based on the *PartialSolution* in the previous generation and the velocity. It could be noted that for the position, the sigmoid function is used to restrict the new values into 0 or 1. The last part of this step is to update the *PartialSolutions* in the sub-population. If the condition of accepting *PartialSolution* is met (i.e., loudness (A_i) is less than random number from 0 to 1, and the new *PartialSolution* is better than the previous one based on the adapted dependency measure of rough set theory, which was utilized as fitness function), then, the *PartialSolution* is accepted and the pulse rate and loudness are updated.

3.3. Cooperative Stage: FullSolutions Evaluation

This stage takes place after every 10 generations (i.e., *Evaluate-fullSol-rate* = 10, as explained in Section 3.1) of the algorithm. The parameter *Evaluate-fullSol-rate* determines which generations of the *PartialSolutions* in different sub-populations will be concatenated and evaluated as *FullSolutions*. The purpose of this stage is to cooperate between all sub-populations. The cooperation is achieved by concatenating the *PartialSolution* in hand (e.g., a *PartialSolution* in sub-population #1) with the best *PartialSolutions* of other sub-populations (e.g., sub-population #2 to #15) in generation g, so that the whole sub-populations cooperate with each other to evaluate the *PartialSolutions*. If the condition of evaluating *FullSolution* is met (i.e., generation % *Evaluate-fullSol-rate* = 0), then the *PartialSolutions* are evaluated by following the pseudocode in Algorithm 5, as shown in Figure 5.

As an example for the cooperative stage, suppose that there are three sub-populations, each with three *PartialSolutions*. Each *PartialSolution* in sub-population #1 focuses on the best *PartialSolutions* in sub-populations #2 and #3. Then, the *FullSolution* with the full dimension (i.e., length = 20) is evaluated using RST-based fitness function. The quality of the *PartialSolution* is updated based on the new evaluation. The same steps explained in this subsection, are repeated for the *PartialSolutions* of the other sub-populations. The last

step in this stage is to update the best *PartialSolution* of each sub-population depending on the new fitness values of the *PartialSolutions*.

Algorithm 5. Pseudocode of the cooperative stage.

```
Cooperative stage
    if (generation % 10 == 0)
        for each sub-population (i)
            for each PartialSolution
                concatenate with the best solutions of other sub-populations
                evaluate FullSolution
                update the fitness of the PartialSolution
                    update the best PartialSolution x_best^i
```

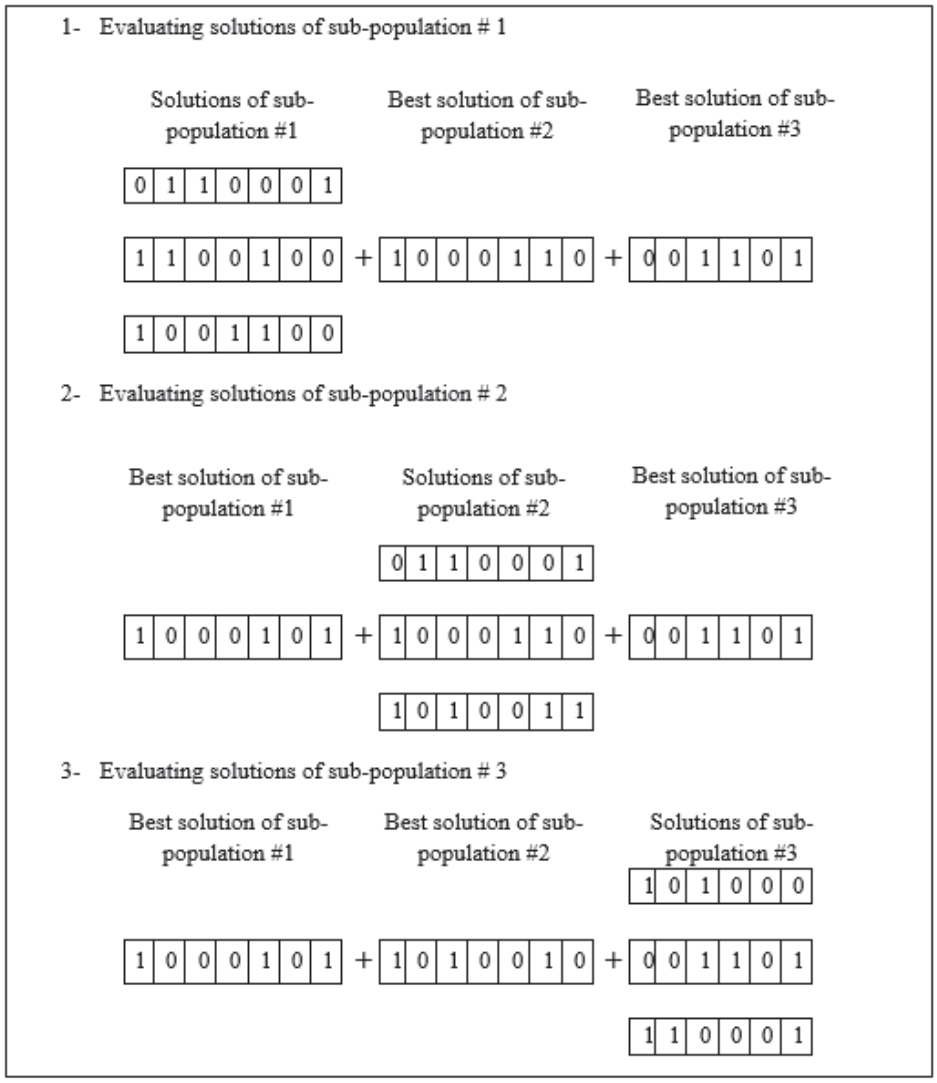

Figure 5. *PartialSolutions* evaluation of all sub-population as *FullSolutions*.

3.4. Selection of FinalSolution

This stage takes place when the stopping criteria (i.e., *Iter-no* >100), is met. In this stage, the saved *FullSolutions* after each generation are evaluated and the best one is selected as a final solution. As mentioned in the previous sections, after each generation, the best *PartialSolutions* of all sub-populations are concatenated and saved in the memory. The importance of this stage is due to the nature of the saved *FullSolutions* as they consist of several parts from several sub-populations. Although each part of these *FullSolutions* was

evaluated and selected as the best *PartialSolution* within its sub-population in a certain generation, the *FullSolution* was not evaluated as one set of features. Therefore, to ensure that the final solution is the best one obtained by the algorithm, the saved *FullSolutions* are evaluated and the best *FullSolution* is selected as the final solution. Algorithm 6 shows the steps of selecting the final solution.

Algorithm 6. Steps of selection of the final solution.

Selection of final solution

```
If (iter-no >100)
    For i = 1 to 100
        Evaluate the saved FullSolution #i
  Select the best FullSolution as the final solution
```

4. Experimental Setup

This section presents the experimental setup where the BBA_{CO} as a text feature selection optimizer is tested on two standard corpora of English text datasets, two Malay datasets, and one Arabic corpus. The pre-processing process, classifier and evaluation metrics used are also presented.

4.1. Pre-Processing

The pre-processing tasks are employed before the BBA_{CO} is tested on the text feature selection problem that involve normalization, tokenization, stop words removal and removal of the less frequent words. The normalization process removes non-letters and punctuation marks. Then, the capital letters are converted to small letters. In the tokenization process, the documents are divided into terms. The stop words (words without discriminative meaning) are then removed. Next, the words that rarely appear in the whole corpus are removed as they seem not to be significant to the classification process.

4.2. The Dataset

Three standard datasets including two English corpora namely Reuters-21578 and WebKB, two Malay corpora namely Mix-DS and Harian-Metro and one Arabic corpus namely, Al-Jazeera news are used in order to assess the performance of the proposed approach.

4.2.1. Reuters-21578 Dataset

This dataset contains 21,578 text files, which were collected from the Reuters newswire. These files were non-uniformly divided into 135 classes. This work utilizes the top 10 classes namely *earn*, *acquisition*, *trade*, *ship*, *grain*, *crude*, *interest*, *money-fx*, *corn* and *wheat*, which contain 4808 documents.

4.2.2. WebKB Dataset

This dataset is a collection of web pages from four different college websites, contains 8282 web pages assigned to 7 classes. In this work, only four classes are used as in the literature, which contains 2803 documents. The utilized classes are *student*, *faculty*, *course* and *project*.

4.2.3. Mix-DS

This dataset was manually collected from several websites. The total number of documents in this dataset is 12,269, distributed unevenly among 6 categories. Table 2 shows the number of documents in each class and the websites where the documents were collected.

Table 2. Number of collected documents from each website for the classes of Mix-DS.

	Sukan	Bisnes	Pendidikan	Sains-Teknologi	Hiburan	Politik	Total
utusan online	170	238	300	220	1030	0	1958
mstar	3392	10	0	0	4533	0	7935
astroawani	96	96	0	96	96	96	480
bharian	527	1292	25	0	27	25	1896
Total	4185	1636	325	316	5686	121	12,269

4.2.4. Harian Metro Dataset

The Harian Metro dataset has been collected from the Harian Metro website, and it consists of 7920 documents, distributed evenly (720 documents) among 11 categories namely; Sukan, Bisnes, Pendidikan, Teknologi, Hiburan, Dekotaman, Global, Vroom, Sihat, Sanati and Addin.

4.2.5. Al-Jazeera News Dataset

This dataset consists of 1500 text documents distributed equally among five categories (Economy, Science, Politics, Sport, and Art) and each category has 300 text documents. This dataset was collected from the Al-Jazeera news channel website (www.aljazeera.net) (accessed on 26 March 2022).

In this work, three classification algorithms are used in the experiments, which are Support Vector Machine (SVM), Naïve Bayes (NB) and K-Nearest Neighbor (KNN).

4.3. Evaluation Metric

In this work, the proposed methods are evaluated internally and externally. The internal evaluation concerns itself with the evaluation of the feature selection method, such as the quality and diversity of the population. On the other hand, the external evaluation evaluates the resulting feature set when utilizing it for classification. Although multiple evaluation metrics are utilized for evaluation, the computational cost is not considered as it is one time cost. Thus, the evaluation focuses on the quality of the resulted feature set and the classification results.

4.3.1. Internal Evaluation Metric

For the internal evaluation of the text-feature selection method presented in this work, different evaluation metrics will be used. These metrics are:

Solution Quality

This is also called the fitness value or the objective value of the solution. This value is calculated using the adapted dependency degree measure of RST and the number of features. The following equation is used to evaluate each candidate solution:

$$Fitness(x_i) = p \times dep(x_i) + (1 - p) \times (1/size(x_i))$$

where x_i is the feature subset found by solution i. *Fitness* is calculated based on both the dependency measure of rough set theory ($dep(x_i)$), and the length of the feature subset ($size(x_i)$). p is a parameter that controls the relative weight of dependency value and feature subset length, where $p \in [0, 1]$. This formula denotes that the dependency value and feature subset size have a different effect on the evaluation. In this study, the dependency value is considered to be more important than the subset length, so p is set to 0.8, as in [69,70].

Size of the Selected Feature Set

This metric evaluated the reduction ability of the text-feature selection method. A good method should be able to produce a high-quality feature set with a smaller number of features. The number of features in the resulting feature set is compared with the original number of features before the feature selection process to evaluate the reduction rate.

Reduction Rate

This metric is also used to evaluate the reduction ability of the TFS method in percentage. The reduction rate is calculated based on the number of original features and the number of selected features as:

$$Reduction\ rate\ (\%) = 100 \times \frac{\#original\ features - \#selected\ features}{\#original\ features}$$

Diversity during the Search Process

The population's diversity of different generations in the algorithm (i.e., generations 1, 20, 50, 80 and 100) is measured graphically to evaluate the exploration ability of the algorithm during the search process and its ability to maintain the population's diversity to later generations.

Convergence Behavior

The convergence of the population is shown graphically to evaluate the ability of the method to keep improving the population and avoid premature convergence. The convergence of the whole population is shown in average, and the convergence of five randomly selected solutions is also shown to show their improvement progress during the search process.

Statistical Tests

The significance test (the *t*-test: two-sample assuming unequal variances) is conducted as a statistical analysis to compare the algorithms. The *t*-test is a statistical check of two population means. The *t*-test was successfully used for comparing two groups of results over multiple datasets for its simplicity, safety and robust results [71]. To perform the *t*-test, the *t State* and *t Critical two tail* values are calculated by Microsoft Excel software. The first group of results is considered significantly higher than the second group if the *t State* value is greater than the *t Critical two tail*. The second group of results is considered significantly higher than the first group if the *t State* value is less than -*t Critical two tail*. The difference between the two groups' results is considered not significant if the *t State* value is in the interval [−*t Critical two tail*, *t Critical two tail*].

In this work, the *t*-test is used for internal evaluation to measure the diversity and quality of the population. The population diversity is measured using the standard deviation values of 32 populations generated. The *t*-test is also used to measure the average quality of the population. The population's quality is also measured using Best Relative Error (BRE), Average Relative Error (ARE) and Worst Relative Error (WRE) of the populations, where lesser values represent a better quality of the population.

4.3.2. External Evaluation Metric

The external evaluation concerns the classification performance by employing the selected feature set. The classification performance is measured by multiple evaluation criteria, which are discussed in the following subsections.

Classification Performance

To evaluate the classification performance using the selected feature sets, two widely used performance measures are used namely *Micro Average Fl* and *Macro Average F1*. The *Macro Average F1* measure depends on *precision* (*P*), *recall* (*R*) and *F-measure*, which are calculated for each class as follows:

$$P = a/(a + b)$$

$$P = a/(a + b)$$

$$F_1 = \frac{2p \times r}{p + r}$$

where a is the number of documents correctly classified; b is the number of documents incorrectly classified and c is the number of documents in the class. *Macro Average F1* is calculated as below:

$$F_1^{macro} = \frac{1}{m}\sum_{i=1}^{m} F_1(c_i)$$

where m is the number of classes and $F1(c_i)$ is the *F1 measure* for the i^{th} class.

Micro Average F1 is calculated globally based on the global precision and recall. Calculations for precision and recall for micro averaging as given by [72] are shown below:

$$P^{\mu} = \frac{\sum_{i=1}^{m} a_i}{\sum_{i=1}^{m}(a_i+b_i)}$$
$$R^{\mu} = \frac{\sum_{i=1}^{m} a_i}{\sum_{i=1}^{m}(a_i+c_i)}$$

where a, b, c, m are the same variables used in the previous equations of *precision* (P) and *recall* (R), and μ indicates Micro Averaging. *Micro Average F1* is calculated as follows:

$$F_1^{Micro} = \frac{2 \times p^{\mu} \times R^{\mu}}{p^{\mu} \times R^{\mu}}$$

Statistical Test

The *t*-test is conducted as a statistical analysis to compare two groups of classification results. If the difference in the results is above a certain value, this indicates that the text-feature selection method is significantly efficient.

5. Results and Discussion

In this work, the performance of the proposed approach is measured based on the internal and external evaluation metrics as discussed in Section 4.3.

5.1. Internal Evaluation

The performance of the BBA_{CO} is compared to BBA_{LHS} in terms of population diversity, convergence behavior and the solution quality. Note that BBA_{LHS} is a binary Bat algorithm that is modelled based on one population, contrary to the BBA_{CO} where it is modelled as a multi-population binary Bat algorithm. In addition, please note that the modified LHS is used to generate initial population(s) for both BBA_{LHS} and BBA_{CO}.

5.1.1. Population Diversity

Population diversity is represented in the form of a distribution of solutions during the optimization process (i.e., at the 1st, 20th, 50th, 80th and 100th generations). Figures 6 and 7 show the distribution of the solutions for the Reuters and WebKB datasets, respectively. It can be seen that the diversity of the population is well controlled by the BBA_{CO} in comparison to BBA_{LHS} where poor diversity can be noted after the first quarter of the search process, contrary with the BBA_{CO} where the solutions in the population are fairly distributed across the feature space.

5.1.2. Convergence Behavior

A comparison between the results achieved by BBA_{CO} and BBA_{LHS} is depicted in graphical form to show the convergence behavior as shown in Figures 8 and 9 on the Reuters and WebKB datasets, respectively.

It can be noted that the population of BBA_{CO} converged slower than the population of BBA_{LHS} due to controlling diversity during the search process. However, the average quality of the solutions at the end of the optimization in BBA_{CO} is higher than in BBA_{LHS}. From the results of the convergence, it is clear that in BBA_{LHS}, the population converge faster and stagnate in the first third of the search process, while in BBA_{CO} the convergence is slower and the stagnation occurs by the last third of the search process. As a conclusion,

BBA_{LHS} initialization could be the best choice if the purpose was improving the whole population within a small number of generations, while BBA_{CO} is better if the purpose is controlling the diversity and obtaining a better final solution. Figures 10 and 11 compare the convergence of five randomly selected solutions using BBA_{LHS} and BBA_{CO} in Reuters and WebKB datasets, respectively.

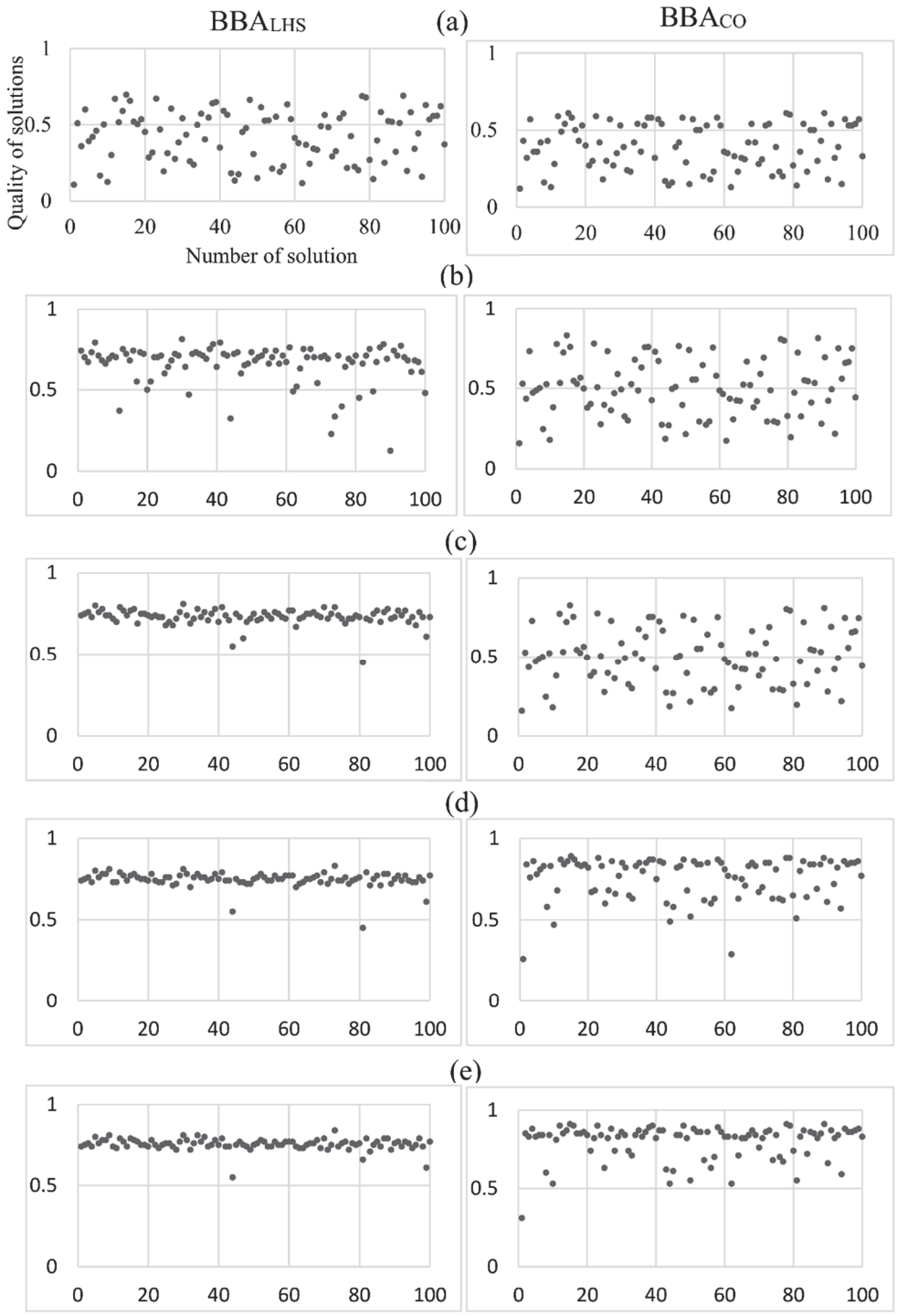

Figure 6. Diversity of solutions in (**a**) first generation, (**b**) generation 20, (**c**) generation 50, (**d**) generation 80 and (**e**) generation 100, using BBA_{LHS} (left) and BBA_{CO} (right) for Reuters dataset.

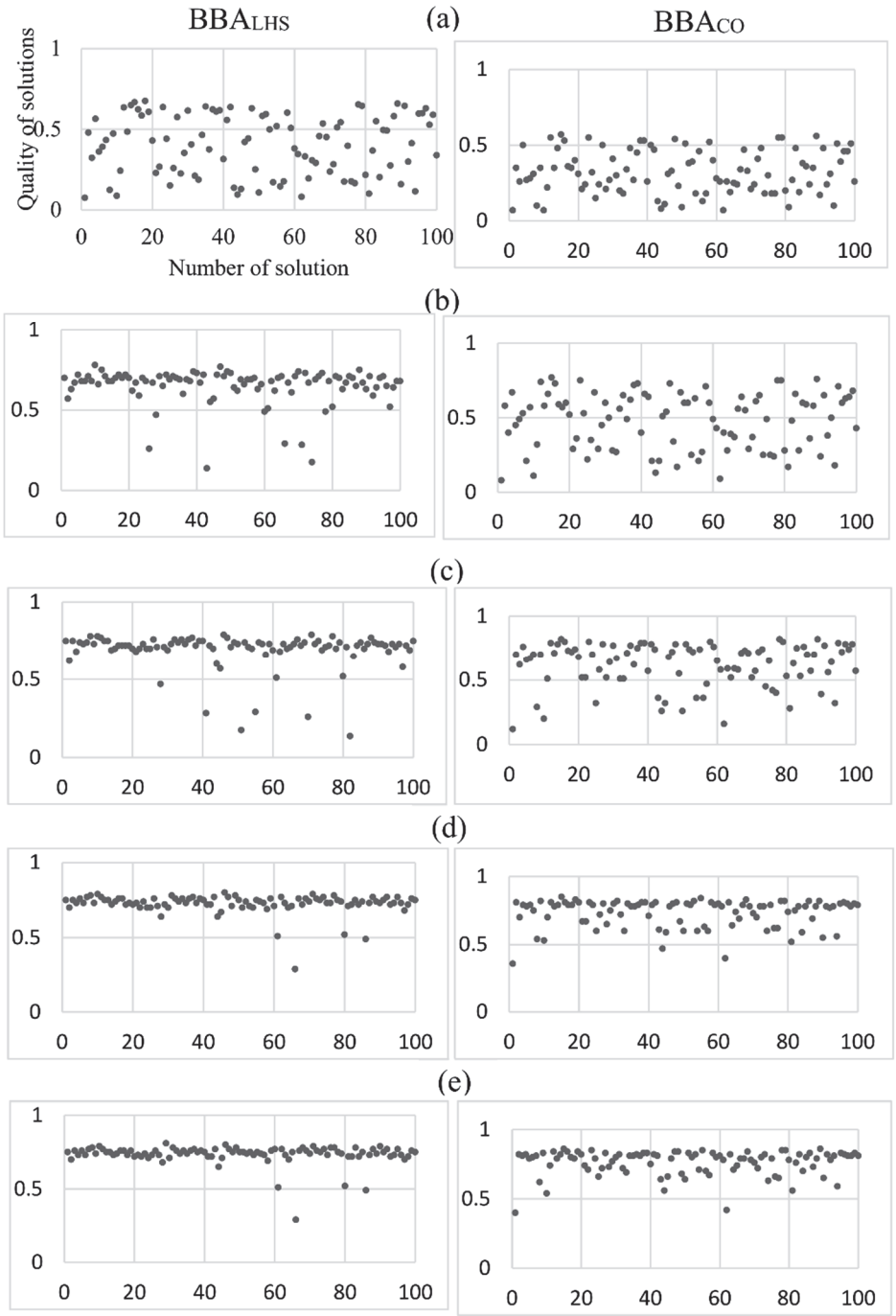

Figure 7. Diversity of solutions in (**a**) first generation, (**b**) generation 20, (**c**) generation 50, (**d**) generation 80 and (**e**) generation 100, using BBA_{LHS} (left) and BBA_{CO} (right) for WebKB dataset.

5.1.3. Statistical Test

The *t*-test is conducted for multiple groups of results to compare BBA_{LHS} and BBA_{CO} in terms of the population's diversity and quality in different generations. The population's diversity is measured using standard deviation, which is greater for more diverse populations. Moreover, the population's quality is measured by the average quality of its solutions and the relative errors as shown in the following subsections.

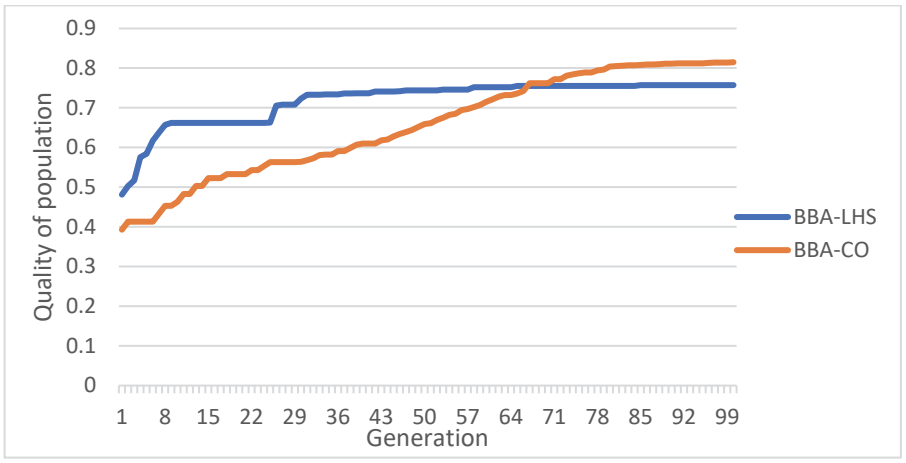

Figure 8. Convergence behavior of the population using BBA_{LHS} and BBA_{CO} for Reuters dataset.

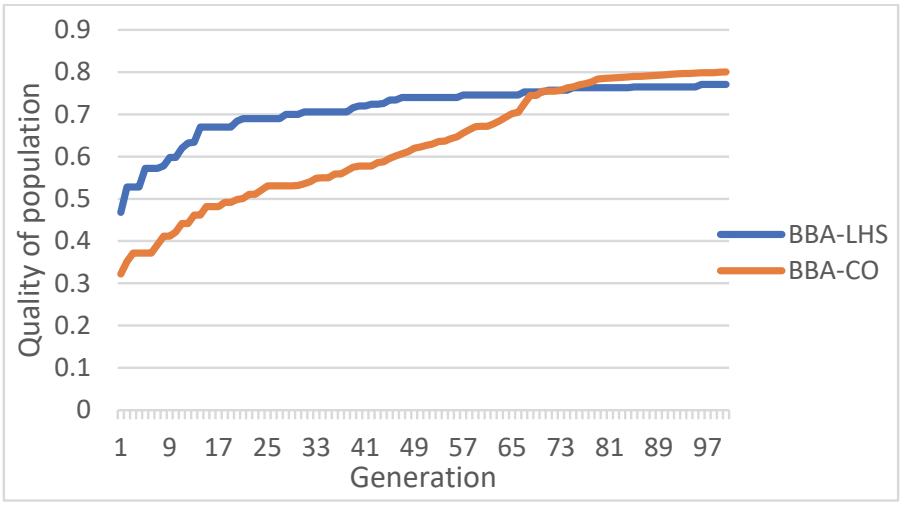

Figure 9. Convergence behavior of the population using BBA_{LHS} and BBA_{CO} for WebKB dataset.

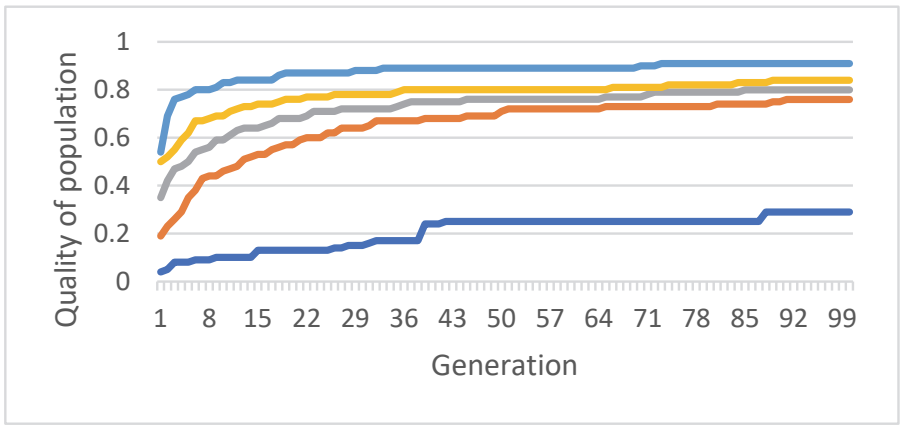

Figure 10. Convergence behavior of 5 solutions using BBA_{CO} for Reuters dataset.

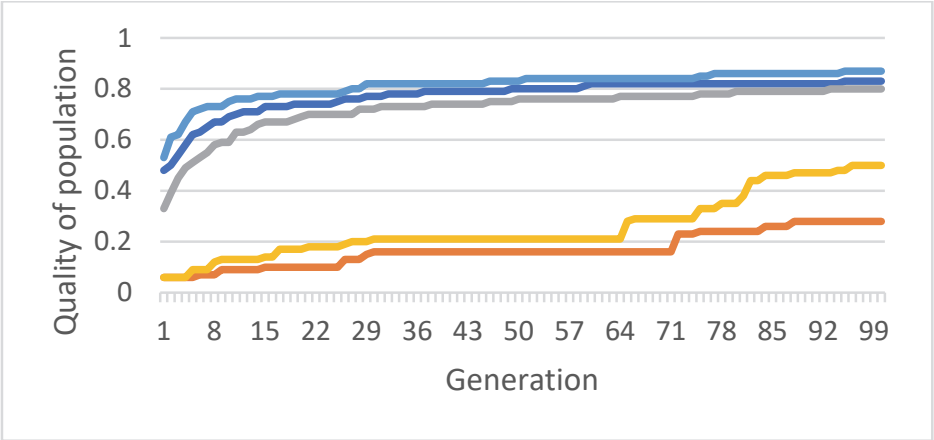

Figure 11. Convergence behavior of 5 solutions using BBA_{CO} for WebKB dataset.

Standard Deviation

The quality dispersion of the solutions in the population is measured on the basis of the standard deviation (SD), which is higher for more diverse distributions and lower for less diverse distributions. SD is calculated as in the equation below. Consequently, the SD of the population in certain generations (1, 20, 50, 80, 100) is calculated and recorded for 32 different runs in BBA_{LHS} and BBA_{CO} for each dataset. Table 3 shows the results of the *t*-test which is conducted between the results obtained using BBA_{LHS} and BBA_{CO} to investigate whether or not they are statistically different:

$$SD = \sqrt{\frac{\sum_{i=1}^{n}\left(fitness_i - \overline{fitness}\right)^2}{n}}$$

where SD stands for standard deviation; n is the population size; $fitness_i$ is the fitness of solution i; $\overline{fitness}$ is the average fitness of the initial population.

Table 3 shows that the SD of the population in the selected generations for both datasets using BBA_{CO} is significantly higher than the SD when BBA_{LHS} is used. Additionally, it could be stated that when using the *t*-test, the first group of results is significantly higher than the second group of results when *t State* is greater than *t Critical two-tail*. On the other hand, if *t State* is less than $-t$ *Critical two-tail*, this means that the results in the second group are significantly higher than the results in the first group. Meanwhile, if *t State* is within the interval [$-t$ *Critical two-tail*, *t Critical two-tail*], the difference is not considered to be significant.

The results in Table 3 show that *t State* is higher than *t Critical two-tail* in all the selected generations. The value of *t State* is higher than *t Critical two-tail* in all generations. The results of the *t*-test indicate that the population diversity which is expressed by standard deviation, is higher in BBA_{CO} during the search process. This reveals the ability of BBA_{CO} to control the population's diversity during the search process.

Population Quality

In order to measure the population quality, the average quality of the solutions will be used. This measure was utilized by the authors of [73] to measure the population quality. It is defined as the average quality of solutions in the population, as given in the following equation:

$$population\ quality\ (\%) = \left(1 - \frac{\overline{fitness} - best_{known}}{best_{known}}\right)$$

where $\overline{fitness}$ is the average quality (i.e., dependency value; fitness) of solutions in the population; $best_{known}$ is the best-known quality that the solution could reach.

Table 3. The *t*-test of standard deviation of population in different generations using BBA_{CO} and BBA_{LHS}.

Dataset	**Reuters**		**WebKB**	
Method	**BBA_{CO}**	**BBA_{LHS}**	**BBA_{CO}**	**BBA_{LHS}**
		generation 1		
Mean	0.16	0.14	0.14	0.13
p-Value	0.13	0.36	0.74	0.29
t State	3.75		4.38	
t Critical two-tail	2.00		2.01	
		generation 20		
Mean	0.15	0.12	0.14	0.13
p-Value	0.58	0.16	0.36	0.12
t State	13.96		1.94	
t Critical two-tail	2.03		2.01	
		generation 50		
Mean	0.14	0.08	0.11	0.09
p-Value	0.12	0.18	0.53	0.13
t State	7.93		3.61	
t Critical two-tail	2.01		2.00	
		generation 80		
Mean	0.12	0.06	0.09	0.08
p-Value	0.20	0.29	0.18	0.08
t State	15.67		2.98	
t Critical two-tail	2.01		2.00	
		generation 100		
Mean	0.10	0.05	0.07	0.06
p-Value	0.17	0.13	0.09	0.07
t State	11.72		2.24	
t Critical two-tail	2.01		2.02	

The population quality in certain generations (1, 20, 50, 80, 100) is calculated and recorded for 32 different runs in BBA_{LHS} and BBA_{CO} for each dataset. Table 4 shows the results of the *t*-test which is conducted for average population quality that are obtained using BBA_{LHS} and BBA_{CO} on the Reuters and WebKB datasets.

Table 4 shows that the average population quality in the early stages of the search process (i.e., generation 1 and 20) is significantly better with BBA_{LHS} in both datasets. By the second half of the search process (i.e., generation 50) the average population quality with BBA_{CO} improved and statistically outperformed the quality with BBA_{LHS}, in the Reuters dataset. In WebKB, the average population quality obtained by BBA_{CO} statistically outperformed the quality obtained by BBA_{LHS} by the last quarter of the search process (i.e., generation 80). In both datasets, the quality of the population in the end of search process is statistically better with BBA_{CO} than BBA_{LHS}.

From the results of the statistical test, it is clear that the population with BBA_{CO} improves more slowly than that of BBA_{LHS}. However, the population with BBA_{CO} converges with better quality than that of BBA_{LHS}. The reason behind the slow improvement of the population with BBA_{CO} is the controlling of diversity by the cooperative coevolving strategy. In this way, the algorithm is allowed to continue exploring a wider range of the search space, which significantly improves the final solution.

Table 4. The *t*-test of population quality in different generations using BBA_{CO} and BBA_{LHS}.

Dataset	**Reuters**		**WebKB**	
Method	**BBA_{CO}**	**BBA_{LHS}**	**BBA_{CO}**	**BBA_{LHS}**
		generation 1		
Mean	37.28	44.66	32.08	45.86
p-Value	0.39	0.35	0.26	0.31
t State	−7.60		−18.59	
t Critical two-tail	2.02		2.00	
		generation 20		
Mean	52.33	55.36	56.48	64.28
p-Value	0.32	0.16	0.13	0.36
t State	−3.26		−7.38	
t Critical two-tail	2.04		2.04	
		generation 50		
Mean	60.23	57.54	64.15	70.78
p-Value	0.64	0.06	0.19	0.15
t State	2.14		−8.38	
t Critical two-tail	2.00		2.04	
		generation 80		
Mean	72.98	70.83	73.09	71.47
p-Value	0.43	0.13	0.16	0.29
t State	3.48		2.58	
t Critical two-tail	2.03		2.03	
		generation 100		
Mean	73.98	71.16	74.27	72.79
p-Value	0.51	0.29	0.22	0.18
t State	2.79		2.82	
t Critical two-tail	2.00		2.01	

Relative Errors

The relative errors include Best Relative Error (BRE), Worst Relative Error (WRE) and Average Relative Error (ARE). BRE, WRE and ARE can be used to measure how far the distance between the best and worst solutions and the average quality of the population from the best-known quality that could be achieved by the solution. Whenever the error rate of a solution decreases, the solution becomes closer to the optimum solution. BRE, WRE and ARE have been used to compare BBA_{CO} with BBA_{LHS} in certain generations (i.e., 1, 20, 50, 80, 100). The *t*-tests of BRE in the selected generations with BBA_{LHS} and BBA_{CO} are shown in Table 5.

Observing the values of *t State* and *t Critical two-tail* in Table 5, it is clear that the BRE of the populations of BBA_{CO} are significantly higher in the beginning of the search process, in both datasets. By the end of the first half (i.e., generation 50), the difference between BRE in the two groups of results reveals no significance in both of the datasets. By the later generations of the search process (i.e., generations 80 and 100), the BRE with BBA_{CO} reduced and became significantly less than that of BBA_{LHS} in both datasets. The results of the statistical test for BRE indicate that in the beginning of the search process, the best solution in BBA_{LHS} outperforms that of BBA_{CO}. Then, with the progress of the search process, the best solution of BBA_{CO} improves until it outperforms that of BBA_{LHS}. This shows that BBA_{CO} is less subjected to stagnation than BBA_{LHS}. The *t*-test of ARE in the selected generations with BBA_{LHS} and BBA_{CO} are shown in Table 6.

Table 5. The *t*-test of BRE in different generations using BBA_{CO} and BBA_{LHS}.

Dataset	**Reuters**		**WebKB**	
Method	**BBA_{CO}**	**BBA_{LHS}**	**BBA_{CO}**	**BBA_{LHS}**
		generation 1		
Mean	35.83	31.96	47.16	34.30
p-Value	0.63	0.39	0.08	0.15
t State	2.11		18.56	
t Critical two-tail	2.03		2.03	
		generation 20		
Mean	24.03	20.50	23.75	23.50
p-Value	0.79	0.38	0.13	0.15
t State	3.14		0.21	
t Critical two-tail	2.03		2.01	
		generation 50		
Mean	20.42	20.00	21.86	23.37
p-Value	0.72	0.25	0.42	0.27
t State	0.45		−1.04	
t Critical two-tail	2.03		2.00	
		generation 80		
Mean	14.80	18.50	18.98	21.50
p-Value	0.54	0.29	0.36	0.16
t State	−7.24		−3.22	
t Critical two-tail	2.01		2.04	
		generation 100		
Mean	14.60	18.50	15.89	21.00
p-Value	0.84	0.27	0.45	0.19
t State	−4.57		−7.05	
t Critical two-tail	2.02		2.03	

As shown in Table 6, the ARE of BBA_{CO} population is significantly higher than that of BBA_{LHS} at the beginning of the search process in both datasets. In later generations (i.e., generations 20 and 50), the ARE of the population of BBA_{CO} keeps reducing but the difference with the other group (i.e., ARE of the population of BBA_{LHS}) is not statistically significant, in the Reuters dataset. By the last quarter of the search process (i.e., generations 80 and 100), the ARE of the population of BBA_{CO} became statistically lower than that of BBA_{LHS}, in the Reuters dataset. In the WebKB dataset, the ARE of BBA_{CO} was significantly higher than that of BBA_{LHS} within the first half of the search process. Later, the ARE of the BBA_{CO} population keeps reducing, but the difference with the other group is not significant. The results of ARE are consistent with those of BRE, as the population in BBA_{CO} improves slower than that of BBA_{LHS} because of controlling diversity. The T-test of WRE in the selected generations with BBA_{LHS} and BBA_{CO} is shown in Table 7.

The results in Table 7 indicate that WRE is significantly higher in the populations of BBA_{CO} until the end of the search process in the Reuters dataset. In the WebKB dataset, WRE remains significantly higher with BBA_{CO} than BBA_{LHS} within the first quarter of the search process. Then, the difference of WRE between both populations of BBA_{CO} and BBA_{LHS} became not significant until later generations. By the end of the search process, WRE of the BBA_{CO} population became significantly higher than that of BBA_{LHS} population. The reason behind that is that the populations of BBA_{CO} are more diverse, and thus, the solutions are more distributed in the search space until the later generations. In contrast, the solutions of BBA_{LHS} population in the later generations are found to be closer to each other. In this way, the quality of the worst solution with BBA_{LHS} remains better than the one with BBA_{CO} in Reuters dataset.

Table 6. The *t*-test of ARE in different generations using BBA_{CO} and BBA_{LHS}.

Dataset	**Reuters**		**WebKB**	
Method	**BBA_{CO}**	**BBA_{LHS}**	**BBA_{CO}**	**BBA_{LHS}**
		generation 1		
Mean	90.85	86.90	90.72	86.78
p-Value	0.14	0.38	0.40	0.16
t State	7.45		3.50	
t Critical two-tail	2.00		2.02	
		generation 20		
Mean	87.91	86.25	88.00	85.24
p-Value	0.18	0.21	0.32	0.06
t State	1.88		2.51	
t Critical two-tail	2.03		2.00	
		generation 50		
Mean	84.73	85.61	84.00	81.65
p-Value	0.09	0.19	0.10	0.13
t State	−0.90		1.86	
t Critical two-tail	2.00		2.00	
		generation 80		
Mean	78.07	68.95	75.12	79.15
p-Value	0.17	0.32	0.07	0.16
t State	6.88		−2.00	
t Critical two-tail	2.02		2.00	
		generation 100		
Mean	74.52	62.95	70.85	77.64
p-Value	0.24	0.14	0.21	0.08
t State	5.64		−2.84	
t Critical two-tail	2.00		2.02	

Table 7. The *t*-test of WRE in different generations using BBA_{CO} and BBA_{LHS}.

Dataset	**Reuters**		**WebKB**	
Method	**BBA_{CO}**	**BBA_{LHS}**	**BBA_{CO}**	**BBA_{LHS}**
		generation 1		
Mean	90.85	86.90	90.72	86.78
p-Value	0.14	0.38	0.40	0.16
t State	7.45		3.50	
t Critical two-tail	2.00		2.02	
		generation 20		
Mean	87.91	86.25	88.00	85.24
p-Value	0.18	0.21	0.32	0.06
t State	1.88		2.51	
t Critical two-tail	2.03		2.00	
		generation 50		
Mean	84.73	85.61	84.00	81.65
p-Value	0.09	0.19	0.10	0.13
t State	−0.90		1.86	
t Critical two-tail	2.00		2.00	

Table 7. *Cont.*

Dataset	Reuters		WebKB	
Method	**BBA_{CO}**	**BBA_{LHS}**	**BBA_{CO}**	**BBA_{LHS}**
		generation 80		
Mean	78.07	68.95	75.12	79.15
p-Value	0.17	0.32	0.07	0.16
t State	6.88		−2.00	
t Critical two-tail	2.02		2.00	
		generation 100		
Mean	74.52	62.95	70.85	77.64
p-Value	0.24	0.14	0.21	0.08
t State	5.64		−2.84	
t Critical two-tail	2.00		2.02	

The Quality and Reduction Rate of the Selected Feature Set

This subsection compares BBA_{CO} and BBA_{LHS} in terms of the quality of the final solution (i.e., the resultant feature set), the size of the selected feature set and the reduction rate. Tables 8 and 9 compare the two methods based on the mentioned metrics in the Reuters and WebKB datasets, respectively.

Table 8. Quality of the final solution, number of features and reduction rate of BBA_{CO} in the Reuters dataset.

Class Name	Quality of Final Solution	# of the Selected Features	Reduction Rate (%)
Earn	0.87	307	82.87
Acquisition	0.88	394	84.76
Trade	0.90	103	89.24
Ship	0.88	56	84.09
Grain	0.73	26	82.31
Crude	0.75	131	85.11
Interest	0.83	45	88.64
Money-fx	0.76	78	87.62
Corn	0.74	89	88.35
Wheat	0.79	157	81.06

Table 9. Quality of the final solution, number of features and reduction rate of BBA_{CO} in WebKB dataset.

Class Name	Quality of Final Solution	# of the Selected Features	Reduction Rate (%)
Student	0.81	226	90.23
Faculty	0.87	179	93.39
Course	0.74	105	94.41
Project	0.71	103	93.51

Tables 8 and 9 clearly show that BBA_{CO} improves the quality and the reduction rate of the selected feature set compared with the BBA_{LHS} results. The improvement is likely attributed to the ability of BBA_{CO} to combine the advantages of dividing the solutions into smaller parts and to maintain the populations' diversity. Dividing the solution into smaller solutions allows BBA_{CO} to better optimize the solution components (i.e., parts) resulting in a better final solution. The cooperative coevolving strategy in BBA_{CO} directs the algorithm to better convergence.

5.2. External Evaluation

This section compares the classification results obtained by the feature sets generated by BBA_{CO} and BBA_{LHS} for the Reuters and WebKB datasets. In order to compare the

classification performance using BBA-based methods, the classification is also conducted using Chi-Square (CHI), Information Gain (IG) and Gini Index (GI) feature-selection methods, which were successfully used in the literature for TFS [74–76]. Micro Average F1 and Macro Average F1 are used as performance measures for text classification. Some of the best classification results are shown in Table 10 comparing BBA_{CO} with BBA_{LHS}, CHI, IG and GI.

Table 10. Classification results using the feature sets generated by BBA_{LHS} and BBA_{CO}.

Dataset	Metric	Classifier	CHI	IG	GI	BBA_{LHS}	BBA_{CO}
Reuters	Micro average F1	NB	88.80	90.80	90.50	92.78	93.76
		SVM	86.70	89.40	89.80	92.55	94.08
		KNN	86.30	89.90	90.30	92.63	93.17
	Macro average F1	NB	79.50	78.50	77.20	89.87	90.03
		SVM	81.70	77.20	75.90	88.76	90.05
		KNN	66.60	68.30	69.10	88.04	89.49
WebKB	Micro average F1	NB	79.50	78.20	77.50	91.79	92.72
		SVM	88.30	89.30	89.10	91.64	92.84
		KNN	65.30	66.70	65.70	90.94	92.06
	Macro average F1	NB	78.20	76.90	75.90	89.82	91.67
		SVM	87.00	87.90	87.80	89.37	90.51
		KNN	60.70	62.50	61.40	88.03	90.87

Table 10 shows the classification results of the Reuters and WebKB datasets using three classifiers (i.e., NB, SVM and KNN) in Micro Average F1 and Macro Average F1. The results clearly show the improvement in the classification performance with all classifiers when using BBA-based TFS methods. In addition, BBA_{CO} improves the classification performance over BBA_{LHS} as a result of the improved feature sets selected by BBA_{CO}. Tables 11–14 show the classification results in terms of Precision, Recall and F-measure for the Reuters and WebKB datasets using BBA_{LHS} and BBA_{CO}, respectively.

Table 11. Precision (P), Recall (R) and F-measure (F) of each class in the Reuters dataset using the BBA_{LHS}.

Class	NB			SVM			KNN		
	P	R	F	P	R	F	P	R	F
Earn	99.18	98.60	98.89	98.09	95.35	96.70	94.02	92.25	93.13
Acquisition	93.16	98.60	95.80	85.95	96.78	91.04	90.54	82.68	86.43
Trade	96.36	91.98	94.12	97.05	87.97	92.29	87.17	92.04	89.54
Ship	97.56	78.43	86.96	96.76	75.29	84.69	82.25	88.39	85.21
Grain	91.96	71.53	80.47	94.20	78.14	85.42	81.02	89.03	84.84
Crude	81.70	94.58	87.67	87.76	82.41	85.00	97.79	91.19	94.37
Interest	95.08	94.79	94.93	96.46	88.02	92.05	82.32	87.12	84.65
Money-fx	97.37	79.20	87.35	91.10	81.99	86.31	89.20	98.67	93.70
Corn	89.63	82.59	85.97	90.15	86.23	88.15	87.87	78.35	82.84
Wheat	88.76	84.50	86.58	88.63	83.39	85.93	89.67	82.12	85.73
Average	93.08	87.48	89.87	92.61	85.56	88.76	88.18	88.18	88.04

Table 12. Precision (P), Recall (R) and F-measure (F) of each class in the WebKB dataset using BBA_{LHS}.

Class	NB			SVM			KNN		
	P	R	F	P	R	F	P	R	F
Project	96.97	85.69	90.98	83.91	97.75	90.30	88.68	97.38	92.83
Course	92.18	86.14	89.06	91.56	83.66	87.43	90.21	84.93	87.49
Faculty	84.85	91.16	87.89	88.98	82.51	85.62	85.96	76.44	80.92
Student	89.48	93.33	91.36	98.42	90.19	94.13	98.11	84.62	90.87
Average	90.87	89.08	89.82	90.72	88.53	89.37	90.74	85.84	88.03

Table 13. Precision (P), Recall (R) and F-measure (F) of each class in the Reuters dataset using BBA_{CO}.

Class	NB			SVM			KNN		
	P	R	F	P	R	F	P	R	F
Earn	94.31	92.66	93.48	96.54	91.41	93.90	91.74	93.94	92.83
Acquisition	90.87	97.25	93.95	89.17	98.36	93.54	91.77	96.56	94.10
Trade	94.22	91.18	92.68	96.06	91.28	93.61	94.59	88.91	91.66
Ship	97.56	78.43	86.96	93.62	77.39	84.73	95.37	70.55	81.10
Grain	94.92	77.78	85.50	93.73	82.64	87.84	89.13	76.47	82.32
Crude	96.49	92.12	94.25	95.16	91.57	93.33	93.99	95.54	94.76
Interest	93.29	89.87	91.55	91.53	88.87	90.18	93.29	93.86	93.57
Money-fx	93.80	90.81	92.28	92.07	87.69	89.83	95.62	92.49	94.03
Corn	90.31	79.52	84.57	90.64	82.59	86.43	86.89	82.98	84.89
Wheat	82.87	87.45	85.10	88.26	85.98	87.11	86.97	84.35	85.64
Average	92.86	87.71	90.03	92.68	87.78	90.05	91.94	87.56	89.49

Table 14. Precision (P), Recall (R) and F-measure (F) of each class in the WebKB dataset using BBA_{CO}.

Class	NB			SVM			KNN		
	P	R	F	P	R	F	P	R	F
Project	95.88	92.29	94.05	96.67	91.57	94.05	91.76	93.35	92.55
Course	87.72	93.57	90.55	97.18	89.33	93.09	93.55	86.74	90.02
Faculty	88.61	79.89	84.02	81.13	88.22	84.53	85.57	92.17	88.75
Student	97.69	98.42	98.05	86.85	94.20	90.38	90.79	93.58	92.16
Average	92.48	91.04	91.67	90.46	90.83	90.51	90.42	91.46	90.87

Tables 13 and 14 clearly show the improvement in the classification results when using BBA_{CO} as a TFS method. Comparing the classification results achieved by BBA_{CO} with those of BBA_{LHS} (i.e., Tables 11 and 12), it is found that the average F is improved with the three classifiers in both datasets. From the experimental results, it is shown that dividing the solutions into smaller ones and optimizing each part then improved the resulting final solution, controlled the population's diversity and improved the classification performance as a result of the improved selected feature set. In order to test if the performance of BBA_{CO} is significantly better than that of BBA_{LHS}, the *t*-test needs to be applied to their classification results. The results of the statistical test are presented in Table 15.

As shown in Table 15, the value of *t State* is less than *-t critical two-tail* in all cases. These results indicate that the second group of the classification results is significantly higher than the first group (i.e., the classification results obtained by using the selected feature set by BBA_{LHS}). The reason behind that is attributed to the cooperative coevolving strategy that is utilized in BBA_{CO}, which improves the performance of the text-feature selection method by dividing the solution into smaller ones with a smaller dimension than the original full solution. This way, optimizing each part separately improves the performance and controls the diversity of the population.

Table 15. The *t*-test of classification results of BBA_{CO} vs. BBA_{LHS} methods.

Dataset	**Reuters**		**WebKB**	
Method	**BBA_{LHS}**	**BBA_{CO}**	**BBA_{LHS}**	**BBA_{CO}**
	NB		NB	
Mean	92.25	93.41	91.35	92.22
p-Value	0.08	0.18	0.49	0.33
t State	−8.23		−5.83	
t Critical two-tail	2.02		2.00	
	KNN		KNN	
Mean	92.17	93.09	90.59	91.88
p-Value	0.10	0.59	0.62	0.99
t State	−6.37		−9.74	
t Critical two-tail	2.00		2.02	
	SVM		SVM	
Mean	91.83	93.67	91.20	92.14
p-Value	0.56	0.59	0.42	0.98
t State	−16.54		−7.41	
t Critical two-tail	2.00		2.03	

5.3. Results on Non-English Datasets

This section investigates the ability of the proposed text-feature selection method to be generalized for different languages. Therefore, the two variants of the proposed text-feature selection method i.e., BBA_{LHS} and BBA_{CO}, are tested on two Malay and one Arabic datasets. The quality of the selected feature sets, the number of the selected features and reduction rate are reported in this section. Furthermore, the classification results utilizing three classifiers (i.e., NB, SVM and KNN) are reported and discussed.

5.3.1. Results of Malay Datasets

The classification results of Malay datasets utilizing three classifiers and the two versions of the proposed method are reported in this subsection. The results are also compared with the classification results using no feature selection method and using Chi-Square feature selection method. Chi-Square is a well-known method that has been successfully utilized for feature selection [9,74,77]. Firstly, the characteristics of the resulting feature sets by BBA_{LHS} and BBA_{CO} are displayed in Tables 16 and 17 for Mix-DS and Harian Metro dataset, respectively. Then, the classification results are reported in Table 18.

Table 16. Characteristics of the selected feature sets in Mix-DS, where Q indicates the quality of feature set, #F indicates to number of features, R indicates reduction rate (%).

Class	**Q-BBA_{LHS}**	**Q-BBA_{CO}**	**#F**	**#F-BBA_{LHS}**	**#F-BBA_{CO}**	**R-BBA_{LHS}**	**R-BBA_{CO}**
Bisnes	0.55	0.63	4785	1058	1014	77.89	78.81
Hiburan	0.56	0.60	10,357	1468	1328	85.83	87.18
Pendidikan	0.63	0.71	2212	564	428	74.50	80.65
Politik	0.54	0.54	1534	341	325	77.77	78.81
Sains-Teknologi	0.59	0.65	3023	784	751	74.07	75.16
Sukan	0.61	0.67	8320	1127	946	86.45	88.63

Table 17. Characteristics of the selected feature sets in Harian Metro dataset, where Q indicates the quality of feature set, #F indicates to number of features, R indicates reduction rate (%).

Class	Q-BBA$_{LHS}$	Q-BBA$_{CO}$	#F	#F-BBA$_{LHS}$	#F-BBA$_{CO}$	R-BBA$_{LHS}$	R-BBA$_{CO}$
Addin	0.416	0.483	3925	315	256	91.97	93.48
Bisnes	0.536	0.547	2564	335	184	86.93	92.82
Dekotaman	0.564	0.58	3396	358	314	89.46	90.75
Global	0.667	0.638	2052	259	237	87.38	88.45
Hiburan	0.645	0.651	2947	356	330	87.92	88.80
Pendidikan	0.655	0.685	3284	365	303	88.89	90.77
Santai	0.517	0.592	3940	428	402	89.14	89.80
Sihat	0.662	0.688	3943	382	374	90.31	90.51
Sukan	0.406	0.489	2041	363	327	82.21	83.98
Teknologi	0.537	0.669	2818	201	194	92.87	93.12
Vroom	0.616	0.668	3150	362	342	88.51	89.14

Table 18. Classification results achieved by using the original dataset, the feature sets selected by Chi-square, BBA$_{LHS}$ and BBA$_{CO}$.

Dataset	Metric	Classifier	No FS	Chi-Square	BBA$_{LHS}$	BBA$_{CO}$
Mix-DS	Micro average F1	NB	76.32	84.32	88.50	89.34
		SVM	76.01	84.6	88.04	88.72
		KNN	75.94	84.07	87.77	88.24
	Macro average F1	NB	73.40	83.58	86.61	88.42
		SVM	74.72	82.15	86.98	87.38
		KNN	73.35	82.42	86.35	87.47
Harian-Metro dataset	Micro average F1	NB	68.20	78.94	82.38	83.56
		SVM	67.38	79.19	82.16	83.71
		KNN	66.77	78.51	81.64	83.18
	Macro average F1	NB	67.49	76.09	81.10	83.61
		SVM	66.29	77.13	80.86	82.16
		KNN	66.02	75.84	80.55	81.98

Tables 16 and 17 compare the performance of BBA$_{LHS}$ and BBA$_{CO}$ in terms of quality, number of selected features and reduction rate. The tables show that both methods are able to reduce the dimensionality of the feature space efficiently. In addition, it is clear that BBA$_{CO}$ outperforms BBA$_{LHS}$ in terms of the quality of the feature set and the reduction rate. Table 18 reports the classification results using the selected feature sets by BBA$_{CO}$ compared with the selected sets by Chi-square, BBA$_{LHS}$ and with the original feature set. The impact of using an efficient feature selection method is clear, which is consistent with what has been concluded in the literature. It is clear how Chi-square has improved the classification results over the original dataset. However, the classification performance with BBA$_{LHS}$ and BBA$_{CO}$ is clearly improved over Chi-square. The results demonstrate the ability of BBA-based methods to select discriminative feature sets. Furthermore, using a coevolutionary technique has clearly improved the performance of the TFS method, which has been reflected in the improved classification results. Tables 19 and 20 show the classification results in terms of precision, recall and F-measure for Mix-DS and Harian Metro dataset, respectively.

Table 19. Classification results of Mix-DS using NB, SVM and KNN classifiers in terms of precision (P), recall (R) and F-measure (F).

Class	NB-No FS			SVM-No FS			KNN-No FS			NB-Chi-Square			SVM-Chi-Square			KNN-Chi-Square		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Bisnes	0.93	0.79	0.85	0.91	0.80	0.85	0.93	0.78	0.85	0.93	0.91	0.92	0.91	0.89	0.90	0.95	0.93	0.94
Hiburan	0.81	0.99	0.89	0.82	0.93	0.87	0.59	0.83	0.69	0.88	0.96	0.92	0.97	0.96	0.96	0.83	0.86	0.84
Pendidikan	0.93	0.43	0.59	0.87	0.49	0.62	0.86	0.56	0.68	0.59	0.82	0.69	0.64	0.75	0.69	0.76	0.87	0.81
Politik	0.94	0.54	0.68	0.94	0.62	0.75	0.83	0.64	0.72	0.89	0.83	0.86	0.81	0.73	0.77	0.83	0.96	0.89
Sains-Teknologi	0.65	0.41	0.50	0.59	0.43	0.50	0.67	0.69	0.68	0.64	0.65	0.64	0.62	0.64	0.63	0.73	0.48	0.58
Sukan	0.99	0.82	0.90	0.96	0.84	0.89	0.85	0.73	0.78	0.98	0.98	0.98	0.97	0.98	0.98	0.82	0.93	0.87
Average	0.87	0.66	0.73	0.85	0.68	0.75	0.79	0.71	0.73	0.82	0.86	0.84	0.82	0.83	0.82	0.82	0.84	0.82
Class	NB-BBA$_{LHS}$			SVM-BBA$_{LHS}$			KNN-BBA$_{LHS}$			NB-BBA$_{CO}$			SVM-BBA$_{CO}$			KNN-BBA$_{CO}$		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Bisnes	0.99	0.89	0.94	0.99	0.89	0.93	0.88	0.93	0.90	0.98	0.93	0.95	0.96	0.94	0.95	0.89	0.89	0.89
Hiburan	0.77	0.99	0.86	0.73	0.94	0.82	0.74	0.89	0.81	0.92	0.89	0.90	0.93	0.89	0.91	0.89	0.86	0.88
Pendidikan	0.94	0.73	0.82	0.96	0.75	0.84	0.79	0.92	0.85	0.78	0.91	0.84	0.77	0.90	0.83	0.80	0.88	0.83
Politik	0.99	0.82	0.90	0.99	0.80	0.89	0.85	0.95	0.89	0.86	0.96	0.91	0.83	0.96	0.89	0.90	0.81	0.85
Sains-Teknologi	0.81	0.77	0.79	0.76	0.93	0.84	0.83	0.72	0.77	0.83	0.70	0.76	0.82	0.66	0.73	0.78	0.88	0.83
Sukan	0.94	0.85	0.89	0.94	0.86	0.90	0.95	0.97	0.96	0.94	0.95	0.95	0.93	0.95	0.94	0.97	0.97	0.97
Average	0.91	0.84	0.87	0.89	0.86	0.87	0.84	0.89	0.86	0.88	0.89	0.88	0.87	0.88	0.87	0.87	0.88	0.87

Table 20. Classification results of Harian Metro dataset using NB, SVM and KNN classifiers in terms of precision (P), recall (R) and F-measure (F).

Class	NB-No FS			SVM-No FS			KNN-No FS			NB-Chi-Square			SVM-Chi-Square			KNN- Chi-Square		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Addin	0.93	0.57	0.70	0.80	0.57	0.66	0.45	0.69	0.55	0.98	0.73	0.84	0.71	0.96	0.82	0.94	0.76	0.84
Bisnes	0.62	0.76	0.69	0.80	0.55	0.65	0.86	0.50	0.63	0.68	0.78	0.73	0.88	0.65	0.75	0.84	0.64	0.73
Dekotaman	0.88	0.51	0.64	0.77	0.53	0.62	0.93	0.53	0.68	0.98	0.68	0.80	0.65	0.88	0.75	0.66	0.73	0.70
Global	0.93	0.50	0.65	0.65	0.64	0.64	0.95	0.52	0.67	0.81	0.74	0.77	0.73	0.81	0.77	0.87	0.66	0.75
Hiburan	0.91	0.50	0.64	0.76	0.52	0.62	0.93	0.51	0.66	0.90	0.69	0.78	0.90	0.58	0.70	0.69	0.85	0.76
Pendidikan	0.86	0.60	0.71	0.74	0.66	0.69	0.88	0.63	0.73	0.65	0.73	0.69	0.76	0.65	0.70	0.78	0.69	0.73
Santai	0.81	0.51	0.63	0.56	0.55	0.55	0.86	0.54	0.66	0.75	0.86	0.80	0.66	0.71	0.68	0.62	0.82	0.71
Sihat	0.87	0.57	0.69	0.54	0.79	0.64	0.42	0.79	0.54	0.74	0.62	0.68	0.75	0.87	0.81	0.87	0.76	0.81
Sukan	0.59	0.98	0.74	0.66	0.91	0.77	0.59	0.97	0.73	0.56	0.72	0.63	0.97	0.76	0.85	0.74	0.99	0.85
Teknologi	0.78	0.64	0.70	0.85	0.58	0.69	0.78	0.66	0.71	0.98	0.68	0.81	0.91	0.73	0.81	0.88	0.69	0.77
Vroom	0.63	0.63	0.63	0.89	0.63	0.74	0.91	0.55	0.69	0.98	0.76	0.85	0.93	0.77	0.84	0.86	0.59	0.70
Average	0.80	0.62	0.67	0.73	0.63	0.66	0.78	0.63	0.66	0.82	0.73	0.76	0.80	0.76	0.77	0.80	0.74	0.76
Class	NB-BBA$_{LHS}$			SVM-BBA$_{LHS}$			KNN-BBA$_{LHS}$			NB-BBA$_{CO}$			SVM-BBA$_{CO}$			KNN-BBA$_{CO}$		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Addin	0.84	0.77	0.80	0.77	0.86	0.81	0.82	0.76	0.79	0.71	0.95	0.81	0.90	0.82	0.86	0.86	0.84	0.85
Bisnes	0.77	0.94	0.85	0.87	0.77	0.82	0.85	0.76	0.80	0.89	0.73	0.80	0.82	0.78	0.80	0.84	0.79	0.82
Dekotaman	0.92	0.76	0.83	0.91	0.81	0.86	0.94	0.81	0.87	0.89	0.92	0.90	0.75	0.86	0.80	0.83	0.87	0.85
Global	0.78	0.88	0.82	0.72	0.86	0.78	0.75	0.85	0.80	0.85	0.71	0.77	0.84	0.77	0.80	0.79	0.84	0.81
Hiburan	0.73	0.79	0.76	0.83	0.73	0.78	0.71	0.80	0.76	0.87	0.92	0.89	0.77	0.89	0.83	0.86	0.82	0.84
Pendidikan	0.84	0.89	0.86	0.84	0.78	0.81	0.88	0.89	0.88	0.81	0.85	0.83	0.75	0.87	0.81	0.88	0.77	0.82
Santai	0.71	0.77	0.74	0.74	0.81	0.78	0.71	0.85	0.77	0.93	0.77	0.84	0.89	0.76	0.82	0.82	0.79	0.80
Sihat	0.87	0.79	0.83	0.82	0.79	0.81	0.76	0.76	0.76	0.76	0.87	0.81	0.88	0.92	0.90	0.81	0.81	0.81
Sukan	0.79	0.83	0.81	0.89	0.74	0.81	0.74	0.88	0.81	0.89	0.92	0.90	0.90	0.74	0.81	0.88	0.74	0.80
Teknologi	0.88	0.73	0.80	0.84	0.79	0.82	0.83	0.79	0.81	0.84	0.78	0.81	0.83	0.80	0.81	0.88	0.81	0.84
Vroom	0.84	0.79	0.81	0.91	0.76	0.83	0.88	0.76	0.81	0.78	0.85	0.81	0.72	0.89	0.80	0.84	0.73	0.78
Average	0.82	0.81	0.81	0.83	0.79	0.81	0.81	0.81	0.81	0.84	0.84	0.84	0.82	0.83	0.82	0.84	0.80	0.82

5.3.2. Results of Arabic Dataset

The classification results of the Arabic dataset utilizing three classifiers are reported in this subsection. The quality of the selected feature sets, the number of the selected features and reduction rate are also reported in Table 21.

Table 21 compares the performance of BBA$_{LHS}$ and BBA$_{CO}$ in terms of quality, number of selected features and reduction rate. It is noticeable that both methods were efficient in reducing the feature set. It is also clear that BBA$_{CO}$ outperforms BBA$_{LHS}$ with all classes, which indicates the efficiency of the proposed coevolutionary technique.

Table 21. Characteristics of the selected feature sets in Al-Jazeera news dataset, where Q indicates the quality of feature set, #F indicates number of features, R indicates reduction rate (%).

Class	Q-BBA$_{LHS}$	Q-BBA$_{CO}$	#F	#F-BBA$_{LHS}$	#F-BBA$_{CO}$	R-BBA$_{LHS}$	R-BBA$_{CO}$
Economy	0.54	0.61	1541	537	397	65.15	74.24
Politics	0.59	0.64	1427	343	152	75.96	89.35
Sport	0.62	0.68	1874	294	199	84.31	89.38
Science	0.63	0.68	1505	235	176	84.39	88.31
Art	0.66	0.71	1693	544	267	67.87	84.23

The classification results in terms of average precision, average recall and Macro average F1 are reported and compared with the state-of-the-art results in Table 22. The state-of-the-art studies in the TFS field utilized different datasets, classifiers and evaluation measures. However, the experimental setting such as the utilized database, classifiers and evaluation metrics should be same in order to reach fair comparison. Thus, few studies are comparable with the proposed method for using the same experimental setting. The methods used for comparison are:

I. Binary Particle Swarm Optimization with k-nearest neighbor (BPSO-KNN) [78];
II. Enhanced Genetic Algorithm (EGA) [69];
III. Category relevant feature measure (CRFM) [79].

The authors of [78] combined Binary PSO and k-nearest neighbor to select a feature set for Arabic text classification. They used three Arabic datasets and three classifiers to evaluate the performance of their method. A feature selection method based on Enhanced Genetic Algorithm (EGA) was proposed in the study by [69]. Their method was evaluated using three Arabic datasets and two classification algorithms namely NB and AC. The same datasets and classification algorithms were used in the study of [79] which proposed three enhanced feature selection methods. However, the results of CRFM have been used in the comparison as it performs better than the other two methods. In the current study, the classification was performed using NB and SVM that where successfully utilized in previous studies for text classification [69,78–80].

Table 22. Classification results of BBA$_{R}$, BBA$_{LHS}$, and BBA$_{CO}$ on Al-Jazeera news dataset compared with state-of-the-art results.

Metric	Classifier	BPSO-KNN	EGA	CRFM	BBA$_{LHS}$	BBA$_{CO}$
Macro Average precision	NB	85.76	91	88.77	91.34	91.64
	SVM	93.7	-	-	90.86	92.22
Macro Average Recall	NB	84.34	90.66	88.33	90.42	91.09
	SVM	92.98	-	-	91.58	92.88
Macro Average F1	NB	84.63	90.83	88.55	90.73	91.24
	SVM	93.12	-	-	91.07	92.42

Table 22 reports the classification results of the two variants of BBA-based TFS methods compared with three state-of-the-art results. Based on Table 22, it is obvious that the results of NB with BBAco outperform the results of all other methods. On the other hand, the results of SVM with BPSO-KNN outperform the results of the other methods. However, unlike BPSO-KNN, the proposed BBA-based methods likely perform above 90% with both utilized classifiers. Hence, it is clear that the proposed BBA-based TFS methods are efficient with the Arabic dataset. Table 23 shows the classification results of the Al-Jazeera news dataset with BBA$_{LHS}$ and BBAco in terms of precision, recall and F-measure.

Table 23. Classification results of Al-Jazeera news dataset using NB and SVM in terms of precision (P), recall (R) and F-measure (F).

Class	NB-BBA$_{LHS}$			SVM-BBA$_{LHS}$			NB-BBA$_{CO}$			SVM-BBA$_{CO}$		
	P	R	F	P	R	F	P	R	F	P	R	F
Economy	94.85	83.65	88.90	91.13	95.20	93.12	89.04	91.51	90.26	94.17	99.59	96.80
Politics	92.28	97.32	94.73	85.25	97.55	90.99	86.92	97.85	92.06	88.50	97.63	92.84
Sport	91.10	85.55	88.24	93.87	85.98	89.75	97.55	88.90	93.02	92.66	83.98	88.11
Science	90.94	89.11	90.02	86.30	87.50	86.90	91.05	86.72	88.83	86.44	90.50	88.42
Art	87.51	96.46	91.77	97.73	91.69	94.61	93.62	90.46	92.01	99.34	92.69	95.90
Average	91.34	90.42	90.73	90.86	91.58	91.07	91.64	91.09	91.24	92.22	92.88	92.42

Table 23 shows that the average precision, recall and F-measure with BBA$_{CO}$ are clearly higher than those of BBA$_{LHS}$ in terms of average precision, recall and F-measure in all cases. These results demonstrate the efficiency of the proposed TFS method with the Arabic dataset. This successfully shows that the proposed BBA-based text-feature selection method is applicable with the Arabic language.

In order to test whether the improvement of BBA$_{CO}$ over BBA$_{LHS}$ is significant or not, the statistical test (i.e., the *t*-test) is performed to the classification results of both methods. The statistical test compares the results of BBA$_{LHS}$ with those of BBA$_{CO}$ using NB and SVM classifiers. The *t*-test is conducted for the results of BBA$_{LHS}$ vs. BBA$_{CO}$, to test whether the difference is significant or not. The results of the *t*-test are reported in Table 24.

Table 24. The *t*-test of classification results of BBA$_{LHS}$ vs. BBA$_{CO}$ methods.

	BBA$_{LHS}$		BBA$_{CO}$
		NB	
Mean	90.78		91.23
p-Value	0.20		0.96
t Stat		−6.89	
t Critical two-tail		2.00	
		SVM	
Mean	91.11		92.52
p-Value	0.07		0.46
t Stat		−23.04	
t Critical two-tail		2.00	

As shown in Table 24, the value of *t State* is less than *-t Critical two-tail* with both classifiers. These results indicate that the second group of classification results (i.e., the classification results obtained using the selected feature set by BBA$_{CO}$) is significantly higher than the first group (i.e., the classification results obtained using the selected feature set by BBA$_{LHS}$). These results demonstrate the efficiency of BBA$_{CO}$ to improve the resulting feature set, and therefore improve the classification performance by utilizing the coevolutionary technique as explained above in order to generating a better feature set.

Observing the results of the five utilized datasets in this study, namely Reuters, WebKB, Mix-DS, Harian Metro dataset and Aljazeera news dataset, it is noticed that the proposed text-feature selection method has a similar effect to all datasets, regardless of the language. The reason behind that is mostly attributed to the nature of BBA-based TFS methods, which depend on the mathematical calculations and do not consider the semantic attributes. Therefore, the performance of the proposed methods is independent from the dataset language.

6. Conclusions

Feature selection is a crucial step in text classification in order to overcome the high dimensionality of the feature space and improve the classification accuracy. Thus, many feature selection methods have been proposed in the literature. Those methods could be ranking methods, which rank the features and select the top ranked ones, or meta-heuristic-based methods that work as a wrapper around certain classification algorithms. However, the ranking methods ignore the correlation between features, while the wrappers are classifier-dependent. To overcome those limitations, this paper introduced the cooperative binary Bat algorithm (BBA_{CO}) and investigated its performance for the text feature selection. The quality of the final solution and the number of selected features are also compared with those of BBA_{LHS}; in order to test the impact of cooperative BBA.

The text classification was performed on two slandered English datasets, namely Reuters and WebKB, to evaluate the discriminative ability of the produced feature set using BBA_{CO}. The best classification results obtained with BBA_{CO} were 94% and 92.8% in terms of Micro Average for the Reuters and WebKB datasets, respectively. In comparison, the best results obtained by the other methods were 90.5% and 89.3% for the Reuters and WebKB datasets, respectively, and those results were obtained by IG. The text classification was also performed for two Malay and one Arabic datasets. The statistical test demonstrated that the improvement of the classification performance was significant. The experimental results in this work have shown the ability of the proposed method to improve the final solution, control the population's diversity and improve the text classification accuracy. In addition, the proposed method was approved to be general in terms of dataset language.

For future work, the proposed coevolutionary method could be tested on a different population-based meta-heuristic algorithms to evaluate its performance. Additionally, the proposed coevolutionary method could be adapted to different high dimensional optimization problems. Moreover, designing a feature selection method based on parallel BBA could also be a better option for very high-dimensional datasets.

Author Contributions: Conceptualization, S.A.; Funding acquisition, N.O. and S.A.; Investigation, A.A.; Project administration, N.O.; Resources, A.A.-S.; Software, A.A. and A.A.-S.; Supervision, N.O. and S.A.; Writing—original draft, A.A.; Writing—review & editing, N.O. and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the grant FRGS/1/2020/ICT02/UKM/02/6 and TAP-K007009.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not Applicable, the study does not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Deng, X.; Li, Y.; Weng, J.; Zhang, J. Feature selection for text classification: A review. *Multimed. Tools Appl.* **2018**, *78*, 3797–3816. [CrossRef]
2. Namous, F.; Faris, H.; Heidari, A.A.; Khalafat, M.; Alkhawaldeh, R.S.; Ghatasheh, N. Evolutionary and swarm-based feature selection for imbalanced data classification. In *Evolutionary Machine Learning Techniques*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 231–250.
3. Pervaiz, U.; Khawaldeh, S.; Aleef, T.A.; Minh, V.H.; Hagos, Y.B. Activity monitoring and meal tracking for cardiac rehabilitation patients. *Int. J. Med. Eng. Inform.* **2018**, *10*, 252–264. [CrossRef]
4. Elminaam, D.S.A.; Nabil, A.; Ibraheem, S.A.; Houssein, E.H. An Efficient Marine Predators Algorithm for Feature Selection. *IEEE Access* **2021**, *9*, 60136–60153. [CrossRef]
5. Abdel-Basset, M.; El-Shahat, D.; El-Henawy, I.; de Albuquerque, V.H.C.; Mirjalili, S. A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Syst. Appl.* **2020**, *139*, 112824. [CrossRef]
6. Qaraad, M.; Amjad, S.; Hussein, N.K.; Elhosseini, M.A. Large scale salp-based grey wolf optimization for feature selection and global optimization. *Neural Comput. Appl.* **2022**, *34*, 8989–9014. [CrossRef]
7. Labani, M.; Moradi, P.; Jalili, M. A multi-objective genetic algorithm for text feature selection using the relative discriminative criterion. *Expert Syst. Appl.* **2020**, *149*, 113276. [CrossRef]

8. Al-Dyani, W.Z.; Ahmad, F.K.; Kamaruddin, S.S. Binary Bat Algorithm for text feature selection in news events detection model using Markov clustering. *Cogent Eng.* **2021**, *9*, 2010923. [CrossRef]
9. BinSaeedan, W.; Alramlawi, S. CS-BPSO: Hybrid feature selection based on chi-square and binary PSO algorithm for Arabic email authorship analysis. *Knowl.-Based Syst.* **2021**, *227*, 107224. [CrossRef]
10. Feng, J.; Kuang, H.; Zhang, L. EBBA: An Enhanced Binary Bat Algorithm Integrated with Chaos Theory and Lévy Flight for Feature Selection. *Future Internet* **2022**, *14*, 178. [CrossRef]
11. Hashemi, A.; Joodaki, M.; Joodaki, N.Z.; Dowlatshahi, M.B. Ant colony optimization equipped with an ensemble of heuristics through multi-criteria decision making: A case study in ensemble feature selection. *Appl. Soft Comput.* **2022**, *124*, 109046. [CrossRef]
12. Ibrahim, A.M.; Tawhid, M.A. A new hybrid binary algorithm of bat algorithm and differential evolution for feature selection and classification. In *Applications of bat Algorithm and Its Variants*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 1–18.
13. Li, A.-D.; Xue, B.; Zhang, M. Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies. *Appl. Soft Comput.* **2021**, *106*, 107302. [CrossRef]
14. Ma, W.; Zhou, X.; Zhu, H.; Li, L.; Jiao, L. A two-stage hybrid ant colony optimization for high-dimensional feature selection. *Pattern Recognit.* **2021**, *116*, 107933. [CrossRef]
15. Paul, D.; Jain, A.; Saha, S.; Mathew, J. Multi-objective PSO based online feature selection for multi-label classification. *Knowl.-Based Syst.* **2021**, *222*, 106966. [CrossRef]
16. Tripathi, D.; Reddy, B.R.; Reddy, Y.P.; Shukla, A.K.; Kumar, R.K.; Sharma, N.K. BAT algorithm based feature selection: Application in credit scoring. *J. Intell. Fuzzy Syst.* **2021**, *41*, 5561–5570. [CrossRef]
17. Xue, Y.; Zhu, H.; Liang, J.; Słowik, A. Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification. *Knowl.-Based Syst.* **2021**, *227*, 107218. [CrossRef]
18. Yasaswini, V.; Baskaran, S. An Optimization of Feature Selection for Classification Using Modified Bat Algorithm. In *Advanced Computing and Intelligent Technologies*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 389–399.
19. Alim, A.; Naseem, I.; Togneri, R.; Bennamoun, M. The most discriminant subbands for face recognition: A novel information-theoretic framework. *Int. J. Wavelets Multiresolution Inf. Process* **2018**, *16*, 1850040. [CrossRef]
20. Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
21. Al-Betar, M.A.; Alomari, O.A.; Abu-Romman, S.M. A TRIZ-inspired bat algorithm for gene selection in cancer classification. *Genomics* **2019**, *112*, 114–126. [CrossRef]
22. Alsalibi, B.; Abualigah, L.; Khader, A.T. A novel bat algorithm with dynamic membrane structure for optimization problems. *Appl. Intell.* **2020**, *51*, 1992–2017. [CrossRef]
23. Devi, D.R.; Sasikala, S. Online Feature Selection (OFS) with Accelerated Bat Algorithm (ABA) and Ensemble Incremental Deep Multiple Layer Perceptron (EIDMLP) for big data streams. *J. Big Data* **2019**, *6*, 103. [CrossRef]
24. Dhal, K.G.; Das, S. Local search-based dynamically adapted bat algorithm in image enhancement domain. *Int. J. Comput. Sci. Math.* **2020**, *11*, 1–28. [CrossRef]
25. Gupta, D.; Arora, J.; Agrawal, U.; Khanna, A.; de Albuquerque, V.H.C. Optimized Binary Bat algorithm for classification of white blood cells. *Measurement* **2019**, *143*, 180–190. [CrossRef]
26. Lu, Y.; Jiang, T. Bi-Population Based Discrete Bat Algorithm for the Low-Carbon Job Shop Scheduling Problem. *IEEE Access* **2019**, *7*, 14513–14522. [CrossRef]
27. Nakamura RY, M.; Pereira LA, M.; Rodrigues, D.; Costa KA, P.; Papa, J.P.; Yang, X.S. Binary bat algorithm for feature selection. In *Swarm Intelligence and Bio-Inspired Computation*; Elsevier: Amsterdam, The Netherlands, 2013; pp. 225–237.
28. Ebrahimpour, M.K.; Nezamabadi-Pour, H.; Eftekhari, M. CCFS: A cooperating coevolution technique for large scale feature selection on microarray datasets. *Comput. Biol. Chem.* **2018**, *73*, 171–178. [CrossRef]
29. Elaziz, M.A.; Ewees, A.A.; Neggaz, N.; Ibrahim, R.A.; Al-Qaness, M.A.; Lu, S. Cooperative meta-heuristic algorithms for global optimization problems. *Expert Syst. Appl.* **2021**, *176*, 114788. [CrossRef]
30. Karmakar, K.; Das, R.K.; Khatua, S. An ACO-based multi-objective optimization for cooperating VM placement in cloud data center. *J. Supercomput.* **2021**, *78*, 3093–3121. [CrossRef]
31. Li, H.; He, F.; Chen, Y.; Pan, Y. MLFS-CCDE: Multi-objective large-scale feature selection by cooperative coevolutionary differential evolution. *Memetic Comput.* **2021**, *13*, 1–18. [CrossRef]
32. Rashid, A.N.M.B.; Ahmed, M.; Sikos, L.F.; Haskell-Dowland, P. Cooperative co-evolution for feature selection in Big Data with random feature grouping. *J. Big Data* **2020**, *7*, 107. [CrossRef]
33. Valenzuela-Alcaraz, V.M.; Cosío-León, M.; Romero-Ocaño, A.D.; Brizuela, C.A. A cooperative coevolutionary algorithm approach to the no-wait job shop scheduling problem. *Expert Syst. Appl.* **2022**, *194*, 116498. [CrossRef]
34. Jarray, R.; Al-Dhaifallah, M.; Rezk, H.; Bouallègue, S. Parallel Cooperative Coevolutionary Grey Wolf Optimizer for Path Planning Problem of Unmanned Aerial Vehicles. *Sensors* **2022**, *22*, 1826. [CrossRef]
35. Jafarian, A.; Rabiee, M.; Tavana, M. A novel multi-objective co-evolutionary approach for supply chain gap analysis with consideration of uncertainties. *Int. J. Prod. Econ.* **2020**, *228*, 107852. [CrossRef]
36. Zhang, G.; Liu, B.; Wang, L.; Yu, D.; Xing, K. Distributed Co-Evolutionary Memetic Algorithm for Distributed Hybrid Differentiation Flowshop Scheduling Problem. *IEEE Trans. Evol. Comput.* **2022**, *26*, 1043–1057. [CrossRef]

37. Peng, X.; Jin, Y.; Wang, H. Multimodal Optimization Enhanced Cooperative Coevolution for Large-Scale Optimization. *IEEE Trans. Cybern.* **2018**, *49*, 3507–3520. [CrossRef]
38. Camacho-Vallejo, J.-F.; Garcia-Reyes, C. Co-evolutionary algorithms to solve hierarchized Steiner tree problems in telecommunication networks. *Appl. Soft Comput.* **2019**, *84*, 105718. [CrossRef]
39. Xue, X.; Pan, J.-S. A Compact Co-Evolutionary Algorithm for sensor ontology meta-matching. *Knowl. Inf. Syst.* **2017**, *56*, 335–353. [CrossRef]
40. Akinola, A. *Implicit Multi-Objective Coevolutionary Algorithm*; University of Guelph: Guelph, ON, Canada, 2019.
41. Costa, V.; Lourenço, N.; Machado, P. Coevolution of generative adversarial networks. In Proceedings of the International Conference on the Applications of Evolutionary Computation (Part of EvoStar), Leipzig, Germany, 24–26 April 2019; Springer: Berlin/Heidelberg, Germany, 2019.
42. Wen, Y.; Xu, H. A cooperative coevolution-based pittsburgh learning classifier system embedded with memetic feature selection. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011; IEEE: Piscataway, NJ, USA, 2011.
43. Bergh, F.V.D.; Engelbrecht, A. A Cooperative Approach to Particle Swarm Optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 225–239. [CrossRef]
44. Krohling, R.A.; Coelho, L.D.S. Coevolutionary Particle Swarm Optimization Using Gaussian Distribution for Solving Constrained Optimization Problems. *IEEE Trans. Syst. Man Cybern. Part B* **2006**, *36*, 1407–1416. [CrossRef]
45. Yang, Z.; Tang, K.; Yao, X. Large scale evolutionary optimization using cooperative coevolution. *Inf. Sci.* **2008**, *178*, 2985–2999. [CrossRef]
46. Goh, C.; Tan, K.; Liu, D.; Chiam, S. A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design. *Eur. J. Oper. Res.* **2010**, *202*, 42–54. [CrossRef]
47. Li, X.; Yao, X. Cooperatively Coevolving Particle Swarms for Large Scale Optimization. *IEEE Trans. Evol. Comput.* **2011**, *16*, 210–224. [CrossRef]
48. Jiao, L.; Wang, H.; Shang, R.; Liu, F. A co-evolutionary multi-objective optimization algorithm based on direction vectors. *Inf. Sci.* **2013**, *228*, 90–112. [CrossRef]
49. Wang, M.; Wang, X.; Wang, Y.; Wei, Z. An Adaptive Co-evolutionary Algorithm Based on Genotypic Diversity Measure. In Proceedings of the 2014 Tenth International Conference on Computational Intelligence and Security, Kunming, China, 15–16 November 2014; IEEE: Piscataway, NJ, USA, 2014.
50. Jiang, W.-Y.; Lin, Y.; Chen, M.; Yu, Y.-Y. A co-evolutionary improved multi-ant colony optimization for ship multiple and branch pipe route design. *Ocean Eng.* **2015**, *102*, 63–70. [CrossRef]
51. Pan, Q.-K. An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling. *Eur. J. Oper. Res.* **2016**, *250*, 702–714. [CrossRef]
52. Gong, M.; Li, H.; Luo, E.; Liu, J.; Liu, J. A Multiobjective Cooperative Coevolutionary Algorithm for Hyperspectral Sparse Unmixing. *IEEE Trans. Evol. Comput.* **2016**, *21*, 234–248. [CrossRef]
53. Atashpendar, A.; Dorronsoro, B.; Danoy, G.; Bouvry, P. A scalable parallel cooperative coevolutionary PSO algorithm for multi-objective optimization. *J. Parallel Distrib. Comput.* **2018**, *112*, 111–125. [CrossRef]
54. Jia, Y.-H.; Chen, W.-N.; Gu, T.; Zhang, H.; Yuan, H.-Q.; Kwong, S.; Zhang, J. Distributed Cooperative Co-Evolution With Adaptive Computing Resource Allocation for Large Scale Optimization. *IEEE Trans. Evol. Comput.* **2018**, *23*, 188–202. [CrossRef]
55. Yaman, A.; Mocanu, D.C.; Iacca, G.; Fletcher, G.; Pechenizkiy, M. Limited evaluation cooperative co-evolutionary differential evolution for large-scale neuroevolution. In Proceedings of the Genetic and Evolutionary Computation Conference, Kyoto, Japan, 15–19 July 2018; pp. 569–576. [CrossRef]
56. Sun, L.; Lin, L.; Gen, M.; Li, H. A Hybrid Cooperative Coevolution Algorithm for Fuzzy Flexible Job Shop Scheduling. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 1008–1022. [CrossRef]
57. Sun, W.; Wu, Y.; Lou, Q.; Yu, Y. A Cooperative Coevolution Algorithm for the Seru Production With Minimizing Makespan. *IEEE Access* **2019**, *7*, 5662–5670. [CrossRef]
58. Fu, G.; Wang, C.; Zhang, D.; Zhao, J.; Wang, H. A Multiobjective Particle Swarm Optimization Algorithm Based on Multipopulation Coevolution for Weapon-Target Assignment. *Math. Probl. Eng.* **2019**, *2019*, 1424590. [CrossRef]
59. Xiao, Q.-Z.; Zhong, J.; Feng, L.; Luo, L.; Lv, J. A Cooperative Coevolution Hyper-Heuristic Framework for Workflow Scheduling Problem. *IEEE Trans. Serv. Comput.* **2019**, *15*, 150–163. [CrossRef]
60. Derrac, J.; García, S.; Herrera, F. IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule. *Pattern Recognit.* **2010**, *43*, 2082–2105. [CrossRef]
61. Derrac, J.; García, S.; Herrera, F. A first study on the use of coevolutionary algorithms for instance and feature selection. In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, Salamanca, Spain, 10–12 June 2009; Springer: Berlin/Heidelberg, Germany, 2009.
62. Tian, J.; Li, M.; Chen, F. Dual-population based coevolutionary algorithm for designing RBFNN with feature selection. *Expert Syst. Appl.* **2010**, *37*, 6904–6918. [CrossRef]
63. Ding, W.-P.; Lin, C.-T.; Prasad, M.; Chen, S.-B.; Guan, Z.-J. Attribute Equilibrium Dominance Reduction Accelerator (DCCAEDR) Based on Distributed Coevolutionary Cloud and Its Application in Medical Records. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *46*, 384–400. [CrossRef]

64. Cheng, Y.; Zheng, Z.; Wang, J.; Yang, L.; Wan, S. Attribute Reduction Based on Genetic Algorithm for the Coevolution of Meteorological Data in the Industrial Internet of Things. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 3525347. [CrossRef]
65. Pawlak, Z. Rough sets. *Int. J. Comput. Inf. Sci.* **1982**, *11*, 341–356. [CrossRef]
66. Taguchi, G. System of Experimental Design; Engineering Methods to Optimize Quality and Minimize Costs. 1987. Available online: https://openlibrary.org/books/OL14475330M/System_of_experimental_design (accessed on 25 October 2022).
67. Conover, W. On a Better Method of Selecting Values of Input Variables for Computer Codes. 1975. Unpublished Manuscript. Available online: https://www.tandfonline.com/doi/abs/10.1080/00401706.2000.10485979 (accessed on 25 October 2022).
68. Hamdan, M.; Qudah, O. The initialization of evolutionary multi-objective optimization algorithms. In Proceedings of the International Conference in Swarm Intelligence, Beijing, China, 25–28 June 2015; Springer: Berlin/Heidelberg, Germany, 2015.
69. Ghareb, A.S.; Abu Bakar, A.; Hamdan, A.R. Hybrid feature selection based on enhanced genetic algorithm for text categorization. *Expert Syst. Appl.* **2016**, *49*, 31–47. [CrossRef]
70. Aghdam, M.H.; Heidari, S. Feature Selection Using Particle Swarm Optimization in Text Categorization. *J. Artif. Intell. Soft Comput. Res.* **2015**, *5*, 231–238. [CrossRef]
71. Abdul-Rahman, S.; Bakar, A.A.; Mohamed-Hussein, Z.-A. An Improved Particle Swarm Optimization via Velocity-Based Reinitialization for Feature Selection. In Proceedings of the International Conference on Soft Computing in Data Science, Putrajaya, Malaysia, 2–3 September 2015; Springer: Berlin/Heidelberg, Germany, 2015.
72. Rehman, A.; Javed, K.; Babri, H.A. Feature selection based on a normalized difference measure for text classification. *Inf. Process. Manag.* **2017**, *53*, 473–489. [CrossRef]
73. Paul, P.V.; Dhavachelvan, P.; Baskaran, R. A novel population initialization technique for genetic algorithm. In Proceedings of the 2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT), Nagercoil, India, 20–21 March 2013; IEEE: Piscataway, NJ, USA, 2013.
74. Zhai, Y.; Song, W.; Liu, X.; Liu, L.; Zhao, X. A chi-square statistics based feature selection method in text classification. In Proceedings of the 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 November 2018; IEEE: Piscataway, NJ, USA, 2018.
75. Ahmad, I.S.; Abu Bakar, A.; Yaakub, M.R. A review of feature selection in sentiment analysis using information gain and domain specific ontology. *Int. J. Adv. Comput. Res.* **2019**, *9*, 283–292. [CrossRef]
76. Algehyne, E.A.; Jibril, M.L.; Algehainy, N.A.; Alamri, O.A.; Alzahrani, A.K. Fuzzy Neural Network Expert System with an Improved Gini Index Random Forest-Based Feature Importance Measure Algorithm for Early Diagnosis of Breast Cancer in Saudi Arabia. *Big Data Cogn. Comput.* **2022**, *6*, 13. [CrossRef]
77. Thaseen, I.S.; Kumar, C.A.; Ahmad, A. Integrated Intrusion Detection Model Using Chi-Square Feature Selection and Ensemble of Classifiers. *Arab. J. Sci. Eng.* **2018**, *44*, 3357–3368. [CrossRef]
78. Chantar, H.K.; Corne, D.W. Feature subset selection for Arabic document categorization using BPSO-KNN. In Proceedings of the 2011 Third World Congress on Nature and Biologically Inspired Computing, Salamanca, Spain, 19–21 October 2011; IEEE: Piscataway, NJ, USA, 2011.
79. Ghareb, A.S.; Abu Bakar, A.; Al-Radaideh, Q.A.; Hamdan, A.R. Enhanced Filter Feature Selection Methods for Arabic Text Categorization. *Int. J. Inf. Retr. Res.* **2018**, *8*, 1–24. [CrossRef]
80. Adel, A.; Omar, N.; Abdullah, S.; Al-Shabi, A. Feature Selection Method Based on Statistics of Compound Words for Arabic Text Classification. *Int. Arab J. Inf. Technol.* **2019**, *16*, 178–185.