Document Clustering with Optimized Unsupervised Feature Selection and Centroid Allocation

Ibraheem Al-Jadir

This thesis is presented for the degree of Doctor of Philosophy of

Murdoch University, Perth, WA, Australia

2018

Declaration

I declare that this is my own account of my research and contains as its main content work that has not previously been submitted for a degree at any tertiary education institution.

Ibraheem Al-Jadir

Abstract

An effective document clustering system can significantly improve the tasks of document analysis, grouping, and retrieval. The performance of a document clustering system mainly depends on document preparation and allocation of cluster positions. As achieving optimal document clustering is a combinatorial NP-hard optimization problem, it becomes essential to utilize non-traditional methods to look for optimal or near-optimal solutions. During the allocation of cluster positions or the centroids allocation process, the extra text features that represent keywords in each document have an effect on the clustering results. A large number of features need to be reduced using dimensionality reduction techniques. Feature selection is an important step that can be used to reduce the redundant and inconsistent features. Due to a large number of the potential feature combinations, text feature selection is considered a complicated process.

The persistent drawbacks of the current text feature selection methods such as local optima and absence of class labels of features were addressed in this thesis. The supervised and unsupervised feature selection methods were investigated. To address the problems of optimizing the supervised feature selection methods so as to improve document clustering, memetic hybridization between filter and wrapper feature selection, known as Memetic Algorithm Feature Selection, was presented first. In order to deal with the unlabelled features, unsupervised feature selection method was also proposed. The proposed unsupervised feature selection method integrates Simulated Annealing to the global search using Differential Evolution. This combination also aims to combine the advantages of both the wrapper and filter methods in a memetic scheme but on an unsupervised basis. Two versions of this hybridization were proposed. The first was named Differential Evolution Simulated Annealing, which uses the standard mutation of Differential Evolution, and the second was named Dichotomous Differential Evolution Simulated Annealing, which used the dichotomous mutation of the differential evolution.

After feature selection two centroid allocation methods were proposed; the first is the combination of Chaotic Logistic Search and Discrete Differential Evolution global

search, which was named Differential Evolution Memetic Clustering (DEMC) and the second was based on using the Gradient search using the *k*-means as a local search with a modified Differential Harmony global Search. The resulting method was named Memetic Differential Harmony Search (MDHS). In order to intensify the exploitation aspect of MDHS, a binomial crossover was used with it. Finally, the improved method is named Crossover Memetic Differential Harmony Search (CMDHS). The test results using the F-measure, Average Distance of Document to Cluster (ADDC) and the non-parametric statistical tests showed the superiority of the CMDHS over the baseline methods, namely the HS, DHS, *k*-means and the MDHS.

The tests also show that CMDHS is better than the DEMC proposed earlier. Finally the proposed CMDHS was compared with two current state-of-the-art methods, namely a Krill Herd (KH) based centroid allocation method and an Artifice Bee Colony (ABC) based method, and found to outperform these two methods in most cases.

List of Publications Related to this Thesis

The following seven publications reported the results of this thesis.

C1. Al-Jadir, I., K. W. Wong, C. C. Fung and H. Xie (2017). Text Document Clustering Using Memetic Feature Selection. Proceedings of the 9th International Conference on Machine Learning and Computing. Singapore, Singapore, ACM**:** 415-420.

C2. Al-Jadir, I., K. W. Wong, C. C. Fung and H. Xie (2017). Text Dimensionality Reduction for Document Clustering Using Hybrid Memetic Feature Selection. Multi-disciplinary Trends in Artificial Intelligence: 11th International Workshop, MIWAI 2017, Gadong, Brunei, November 20-22, 2017, Proceedings. S. Phon-Amnuaisuk, S.-P. Ang and S.-Y. Lee. Cham, Springer International Publishing**:** 281-289.

C3. Al-Jadir, I., K. W. Wong, C. C. Fung and H. Xie (2017). Differential Evolution Memetic Document Clustering Using Chaotic Logistic Local Search. Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part I. D. Liu, S. Xie, Y. Li, D. Zhao and E.-S. M. El-Alfy. Cham, Springer International Publishing**:** 213-221.

C4. Al-Jadir, I., K. W. Wong, C. C. Fung and H. Xie (2018). Enhancing Digital Forensic Analysis using Memetic Algorithm Feature Selection Method for Document Clustering. IEEE International Conference on Systems, Man, and Cybernetics (SMC2018). Miyazaki, Japan 2018

C5. Al-Jadir, I., K. W. Wong, C. C. Fung and H. Xie (2017). Adaptive Memetic Differential Harmony Search for Optimising Document Clustering. Neural Information Processing: 24th International Conference, ICONIP 2018,

J6. Al-Jadir, I., K. W. Wong, C. C. Fung and H. Xie (2018). (under review) "Document Clustering using Crossover Memetic Differential Harmony Search, Submitted to the Memetic Computing Journal.

J7. Al-Jadir, I., K. W. Wong, C. C. Fung and H. Xie (2018). "Unsupervised Text Feature Reduction using Memetic-Based Feature Selection for Document Clustering" submitted to the International Journal of Artificial Intelligence (IJAI).

Summary of Publications with Respect to the Chapters and Aims of the Thesis

Chapter	Contributions	Paper No
Chapter 2: Literature Review and Theoretical Background	This chapter presents a comprehensive literature review regarding text pre-processing, supervised and unsupervised feature selection modes, optimization methods and hybrid evolutionary algorithms used for document clustering.	C1, C2, C3, C4, C5, J6, J7
Chapter 3: Document Clustering using Partitioning Methods with Supervised Feature Selection	Successfully developed memetic text feature selection method using a supervised wrapper-filter style using the k-nearest neighbor classifier (KNN) to reduce the extra text features. With the aid of using the original document class labels, the KNN can classify the features according to their accuracy.	C1, C2, C3
Chapter 4: An Unsupervised Feature Selection Using a Memetic Hybridization of a Wrapper and Filter Methods	In this chapter, a memetic text feature selection was also used, but this time the wrapper and filter components did not rely on the class labels. Using the unsupervised fitness function, Mean Absolute deviation (MAD) was helpful to handle the unlabeled features.	J6
Chapter 5: Document Clustering using Memetic Optimization	Successfully developed two distinct intelligent methods for the cluster centroid selection to enhance the performance of the traditional text document clustering using the reduced feature subsets.	C4, C5, J7

Abbreviations and Acronyms

Abbreviation & Acronym	Description
ABC	Artificial Bee Colony
AC	Absolute Cosine
ACDE	Automatic Clustering Differential Evolution
ACMDHS	Adaptive Crossover MDHS
ACO	Ant Colony Optimization
ADDC	Average Distance to Cluster Centroid
AI	Artificial Intelligence
BB	Branch and Bound
BIC	Bayesian Information Criterion
BCO	Bee Colony Optimization
BW	Bandwidth
CLS	Chaotic Logistic Search
CGABC	Chaotic Gradient Artificial Bee Colony
CMDHS	Crossover Memetic Differential Harmony Search
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DC	Document Clustering
DDE	Discrete Differential Evolution
DE	Differential Evolution
DEMC	Differential Evolution Memetic Clustering
DEKM	Differential Evolution K-Means
DDESA	Dichotomous DESA
DHS	Differential HS
DESA	Differential Evolution Simulated Annealing
EA	Evolutionary Algorithms
EP	Evolutionary Programming
FEF	Feature Evaluation Function
FS	Feature Selection
FSHSTC	Feature Selection using Harmony Search for Text Clustering
FSGATC	Genetic Algorithm for Improving Text Clustering
GA	Genetic Algorithm
GHS	Global-Best HS
GR	Gain Ratio
HM	Harmony Memory
HMCR	Harmony Memory Consideration Rate
HMS	Harmony Memory Size
HS	Harmony Search
IFCWR	Iterative Fuzzy C-means algorithm for Clustering of Web Results

IG	Information Gain
IGFSS	Improved Global FS Scheme
IG-PCA	Information Gain- Principle Component Analysis
IHS	Improved Harmony Search
KH	Krill Herd
LSI	Latent Semantic Indexing
MA	Memetic Algorithms
MAD	Mean Absolute Deviation
MAFS	Memetic Algorithm Feature Selection
MDHS	Memetic Differential Harmony Search
MFD	Maximum Features per Document
MFDR	Maximum Features per Document – Reduced
MI	Mutual Information
MRMR	Maximum Relevance Minimum Redundancy
MM	Mean-Median
NB	Naive Bayes
NP	Nondeterministic Polynomial
PAR	Pitch Adjustment Rate
PCA	Principal Component Analysis
PQ	Power Quality
PSO	Particle Swarm Optimization
PSOKHM	Particle Swarm Optimization K Harmonic Mean
Plus-l-Take-Away-r	PTA
QPSO	Quantum PSO
SBS	Sequential Backward Selection
SFBS	Sequential Floating Backward Selection
SFFS	Sequential Floating Forward Selection
SI	Swarm Intelligence
SSE	Sum of Square Error
SU	Symmetrical Uncertainty
SVM	Support Vector Machine
TC	Text Classification
TF.IDF	Term-Frequency Inverse Document Frequency
TV	Term Variance
VSM	Vector Space Model
WDC-NMA	Web Document Clustering based Niching Memetic Algorithm
WOA	Wale Optimization Algorithm

List of Tables

List of Figures

Contents

Acknowledgments

My sincerest thanks to my supervisor, Associate Professor Dr. Kevin Wong for his continuous support, patience, guidance throughout my research studies. His willingness to let me pursue my interests has allowed me to grow as a graduate student and as a researcher and to be able to deal logically with any scientific issue I may face in my future academic and research career. I would like also to thank my co-supervisors, Emeritus Professor Lance Fung, and Dr. Hong Xie for guiding me through my academic journey with my main supervisor.

I also would like to thank my wife for encouraging me to do my research. I could not have achieved any of this successfully without her love and continual giving. I would like also to thank my son Sadiq who was born at the end of my PhD; his smiles and companionship have changed my life and made me more committed to keep going forward. I would like to dedicate this work to him.

My sincere appreciation is extended to the Higher Committee for Education Development in Iraq for giving me this chance to undertake my doctoral studies at Murdoch University.

Chapter 1

Introduction

1.1 Overview

An effective document clustering system can significantly improve the tasks of document analysis, grouping, and retrieval. The performance of document clustering systems depends on document preparation and allocation of cluster positions (Zaw and Mon 2015). A document clustering system automatically organizes documents according to their content. This must be done in such a way that documents are more similar in one cluster than those belonging to other clusters (Mecca, Raunich et al. 2007). The process of document cluster centroids distribution in document clustering is considered as an unguided or unsupervised machine learning task (Dhillon, Mallela et al. 2003, Premalatha and Natarajan 2010).

Traditional clustering methods deal with documents as numeric vectors. Numeric vectors are clustered according to distance measures such as cosine or Euclidean distance criteria. These measures find the distance between each pair of document vectors and between each document vector and particular cluster center (centroid) (Jain 2010).

As the number of text documents increases, the process of allocating these documents to their right clusters becomes more complicated. This is especially more challenging in the current digital environment, given the huge amount of digital text available. In this context, traditional clustering methods that use distance measures might fail to perform the clustering optimally (Forsati, Keikha et al. 2015). In order to find optimal document clusters, it becomes necessary to apply optimization methods (Patil and Thakur 2016) capable of enhancing centroids allocation process. Consequently, using these methodologies in that context has become an active research area in the last few years (Forsati, Keikha et al. 2015, Abualigah, Khader et al. 2018).

Since the introduction of Evolutionary Algorithms (EA) (Hruschka, Campello et al. 2009), researchers have begun to use this branch of Artificial Intelligence (AI) to formulate document

clustering as a typical combinatorial optimization problem (Patil and Thakur 2016). EA methods have been used for document clustering, however due to their global search nature, it has become necessary to look for other variants to perform the local search in addition to performing the global search. Recently, Memetic Algorithms (MA) are a particular type of algorithm that belongs to the EA category, and can strike a balance in the search space between the exploration performed by the global search and the exploitation performed by the local search (Ning, Ong et al. 2003). Therefore, an investigation of MA algorithms to enhance the task of centroid allocation is justified for this present research.

Moreover, in every document each keyword corresponds to a feature. The distribution of features among documents can give an idea of categorization patterns in different documents (Ghareb, Bakar et al. 2016). The number of features selected has increased at the same rate as the volume of documents and is thus unwieldy. In this case, optimization methods can play a significant role to select the best features to enhance the document clustering process. The selection of best representative features will also help to reduce the complexity of centroids allocation processing in document clustering and to increase the accuracy of the resulted clusters (Hong, Lee et al. 2015).

It is important to mention that text feature selection for document clustering differs from many other feature selection methods used for Text Classification (TC) (Chandrashekar and Sahin 2014, Xue, Zhang et al. 2016). The latter relies on the classification accuracy of the chosen subset of features. Therefore, it is necessary to research ways to create a more efficient and automated unsupervised feature selection method to enhance the centroid allocation process that will improve the document clustering process. In that context MA was also applied for feature selection. However, the way that the MA applied to feature selection differs from that used with the centroids allocations. In feature selection the MA is a combination of both wrapper and filter methods (Abualigah, Khader et al. 2018). It is worthy to mention that two MA methods were proposed for feature selection in this thesis. One is to handle labelled documents while the other is to handle unlabelled documents. In summary, three main issues in this thesis were taken care by the MA which are the centroids allocation, supervised feature selection, and the unsupervised feature selection.

1.2 Text Document Clustering System

Typically, document clustering system has four major phases. These phases are all derived from a typical data clustering system, but they differ in techniques used to perform them. These phases are sequentially conducted (Forsati, Mahdavi et al. 2013, Zaw and Mon 2015). They are the pre-processing, feature selection, centroids allocation, and clusters evaluation (Patil and Thakur 2016). Figure. 1.1 shows the architecture of a typical document clustering system.

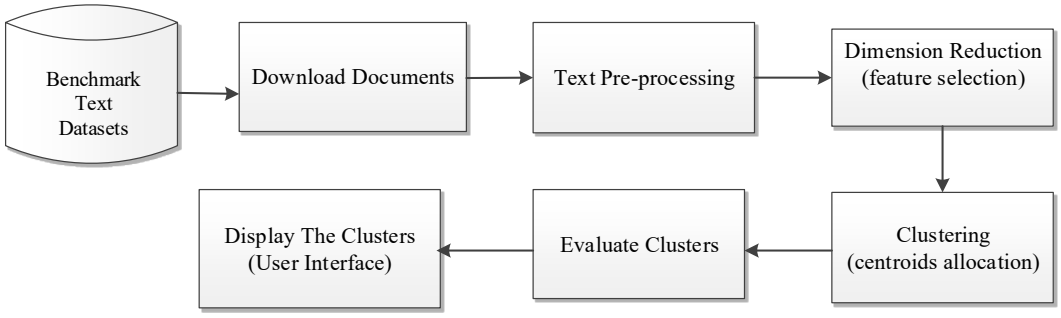

Figure 1. 1 General System Architecture

Despite the large number of research conducted on document clustering, there are still many challenges that need to be handled, in particular with the increasing number of documents and their features. To address the problems different researchers have proposed solutions (Forsati, Mahdavi et al. 2013, Forsati, Keikha et al. 2015, Song, Qiao et al. 2015) that will be discussed in the next chapter. The two most crucial parts in the document clustering process are feature selection, which is responsible in selecting representative, meaningful and non-redundant features, and the centroid allocation process (Jun, Park et al. 2014).

1.2.1 Document Pre-Processing

The text pre-processing phase is responsible for transforming text corpus into a structured format. Typically, each document will be converted into a numeric vector after this process (Tran, Vo et al. 2017). The pre-processing starts by analyzing the content of each document to generate a feature space referencing their original documents (Uysal and Gunal 2014). This phase involves tokenization, stop words removal, stemming, and term weighting. After tokenizing each word in each document, unnecessary words are excluded by comparing their

values with a standard stop words list in English. The stemming is then applied to combine the words that share the same root regardless of their conjugations. Finally, the weighting step is applied to find the numeric value corresponding to each keyword, and is usually calculated according to their co-occurrence within text corpus to produce weights (Ghareb, Bakar et al. 2016). The weighted keywords are named features. The stop words removal is insufficient to eliminate all the unimportant features, especially in large documents. Therefore, dimensionality techniques such as the feature selection and feature extraction techniques can be used. The main text pre-processing operations involved are text tokenization, stop word removal and stemming steps before the weighting step (Uysal and Gunal 2014).

1.2.2 Text Feature Selection

High dimensional feature space for text documents is associated with several problems such as the overfitting problem that occurs due to the usage of the entire existing features, which leads to all results returned as true positives. Also, inconsistency occurs when two documents have the same features even though they belong to two distinct groups. Last, the entire feature space leads to a degraded clustering performance as the computational complexity will increase (Zong, Wu et al. 2015).

Fundamentally, text feature selection is based on the same principles and ideas of data feature selection. Data and text feature selection can be classified into the filter, wrapper, or hybrid methods (Liu and Motoda 2012). Despite their simplicity, filter methods are incapable of selecting the optimal features on their own (Hua, Tembe et al. 2009). Several studies have reported that wrapper methods outperform filter methods in many cases, in particular when datasets become larger (Oreski, Oreski et al. 2016, Abualigah and Khader 2017).

Stochastic search methods have been used as successfully as wrapper methods (Liu and Yu 2005). In recent years, researchers have used those techniques in two different ways: supervised and unsupervised methods. The supervised method is more commonly tested, and is widely studied in the field of text categorization (Bharti and Singh 2014). Supervised feature selection depends on the availability of class labels, which are necessary for classifiers to use; the class labels are used to group features according to their classification accuracy. In contrast to the supervised method, unsupervised feature selection depends on measuring the relationship

between features without class labels, and therefore is an ideal feature selection used with document clustering in which class labels are not available (Dadaneh, Markid et al. 2016).

Genetic algorithm (GA) is an example of stochastic methods frequently used for feature selection. It has been used for feature selection in the two decades prior to this present study (Hong, Lee et al. 2015, Ghareb, Bakar et al. 2016). However, due to some drawbacks associated with the GA in terms of feature selection, such as parameter tuning and the random effect of initial population, GA has been improved in different ways, and other population-based methods are proposed (Ong, Lim et al. 2006). One of the most successful ways to enhance the performance of GA is to use MA (Radcliffe and Surry 1994). Recently, a large body of research has been devoted to further exploring the prospects of MA in feature selection. However, most of the MA research in text feature selection only addressed supervised feature selection (Kannan and Ramaraj 2010, Montazeri, Naji et al. 2013, Lee and Kim 2015).

1.2.3 Centroids Allocation

The process of centroids allocation means the distribution of cluster centers to their optimal position throughout the search (Forsati, Keikha et al. 2015). These centroids are distributed in the space according to their distance from their relevant documents, calculated using a distance function (Celebi 2015). The centroid allocation process can be performed using statistical methods such as *k*-means and its variants (Jain 2010). However, because the centroids allocation process is a combinatorial optimization problem, it becomes necessary to look for more sophisticated methods to find optimal or near-optimal solutions (Forsati, Mahdavi et al. 2013, Forsati, Keikha et al. 2015, Patil and Thakur 2016).

In order to enhance the distribution of centroids, optimization methods could also be utilized (Song, Qiao et al. 2015). A large number of optimization methods has been proposed and reused in various scientific and engineering problems, such as data and document clustering (Hruschka, Campello et al. 2009) (Nanda and Panda 2014). These methods are derivative-free methodologies, which means they are incapable of finding optimal solutions in the neighborhood of any particular region (Kramer, Ciaurri et al. 2011). MA has been used to enhance the task of centroids allocation.

A memetic-based clustering method named Web Document Clustering-based Niching Memetic Algorithm (WDC-NMA) was proposed in (Cobos, Montealegre et al. 2010). The *k*-means was used as a local search and hybridized with the GA that performs the global search. On the other hand, another web-based document clustering method called Iterative Fuzzy C-means algorithm for Clustering of Web Results (IFCWR) was proposed in (Cobos, Mendoza et al. 2013); it also used the memetic hybridization. The IFCWR selects the initial centroids using the Fuzzy C-Means algorithm; the Bayesian Information Criterion (BIC) was used as a fitness function. The limitation of both WDC-NMA and IFCWR is the use of the BIC classifier. The Bayes factors depend on prior assumptions about the distribution of cluster centroids. Nonetheless, there is no guarantee that BIC will be close to the first assumption. Therefore, the use of this criterion could mislead the process of centroids allocation.

In addition, several other global search methods other than GA could also be hybridized with the local search method used for an efficient centroids allocation for document clustering. For instance, other global search methods include the Ant Colony Optimization (ACO) (Saatchi and Hung 2005), Artificial Bees Colony (ABC) (Bharti and Singh 2016), Harmony Search (HS) (Forsati, Mahdavi et al. 2013), and Bee Colony Optimization (BSO) (Forsati, Keikha et al. 2015). Some of these hybrid methods are equivalent to the MA such as in (Saatchi and Hung 2005), (Forsati, Mahdavi et al. 2013) and (Forsati, Keikha et al. 2015). These methods will be discussed in more detail in the next chapter.

1.2.4 Evaluation Measures for Text Document Clustering

Evaluation measures are used to assess the quality of the resulting clusters. These measures can be divided into both external and internal evaluation measures. Internal measures are used to verify the degree of closeness of documents within every single cluster as a fitness function. For instance, the Average Distance to Cluster Centroid (ADDC) was one of the internal measures used to assess the compactness of clusters (Forsati, Mahdavi et al. 2013, Forsati, Keikha et al. 2015). It uses Euclidean distance or cosine distance. Such measures are called internal because they depend only on the intrinsic characteristics of clusters. External measures, however, use an external knowledge source to check the accuracy of the resulted clusters (Forsati, Mahdavi et al. 2013, Forsati, Keikha et al. 2015, Al-Jadir, Wong et al. 2017).

The objective of this present research was to develop an improved approach for text feature selection and optimally allocate the cluster centroids for an efficient document clustering system.

1.3 Problem Statement

The research scopes of both document clustering and feature selection are given in this chapter. It is necessary to provide an outline of achievements and contributions accomplished regarding these two problems in order to develop the currently available techniques and to enhance the performance further. A list of research questions realized after the literature review is as follows:

A. Missing Labels of Text Features

With the increasing number of irrelevant, redundant, inconsistent text features, the clustering quality could be degraded. The primary challenge of unlabeled data in feature selection is that no external knowledge source helps to predict the real distribution of features to guide the ranking process. The majority of the current techniques available for text feature selection deal with the supervised text classification problems (Xue, Zhang et al. 2016). Studies relating to semi-supervised and unsupervised text feature selection have been relatively sparse. Given the increasing number of digital documents, it is a challenge to obtain significant number of labelled information. Therefore, developing efficient unsupervised text feature selection method is increasingly important due to the fact that not all documents are labelled. Predicting the classes of these documents via their features is necessary in that case. Conducting unsupervised feature selection is more challenging than both supervised and semi-supervised feature selection as unsupervised technique is unguided by any extra knowledge sources i.e. the class labels. Unlike the supervised and the semi-supervised, the unsupervised feature selection is unbiased as there is no necessity to use an expert view or data labels for the feature categorization (Ang, Mirzal et al. 2016). In that case, in this thesis by using the unsupervised approach, is more compatible with the next stage of study which is the unsupervised centroids allocation for clustering the text documents.

B. Local Optima in Feature Selection

One of the problems associated with the limited number of methods that handle unsupervised text feature selection is that these methods might become stuck in the local optimum during the search. That result is because of global wrapper search methods (Abualigah and Khader 2017). Wrapper methods are capable of exploring feature space, but are incapable of exploring regions of interest within that space by performing only short leaps. It is therefore necessary to develop robust optimizing algorithms to assist the unsupervised feature selection process.

C. No Optimal Centroids Allocation Strategy

An optimization method is needed to enhance the distribution of centroids in document clustering. Although global search methods of optimization have been widely used instead of statistical clustering methods to improve the distribution for cluster centroids, hybrid methods such as memetic searches have been used by combining the global and local searches. The majority of the existing memetic search methods have used Gradient local search, which only searches within a narrow area within the search space (Abualigah, Khader et al. 2018). Other local search patterns such as the Chaotic search that has different search patterns, could help the local search to converge to more promising regions within the search space. It becomes important to develop a centroids allocation method that uses memetic optimization for local searches capable of performing a search over wider distances than the Gradient search.

D. Harmony Search Parameter Dependency

The calculation of the cluster centroids allocation is crucial in a document clustering system. The Harmony Search (HS) optimization has been applied successfully for that problem (Forsati, Mahdavi et al. 2013). However, the HS is considered a Bandwidth-dependent method. The Bandwidth parameter is a parameter used to modify solutions. This parameter could potentially affect the performance due to its higher sensitivity (Abedinpourshotorban, Hasan et al. 2016). Thus, a modified version of the HS that bypasses the need to use the Bandwidth parameter is important to perform the global search in the memetic optimization properly.

1.4 Aims and Objectives

The research presented in this thesis aims to enhance document clustering by efficiently allocating cluster centroids to their optimal positions. The research further improves the document clustering system by applying feature selection techniques that are capable of selecting a reduced feature space. In that sense, the concept of the memetic optimization, which hybridizes global and local search methods, is applied to enhance the performance of both feature selection and document clustering.

1.4.1 Improving Text Feature Selection

Feature selection methods are intended to reduce the extra text that affects the centroids allocation process with the following two sub-objectives:

1. To develop a supervised text feature selection method that combines the global search wrapper and a ranking method to reduce the original feature subsets into smaller feature spaces with an eliminated chance of falling into local optima.
2. To develop an unsupervised text feature selection method that selects a reduced feature space using a global search method and unsupervised local search. This method aims to resolve the problem of missing class labels in text features.

1.4.2 Improving Centroids Allocation

To develop two centroid allocation methods, which are Differential Evolution Memetic Clustering (DEMC) and the Crossover Memetic Differential Harmony Search (CMDHS), that address the following:

1. Regarding DEMC: it overcomes the problem of local search using the chaotic logistic search with randomicity and ergodicity properties. Using these two properties is important to exploit the search in the vicinity to the best solution.

2. Regarding the CMDHS: it overcomes the problem of bandwidth (BW) dependency in the existing harmony global search. This aims to reduce the

adverse effects of using the BW with existing methods. It aims to overcome the problem of the parameters' dependancy in the HS. Using the modified version of the HS only improves the control parameters of this method, but the local search problem remains. Therefore, this method aims to enrich the modified HS with a local search method that overcomes the local search problem.

3. To develop an adaptive version of the produced modified memetic HS document clustering CMDHS method in order to ensure that the values of these parameters are automatically set to their best values.

1.5 Contributions and Significance

It is expected that this thesis will contribute to more efficient document clustering by improving the text feature selection process before clustering, and by improving the distribution of the cluster centroids. The following contributions of this study are expected:

1. Unsupervised text feature selection methods are capable of ranking features without previous categorization. However, most previous research focused only on supervised-based feature selection for classification problems and paid less attention to unsupervised-based methods, especially with text. The present study will investigate the gap more completely.

2. Unsupervised text feature selection needs to be capable of skipping entrapment in local optima. The few studies that handled the unsupervised text feature selection focused on using wrapper schemes only, and these methods are more likely to be stuck in local optima. This study will cover this gap by using the hybrid memetic schemes.

3. The calculation of the centroids for text document clustering using the HS have been successful; still, the HS suffers from the effect of using bandwidth parameters on performance. The previous literature did not explore the prospect of using modified versions of HS in centroids allocation for document clustering. This study will propose a modified HS method that overcomes the problem of the native HS and another version of a modified differential

harmony search. These proposed methods can efficiently be used with centroid allocations in document clustering.

1.6 Thesis Outline

The thesis is organized as follows:

Chapter 2 covers the existing Document Clustering (DC) methods, supervised and unsupervised Feature Selection (FS) methods for text data. This chapter also highlights the main limitations and restrictions that exist in the DC, supervised and unsupervised FS methods.

Chapter 3 discusses the proposed supervised FS that uses the memetic wrapper-filter hybridization. It gives various comparisons of the proposed method with other existing methods, in order to highlight the performance of the proposed method with the other baseline and state-of-the-art methods.

Chapter 4 proposes the unsupervised FS that also uses memetic hybridization with comparisons of performance with other methods.

Chapter 5 presents the centroids allocation process of document clustering using the proposed methods. Two approaches are given: the DEMC and the MDHS, with two variants which are the Crossover MDHS (CMDHS) and the Adaptive Crossover MDHS (ACMDHS).

Chapter 6 presents the overall outcomes of the research and explains achievements obtained during the investigation. It also describes future work, which can be continued in the same domain.

Chapter 2

Literature Review and Theoretical Background

2.0 Overview

This chapter provides a literature review of the fundamental and background concepts that support the justification of the aims and objectives presented in Chapter 1. An overview of the theoretical background of Document Clustering (DC) and Feature Selection (FS) will be presented first, followed by a detailed explanation of optimization methods used to enhance the traditional techniques used for DC. Recent research advancements and trends in DC and FS areas will be described. This chapter will also highlight the drawbacks associated with FS and DC.

In this chapter, DC, supervised feature selection and unsupervised feature selection are reviewed critically, starting with DC as it is the main problem addressed by the present research. The use of traditional data clustering is first discussed. It is followed by examining feature selection of text data from the two different approaches of supervised and unsupervised methods. In this literature review, the main focus will be concerned with optimization methods used for both document clustering and feature selection.

2.1 Formal Description of Document Clustering

The field of document clustering is illustrated in Figure 2.1. Document clustering can be represented as a document corpus named D, such that $d_i \in D$ where d_i represents a particular document. The d_i document is transformed into a vector, v_i, which is composed of a number of components named weights. In other words, the document corpus can be formalized as a *2-D* matrix (M). This matrix has D rows (the same number as the documents number) and V columns where each m_{ij} is an element of Matrix M that represents the weight of the j^{th} feature of the i^{th} document. Thus, in this matrix, each row is a vector representing a document and each column represents a feature of that documents.

The clustering works on this matrix to find the most relevant documents and label them as one cluster (C)), where $c_i \subseteq C$. Thus, for a document subset C_1 where $C_1 \subseteq C$, it must have more relevant documents and be distinct from other subsets $C_n \subseteq C$. In this case, clustering aims to find the optimal representation by considering the minimal distance of documents within the same class and a maximum distance between documents located in different classes. Thus, the clustering objective is to find out the representation that has highest adequacy in relation to the large number of potential candidate solutions. That could be represented using Stirling number of second kind which is usually represented in the notion $S(n,k)$ where S represents the number of representations of the n objects into a nonempty clusters (k) (Sharp 1968). This representation shows the complicated nature of the clustering problem as can be seen in equation 2.1 where the solutions (groups) $S_i \in S$ are represented as $S = \{S^1, S^2, \ldots, S^{N(n,k)}\}$ and N(n,k) is calculated as follows in Equation 2.1, where k is the number of the clusters, n is the number of the document vectors and i is the index of a particular cluster.

$$N(n,k) = \frac{1}{k!}\sum_{i=1}^{k}(-1)^{k-i}\binom{k}{i}i^n \qquad \textit{Equation (2.1)}$$

It has been proven that the clustering is NP-hard optimization problem even with the cases where there is no more than two classes (Dasgupta 2008). This shows that the clustering by testing all possible solutions of n vectors of d-dimensions into any number of clusters is computationally infeasible. This problem is far more complex when the number of clusters becomes unknown. Then the number of combinations equals to the sum of the Stirling numbers of the second kind. Therefore, the optimization is used to reduce the search space. However, obtaining the optimal solution is not guaranteed (Forsati, Keikha et al. 2015).

2.2 Document Clustering System Using Traditional Methods

Document clustering is an important tool that plays a significant role in document archiving, organization, summarization, and retrieval. It is also one of the effective ways of knowledge management, via the categorization of text documents with minimal human intervention (Song, Qiao et al. 2015). It is a process of grouping documents into distinct clusters (Saiyad, Prajapati et al. 2016) in a way that similarities of documents within any single cluster are to be maximized while similarities between different cluster centers (centroids) are to be minimized (Feng 2007). The clustering problem can be categorized as an unsupervised machine learning

approach. The problem differs from supervised machine learning (classification), in that the latter depends on the previous categorization of documents and divides documents sets into training and testing subsets (Tang, He et al. 2016). On the other hand, clustering uses the entire document set. The clustering system predicts the best representation of documents according to their intrinsic properties (Rafi, Shahid et al. 2017).

Traditional clustering can be divided into hierarchical (Lee, Hsu et al. 2017), partitional (Nanda and Panda 2014), density-based (Geng, Li et al. 2018), and other clustering techniques. Partitional clustering allocates documents into *K* groups by minimizing the distance between objects within one group and maximizing it between objects located in other groups using an objective function that is capable of capturing a true notion of clustering, whereas hierarchical clustering creates a hierarchy of clusters by conducting a series of merge and split operations (Aggarwal and Zhai 2012).

Document clustering uses almost the same methodologies and techniques that are used for data clustering (Zaw and Mon 2015). However, traditional clustering, in general, suffers from several drawbacks, such as random centroid initial distribution and data variance caused by the usage of mean values to calculate the distance between objects. Moreover, the average-based centroid calculation is probably not the most efficient way to reflect the best representation of clusters. Finally, when the number of clusters increase, traditional methods such as *k*-means are not so capable of handling that increase (Liu, Li et al. 2012). Consequently, these drawbacks could potentially be propagated, in turn, to text document clustering as the text data will have the same data format as any other numeric dataset after transformation.

2.2.1 Hierarchical Methods

These methods recursively build cluster groups by grouping documents (or any objects) in top-down or bottom-up fashion. In turn, the top-down and bottom-up approaches can be classified further into agglomerative hierarchical clustering and divisive hierarchical clustering. In the agglomerative hierarchical clustering, each document is considered as a separate cluster, and then single clusters are successively combined until the desired number of clusters is reached (Lee, Hsu et al. 2017). In divisive hierarchical clustering, all documents are treated as if they belong to a unique supercluster. This cluster is then broken up into smaller clusters and

repeatedly subdivided until the desired number of clusters is generated (Zhao and Karypis 2002).

Dissimilarity measures in hierarchical clustering determine both merge and split operations. These measures are selected to optimize some criteria (e.g., sum of squares). Hierarchical clustering methods can also be classified according to the way these dissimilarity measures are computed. For instance, single-link clustering (Murtagh and Contreras 2017) (also named connectedness, minimum or nearest neighbor methods) considers the path running between any two cluster centers to be equal to the shortest path between any object of one cluster to any other object of another cluster. Unlike single-link clustering, complete-link clustering (also named maximum, diameter or further neighbor methods), considers a path running between any two cluster centers to be equal to the longest path between any object of one cluster to any other object of another cluster (Pal and Bhattacherjee 2018). Besides both single and complete-link methods, the average-link clustering method, also known as the minimum variance method, considers the path linking any two cluster centers to be equal to the average path between an object of one cluster to any other object of another cluster (Murtagh and Contreras 2017).

Both single-link and average-link methods have several drawbacks. The single-link suffers from a problem called the chaining effect. This happens when the single-link can produce straggling clusters. As the merging criterion is strictly local, a chain of points could be prolonged for wider distances with no consideration to the final shape of the constructed cluster.

On the other hand, in average-link clustering, the distance between any two clusters is calculated by taking the average distance of each point in one cluster to each point in another cluster. The general problems with all types of hierarchical methods of clustering is that hierarchical methods are incapable of scaling up, because of the time complexity associated with it. Regarding document centroids allocation, the non-linear relationship with the number of documents might potentially lead to the need of more computational requirements (Forsati, Mahdavi et al. 2013).

2.2.2 Partitional Methods

Partitional clustering methods reallocate documents (or any other objects) by rearranging them in different clusters, beginning with a random initial distribution of cluster centroids. These methods ideally need a predefined number of cluster centers. One of the main aims of this thesis is to optimize partitional document clustering; the main focus is dedicated to exploring methods used to optimize the traditional methods of clustering. As mentioned earlier, partitional clustering divides documents into distinct clusters depending on some distance functions. It can further be categorized into traditional and non-traditional approaches (Figure 2.1). Traditional-based clustering uses statistical methods and similarity distance measures to find the distance between documents and their corresponding central points (centroids) (Rafi, Shahid et al. 2017). Error minimization methods are first used as partitional clustering methods. These methods are based on the idea of finding the least error after using particular minimization criteria, which are usually distance-based. The widely used criterion is the Sum of Square Error (SSE) using the Euclidian distances of document vectors to their corresponding clusters. The SSE value might be globally optimized via the use of the exhaustive enumeration of all clusters, an inefficient process. As another enumeration process used in error minimization methods is by setting approximated solutions (not mandatorily leading to global minima) using some heuristics such as *k*-means and its variants (Deelers and Auwatanamongkol 2007).

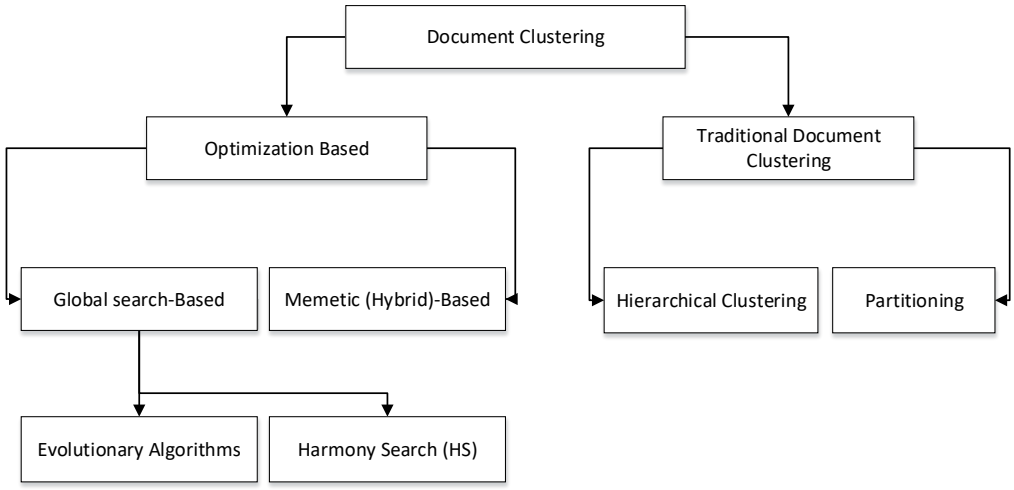

Figure 2.1 Clustering Hierarchy

The *k*-means approach has been widely used in data clustering and later in document clustering because of its simplicity and easy implementation (Jain 2010). The *k*-means can optimize an objective function in a two-step procedure. Often, documents are transformed into vectors that contain weights of terms in that document. By using term-frequency or term-frequency-inverse-document-frequency weighting schemes, different objective functions of clustering can be optimized (Zhao and Karypis 2001). The objective functions quantify cluster properties such as compactness, separability, or both. The *k*-means is capable of maximizing the similarity of documents within the same cluster documents in a reliable way.

In (Farnstrom and Lewis 2008) a comparison was made of scalable *k*-means, complete *k*-means, and traditional *k*-means. The comparison outcomes showed that traditional *k*-means outperformed other methods. In (Bouras and Tsogkas 2010), the authors tested hierarchical clustering using single, maximum, link and centroid link. The *k*-means, *k*-medians, and *k*-means++ were also tested. The results revealed that clustering using *k*-means outperformed other methods in terms of the internal measurement index. Moreover, the results showed the superiority of *k*-means using the external evaluation measurement. Additionally, in (Jo 2009), a comparison was made which involved *k*-means and single-pass hierarchical clustering and some other methods on text news datasets. The authors showed that *k*-means performed better than hierarchical clustering, which used the single-pass method.

Still, solutions from the *k*-means approach and its variants may not be optimized and remain stuck in local optima for several reasons, (Forsati, Mahdavi et al. 2013) such as random centroid initial distribution and data variance that occurs because of average values used to calculate the distance between objects. Moreover, the average-based centroid calculation is not an efficient way to reflect the best representation of clusters. Finally, when the number of clusters increases, *k*-means is not capable of handling the increase efficiently (Abualigah, Khader et al. 2018).

K-medoid is a variant of the *k*-means. Although its main idea is similar to that of *k*-means, the main difference between them is that *k*-medoid clustering uses the closest documents to the cluster centroid while *k*-means uses the average value of document vectors in order to calculate cluster centroids (Nanda and Panda 2014). A *k*-medoid clustering algorithm is capable of effectively enhancing the performance of the clustering, but two drawbacks are associated with its performance. First, a large number of iterations is required to converge due to the large

number of distance measure computations. Second, the *k*-medoid clustering method is incapable of handling sparsity in text data (Aggarwal and Zhai 2012).

2.2.3 Density-Based Clustering

A density-based clustering is another example of traditional clustering. It assumes that the documents (objects) belonging to any particular cluster are the result of a specific distribution of probability. The overall distribution is considered as a combination of many single distributions. The objective of density-based clustering is to find the clusters and their associated distributional probabilities. Such methods are designated to discover the clusters of random shapes, but not limited to finding those residing in convex areas. Density-based clustering suffers from several drawbacks. For example, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering method, despite its ability for noise-resistance and high accuracy, still suffers from time complexity. Additionally, its threshold parameters can be difficult to set up (Chen, Yuen et al. 2017).

Subsequent sections will explore the techniques proposed in the literature to enhance document clustering using optimization methods such as evolutionary and memetic optimization methods. A broad spectrum of evolutionary-based methods has been proposed. Such techniques have been used in various scientific and engineering problems (Hruschka, Campello et al. 2009, Jensi and Jiji 2014, Saiyad, Prajapati et al. 2016). The use of these methods has been noted to obtain better clustering results not only with text, but also with other data types such as microarray data and image processing. However, this thesis only focuses on text data and text documents processing. Due to the similarities existing among the methods used with other data types, some of those optimization methods will also be reviewed. The use of optimization methods could make the clustering process more efficient via the best distribution of cluster centers in the search space (Forsati, Keikha et al. 2015). The following sections will focus on studies conducted to select the best representation of cluster centroids using different optimization techniques, especially the memetic-based techniques.

2.3 Optimization Methods for Centroid Allocation in Document Clustering

Optimization methods are used as an alternative to traditional methods as explained in the last section for document clustering. These methods can perform the clustering in a non-traditional manner when the centroids are calculated and allocated. EA, Swarm Intelligence (SI) or HS are examples of those methods (Feng 2007, Kohli and Mehrotra 2016, Patil and Thakur 2016, Daoud, Sallam et al. 2017). Mainly, they are used for the selection of the best centroid positions with respect to their surrounding documents (Kohli and Mehrotra 2016). The main reason behind for using these methods is their ability to choose the most optimal solution among many options.

After modelling the centroids distribution as an optimization problem, these optimization methods repeatedly iterate to obtain the optimal solution. Each solution contains a number of cluster centroids, which are calculated as a function of distance from their current location to the surrounding documents. These centroids are repositioned according to the fitness function used by the optimization methods. The common denominator among the optimization methods used in centroids allocation for document clustering is the effect of their parameters on the performance. Moreover, these methods may fall in local minima due to their deficiency in the local search. The global search refers to the search to the best individual solution or set of solutions while the local search is responsible of modifying only one single solution (Nanda and Panda 2014, Joyce and Herrmann 2018).

As a result, these methods are confronted with the problem of premature convergence that downgrades the accuracy of the global search. The premature convergence problem might be the result of the poor distribution of initial solutions (Jensi and Jiji 2014).

2.3.1 Global-Based Centroid Allocation in Document Clustering

The first category of the optimization-based methods includes those methods that perform the global search. They are classified as follows:

2.3.1.1 Evolutionary-Based Methods

These methods are inspired by either natural phenomenon and/or natural processes (Nanda and Panda 2014). EA methods incorporate a number of algorithms such as GA, Evolutionary Strategies (ES) (Beyer and Schwefel 2002), Evolutionary Programming (EP) and Differential

Evolution (DE). From these many variations, the main focus will be on the GA-based and the DE-based centroids allocation methods, it is because the GA-based and the DE-based are most used algorithms in this category.

A. Genetic Algorithm for Centroids Allocation

GA is considered as a baseline evolutionary method. It was inspired by Darwin's concepts of evolution and natural selection. It starts with an initial population of solutions (individuals) thriving for survival. The fittest individuals survive to the next generations. The older population passes merits of survival to successor generations by different means of evolution through using genetic crossover and mutation operators. The evolution process iterates until a specified number of generations is reached, with the evolutionary goal being the best solution to minimize or maximize the fitness function that determines survival (Goldberg and Holland 1988).

In document clustering, the same concepts presented by Goldberg in his original paper can also be applied. For instance, since Ravindra Krovi (Krovi 1992) first studied the possibility of obtaining promising clusters centroids by GA algorithms, optimization methods for clustering (mostly GA) have continued to develop. The research proposed in (Bezdek, Boggavarapu et al. 1994) was a leading study to incorporate genetic-based searching for the optimization of centroids allocation of the clustering problems. Later, GA was used for the document clustering problem in (Casillas, de Lena et al. 2003), the binary encoding scheme represented the population with a predefined number of centroids. Uniform crossover and cluster-orientated mutations were used to modify the population (alternating the order of bits in each solution).

After many years, in (Song and Park 2009), another GA-based clustering system that uses a gene index to encode chromosomes in a semantic space was proposed. Gene index refers to the position of every single gene in the chromosome. As for text document centroids allocation, the gene index should point out the right positions for them. Latent Semantic Indexing (LSI) was used to find the minimum features of text. In (Karaa, Ashour et al. 2016) a document clustering approach was devised to handle text data. This approach was based on GA and agglomerative clustering. Despite the success of the document clustering systems using the genetic search, just mentioned, the genetic-based centroids allocation has a deficiency: they fall in local optima (Song and Park 2009), which makes it more likely for performance

impairment, in comparison to some other modified versions and other meta-heuristic methods in numerical optimization.

B. Differential Evolution for Centroids Allocation

DE (Storn and Price 1997) is another EA method. It has been reported that it outperformed GAs in terms of the highest scores achieved by their fitness function in different problems (Das, Mullick et al. 2016). The DE is simple, robust, and converges fast. In addition, it has few parameters to tune, and the same settings could be utilized for different problems. DE has shown its worth in real-world problems, and in (Vesterstrom and Thomsen 2004) it outperformed PSO and EA in the majority of numerical benchmark problems. Among the tested algorithms, DE could be an optimal choice when confronted with a new optimization problem to resolve. In order to understand how DE is used for document clustering, its main fundamentals are described in this section. DE is a population-based evolutionary optimization method. It is considered a modification to GA. The main steps of DE can be summarized as below:

A. The initialization phase: A population consists of S solutions, where population size is S and the size of each solution is N. The population in differential evolutions can be represented by $S*N$ matrix P. Each row of matrix P is a solution. We use the notion x_i to present i^{th} row of P.

B. Population update: the mutation is first used to create a new trial solution v through the addition of the weighted difference of a randomly chosen pair of solutions to a third one. The operation is illustrated in Equation 2.2.

$$v = x_{r1} + F \times (x_{r2} - x_{r3}) \qquad i=1,2,\ldots,S \quad \textit{Equation (2.2)}$$

where row numbers r_1, r_2 and r_3 are three randomly selected numbers between 1 and S and these three numbers, differ from each other. x_{r1}, x_{r2}, and x_{r3} are r_1^{th}, r_2^{th} and r_3^{th} solutions selected from P respectively. F, which is between 0 and 1, is a scaling factor that modifies the difference between solution x_{r2}, and solution x_{r3}.

The crossover is applied later to diversify the population by the perturbation of the current population. The crossover in DE is performed as shown in

Equation 2.3. The perturbation is carried out in accordance with a specific probability $Cr \in [0, 1]$.

$$u_j = \begin{cases} v_j & rand(0,1) \le CR_i \\ x_k & otherwise \end{cases} \quad j=1,2...,N \qquad \textit{Equation (2.3)}$$

where k is the index of the best solution from the current population and *rand* (0,1) is a random number between 0 and 1. At the beginning of each iteration, the best solution x_k, from the current population is found, and is used to create vector u using the above equation. The target solution is x_k , x_i is the mutant vector obtained from Equation 2.4, and Cr is the crossover probability. The number of solutions is S and N is the number of documents in each solution.

C. Selection: the target solution (x_k) is then compared with vector u, and both x_k and u are evaluated; the one that obtains a better fitness value is transferred to the next generation. Equation 2.4 shows the selection operation.

$$x_i = \begin{cases} u & f(u) \le f(x_k) \\ x_k & otherwise \end{cases} \qquad \textit{Equation (2.4)}$$

The DE was used in (Abraham, Das et al. 2006) in document clustering for the centroids allocation process, and it outperformed both Particle Swarm Optimization (PSO) and genetic-based clustering methods. That study was the first study to incorporate DE explicitly for clustering allocation problem. However, in (Das, Abraham et al. 2008) a modified version of the DE was proposed. To enhance the convergence of DE, the authors intended to modify the scaling parameter F (shown in Equation 2). In normal cases, the F parameter is usually selected between [0, 1] as a random number. In contrast, the authors in (Das, Abraham et al. 2008) used a different way to generate this parameter, which is shown in Equation 2.5. Equation 2.5 is simply given a systematic way to generate this parameter.

$$F = 0.5 \times (1 + rand(0,1)) \qquad \textit{Equation (2.5)}$$

This allows for stochastic variations in the amplification of the difference vector and therefore that will preserve the population diversity while the search advances. This method performed the clustering using an automatically determined number of clusters centroids with an unlabeled document set. Moreover, it showed that DE could obtain promising clustering results with negligible control parameters. This modified DE method was proposed to improve its convergence, and named the Automatic Clustering Differential Evolution (ACDE) algorithm.

DE mimics the GA in its search global nature. Therefore, it has been modified further. In recent years, to intensity its local search capability. It has been modified in two ways:

A. Using it as a global search and combine it with a local search (Peng, Zhang et al. 2017) which is a key candidate solution to boost its performance in different optimization problems.
B. The second way is by combining it with another global search such as the Differential HS (DHS). For instance, the differential operators are used in the HS (Abedinpourshotorban, Hasan et al. 2016).
C. Using both 2 previous approaches mentioned in points one and two.

Thus, in order to achieve the second and third modifications of the DE, in the next section we discover the strength points of the HS to aid with optimizing the performance of the DE for better document clustering.

2.3.1.2 Swarm Intelligence- Based Methods

Swarm Intelligence (SI) is another category of global search methods. However, their performance is similar to that of the GA or DE. The SI is the property of a system where the overall behavior of particles (members) that interact in a local manner with their environment results in a coherent functional global search pattern. Unlike EAs, SI algorithms are inspired initially from basic behavioral actions and self-organizing interaction among the swarm members, such as ant colony foraging, bird flocks, animal herds, bacteria foraging and division, honey bees colonies, fish schools, etc. (Mavrovouniotis, Li et al. 2017).

The terminology of SI was first time used by Beni in (Beni 1988) in cellular robotics systems where a self-organization process is conducted among simple agents using neighborhood interactions. These methods do not need constrains on the objective functions such as the continuity or differentiability. Thus, these methods are good candidates to solve different

optimization problems (Rezaee Jordehi, Jasni et al. 2015). Moreover, these methods have a tendency to obtain good quality solutions and also they are capable of handling a large and complex set of NP-hard optimization problems efficiently. It is also noticed that they have a neighboring structure which directs the particles (solutions) in directions that lead to the optimality more than the classical methods (kumar and Sahoo 2017).

For that reason, SI methods have been widely used for the document clustering problem. Some examples of this category are PSO (Karol and Mangat 2013), ACO (Aghdam, Ghasem-Aghaee et al. 2009), Cuckoo Cuckoo Search via Lévy Flight (Yang and Deb 2009), Bee Colony Optimization (BCO) (Bui, Bui et al. 2017), Artificial Bee Colony (ABC) (Xue, Jiang et al. 2017), Krill Herd (KH) (Abualigah, Khader et al. 2016, Abualigah, Khader et al. 2017).

As for the centroids allocation process in document clustering, PSO was first used with document clustering in (Cui, Potok et al. 2005). Later in (Premalatha and Natarajan 2010) it was combined with the generic algorithm to improve the diversity of PSO. In (Song, Qiao et al. 2015) a hybrid method that combines the GA and Quantum PSO (QPSO) that is named (GQPSO) was proposed for document clustering. The GA was used to initialize the population of the QPSO. More specifically, GA was used as a first line optimizer to generate the initial population used later by the QPSO. In this system, the use of the GA and QPSO as two global searchers could be associated with two problems. First using two global searchers is associated with a doubled system complexity, and also both methods would still be suffering from the local search deficiency. In general, PSO in its standard form has a weakness in the non-oscillatory route that can quickly make the particles to stagnate and that may lead to premature convergence on suboptimal solutions.

The ACO method was also used for document clustering as an SI optimizer. For example, in (He, Hui et al. 2006) the ACO was used for an efficient centroids allocation method, this method does not depend on a 2D grid structure. Even though the convergence is guaranteed, still the time to converge to the optimality is uncertain. On the other hand, in (Zaw and Mon 2015) the cuckoo search via Lévy Flight was applied to the document clustering in order to find optimal solutions. Cuckoo search via Lévy flights is based on the obligate brood parasitic behavior of cuckoo birds in combination with Lévy flight of some birds. Still, this method lacks to enough mathematical and theoretical background. The KH optimization was also used for document clustering (Abualigah, Khader et al. 2016). Later, it was discovered that the original KH algorithm is incapable of guaranteeing to reach to optimality. More recently, KH was

modified by using the genetic operators. Authors claimed that using the genetic operators insignificantly enhance the global search capability in the basic KH. Therefore, the authors used a hybrid fitness function that is based on the global best concept to improve the performance of KH in (Abualigah, Khader et al. 2018).

The ABC method (Karaboga and Ozturk 2011), is another variant to the PSO, and is is based on the synergy of intelligent foraging of bees. It is one of the popular methods for numeric optimization. It was used for centroid allocation clustering problems in (Karaboga and Ozturk 2011). As with other SI methods, ABC starts with a random initial population. The exploration of the search space begins in different directions to detect global optimum solutions. Candidate solutions of honey bees are food sources. Each solution is updated by re-positioning in order to diversify the exploration of the search space. Still, the ABC as used in experiments by (Kumar, Sharma et al. 2014) is quite inefficient for exploiting the obtained solution.

2.3.1.3 Harmony Search-Based Methods

Since its discovery, the HS has been successfully used for a large number of optimization problems. HS could be perceived as a simple real-coded GA, as it has almost the same distinctive features of GAs such as mutation, crossover, and selection. In order to understand how HS has been used with centroids allocation, we need to first understand its fundamentals. HS was proposed first by Geem (Geem, Kim et al. 2001) as an optimization method. It belongs to EAs. The population in HS is represented as a set of harmonies stored in a data structure, such as a matrix, called Harmony Memory (HM), and each harmony represents one solution. The method uses the following parameters: Harmony Memory Size (HMS) that is the number of solutions in the HM, Harmony Memory Consideration Rate (HMCR) that controls the selection of the solutions from HM, and Pitch Adjustment Rate (PAR) that resembles the mutation in the GA. Furthermore, a BW parameter is used to modify the harmonies. The optimal value of BW is still unknown, despite much previous research. It is important to note that the terminology of harmony and solution is interchangeable. There are four main phases in the standard HS optimization which are listed below:

A. Initialization of the HM. The method sets a predefined number of clusters, *c*. HM contains a list of potential solutions, which is normally represented by a matrix as shown in Figure 2.2. Each row of the matrix is a solution which includes an assignment of documents to cluster numbers. For instance, if there are six clusters,

values in a solution will be between 1 and 6. The length of each solution is the same as the number of the documents while the number of solutions in HM, or HMS, can be set to any number, normally to twice the number of clusters.

$$\mathrm{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{\mathrm{HMS}-1} & x_2^{\mathrm{HMS}-1} & \cdots & x_{N-1}^{\mathrm{HMS}-1} & x_N^{\mathrm{HMS}-1} \\ x_1^{\mathrm{HMS}} & x_2^{\mathrm{HMS}} & \cdots & x_{N-1}^{\mathrm{HMS}} & x_N^{\mathrm{HMS}} \end{bmatrix}$$

Figure 2.2 Harmony Memory (Forsati, Mahdavi et al. 2013)

The initialization of HM is performed using Equation (2.6), which randomly sets the initial population in a particular range.

$$x_j^i = rand(0,1) \times (c-1) + 1$$

$$, \ i = 1, 2, 3, \ldots, HMS \text{ and } j = 1, 2, 3, \ldots, N \qquad Equation \ (2.6)$$

where c is the predefined number of clusters and x_j^i represents the assignment of a cluster index.

In equation 2.6, the new solution is generated by randomly selecting numbers between zero and one. In equation 2.6 the random number is multiplied by the number of the clusters represented by (c-1).

B. Harmony memory improvising. This step creates new solutions by modifying the existing solutions in HM, as shown in Algorithm 2.1.

C. The next step is to update HM. A comparison is done by checking the fitness value of an improvised solution with the older one. If the fitness value of the newly improvised solution is higher than the older one, the newly updated solution will replace the older one.

D. The termination condition is satisfied when the maximum number of iterations is reached, or no further improvement is observed.

Algorithm 2.1: Improvising the population

1. for $i = 1$ to HMS do
2. for $j = 1$ to N do
3. if $rand\ (0, 1) \leq HMCR$ then
4. $x = HM(i, j);$
5. if $rand\ (0, 1) \leq PAR$ then
6. $x = x + rand(0, 1) \times BW$
7. else
8. $x = x - rand(0, 1) \times BW$
9. end if
10. *else*
11. $x = rand(0,1)*(c-1) + 1;$
12. end if
13. $HM(i, j) = x$;
14. end for
15. end for

In algorithm 2.1, the HMS represents the first population while N represents the size of each member (harmony) in the population. Therefore, the first two loops determine the harmonies number and the size of each harmony. The rest of the algorithm describes the way that each element in each harmony represented by x is created using and adding that element to its position in the Harmony Memory (HM).

2.3.2 Memetic Optimization for Centroid Allocation in Document Clustering

As an improvement to EA optimization methods that performs only the global search, MA have been proposed to merge the advantages of the EAs with local search methods. The global search method is greatly improved when combined with the local search method; this is known as MA. Such a combination has been successfully applied to global optimization of numerical functions (Nguyen, Ong et al. 2009) and it has been utilized to solve many real-world optimization problems (Lee and Kim 2015). The workability of the memetic optimization in its simplest form is shown in Figure 2.3.

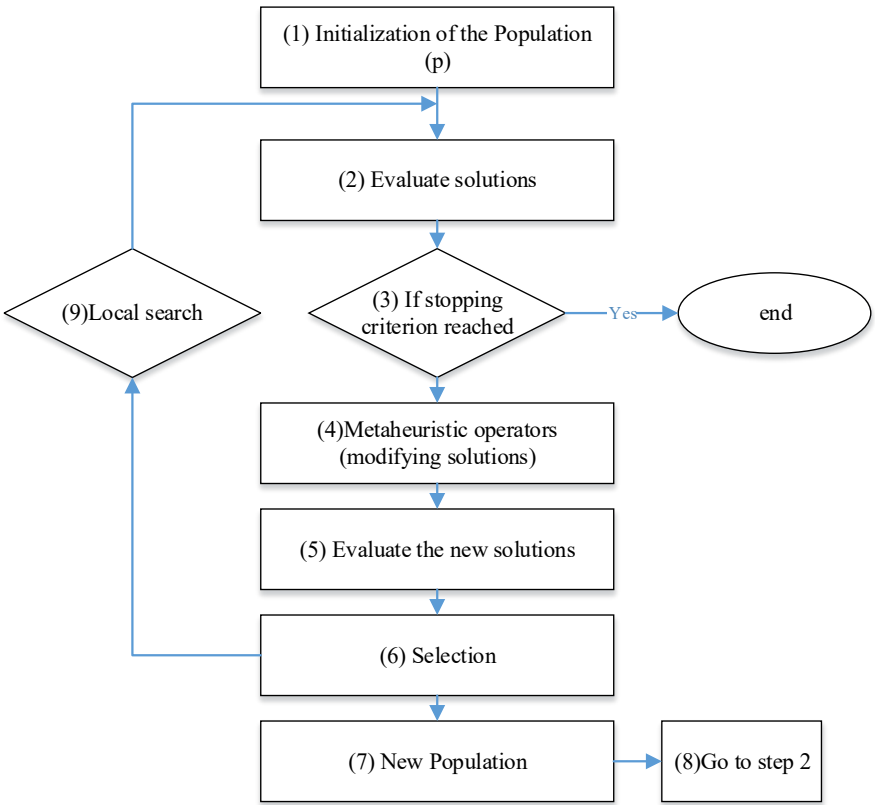

Figure 2.3 Local Search in Memetic Optimization

The term ‘memetic' was used first by Dawkins (Amaya, Porras et al. 2015) to refer to a gene counterpart in terms of cultural evolution in the GA. Unlike EA, MA is concerned with the exploitation of search space; however, this is unfortunately not covered by EAs. On the other hand, MA are population-based algorithms; they maintain a solution set of the problem that mimics the population set in the evolutionary algorithm. Each solution is named as an individual in evolutionary algorithm terminology, whereas they are called agents in MA (Neri and Cotta 2012).

In memetic optimization, the use of the local search is an efficient way to enhance the performance of the global search (Aarts and Laarhoven 1989). The local search starts from a single solution; this solution will be improved until no further optimization makes a difference to the previous versions of that solution. In other words, it stops when the solution is stagnated in one specific area in the search space through iterations (Bolaji, Al-Betar et al. 2016). Some of the well-known local search methods in the clustering or the document clustering problems are simulated annealing (Kirkpatrick, Gelatt et al. 1983), Chaotic Logistic Search (CLS) (Choi and Lee 1998), and gradient-based search methods such as the *k*-means and its variants (Jain

2010). The integration between global search methods with local search methods is highly recommended by researchers, in order to handle various combinatorial optimization problems (Patil and Thakur 2016). The next subsections shows the modification of the previously explained meta-heuristic methods.

2.3.2.1 Memetic Evolutionary-Based Methods

The GA was first used as an evolutionary global search method in a memetic-based clustering method named Web Document Clustering, which the researchers (Cobos, Montealegre et al. 2010) based on a new Niching Memetic Algorithm, Term-Document Matrix and Bayesian Information Criterion (WDC-NMA). The *k*-means was used as a local search and hybridized with the genetic search that performs the global search for the document clustering, whereas the BIC was used as a fitness function. Another web-based document clustering method, the Iterative Fuzzy C-means algorithm for Clustering of Web Results (IFCWR) was proposed in (Cobos, Mendoza et al. 2013), which also used memetic hybridization. The IFCWR method selects the initial centroids using the Fuzzy C-Means algorithm, and it also used the BIC fitness function.

As is described earlier, the DE outperformed GA on different occasions in terms of the centroids allocation, but its global search nature is still dominant. Thus, different memetic techniques have been developed to enhance performance of DE. For example, in (Reynoso-Meza, Sanchis et al. 2011) the DE was integrated with a sequential quadratic programming local search method (Boggs and Tolle 1995) while in (Poikolainen and Neri 2013), the Hooke–Jeeves local search was integrated with the DE global search. In (Poikolainen and Neri 2013) the local search was used to generate the initial population, to reduce the randomness of the first generation.

A distributed memetic that also used the Hooke–Jeeves-based DE, controlled by the Lamarckian and Baldwinian learning, is proposed in (Zhang, Chen et al. 2013). In (Jia, Zheng et al. 2011) the chaotic local search method was hybridized with the DE using the shrinking strategy to stabilize the algorithm through generations. The use of the shrinking strategy in local search is an effective way to help a local search to perform more efficiently via the preservation of the convergence directions of the global search as reported in (Jia, Zheng et al.

2011). In (Guo, Huang et al. 2015) chaotic local search was also used to intensify the exploitation aspect of DE to optimize the benchmark functions set.

Thus, it can be observed that the use of the local search outperforms the canonical global search methods on many occasions. However, the review of literature suggests none of the above research applied the MDHS to enhance the performance of document clustering by a more efficient distribution of the cluster centroids. The advantages of using the MDHS with the centroids allocation is by allocating the centroids with a minimal likelihood of entrapment in local optima due the use of the localized search. On the other hand, another advantage of using this algorithm is by enhancing the performance of the global search to be less dependent on the tuning of the BW parameter, which is a sensitive parameter; incorrect tuning of this parameter could lead the search into undesirable directions.

2.3.2.2 Memetic Swarm Intelligence-Based Methods

Memetic optimization has also been used to optimize the SI methods used for different clustering problems. As is mentioned earlier, PSO is a leading SI method. As a global search in the memetic optimization, it was first used with clustering by Merwe et al. (Van der Merwe and Engelbrecht 2003). Two approaches were proposed, both based on PSO optimization. The first utilized the native PSO to obtain clusters from randomly initialized points whereas the second used *k*-means clustering to initialize the first generation. The performance of both approaches was evaluated on benchmark datasets and a comparison was conducted against the *k*-means. The test results showed that the hybrid approach outperformed both the *k*-means and PSO algorithms when used on their own.

In (Yang, Sun et al. 2009) the authors explored the use of PSO to assist the K-harmonic means algorithm to skip the local optima. A hybrid data clustering algorithm based on KHM and PSO, called PSOKHM, was proposed. This method was incapable of being used with the text data. As another PSO-based clustering method (Daoud, Sallam et al. 2017) an Arabic document clustering system was proposed. A combined PSO *k*-means method was proposed. However, in (Daoud, Sallam et al. 2017), the performance of the PSO *k*-means combined method was not tested against other SI document clustering methods.

As for local search methods, several studies utilized gradient search using the *k*-means as a local search with SI-based global search methods as explained in (Van der Merwe and

Engelbrecht 2003) (Yang, Sun et al. 2009) and (Daoud, Sallam et al. 2017). This technique has been used to enhance the selection of the initial centroids in *k*-means for clustering that follows a random fashion and that in turn affects the clustering results. For instance, in (Mohd, Bsoul et al. 2012) the authors tried to generate new ways of choosing optimal initial centroids for each cluster in the *k*-means. The method proposed in their study achieved a higher F-measure over the traditionally initialized *k*-means. As another example, in (Mahdavi and Abolhassani 2009) the authors tried to propose an initialization method of the *k*-means using the HS to generate an optimal initial centroid selection for each cluster. The authors compared their method with the genetic-based *k*-means, PSO clustering, and Mises-Fisher Generative-based model. The test results showed that the centroids initialization using the HS method outperformed other methods.

2.3.2.3 Memetic Harmony Search-Based Methods

The first use of HS in document clustering was when it had been modified to Global-Best HS (GHS) method. GHS was inspired by the SI as proposed in PSO. In a global-best PSO, a swarm of individuals (or particles) fly in the search space. Each particle can be considered a candidate solution for the problem. The position of an individual is impacted by the best position traversed by itself (i.e. self-experience) as well as the position of the best particle in the entire swarm (i.e. all-experience) (Omran and Mahdavi 2008). However, the global search nature of the HS was still persistent. In (Cobos, Andrade et al. 2010), the GHS was used with the *k*-means for Web document clustering.

Another variation to the HS was the Improved Harmony Search (IHS). This version dynamically modifies the PAR and this version was used later in hybridization with *k*-means. The IHS was used to optimize centroids selection for efficient document clustering along with *k*-means in (Forsati, Mahdavi et al. 2013). The *k*-means was used to refine solutions resulted from HS global search and it was hybridized in three different forms: interleaved, sequential, and one-step-*k*-means. The difference between these three forms is in the position of the local search (i.e. *k*-means in that case) within the global search. The experimental results showed a superiority of all hybridized forms over the HS, GA, and *k*-means. The authors stated that the test result of document clustering using the combination of HS and *k*-means outperformed the result produced after using HS or *k*-means separately. The HS was also used to optimize data

clustering (Senthilnath, Kulkarni et al. 2016), and outperformed many other evolutionary algorithm methods, including the genetic search. In (Rafi, Shahid et al. 2017) three models have been tested for document clustering which are the GA-*k*-means and HS-k-means, and a combination of both methods in which the results of the GA-*k*-means were fed into the input to the HS-*k*-means. The authors claimed this hybridization outperformed the use of HS or GA solely with the *k*-means.

As was explained earlier, hybrid memetic methods for centroids allocation have been developed. The powerful global search ability is combined with some local improver methods. Such combinations have been successfully applied to the global optimization of numerical functions (Nguyen, Ong et al. 2009) and have been utilized to solve many real-world optimization problems (Neri and Mininno 2010). To enhance the performance of global search optimization methods such as the EA, , SI and HS methods, they have been combined with local search. In this way, the global search methods, which are more robust in the exploration aspect of the search space, could perform better using the local search (Gao, Wang et al. 2015).

Moreover, the modification of the memetic global search itself can contribute to better search results, such that combining two or three global search methods might be more efficient than previous methods. For example, the use of the DE mutation with the HS instead of the PAR step can produce a better performance. Furthermore, the use of the adaptive parameter settings can also enhance performance in comparison to the performance of the static parameter setting (Zhang and Sanderson 2009, Reynoso-Meza, Sanchis et al. 2011). These improvements can all be combined to produce an efficient method for document clustering. Table 2.1 shows a detailed summary of all the document clustering methods that utilized the optimization methods since 2006.

From the discussion thus far, it can be said that optimization methods, especially memetic optimization, are capable of enhancing performance document clustering. However, another problem remains: the higher number of text features dimensionality. In most document clustering systems, the issue of the high number of features can negatively affect the performance of the clustering. Even if the method is successful and performs accurately, hyper-dimensionality has the potential for causing failure (Alsaeedi, Fattah et al. 2017).

Therefore, besides the intelligent selection of centroids, the feature selection is still an important issue to be discussed in this thesis. In the upcoming section, the most recent feature

selection methods along with the baseline methods are explained first. In particular, attention will focus on applying feature selection methods for text dimensionality reduction. Moreover, the detection of the gaps in existing techniques also will be explained.

Table 2.1Summary of Document Clustering Using Optimization Methods

#	Name	Document clustering method	Data type	Year	Local search
1	Genetic algorithm for text clustering based on latent semantic indexing (Abe 2005).	GA+LSI feature extraction	Text data	2009	no
2	Automatic clustering using an improved differential evolution algorithm (Das, Abraham et al. 2008).	DE	Text data	2008	no
3	Exploring differential evolution and Particle Swarm Optimization to develop some symmetry-based automatic clustering techniques: application to gene clustering (Saha and Das 2017).	DE+PSO	Data mining dataset	2017	no
4	Document clustering using Particle Swarm Optimization (Cui, Potok et al. 2005).	PSO	Text data	2005	no
5	Unsupervised Text Classification and Search using Word Embeddings on a Self-Organizing Map (Subramanian and Vora 2016).	SOM	Text data	2016	no
6	Toward A Soft Computing Approach to Document Clustering (Rafi, Shahid et al. 2017).	GA+HS	Text data	2017	*k*-means
7	A hybrid evolutionary computation approach with its application for optimizing text document clustering (Song, Qiao et al. 2015).	QPSO+GA	Text data	2015	non
8	Web Document Clustering by Using PSO-Based Cuckoo Search Clustering Algorithm (Zaw and Mon 2015).	PSO+Cuckoo Search	Text data	2015	non
9	Efficient stochastic algorithms for document clustering (Forsati, Mahdavi et al. 2013).	HS	Text data	2013	*k*-means
10	A novel clustering based differential evolution with 2 multi-parent crossovers for global optimization (Liu, Li et al. 2012).	DE	Data mining dataset	2012	non
11	An improved bee colony optimization algorithm with an application to document clustering (Forsati, Keikha et al. 2015).	BCO	Text data	2015	*k*-means
12	Improving Arabic document clustering using *k*-means algorithm and Particle Swarm Optimization (Daoud, Sallam et al. 2017).	PSO	Text data	2017	*k*-means
13	A krill herd algorithm for efficient text documents clustering (Abualigah, Khader et al. 2016).	KH	Text data	2016	no
14	A novel hybridization strategy for krill herd algorithm applied to clustering techniques (Abualigah, Khader et al. 2017).	KH	Text data	2017	*k*-means
15	Hybrid PSO and GA models for document clustering (Premalatha and Natarajan 2010).	PSO+GA	Text data	2010	*k*-means

16	A novel ant-based clustering approach for document clustering (He, Hui et al. 2006).	ACO	Text data	2006	no
17	A novel clustering approach: Artificial Bee Colony (ABC) algorithm (Karaboga and Ozturk 2011).	ABC	Data mining dataset	2011	no
18	Web page clustering using harmony search optimization (Forsati, Mahdavi et al. 2008).	HS	Text data	2008	no
19	Harmony *k*-means algorithm for document clustering (Mahdavi and Abolhassani 2009).	HS	Text data	2009	no
20	An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization (Yang, Sun et al. 2009).	PSO	Data mining dataset	2009	no
21	Fuzzy document clustering based on ant colony algorithm (Wang, Zhang et al. 2009).	ACO	Text data	2009	no
22	Web document clustering based on a new niching memetic algorithm, term-document matrix and Bayesian Information Criterion (Cobos, Montealegre et al. 2010).	GA	Text data	2010	*k*-means
23	Clustering of web search results based on an Iterative Fuzzy C-means Algorithm and Bayesian Information Criterion (Cobos, Mendoza et al. 2013)	GA	Text data	2013	FCM
24	Web document clustering based on global-best harmony search, *k*-means, frequent term sets and Bayesian Information Criterion (Cobos, Andrade et al. 2010).	HS	text data	2010	*k*-means
25	A novel harmony search-based approach for clustering problems (Senthilnath, Kulkarni et al. 2016).	HS	text data	2016	no
26	Chaotic gradient artificial bee colony for text clustering (Bharti and Singh 2016).	ACO	text data	2016	Chaotic local search+ *k*-means
27	An efficient Particle Swarm Optimization approach to cluster short texts (Cagnina, Errecalde et al. 2014).	PSO	short text data	2014	no
28	Hybrid clustering analysis using improved krill herd algorithm (Abualigah, Khader et al. 2018)	Krill Herd	text and data mining	2018	*k*-means
29	A new hybridization strategy for krill herd algorithm and harmony search algorithm applied to improve the data clustering (Abualigah, Khader et al. 2017).	PSO+HS	data mining dataset	2017	no
30	Hybrid clustering analysis using improved krill herd algorithm (Abualigah, Khader et al. 2018).	KH	text data	2018	no
31	MEDLINE text mining: an enhancement genetic algorithm based approach for document clustering (Karaa, Ashour et al. 2016)	GA	text data	2016	no

2.4 An Overview of Dimensionality Reduction

Knowledge discovery from the text is a challenging task as was explained earlier in this chapter. Due to the increasing numbers of electronic text documents, it becomes necessary to develop

up-to-date tools to handle that increase. With around 80 percent of all data stored electronically in text format, it is a priority to reduce the extra text dimensions (Korde and Mahender 2012). In most cases, processing text data in its original format without feature reduction could affect machine learning accuracy, efficiency and data comprehensibility (Khorsheed and Al-Thubaity 2013). Thus, text feature selection methods are used to reduce the feature space (Diaz-Valenzuela, Loia et al. 2015). Unlike feature extraction methods such as the Principal Component Analysis (PCA) or compression methods using information theory, feature selection methods select a smaller number of features and preserves the original features (Gui, Sun et al. 2017).

In general, machine learning performance can be affected by processing high dimensional features. Therefore, the combination of feature selection with machine learning becomes an important issue in different applications such, as document classification and clustering. After feature selection, the size of the selected feature groups from text is reduced, the size is less than the original. As a result, the storage, processing and time requirements of non-contributing features will be reduced. This will make machine learning more efficient. Moreover, feature selection improves the model performance for obtaining better cluster detection because redundant and non-significant features are eliminated (Gui, Sun et al. 2017).

2.5 Text Dimensionality Reduction Techniques

For document clustering, each document is represented by a set of relevant terms in a Vector Space Model (VSM) (Tang, Kay et al. 2016). Each document has a multi-dimensional feature space, and each dimension is represented by a numeric value (*weight*) corresponding to a specific featured term, which is calculated using various weighting schemes. However, not all the weighted features (*keywords*) are similarly important. Therefore, irrelevant and confusing features should be excluded. That is, for an n feature space the number of possible feature representations reaches 2^n, potentially becoming more complicated over time, because of the increasing number of text documents, leading to an increase in n dimensions (Song, Qiao et al. 2015, Xue, Zhang et al. 2016).

Three critical problems are associated with high dimensional feature space for text data. First, an overfitting problem that occurs, because the use of all the existing features cause all results to be returned as true positives. Second, the inconsistency problem that happens when two objects have the same features, although they belong to two distinct groups. Third, use of the

entire feature space leads to a degraded clustering performance as the computational complexity increases (Zong, Wu et al. 2015).

In general, dimensionality reduction techniques are classified into two types: feature extraction and feature selection. The first aims to generate features from existing ones by merging existing ones. Some widely used extraction methods successfully used with document clustering are Semantic Mapping, PCA, and LSI (Yaghoubyan, Maarof et al. 2016).

An explanation of feature extraction methods is beyond the scope of this literature review; our main focus is feature selection of text documents. Unlike feature extraction, feature selection selects only a subset of features from the original feature space without transforming them and generating new features (Liu and Motoda 2012). It looks for the important subsets according to evaluation criteria that calculates the classification error ratio, divergence, consistency or correlation (Khalid, Khalil et al. 2014). Figure 2.4 shows the difference between the feature selection and feature extraction. Figure 2.4 (a) demonstrates that feature selection selects only a smaller number of subsets. Figure 2.4 (b) shows that the results of extraction techniques are a function of the input data. In both cases m < n (Zhang, Wang et al. 2014)

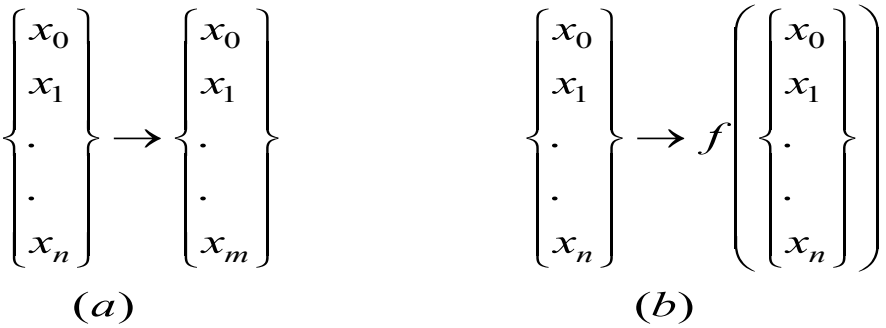

Figure 2.4 (a): Feature Selection of *n* Dimensions to *m* Dimensions.

Figure 2.4 (b)Feature Extraction from *n* Dimensions into new Reduced Feature Space (m<n in both cases).

This literature review will examine the following issues, in particular the use of unsupervised feature selection methods. More specifically, the review will:

A. Discuss the filter, wrapper and the hybrid feature selection methods that reduce the text features using statistical methods.

B. Study the impact of feature selection using different techniques on the performance of unsupervised machine learning.

C. Review the current methods used as internal clustering evaluation measures that are used as filter methods with unsupervised methods, rather than using classifiers to assess the quality of resulted features.

D. Discuss a memetic optimization option to be used with unsupervised combination of wrapper and filter methods. Also, discuss the advantages of using this option over wrapper or filter methods by reviewing studies in that domain.

Table 2.2 reports the most recent published academic works regarding text feature selection. Studies 1 to 9 are unsupervised studies, and studies 10 to 14 concern supervised methods. The table shows that the majority of studies in both the supervised and unsupervised feature selection followed the hybrid or memetic-based schemes. However, filter methods are seldom used by themselves without optimization. Another observation drawn from Table 2.2 is that only a few studies that focused on text data or document clustering dealt with unsupervised feature selection. In the following sections, the studies in Table 2.2 and research conducted in the last decade will be Discussed. On the other hand, Figure 2.5 shows a hierarchy of the dimensionality reduction techniques that will be handled in this literature review.

2.6 Text Feature Selection Development

Text feature selection methods have been used to handle high dimensionality within the text as a pre-processing step (Uysal and Gunal 2014, Hong, Lee et al. 2015). The recent methods proposed to handle the text feature selection are shown in Table 2.2. The methods are classified into three categories: filter, wrapper, and hybrid methods (Liu and Motoda 2012) as can be seen in Figure 2.4. Despite their simplicity, filter methods are incapable of selecting optimal features on their own (Hua, Tembe et al. 2009). These methods are inline methods. The availability of the optimization methods or classifiers is not required (Lazar, Taminau et al. 2012). There is a large number of filter methods such as Mutual Information (MI), Gain Ratio (GR), Symmetrical Uncertainty (SU), Relief and Relief-F (Lazar, Taminau et al. 2012). However, all these methods are used with supervised feature selection with the availability of the class labels.

Recently, an improved global feature selection filter method, Improved Global FS Scheme (IGFSS), was proposed to enhance the performance of text classification (Uysal 2016). In (Pinheiro, Cavalcanti et al. 2015) two other filter methods were introduced. The first is named Maximum Features per Document (MFD), and the second is Maximum Features per Document–Reduced (MFDR). The number of selected features is determined in a data-driven way by using a global ranking Feature Evaluation Function (FEF). To ensure that each document in the training set is represented in the final feature vector, the MFD filter analyses all documents while the MFDR analyses only the documents with high FEF values to select fewer features and to avoid insignificant ones.

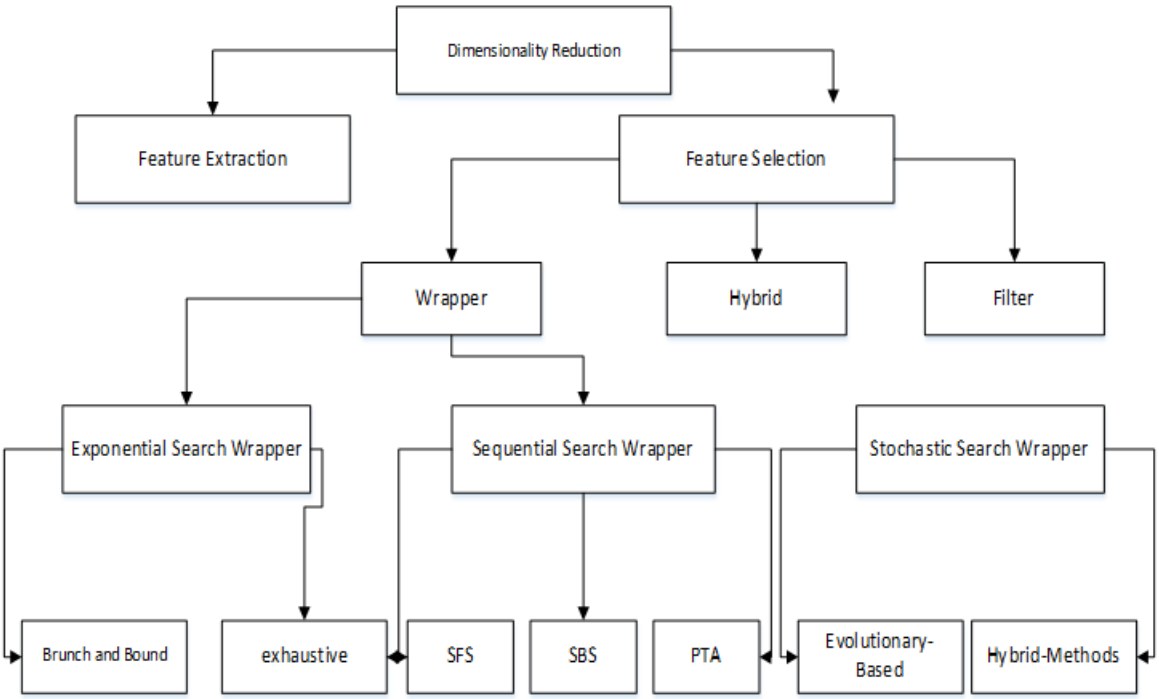

Figure 2.5 Dimensionality Reduction Hierarchy

Several studies (Table 2.2) have reported that wrapper methods outperform filter methods in many cases, especially when the datasets become larger. Wrapper search strategies can be

classified into three categories: exponential, sequential and stochastic methods. These feature selection methods are shown in Figure 2.5.

Table 2.2 Supervised and Unsupervised Feature Selection

#	Ref.	Datasets	Extraction	FS	Optimization Method	Abbreviation	Categorization	Parameter Settings	Domains
1	(Dadaneh, Markid et al. 2016) 2016	Benchmark dataset	Unsupervised	Wrapper	ACO	--	Data FS	No	Data Mining
2	(Tabakhi, Moradi et al. 2014) 2014	Benchmark dataset	Unsupervised	Wrapper	ASO	--	Data FS	No	Data Mining
3	(Abualigah, Khader et al. 2016) 2016	Benchmark text dataset	Unsupervised	Wrapper	GA	FSGATC	Text FS	No	Text Clustering
4	(Abualigah and Khader 2017) 2017	Text dataset	Unsupervised	Wrapper	GA+PSO	H-FSPSOTC	Text Clustering	No	Text Clustering
5	(Tabakhi and Moradi 2015) 2015	Benchmark dataset	Unsupervised	Wrapper	ACO	RRFSACO_1 and RRFSACO_2	Data FS	No	Text Clustering
6	(Bharti and Singh 2016) 2016	Benchmark text dataset	Unsupervised	Wrapper	PSO	BPSO	Text FS	No	Text Clustering
7	(Kumar, Chhabra et al. 2015) 2015	Benchmark dataset	Unsupervised	Wrapper	Gravitational Search Algorithm	AFSGSA	Data FS	No	Data Mining
8	(Chen, Li et al. 2017) 2017	Benchmark dataset	Supervised	Wrapper	BFO	ACBFO and ISEDBFO	Data FS	No	Data Mining
7	(Apolloni, Leguizamón et al. 2016) 2016	Microarray dataset	Supervised	Hybrid	BDE	--	Microarray FS	No	Microarray clustering
8	(Tang, Kay et al. 2016) 2016	Benchmark text dataset	Supervised	Filter	Jeffrey's - Multi-Hypothesis divergence	--	Text FS	No	Text Classification
9	(Qian and Shu 2015) 2015	Benchmark dataset	Supervised	Filter	MI and Rough sets	--	Data FS	No	Incomplete data
10	(Lee and Kim 2015) 2015	Benchmark dataset	Supervised	Memetic	Memetic	--	Data FS	No	Text Classification
11	(Tran, Xue et al. 2016) 2016	Benchmark dataset	Supervised	PSO + local search	Hybrid	--	Data FS	No	Data classification
12	(Ghareb, Bakar et al. 2016) 2016	Arabic Text Datasets	Supervised	GA + local search	Hybrid	--	Text FS	No	Text Classification
13	(Shreem, Abdullah et al. 2016) 2016	Benchmark dataset	Supervised	HS + Filter	Hybrid	--	Data FS	No	Data classification
14	(Han and Ren 2015) 2015	Benchmark dataset	Supervised	MI	Filter and Hybrid	--	Data FS	No	Data classification

2.6.1 Exponential Wrapper Search

The exponential search can be divided into an exhaustive (breadth-first) heuristic search and Branch and Bound search (BB) (Narendra and Fukunaga 1977). Although the exhaustive search guarantees an optimal search state, it works only with countable feature sets. If the

number of features increases, the computational complexity will increase accordingly, rendering the exhaustive search to be impracticable for many cases (Chen 2003). Similarly, the BB method is considered impractical as its performance downgrades when it deals with non-linear classifiers (e.g., Neural Networks) (Ripley 2007). Furthermore, the BB method has exponential complexity. Therefore, it is considered inapplicable to large (or even moderate) feature spaces (Zhang and Sun 2002).

For NP-problems, it is important to create a compromise between effectiveness and optimality, because optimality is not the only factor that has an impact on performance. Therefore, with increased robustness and efficacy, sub-optimality can become an acceptable option (Chandrashekar and Sahin 2014). Thus, sequential wrapper search methods were suggested.

2.6.2 Sequential Wrapper Search

The sequential search looks for features in two directions, either forward or backward (Guan, Liu et al. 2004). The Sequential Forward Selection (SFS) starts from an empty set of features and continuously adds features in that list, while in the Sequential Backward Selection (SBS), the search begins with all features and then deletes those features from the list repetitively (Aha and Bankert 1996, Kudo and Sklansky 2000).

Although the sequential search seems to be straightforward and easy to implement, it suffers from the nesting effect, which means that features cannot be updated in case of the existence of redundant features in the list (Pudil, Novovičová et al. 1994). Therefore, *the Plus-l-Take-Away-r* (PTA) method was proposed, which combines both the forward and backward strategies. It was primarily suggested as a remedy to overcome the nesting effect of the previously explained methods. The PTA search strategy moves L stages forward by adding L elements to the list, and similarly, it moves R stages backward and removes R elements from the list. Consequently, this method had shown no visible improvement over the SFS or the SBS methods as noted by (Hao, Liu et al. 2003).

On the other hand, another version of PTA has also been suggested which relies on the floating search strategy (Pudil, Novovičová et al. 1994) where L and R values are not fixed. This helped to approximate the optimal feature solution subsets. The floating search is either a Sequential Floating Forward Selection (SFFS) or a Sequential Floating Backward Selection (SFBS). Theoretically, there is no reliable way to predict the exact values of L and R to achieve

optimality in the floating search (Bolón-Canedo, Sánchez-Maroño et al. 2013). Also, the floating approach has not shown any improvement on the PTA, SFS and the SBS approaches.

2.6.3 Stochastic Wrapper Search

The stochastic search method uses the meta-heuristic algorithms. It has been proposed as an alternative to the sequential search (Liu and Yu 2005). In recent years, researchers have used stochastic methods in two different ways: supervised and unsupervised (Tutkan, Ganiz et al. 2016, Zorarpacı and Özel 2016). The supervised method has been more commonly experimented with, and it has been widely studied in the field of text categorization. The supervised feature selection depends on the availability of the class labels, which are mandatory for classifiers, and the class labels are used to group features according to their classification accuracy. On the other hand, unsupervised feature selection is not well discussed and tested in the text mining field. In contrast to the supervised feature selection, the unsupervised feature selection depends on measuring the relationship between features in a data-driven way. In other words, it uses internal validation measures that derive knowledge from the intrinsic relationships between features. It is also an appropriate choice used with text clustering in which the class labels are unavailable (Tang, Kay et al. 2016).

The stochastic global search can be applied to perform both the supervised and unsupervised feature selection. For instance, the GA is an example of stochastic methods that have frequently been used for feature selection. GAs initially were proposed by Holland (Holland 1975), and inspired by the natural biological evolution of species. It has a well-documented and successful history with applications to many combinatorial NP-hard optimization problems in science and engineering (Dasgupta and Michalewicz 2013). GA (Goldberg and Holland 1988) has been used for feature selection since the last two decades. However, due to some of the drawbacks associated with its structure, such as the parameter tuning and the random effect of the initial population, it has evolved in many different ways, and other population-based methods have been proposed (Ghareb, Bakar et al. 2016).

As stochastic-based wrapper methods, meta-heuristic optimization methods have been used as wrapper feature selection. Those methods can roughly be classified into evolutionary search, SI, HS (Xue, Zhang et al. 2016) and other meta-heuristic methods. For evolutionary search, a genetic-based wrapper search method was used in (Abualigah, Khader et al. 2016) for unsupervised text feature selection problems. The method was named Unsupervised Feature

Selection Technique Based on Genetic Algorithm for Improving Text Clustering (FSGATC). Also, in (Hong, Lee et al. 2015), the authors used the genetic-based wrapper search method for the text feature selection. As was mentioned earlier, the genetic wrapper has some problems therefore other variants have been used. Similarly, the SI methods have been used extensively for text feature selection, such as the ACO method that was used in (Aghdam, Ghasem-Aghaee et al. 2009), and it achieved slightly better results than the genetic wrapper methods. In (Saraç and Özel 2014) the ACO was also used for text feature selection to improve the accuracy of the classification of Web documents.

Moreover, the PSO method was used in (Zahran and Kanaan 2009) to reduce the features of Arabic text. The algorithm showed superiority over the Chi statistical filter method. In (Lu, Liang et al. 2015) an improved PSO method was proposed, based on a functional inertia weight and a constant constriction factor. According to the constant constriction factor, a functional constriction was added to the traditional PSO. The two improved PSO methods developed upon the functional constriction and functional inertia factors to obtain higher accuracy. They were named synchronously and asynchronously improved PSO, respectively. Moreover, an unsupervised text feature selection method that uses a hybrid PSO with genetic wrapper search operators for text clustering was proposed in (Abualigah and Khader 2017).

The HS has also been applied as a wrapper feature selection method. A self-adjustment feature selection approach that uses HS was proposed in (Zheng, Diao et al. 2015) to improve the performance of the traditional HS based method further. On the other hand, unsupervised HS-based text feature selection was proposed by (Abualigah, Khader et al. 2016). The proposed method was named Feature Selection using Harmony Search for Text Clustering (FSHSTC).

All global search methods represented by the meta-heuristic methods mentioned above are more likely to be stuck in local optima due to their global search nature (Al-Jadir, Wong et al. 2017). This implies they are not well-suited to deal with searches in local areas of the feature space. As a result, that could affect the quality of their resulting features. Therefore, hybrid methods have been proposed to overcome the underlying deficiency of wrapper methods by integrating the advantages of filter and wrapper methods together, as will be seen in following subsections.

2.7 Memetic Filter-Wrapper Feature Selection

Memetic optimization combines filter methods as a local search with wrapper methods that are powerful in a global search (Günal 2012, Ghareb, Bakar et al. 2016). The first attempt to hybridize feature selection using memetic-based wrapper-filter method was proposed in (Radcliffe and Surry 1994). The authors used the method to improve the performance of the genetic wrapper search. Many years later, memetic optimization was also used to produce high-quality solutions in different feature selection problems (Lee and Kim 2015). Through reviewing the methods used for the text feature selection, it was discovered that the majority only utilized memetic optimization to optimize feature selection performance in a supervised manner (Günal 2012, Lee and Kim 2013, Lamirel, Cuxac et al. 2015). The literature review suggests little effort has been made to solve the problem of the unsupervised text feature selection by memetic optimization for efficient document clustering.

The hybrid methods could be performed in various ways. Some of them combine either two filters or two wrappers such as the method proposed in (Zorarpacı and Özel 2016), that combines ABC and DE. Still, the computational complexity becomes higher when two wrappers are used. Filter-filter models are not common reported in the literature unless more than one filter is aggregated with some wrapper method, as seen in Figure 2.6.

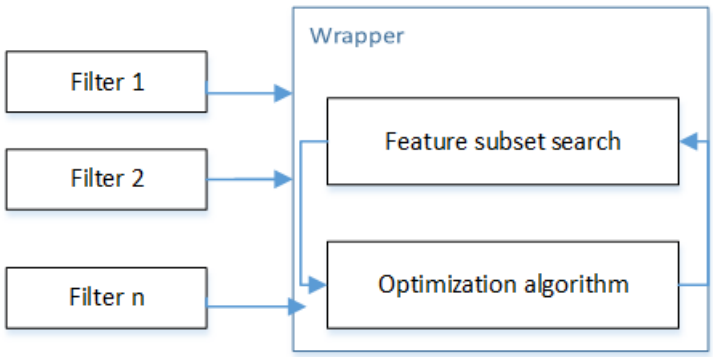

Figure 2.6 Aggregated Hybridization

Therefore, Wrapper-Filter or Filter-Wrapper models are more commonly proposed. Wrapper-Filter hybridizations are mainly constructed with filters embedded inside a wrapper, as can be seen in Figure 2.7. For instance, in a previous work published by the authors in (Al-Jadir, Wong et al. 2017), it was shown that feature selection using a wrapper-filter hybridization, could

improve the performance of traditional clustering methods using *k*-means and Spherical *k*-means traditional clustering methods in terms of the external and internal clustering evaluation measures.

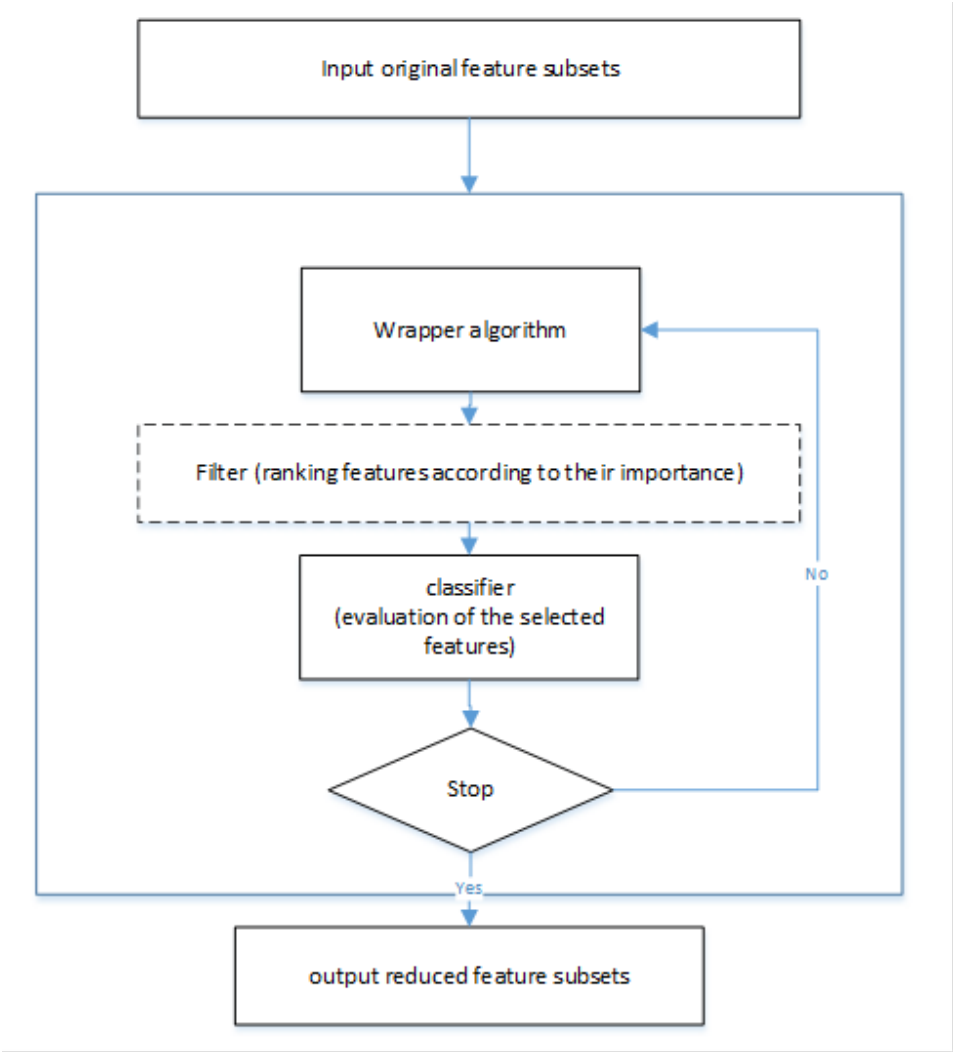

Figure 2.7 Embedded Hybridization

Important examples of the supervised filter methods (classifiers) and unsupervised filter methods used in a memetic scheme are explained in the next two subsections.

2.7.1 Supervised Filter Methods

Using the supervised filter methods with classifiers in hybrid wrapper-filter methods has been dominant in data mining and also text mining (Xue, Zhang et al. 2016). For example, in (Onan and Korukoğlu 2015) a hybrid feature selection method was used to reduce the extra features for the sentiment analysis. The wrapper was applied to aggregate features selected using different filters. It was used to choose the most optimal feature subsets in an ensemble approach. This method composed of a wrapper and multiple filters that are integrated using an ensemble scheme, in which the MI, Information Gain (IG), and Chi filters were used first, before their results were fed to the wrapper. In the same context, in (El Akadi, Amine et al. 2011), a hybrid feature selection method named Maximum Relevance Minimum Redundancy (MRMR) was proposed. The MRMR filter was first applied to exclude noise and redundancy, followed by the genetic wrapper search. The Support Vector Machine (SVM) and Naive Bayes (NB) classifiers were utilized later to serve as fitness functions to evaluate the selected features. Both of (Onan and Korukoğlu 2015) and (El Akadi, Amine et al. 2011) resemble the aggregated hybridization shown in Fig. 2.5.

In (Uğuz 2011), the authors hybridized both filter and wrapper methods to generate a reduced feature space. The authors proposed two models for the dimensionality reduction. The first was based on the Feature Selection-Feature Selection (FS-FS) which is an Information Gain-Genetic Algorithm (IG-GA) that used the Information Gain classifier first and then the genetic wrapper. The latter was used to further refine the selected features. The second model was based on Feature Selection-Feature Extraction (FS-FE) that was named Information Gain-Principle Component Analysis (IG-PCA). This second model differs from the IG-GA in that it extracts features by using PCA. In both models, features are first ranked according to their importance, determined by the IG classifier. Then the feature selection using the genetic wrapper or the Feature Extraction using the PCA is applied separately to reduce the feature space. For evaluation, the effectiveness of both models was assessed using the *k*-nearest neighbor (KNN) and C4.5 decision tree classifiers on Reuters and Classic-3 benchmarks datasets. The test results showed that the two models were effective according to precision, recall and F-measure values.

Furthermore, it was discovered that using the filter-wrapper model, IG-GA had a better performance than IG-PCA. Moreover, despite the success of the classifiers with these two models, the overfitting problem could be generated due to the use of classifier-dependant

features. Also, using the classifiers iteratively might be expensive in terms of computational complexity. As another example, in (Hsu, Hsieh et al. 2011) a hybrid feature selection was proposed, which integrates the IG classifier with a sequential floating search wrapper. The IG classifier was first used individually to select first feature groups, and these features were further selected using the wrapper method. The study concluded that better or at least similar classification performance can be obtained by using the feature selection.

2.7.2 Unsupervised Filter Methods

Unsupervised filter methods differ from supervised filters in that unsupervised methods can rank features without a need to use the original representation (Dadaneh, Markid et al. 2016). However, the challenge of performing unsupervised feature selection is associated with the absence of referencing class labels, which makes it impossible to utilize the same validation criteria used in the supervised feature selection for classification. Because of the lack of research in this domain, there are no standardized measures to assess the performance of unsupervised methods, as the meaning of the best feature subsets might change across different methods. In effect, the limited unsupervised feature selection methods found in the literature have only been used as wrapper methods (Tabakhi, Moradi et al. 2014).

In summary of what has been discussed above, hybrid methods can be more successful than filter or wrapper methods used separately. However, many of the available filter methods used with hybrid methods are not suitable with unsupervised feature selection, because of the necessity of class labels. However, a few examples of unsupervised text filter methods show exceptions. These methods can be categorized into two groups: statistical and approximation-based methods.

2.7.2.1 Statistical Unsupervised Local Search Filter Methods

Statistical unsupervised filter ranking methods could be used as local searches. For instance, the MAD method reported in (Bharti and Singh 2014, Abualigah, Khader et al. 2016, Abualigah, Khader et al. 2016, Abualigah and Khader 2017) is considered a simplification of the Term Variance (TV) filter method proposed in (Liu, Kang et al. 2005). A feature relevancy score is assigned by the MAD method for every feature by calculating the distance of each

feature to the average of the entire set. Mean-Median (MM) (Ferreira, #225 et al. 2012) is another example of the unsupervised filters. Unlike MAD, the MM calculates the absolute distance of each feature between the median and mean values. Similarly, the Absolute Cosine (AC) (Yu and Liu 2003) method serves the same purpose as the MAD and MM methods. These unsupervised filter methods are statistical-based, and their performance relays on the intrinsic properties of the available data. Due to their simplicity and easy implementation, they can efficiently replace the role of classifiers with wrapper methods used in the supervised feature selection as explained in (Bharti and Singh 2014, Abualigah, Khader et al. 2016, Abualigah, Khader et al. 2016, Abualigah and Khader 2017).

2.7.2.2 Approximation-Based Unsupervised Filter Methods

Examples of another category of filter methods that uses optimization and approximation tools are the Simulated Annealing and Chaos theory-based methods, Gradient-based search, Steepest Descent, Newton Raphson, and Gauss-Newton methods (Choi and Lee 1998). All these methods are capable of local search (Jia, Zheng et al. 2011, Merendino and Celebi 2013, Saruhan 2014, Mafarja and Mirjalili 2017). Simulated annealing, for example, can be used for unsupervised feature selection. As an optimization method, simulated annealing iterates until it reaches the best representation of the solution. However, it differs from other optimization methods in that it is not a population-based method; instead, it works on only a single solution at a time, and that makes it a good candidate as a local searcher.

More recently, a Wale Optimization Algorithm (WOA) was used in combination with Simulated Annealing in a hybrid wrapper-filter scheme (Mafarja and Mirjalili 2017). The WOA was hybridized with simulated annealing to improve the quality of the resulted features. Furthermore, simulated annealing was also used by (Mafarja and Abdullah 2013) in a genetic-based wrapper for a hybrid feature selection method. The use of simulated annealing showed improvement using some benchmark datasets in comparison to other state-of-the-art methods. Also, in (Azmi, Pishgoo et al. 2010) a Farsi hand written printed character feature selection used a hybrid GA with a simulated annealing local search. Moreover, in (Manimala, Selvi et al. 2011) also a wrapper using a GA was combined with local simulated annealing to generate a feature selection method that helps in the classification of the Power Quality (PQ) problem, and to improve the parameters of the SVM classifier. It was also used in a timetabling problem along with the GA (Olabiyisi Stephen, Fagbola Temitayo et al.). It can be seen that simulated

annealing can be used as a local search in both supervised and unsupervised modes. However, as an unsupervised local searcher, simulated annealing has not been used for the problem of text feature selection.

In summary, in most document clustering systems the issue of the high number of features could have an adverse impact on the clustering performance. With hyper-dimensionality, even good performing clustering methods could fail. Thus, besides intelligent centroids allocation methods, feature selection methods are still necessary (Khorsheed and Al-Thubaity 2013). Therefore, text feature selection methods are utilized to eliminate unnecessary text features (Diaz-Valenzuela, Loia et al. 2015). Feature selection methods select a smaller number of features without changing them, whereas feature extraction methods such as the PCA reduce features by changing them (Gui, Sun et al. 2017). After feature selection, the size of the selected feature groups becomes less than the original. Consequently, the storage, processing and time requirements of unnecessary features will be far less. Furthermore, feature selection enhances the model to perform and achieve a better clustering outcome, because redundancy and inconsistency is reduced (Gui, Sun et al. 2017).

As in document clustering, documents are represented as a VSM. Every single document has a multi-dimensional feature space, and each dimension is characterized by a numerical weight that corresponds to a particular keyword in VSM. Different weighting schemes, such as the Term-Frequency Inverse Document Frequency (TF.IDF), have been used. Nonetheless, not all the weights are contributing. For that reason, feature selection methods explained in this chapter have been utilized. This chapter explained problems related to high feature dimensionality, such as the overfitting problem that occurs when all feature space is used, causing all documents to be returned as true positives. Moreover, there is a problem of inconsistency, which appears as a result of two objects having the same features whereas they are available in two different groups. Finally, the use of the entire feature space potentially degrades clustering performance, because of an increase in computational complexity (Zong, Wu et al. 2015).

Chapter 3

Document Clustering Using Partitioning Methods with Supervised Feature Selection

3.1 Introduction

In this chapter, a memetic supervised scheme with both wrapper and filter methods for feature selection is discussed. The aim of this chapter is to propose an efficient way of text feature dimensionality reduction that helps to reduce the feature space by eliminating unnecessary text features**.** MA have been successfully applied to different optimization problems. The feature selection problem has been modeled as a population-based problem. This makes it solvable by optimization techniques to look for the best reduced feature space. The main objective of this process is to represent the document sets numerically with no major loss of meaning after post-processing. The feature selection process becomes more important given the large amount of text documents being made available in various digital domains such as the Internet.

The supervised feature selection depends on the availability of class labels which must be used by the classifiers; the class labels are used to group features according to their classification accuracy. The stochastic global search can be applied to perform both the supervised and the unsupervised feature selection. This chapter will focus on the stochastic search using MA to perform the supervised feature selection, and its effect on the document clustering.

As was mentioned in Chapter 2, feature selection can be classified into *filter* (Nie 2005) and *wrapper* (Kohavi and John 1997, Maldonado and Weber 2009) methods. Filters are straightforward in implementation with a higher efficiency than wrapper methods. In most cases, filter methods rank features according to their significance in an ascending order (Souza, Japkowicz et al. 2005). Eventually, clustering or classification is applied to the filtered feature space (Saeys, Inza et al. 2007). Unlike filters, wrapper methods are implemented by using one of the machine learning methods and a classifier. Despite its advantages over filter methods, wrappers may suffer from the issue of overfitting (Saeys, Inza et al. 2007). Therefore, hybrid or memetic schemes that combine both wrapper and filter methods have been introduced (Vergara and Estévez 2014). The proposed memetic feature selection method uses Relief-F

filter and GA-based wrapper to perform the feature selection before the document clustering steps. The method that will be discussed in this chapter is named as Memetic Algorithm Feature Selection (MAFS).

The following sections of this chapter are organized as follows.

- Section 3.2. Feature selection using Memetic Wrapper-Filter hybridization is presented to explain the proposed MAFS method, which is a combination of the global search phase and local search phases.
- Section 3.3. The baseline document clustering methods used to evaluate the resulted feature subsets are explained.
- Section 3.4. The performance evaluation measures used to evaluate the clustering results using the selected feature subsets from the proposed method are explained.
- Section 3.5. Explains the datasets and the experimental results used in this chapter.
- Section 3.6. The document clustering technique using *k*-means and Spherical *k*-means is explained.
- Section 3.7. Explains the impact of various parameter tunings in the proposed MAFS.
- Section 3.8. A comparison between non-tuned MAFS vs. tuned MAFS is presented.
- Section 3.9. Presents a summary of the findings in the chapter.

3.2 Feature Selection Using Memetic Algorithm Feature Selection

The approach for hybridizing a filter within a wrapper is to use the memetic hybridization approach to overcome the GA's premature convergence (Lee and Kim 2015). The memetic search is composed of two components: the local and the global searches. The genetic inductive feature selection method is used to perform the global search while the Relief-F is utilized for fine-tuning the solution (Zhu, Ong et al. 2007). The GA iterates to generate the best feature subset, and Relief-F is used to rank each feature and then order the features in each solution according to their importance.

The proposed hybrid system is illustrated in Figure 3.1. First, the original datasets are transformed into term-document-matrix format. In the next step, the solutions are randomly initialized (random feature subsets), and each solution is evaluated using the KNN classifier and Leave-One-Out Cross Validation. The resulting values represent the classification error ratio that measures the classification predictability of each solution. As illustrated in step 4 of

Figure 3.2, the meta-heuristic operations *(Mutation and Crossover)* are applied to produce new members of the population. Each newly generated solution will replace the older one if its fitness exceeds its ancestors.

Consequently, the selection of the *elite* solutions is performed depending on their *fitness* value. The *elite* population includes the candidate solutions that undergo the local search *(meme)*. Two parameters used to determine the intensity of applying the *meme* are the local search length *(l)* and the local search range *(w)* parameters. The first specifies the maximum number of the additions and deletions performed on the *elite* solutions, and *w* represents the highest number of local search calls to each solution iteratively. There are l^2 times of addition and deletion operations. In other words, *Relief-F* local search that will be described later in this chapter is applied after selecting l^2 adjusting operations on each solution. This process ends when the first newly produced solution has a lower classification error ratio in comparison to the older solution.

In order to clarify this mechanism, it is important first to explain the local search role. We assume an individual *v* with two sets S_1 and S_2. S_1 contains the number of the added features to *v* while S_2 contains the number of the deleted features from *v*. Both lists are ordered based on the features' significance ranks from the best (highest) to the worst (lowest). The deletion moves the features from S_1 to S_2 while the insertion moves features from S_2 to S_1. The deletion and insertion processes are determined by the rank of each feature. Finally, the stopping criterion is satisfied when the maximum number of generations reaches a specific number of iterations. The memetic feature selection in the proposed method works on two modes or phases, which are the wrapper and the filter modes. The wrapper mode is represented by the global search while the local mode is represented by the filter method. Figure 3.1 explain in details these two modes in details.

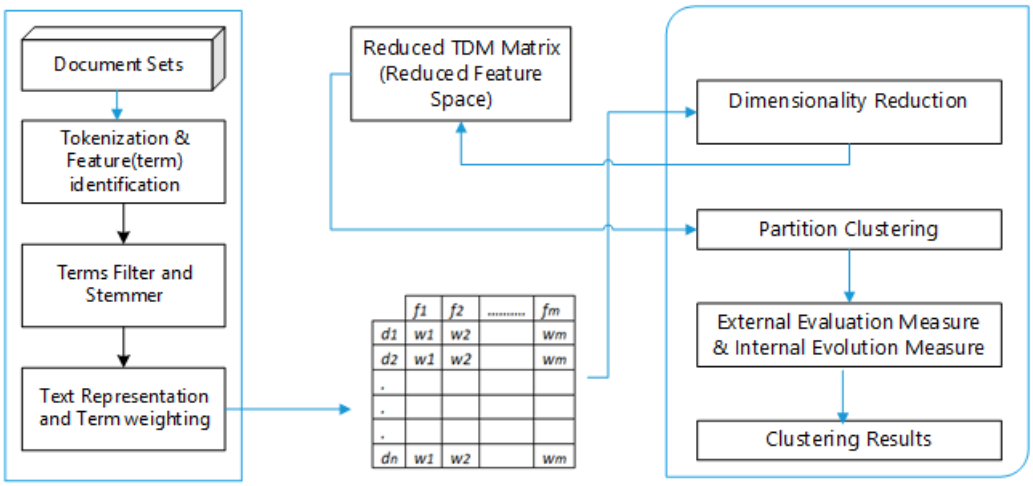

Figure 3.1 General Architecture of the Document Clustering and Feature Selection

3.2.1 Global Search Phase

The global search used the GA-based search, which involves the following three steps while Figure 3.2 explains the flowchart of these steps in details.

A. Initialize the first population randomly; each solution in this population represents a random subset of features. A solution is a binary string with length *n* that represents, at the same time, the number of features. Each bit represents a presence or an absence of a feature. When the feature exists in the solution it is encoded by "1". Otherwise, the feature is encoded by "0". The number of the maximum allowed number of 1s is represented by *m*. In order to obtain a reduced number of features, those features should be less than *m*.

B. Evaluate each one of these random solutions using the fitness function that is the KNN classifier; the resulting fitness values represent the classification error ratio of that solution, measuring its predictability.

C. Apply the meta-heuristic operations *(Mutation and Crossover)* to produce new members of the population. The newly generated solutions will replace the older solutions if their fitness exceeds their ancestors. The meta-heuristic operations used in MAFS are the same as the operations used in the native GA.

3.2.2 Local Search Phase

The local search is applied after the global search starts and initializes the first population as seen in Figure 3.1. This process starts by selecting the *elite* individuals, whose fitness values are the best among the entire population. Then the local search length *(l)* and local search interval values *(w)* are set to their initial values. Where *l* is the maximum number of feature *addition* and *deletion* operations, which are performed on the elite solutions, *w* is the maximum number of local search calls to each solution at a time. Later, the refinement process starts at the elite sub-population using local search. Consequently, the fitness values of the new individuals resulting from local search are calculated. Finally, the comparison of the new fitness values with their corresponding older counterparts is conducted. The classification performance is used to calculate the degree of discriminability of each solution, where the KNN classifier is used. The preceding solutions are replaced with the new optimized solutions if their fitness values are less than the older one, otherwise, the locally optimized solutions are discarded.

The essential idea behind the local search used in this MAFS is based on the Relief-F filter, which is in turn based on the Relief ranking method. The latter estimates the quality of features in respect to how accurately they discriminate between objects, where each object has many features that need to be reduced. In our case the documents are the objects where each document contains many features (weighted keywords). For instance, if there is a random document *Ri* the local search using Relief-F will look for its closest document neighbors. The one that belongs to the same class of that document is named nearest hit *H*, and the other that belongs to the different class, is named nearest miss *M*. The features updating process involves the estimation of the quality vector *W[A]* for all features *A* with respect to their values in R_i, *M*, and *H*. Thus, if documents R_i and *H* have different values of feature $W[A_i]$ then feature $W[A_i]$ discriminates between two documents existing in the same class which is an unwanted case. Thus, that would decrease the quality estimation vector *W[A]*. Another scenario happens when documents R_i and *M* have different values of feature $W[A_i]$; feature $W[A_i]$ separates two documents belonging to different classes which is the target case. Thus, that would increase the quality estimation vector *W[A]*. The entire update process continues for a particular number of times (*m*), where *m* is a predefined parameter. Algorithm 3.1 shows the relief-F steps in details.

Algorithm 3.1 Relief-F

1. *Input (for each training instance a vector of attribute values and the class*
2. *Value).*
3. *Output: the vector W of estimations of the qualities of attributes*
4. set all weights $W[A], N_{dc}, N_{DA\&dA} := 0,$
5. for i: = 1 to *m* do begin
6. Randomly select document R_i
7. Choose k documents I_j closest to R_i ;
8. for j: = 1 to k do begin
9. $N_{dC} := N_{dC} + diff(t(\cdot), R_i, I_j) \cdot d(i, j);$
10. for $A := 1$ to a do
11. $N_{dC\&dA}[A] := N_{dC\&dA}[A] + diff(t(\cdot), R_i, I_j).\ diff(A, Ri, I j) \cdot d(i, j);$
12. end;
13. end
14. end
15. for $A := 1$ to a do
16. $W[A] := N_{dC\&dA}[A]/N_{dC} - (N_{dA}[A] - N_{dC\&dA}[A])/(m - N_{dC});$
17. end

The Relief-F is a modified version of the Relief, but it is not restricted to only two classes. Relief-F can deal with multiple classes, is more robust and is more capable of dealing with incomplete or noisy data. Relief-F randomly selects a document *Ri.* Unlike Relief, Relief-F looks for *k*-nearest documents of the same class of *Ri*, which are called nearest hits H_j and at the same time it looks for *k*-nearest documents existing in other classes which are called nearest misses M_j. The quality estimation *W[A]* is then updated for all features *A* according to their values for *Ri* and hits *Hj* and M_j.

3.3 *k*-means and Spherical *k*-means Document Clustering

The two baseline partitioning clustering algorithms used to test the proposed MAFS are the *k*-means algorithm (Deelers and Auwatanamongkol 2007), and the Spherical *k*-means (Dhillon, Fan et al. 2001):

A. The *k*-means algorithm initialized with a specific number of clusters determined by a randomly initialized centers points (centroids) represented by *k* that means each *k* refers to one cluster. Next, the Euclidian distance between each centroid and each document is calculated to associate each document with its nearest centroid. The next iteration involves a new *k* centroids set to be recalculated again. At this point, a new distance should be found between the documents and the new centroids. The distance

measurement and the centroids updating will continue until the *k* centroid values of the current step have the same positions as the previous step. The distance objective function is stated in equation(3.1):

$$D = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| d_i - c_j \right\|^2 \qquad \textit{Equation (3.1)}$$

where *D* is the distance between any document *(d)* and centroid *(c)* while *k* and *n* are the number of centroids and the number of documents respectively.

B. The Spherical *k*-means is an enhancement to the traditional *k*-means. It is developed to produce more stable results than the *k*-means. It is a fixed point heuristic clustering algorithm that minimizes the angular distance between two documents measured by the cosine similarity. Due to the overrepresentation of the highly weighted terms using the *k*-means, it is suggested in (Dhillon, Fan et al. 2001) to use the projections of the weighted vectors using the Euclidean distance onto a spherical space, or equally, using the cosine similarity which is equivalent to the first as shown in equation(3.2).

$$D = \sum_{i=1}^{n} (1 - \cos(x_i, p_{c(i)})) \qquad \textit{Equation(3.2)}$$

where, the $x_{i\ is}$ an i^{th} document vector, p_c is a centroid vector and *c(i)* is the cluster identifier.

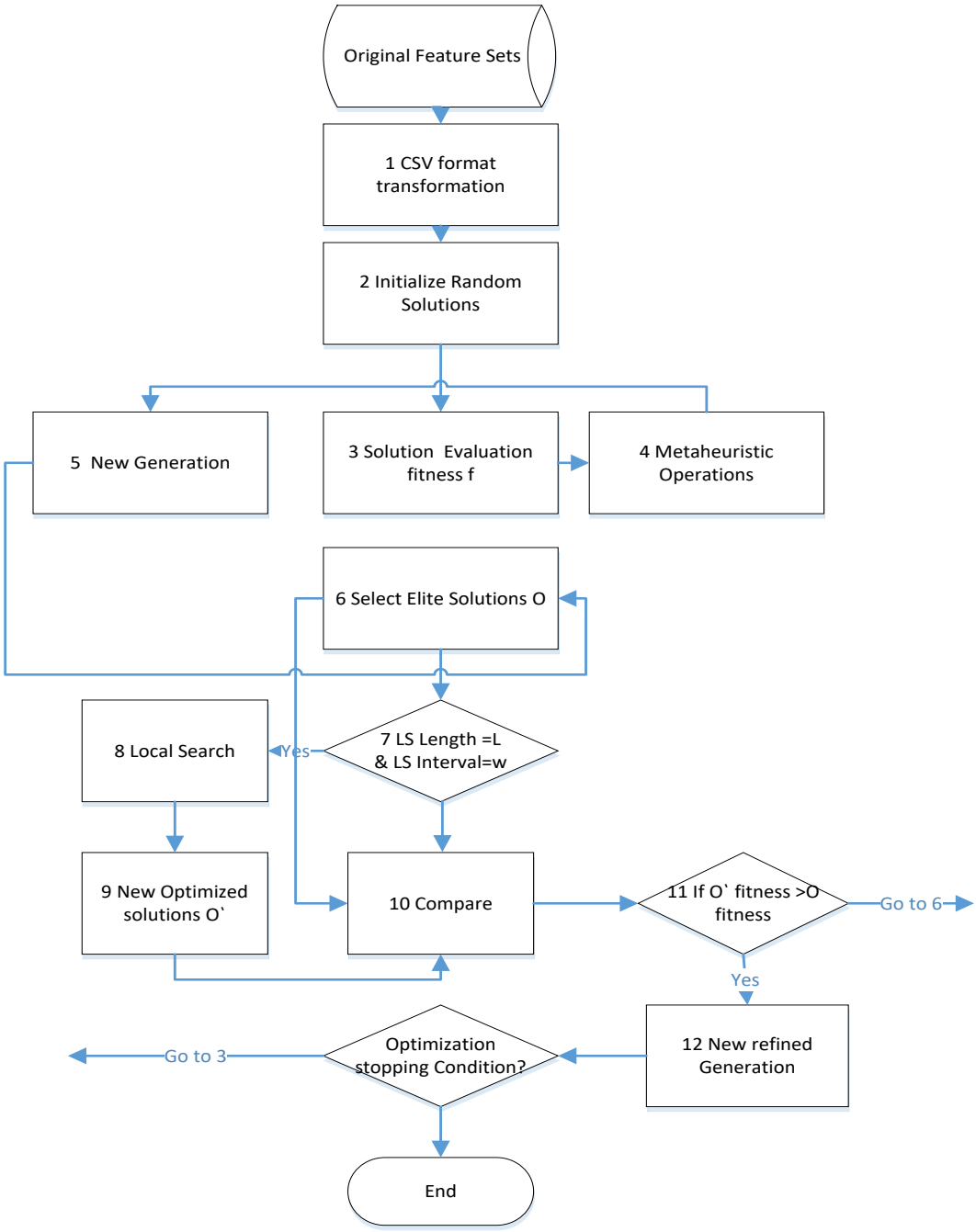

Figure 3.2 MAFS Feature Selection

3.4 Performance Evaluation

The evaluation measures are important to observe the performance of each feature selection method and its effect on the clustering algorithm. Two types of measurements are used in the experiments, which are the internal and the external evaluation measures (Aggarwal and Reddy 2013). In the experiments, the F-measure is used as an external measure while the internal measure used is the Average-Document-Distance-to-the-Cluster-Centroid (ADDC). A thorough analysis is conducted in the next section by observing the maximization of the F-measure and the minimization of ADDC measure using different methods. The F-measure and the ADDC are computed as follows:

A. The F-measure, is calculated after computing the precision *(P)* and the recall *(R)*. *P* is the proportion of documents in group *A* and still in class *B* whereas *R* is the proportion of documents in class *B* and group *A*. The precision and recall and the F-measure are computed as is shown in equations (3.3),(3.4), and (3.5), respectively. In equation 3.5, the value of the F-measure is calculated using the previous two values represented in equations 3.3 and 3.4.

$$P(x,y) = \frac{n(A,B)}{n(A)} \qquad \textit{Equation(3.3)}$$

$$R(x,y) = \frac{n(A,B)}{n(B)} \qquad \textit{Equation (3.4)}$$

$$F-measure(A,B) = \frac{2P(A,B)R(A,B)}{P(A,B)+R(A,B)} \qquad \textit{Equation (3.5)}$$

B. The Objective Function used for evaluation is the Average Distance of Documents to Cluster centroid (ADDC). The cosine similarity is used in ADDC to find the distance of the documents between each other and between themselves and their corresponding cluster centroids. The ADDC can be expressed as is shown equation (3.6):

$$Fitness = \left[\sum_{i=1}^{C} \left(\sum_{j=1}^{Pi} \cos(c_i, x_j) / p_i \right) \right] / C \qquad \textit{Equation (3.6)}$$

where C is the number of centroids while the P_i is the number of documents in each cluster and the c_i, and the d_{ij} are any particular centroid and document pair, and *cos* is the cosine similarity measure between any two vectors.

3.5 Datasets and Experimental Results

The datasets used in the next experiments are listed in Table 3.1 where the number of classes, instances and features of each dataset is shown in that Table.

Table 3.1 Datasets

Dataset	D#	#Classes	Instances	Features
6 Event Crimes	D1	6	223	3864
10 Types Crime	D2	10	2422	15601
Reuters-21578,	D3	10	2277	13310
20news Groups	D5	20	1489	6738
Pair 20news Groups	D4	2	1071	9497

D1. 6 Event Crimes. This dataset is collected from the online news available at (http:// www. bernama. Com /bernama/v8/index.php). The first dataset has six classes of crimes whereas the other dataset has ten categories. In Table 3.1 the number of the documents and the number of classes in each of those datasets are reported.

D2. 10 Types Crime. This contains ten types of criminal reports which contain 2,422 documents and 15,601 features (keywords).

D3.Reuters. This dataset is available at Machine Learning Repository[1]. Although this dataset is diversified and challenging (Debole and Sebastiani 2005), many labels in the documents are missing. There is a large number of multi-labeled documents. Besides, the number of classes is skewed leading to inconsistent class sizes. In order to deal with these drawbacks, the same edition utilized in (Fodeh, Punch et al. 2011) is also used in this present research. The edition includes only the labeled documents and single-labeled documents. Furthermore, the number of documents chosen for each class is 200.

D4. 20news Groups, This dataset consists collected from 20 news sources. It is also available at the Machine Learning Repository[2].

D5. Pair 20news Groups, This sub-dataset contains: the.talk.Politics, Mideast, and talk.Politics.Misc which is a subset of D4.

3.6 Clustering Using *k*-means and Spherical *k*-means

In this subsection, we describe the tests conducted on the datasets explained earlier. In the tests, the algorithms were run 20 times. Running the algorithms for one time is not enough to determine the performance of each clustering algorithm accurately, because of the randomness of the initial solutions of each run. This randomness can affect the results. When multiple runs are used, the general trend of each single algorithm will be identified. Table 3.2 shows the average F-measure results while Table 3.3 shows the average ADDC results after using the entire feature space.

Table 3.2 illustrates the average F-measure values of 20 runs for the *k*-means and the Spherical *k*-means algorithms without a feature selection for each dataset and with feature selection in two cases once with using the Genetic feature selection and once again with the Memetic feature selection. When using the entire features without feature selection is referred as All method, in all the runs for all the datasets the Spherical *k*-means outperformed the traditional *k*-means regarding the F-measure. In D2, the *k*-means has slightly outperformed the Spherical *k*-means; the first achieved (0.33) while the second achieved (0.25). On the other hand, Table 3.3 reports the ADDC values. Unlike the F-measure values, the ADDC values should be minimized, and that means less is better. It is evident that the ADDC measure is not highly

[1]https://archive.ics.uci.edu/ml/machine-learning-databases/reuters21578-mld/
[2] https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups

correlated to the F-measure. However, there might be a slight correlation between these two measurements. For instance, the average F-measure of the D1 dataset is less than that of the Spherical *k*-means, while its ADDC value dropped slightly accordingly for the (All) method. Furthermore, the average F-measure value (0.33) of D2 dataset for the *k*-means is decreased in the Spherical *k*-means (0.25). But the average ADDC value of the D2 dataset (0.75) for the Spherical *k*-means has slightly increased over that of the *k*-means (0.68). That indicates there is only a slight correlation between the two measurements. In general, the ADDC values will be lowered (becomes better) if the feature selection is applied to the data by reducing the extra features.

Table 3.2 Average ADDC Measure Values Using the Entire Feature Set

	Datasets	***k*-means**	**Spherical *k*-means**
ALL	D1	0.57	0.56
	D2	0.68	0.75
	D3	0.5	0.85
	D4	0.59	0.82
	D5	0.65	0.68
GA	Datasets	GA-k-means	GA-Spherical k-means
	D1	0.219	0.24
	D2	0.17	0.84
	D3	0.41	0.75
	D4	0.55	0.63
	D5	0.48	1
MAFS	Datasets	MAFS-k-means	MAFS-Spherical k-means
	D1	0.23	0.24
	D2	0.15	0.38
	D3	0.35	0.62
	D4	0.63	0.80
	D5	0.34	0.52

Table 3.3 Average F-measure Values After Applying GA

	Datasets	**k-means**	**Spherical k-means**
ALL	D1	0.62	0.72
	D2	0.33	0.25
	D3	0.30	0.68
	D4	0.51	0.95
	D5	0.64	0.87

	Datasets	GA-*k*-means	GA-Spherical *k*-means
GA	D1	0.50	0.74
	D2	0.12	0.10
	D3	0.28	0.25
	D4	0.56	0.79
	D5	0.35	0.33
MAFS	Datasets	MAFS-k-means	MAFS-Spherical k-means
	D1	0.62	0.74
	D2	0.29	0.10
	D3	0.38	0.70
	D4	0.56	0.95
	D5	0.77	0.89

Regarding the effect of applying the GA wrapper feature selection on the F-measure and the ADDC measurements, Table 3.2 clearly shows that the F-measure results have dropped in comparison to Table 3.3, which indicates important text features were lost through the process of feature selection. In contrast, the ADDC values of the GA method have dropped in comparison to those of the All method in the Table 2.3; that means the distance of the inter-cluster similarity has been notably minimized. If ADDC values become less, it suggests more compact clusters are produced. In other words, 'within cluster' similarities have been increased despite the fact that the F-measure has slightly dropped for most of the datasets. This is true for all datasets used in the Thesis for D5 (Reuters) dataset, in which the ADDC value increased to 1 for the Spherical *k*-means algorithm for the GA method while it was 0.68 for the All method in Table 2.3. That result is because the F-measure plummeted dramatically from 0.87 for the All method to only 0.33 for the GA method in Table 3.2. The result is considered to be another example of the correlation between these two measurements in some cases.

In Tables 3.2 and 3.3, the F-measure and the ADDC values of MAFS method was also reported similarly as the All and the GA methods.

The MAFS-*k*-means F-measure values in Table 3.2 are higher than those obtained by the GA method in the same table that indicates the feature selection with the MAFS algorithm preserves the important features that are removed by the genetic feature selection that led to an improved F-measure. For instance, the F-measure of the D5 for the *k*-means has increased by almost double after using the MAFS method in comparison to the GA based method. Similarly, the F-measure values of D5 for the Spherical *k*-means increased sharply in the same table in compassion to values obtained by the All and the GA based methods. On the other hand, the

F-measure for D3 has slightly improved using the MAFS method in comparison to results obtained by the other methods in Table 3.2.

With regard to the ADDC measure, the most striking result to emerge from the data in Table 3.3 is that the ADDC values of the MAFS method have been dropped remarkably in comparison to the results of the other two methods. This finding shows that the resulting clusters became more compacted with reduced distances among the class members. It can also be observed that using the MAFS-*k*-means and the MAFS-Spherical *k*-means led to lower ADDC values even if the F-measure values slightly dropped. For example, the *k*-means F-measure value for the D2 dataset has dropped from 0.33 for the All method to 0.29 for the MAFS method as is illustrated in Table 3.2, but the ADDC in for the MAFS shown in Table 3.3 has halved. The most obvious observation is that the ADDC and the F-measure values are slightly dependent. That in turn could lead to the fact that the resulting clusters will be compacted further, if the feature selection is used, even if the F-measure values do not improve or slightly drop (in some cases).

Besides using the average value to determine the best performing algorithm, Table 3.4 shows the best, worst and the difference *(diff.)* between the best and the worst F-measure values in each algorithm, while Table 3.5 reports the ADDC values. In both tables, the average best, average worst and average difference were calculated after running each clustering algorithm multiple times. Using the best, worst and the difference of the values is important to show the stability of the three algorithms throughout multiple runs after and before feature selection. In other words, discussing only the average might not show the individual performance of the algorithms in each single run.

In Table 3.4 the *k*-means difference values between the best and the worst values are higher than the Spherical *k*-means in all datasets used except D5 for the All method. This implies that the values resulted from the *k*-means are more random, and the Spherical *k*-means appears to be more stable. On the other hand, the same observation can also be true after applying MAFS. Nonetheless, for the MAFS, the value difference for D4 slightly increased compared to the results of the All method for the Spherical *k*-means, while the performance was degraded in comparison to the *k*-means as can be seen in both tables. It was decreased for the best from 0.99 to 0.95 and from 0.87 to 0.77 for the worst as reported in Tables 3.4. Also, the values were slightly decreased for D3 and D5, however D2 values plummeted for the best values (from 0.43

to 0.11) and for the worst (from 0.17 to 0.10). Moreover, the *k*-means also had slight degradation with some datasets, but substantial improvements with other datasets in Table 3.4. In general, the *k*-means performance was highly comparable to the performance of the Spherical *k*-means before using the MAFS. Although the external measurement via the F-measure showed that the Spherical *k*-means is more stable than the *k*-means, it showed degradation in performance after using the MAFS.

With regard to the internal measure when using ADDC, Table 3.5 gives a clear idea of the actual impact of using MAFS on the best and worst values of all datasets. The *k*-means algorithm was also improved notably after using MAFS. That is evident in all the datasets used except D4, which showed an insignificant increase for the worst (0.07) and for the best (0.01). In addition, the Spherical *k*-means also dropped noticeably in both tables.

Table 3.4 The Best, Worst, Difference F-measure Values of *k*-means and Spherical *k*-means Algorithms Using Entire Feature Space

Method		***k*-means**			**Spherical *k*-means**		
	Datasets	**Best**	**Worst**	**Diff.**	**Best**	**Worst**	**Diff.**
All (All features)	D1	0.89	0.32	0.57	0.84	0.54	0.29
	D2	0.54	0.15	0.39	0.43	0.17	0.26
	D3	0.84	0.42	0.69	0.94	0.46	0.48
	D4	0.51	0.30	0.21	0.99	0.87	0.12
	D5	0.33	0.25	0.8	0.95	0.55	0.40
	Average	0.62	0.29	0.39	0.83	0.52	0.31
MAFS	D1	0.69	0.21	0.47	0.89	0.56	0.33
	D2	0.51	0.11	0.40	0.11	0.10	0.1
	D3	0.88	0.42	0.45	0.90	0.47	0.43
	D4	0.61	0.51	0.9	0.95	0.77	0.18
	D5	0.45	0.33	0.12	0.96	0.45	0.51
	Average	0.63	0.32	0.31	0.76	0.47	0.29

Table 3.5 The Best, Worst, Difference ADDC Values of *k*-means and Spherical *k*-means Algorithms using Entire Feature Space

Method		***k*-Means**	**Spherical *k*-means**

	Datasets	Best	Worst	Diff.	Best	Worst	Diff.
All (All features)	D1	0.55	0.59	0.04	0.66	0.71	0.05
	D2	0.66	0.71	0.11	0.75	0.75	0
	D3	0.64	0.68	0.04	0.59	0.69	0.1
	D4	0.54	0.6	0.14	0.78	0.83	0.12
	D5	0.47	0.64	0.17	0.86	0.87	0.01
	Average	0.57	0.64	0.10	0.73	0.77	0.06
MAFS	D1	0.20	0.29	0.09	0.09	0.29	0.2
	D2	0.13	0.19	0.06	0.47	0.48	0.01
	D3	0.34	0.45	0.11	0.50	0.57	0.07
	D4	0.56	0.67	0.11	0.74	0.84	0.10
	D5	0.34	0.50	0.16	0.54	0.54	0
	Average	0.31	0.42	0.11	0.47	0.54	0.08

It is important to use the two measures for comparison as there is a small relationship between the internal and the external measures. If the external measures fail to give clear evidence of the performance, it is more reasonable to follow the results of the internal measures.

3.7 The Impact of Various Parameter Tunings on MAFS Performance

The next experiments are intended to show the results after parameter tuning, conducted as a series of empirical experiments on the same datasets used in the previous tests. All previous tests were based on the parameters set by (Zhu, Ong et al. 2007) and shown in Table 3.6. The FSGATC method that corresponds to the GA wrapper method is used for comparison purposes (Abualigah, Khader et al. 2016).

Table 3.6 Parameters Used with MAFS

Parameter	Value
Search Strategy	Genetic Search Method
Population size	50
Solution size	50
Local search range	5

Local search length	8
Number of generations:	200
Probability of crossover:	0.6
Crossover Type:	Uniform Crossover
Probability of mutation:	0.1
Local Search:	TRUE
Local Search Method:	Filter Ranking
Local Search Strategy:	Improvement First
Selection Type:	Linear Rank Selection
Stopping criterion	6000 Objective Function Evaluations

In this section, various experiments are conducted using different parameter combinations to reveal the effects of them on feature subsets. Table 3.7 shows the parameters are tuned based on their significance to MAFS.

The crossover type is tuned first, using the *uniform, one-point, and multi-point* crossover types. For the local search range *w*, five different values were compared. Similarly, for the local search length *l*, three different values were tested. The classification methods used with the wrapper as a fitness function were also tested with three different values. The compared classifiers were the *lwl* (Locally Weighted Learning) classifier, *knn* with $k = 1$ and $k = 3$. The *knn* classifier using $k = 3$ achieved the highest accuracy among the other two options. In order to measure the impact of the local search range, the best value of the local search range equaled 10 as it achieved the minimal error rate in the last generation in comparison to other methods.

Table 3.7 Values Tested for the Parameters Tuning

No.	**Crossover**	**w**	**l**	**Classifier**
1	*One-point*	10	3	*lwl (locally weighted learning)*
2	*Multi points*	1	8	*knn, k = 3*
3	*Uniform*	15	1	*knn, k = 1*
4	_	5	_	_
5	_	25	_	_

Regarding the local search length, the tested values were 3, 8, and 1. It is noteworthy that other values are selected within the range from 1 to 10, but the results were almost similar to the

results of the three tested values. Keeping the local search range to its best value selected previously, the best value chosen for the local search length was 8. Finally, the crossover experiments showed that using the *multi-point* crossover was more productive than the two counterparts. Moreover, as for the other parameters, they were kept as same as shown in Table 3.6.

From the results in Table 3.8 it can be concluded that by using the MAFS method, document clustering performance can be improved. The MAFS method achieved more accurate results for D2, D3, and D5 datasets, whereas the ADDC values of the MAFS method were increased slightly for D1 and D4. However, the corresponding F-measure values for D2 and D4 were much higher than those obtained by ALL and FSGATC methods. The slight increase of ADDC values for D1 and D4 could be tolerated in favor of the higher leap achieved by the MAFS method using the external measure. On the other hand, the external evaluation measures for the D2, D3, and D5 had a higher value for the F-measure after using the MAFS method. At the same time, results for these datasets generated by using MAFS methods provided smaller ADDCs in comparison to those generated by using FSGATC and ALL methods as mentioned earlier.

The results in Table 3.8, it is also noted that in all datasets, the *k*-means performance improved after using the MAFS method by observing the F-measure. An improvement in performance was achieved by using the MAFS method for the *k*-means clustering. For the ADDC values and results of D2, D3, and D5, the MAFS method obtained smaller values but a higher F-measure. On the other hand, for D1 and D4, although the FSGATC method obtained smaller ADDC values than the MAFS method, the corresponding FSGATC external measure values were still less than those achieved by MAFS for both D1 and D4.

The clearest observation to make from Table 3.8 is that the proposed MAFS method is superior when compared to the ALL and FSGATC methods. It appears from the trends of both the ADDC measure and the F-measure that the relationship between them could be stated in three cases which are listed below:

A. When the internal measure decreases the external measure increases, which is an ideal convergence state. For example, this happened with the MAFS method for the D2, D3, and D5 datasets using the spherical k-means while it is also clear for the D1, D3, and D5 using the k-means

B. The second case happens when the internal measure does not significantly decrease while the corresponding external measure increases significantly, which indicates a notable improvement in the clustering accuracy.

C. Finally, the worst case that might happen is when there is no improvement in the external measure, but this was not visible in the results of the proposed MAFS method in any of the datasets used. It can be clearly concluded that the MAFS performed well with more stability than using ALL and FSGATC methods for all datasets.

Table 3.8 Average Results of 20 Spherical *k*-means Runs

		Spherical k-means		k-means	
	Methods	**ADDC**	**F-Measure**	**ADDC**	**F-measure**
D1	ALL	0.57	0.69	0.56	0.52
	MAFS	0.24	0.81	0.22	0.70
	FSGATC	0.22	0.52	0.24	0.56
D2	ALL	0.67	0.31	0.20	0.15
	MAFS	0.15	0.36	0.54	0.66
	FSGATC	0.17	0.17	0.82	0.1
D3	ALL	0.54	0.20	0.86	0.27
	MAFS	0.26	0.28	0.47	0.33
	FSGATC	0.33	0.24	0.73	0.33
D4	ALL	0.85	0.94	0.82	0.83
	MAFS	0.82	0.94	0.76	0.89
	FSGATC	0.63	0.83	0.63	0.83
D5	ALL	0.65	0.74	0.69	0.92
	MAFS	0.37	0.75	0.48	0.93
	FSGATC	0.51	0.47	1	0.33

3.8 Non-Tuned MAFS (MAFS1) vs. Tuned (MAFS2)

The effect of the new parameter settings on the MAFS is shown in Tables 3.9 for the *k*-means and Spherical *k*-means respectively. MAFS1 represents the clustering results with original parameter settings shown in Table 3.6 while MAFS2 represents the results of the clustering after tuning the parameters as mentioned previously in section 3.6.

Table 3.9 Parameter Setting Effect on MAFS for *k*-means using F-measure

	k-means		Spherical k-means	
Datasets	**MAFS1**	**MAFS2**	**MAFS1**	**MAFS2**
D1	0.23	0.70	0.24	0.81
D2	0.15	0.66	0.38	0.36
D3	0.35	0.33	0.28	0.62
D4	0.63	0.89	0.80	0.94
D5	0.34	0.93	0.52	0.75

In Table 3.9 it can be noticed that using the parameters settings had a positive impact on the clustering results. The F-measure increased with almost all datasets except D3, which had only a small degradation. On the other hand, Table 3.9 shows the results increased with all datasets. Therefore, the parameter tuning of the MAFS2 is selected for further experiments presented in the next chapters. For simplicity, the MAFS2 will only be referred to as MAFS in subsequent chapters, which is the tuned version of MAFS.

3.9 Summary

Supervised memetic hybridization between filter and wrapper feature selection methods, identified as MAFS, was presented in this chapter. Traditional clustering methods including the *k*-means and Spherical *k*-means were used to examine the performance of the proposed feature selection method. One of the significant findings to emerge from this study is that the ADDC measure is minimized while the F-measure is maximized with most of the cases after using the MAFS hybrid memetic feature selection method. Moreover, the memetic hybridization using the proposed MAFS performed better than the wrapper feature selection using the genetic wrapper search. In addition, the results showed that the clustering results after using MAFS outperformed those using the entire feature space (ALL). The test results also showed that using the proposed feature selection can enhance the performance of traditional clustering. In the comparison study, the proposed MAFS method outperformed the results obtained by the recently proposed method named FSGATC. The proposed MAFS method also performed better when compared to the results generated using the ALL feature space. The experiments also found a slight correlation between the ADDC and the F-measure. Finally,

tuning of the parameters in MAFS had a positive effect on the results, and was selected for subsequent testing that is reported in following chapters.

Chapter 4

An Unsupervised Feature Selection Using a Memetic Hybridization of a Wrapper and Filter Methods

4.1 Introduction

In chapter 3 a supervised method was proposed for feature selection and results of testing were reported. It was seen that a combination of wrapper and filter methods can help to produce better document clustering results in terms of higher accuracy than wrapper or filter methods alone. In chapter 3, the proposed method handled the feature selection for cases where class labels are available. In this chapter, the method proposed is applied to cases where class labels are missing. The aim of the unsupervised approach is to achieve equivalent performance when class labels could be present.

The method presented in this chapter is an unsupervised feature selection method that combines the DE with simulated annealing. The presented method is named DESA. The challenge of performing unsupervised feature selection is associated with the absence of referencing class labels, which makes it impossible to utilize the same validation criteria used in the supervised FS for classification. Moreover, there are no standardized measures to assess the performance of unsupervised methods, because of a lack of research in this domain, as the meaning of the best feature subsets might differ across different methods. In effect, the limited literature on unsupervised FS methods has only reported wrapper unsupervised FS. In order to address the gaps in the unsupervised FS area, this chapter discusses details of the DESA FS method. SA is used to improve the exploitation aspect and DE is used as a global search to perform the explorative aspect. The MAD filter is used as feature subset internal evaluation criterion. Class labels are not required by the MAD as it is a data-driven evaluation measure that discovers similarities between features according to their intrinsic properties. The MAD resembles the classifier in the supervised feature selection. This paper also investigates ways of generating the performance measure that can be used to assist the optimization process using the internal and the external clustering evaluation measures.

However, another modification to the DESA is also presented in this chapter which uses the dichotomous mutation, which is named Dichotomous DESA (DDESA). This version uses the dichotomous mutation that eliminates the need to use the *F* probability, which can be used for situations such as knapsack problems reported in (Peng, Wu et al. 2016). The same concept is used with the DDESA method for the centroids allocation problem of document clustering. An extensive explanation of DESA and DDESA is given in this chapter with examples that show the workability of these methods. Later, the experimental results of using these methods in comparison to several other methods are also presented.

4.2 Differential Evolution for Feature Selection

The DE is a population-based evolutionary optimization method that was explained in chapter two (Chunming, Yadong et al. 2017). In feature selection, all the generated solution values should be distributed in the range between two values which are either 0 or 1. The values will be rounded to the nearest integers toward 0 using the *fix* function[3]. For instance, in Y = fix(X) it rounds each element of X to the nearest integer toward zero. For positive X , the behavior of fix is the same as floor . The memetic DE is used in this chapter to perform the unsupervised feature selection. Two versions of the memetic DE used in this chapter. The first is the DESA that uses the standard DE combined with the SA. However, another version of the DESA is also experimented with in this chapter, which is named Dichotomous DESA (DDESA). This version uses the Dichotomous mutation that eliminates the need to use the *F* probability. This method was used in (Peng, Wu et al. 2016) to solve binary Knapsack problems. The dichotomous mutation is expressed as equation (4.2). This equation shows the new offspring solution v_{ij} is generated using the three existing solutions x_{r1}, x_{r2}, and x_{r3}. The logical operators were used to in the dichotomous mutation. The example explained in Table 4.1 shows how old solutions are used to generate the new offspring solution.

$$v_{i,j} = ((x_{r1} \; xor \; x_{r2}) \; and \; rand\{0,1\}) \text{ or } (not(x_{r1} \; xor \; x_{r2}) \; and \; x_{r1}) \qquad Equation(4.2)$$

Table 4.1 shows an example to generate a new solution using the dichotomous mutation. In Table 4.1, where x_1 and x_2 are two random vectors selected from the DE population, *r* is a random vector, $x_3 = (x_1 \; or \; x_2)$, $x_4 = not \; (x_3)$, $x_5 = (x_4 \; and \; x_1)$, and $x_6 = (x_4 \; or \; x_5)$. The crossover for both methods is applied later, to diversify the population by the perturbation of the current

[3] https://au.mathworks.com/help/fixedpoint/ref/fix.html

population. The crossover in DE is performed as shown in the following equation (2.3). Figure 4.1 shows an example of the DE crossover. The target solution x_k is then compared with vector u and evaluated using equation (2.4).

Table 4.1 Example of Dichotomous Mutation

x1	x2	r	x3	x4	x4	x5	x6
1	1	1	0	0	1	1	1
1	0	1	1	1	0	0	1
0	1	0	1	0	0	1	1
0	1	0	1	0	0	1	1
1	0	0	1	0	0	0	0
1	1	0	0	1	1	1	1

4.3 Unsupervised Text Feature Selection Using Memetic Optimization

The proposed method has four phases. First, the text documents corpus is transformed into numerical data in the pre-processing phase as was described in section 3.2 in chapter 3. Second, the resulting data are fed into the proposed feature selection method. Third, document clustering is performed using the resulting features. Finally, evaluation measures are used to assess the resulting clusters. In this subsection, the main steps of the proposed method of feature selection are described.

1) The population is first randomly initialized, and then the solutions are refined in each generation. Each solution consists of a random subset of features as shown in Table 4.2. All solutions are encoded using a binary encoding scheme. As the unsupervised feature selection is a discrete binary-based optimization problem, the values range of every solution is limited to only [0, 1]. Each solution is represented as a string of random binary values, and the length of each solution represents features numbers. The presence of a feature is represented by 1 while the absence of it is represented by 0. Although DE uses floating numbers to generate the initial solutions all the values will be rounded to their nearest integers, which are limited to 0 and 1.

2) The fitness calculation is performed first using the MAD. The mean absolute deviation is the average distance runs between each feature and its mean. The MAD fitness is calculated using these steps:

Step 1: compute the average of all features through all documents.
Step 2: compute how far each feature is from the mean using positive distances through the absolute deviations.
Step 3: sum out all resulted deviations.
Step 4: divide the sum by the number of the non-zero features.
Below is explained how the MAD is computed as shown in the equation (4.3).

$$MAD_i = \sum_{i=}^{m} \sum_{j=1}^{n} \frac{\left| f_{i,j} - \bar{f}_i \right|}{n} \quad \textit{Equation (4.3)}$$

where

m is the number of features (one valued in any particular solution).

$f_{i,j}$ is the value of feature i that appears in document j.

n is the number of documents containing feature i.

$\bar{f}_i$ is the average number of features appearing in documents n which is calculated as shown in the equation (4.4).

$$\bar{f}_i = \frac{\sum_{j=1}^{n} f_{i,j}}{a_i} \qquad \textit{Equation (4.4)}$$

The reason behind using the MAD fitness function is to find the score of each feature and to find its distance from the mean values of that feature in all documents with no consideration to the original class labels using a data-driven scheme.

3) Each solution is modified using the DE mutation and crossover operators as shown in section 4.2.

4) Simulated annealing (SA) is used as a local search modifier; it resembles the meta-heuristic operators when applied as a local search in the memetic search. Almost the same idea of using mutation and crossover is followed in the local search. The solution chosen for the local search will have the highest MAD fitness value. SA is used to guide the DE search in the search space. It accepts all new solutions as long as the temperature is high, meaning a random-like search is conducted when the temperature reaches a certain degree. In contrast, when the temperature cools down to near 0, the acceptability of

solutions will be reduced, because the movement of atoms become more restricted. Using the synergy of melting metals in SA works on a particular solution to search in the vicinity of its neighboring area.

A neighboring solution will be created after perturbation of the solution undergoing the local search using SA. The perturbed solution could be considered as a neighboring solution to the original one. The MAD scores of both the original and perturbed solution are calculated. If the neighboring solution outperforms the original, this case is always accepted. However, when the neighboring solution is worse, the acceptance of the new solution would be determined by a particular probability, the Boltzmann probability, which equals $B = e^{-\theta/T}$ where θ is the difference of the original and the perturbed solution while T is the temperature parameter whose value decreases while the search advances. Figure 4.1 shows the use of SA within the DESA and DDESA feature selection. The control parameters required for the SA are the initial temperature (T_0) and the cooling schedule (T), adopted from (Mafarja and Mirjalili 2017).

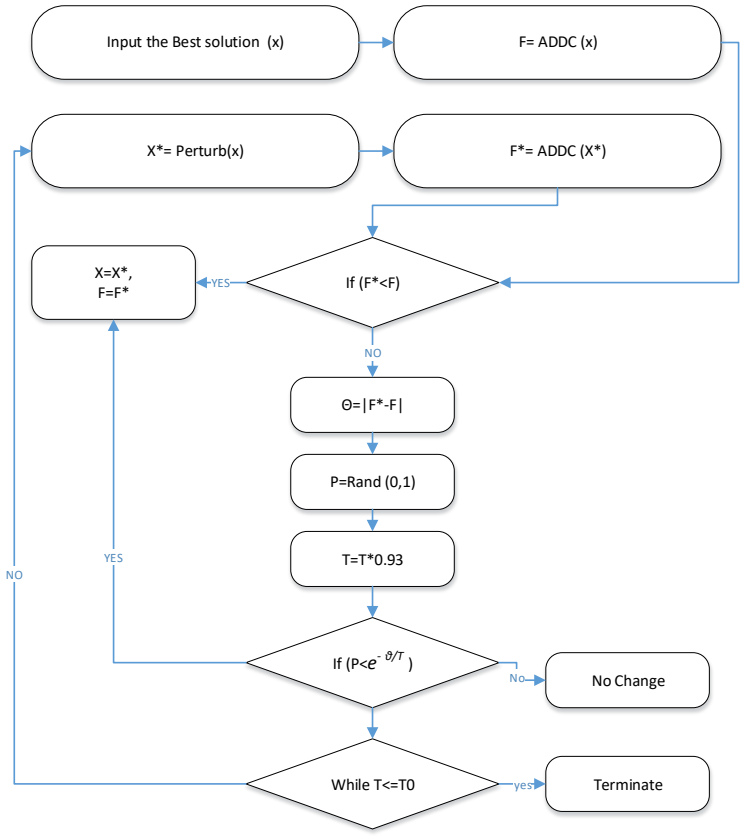

Figure 4.1 Local Search using SA

5) After steps 3 and 4, the new solution will be re-evaluated using the fitness function. Its fitness will be compared with the older solution using the following equation(4.5).

$$x_i^{g+1} = \begin{cases} s_{new_i}^{\ g} & if \quad fitness(s_{old_i}^{\ g}) < fitness(s_{new_i}^{\ g}) \\ s_{old_i}^{\ g} & if \quad fitness(s_{new_i}^{\ g}) <= fitness(s_{old_i}^{\ g}) \end{cases}$$

Equation (4.5)

where $s_{new,}$ is the newly generated solution and s_{old} is the old solution, while g is the generation number and x^{g+1} is the solution that survived into the next generation.

Table 4.2 shows an example of the representation of each solution in the initial population of the features. In this table the rows (S_i) represents the solutions where each solution is a vector that has a number of indexes (n). Each index represents the existence or the absence of that feature. In the case that the feature exists, the index will be 1 whereas 0 represents the absence of that feature. In the last column of Table 4.2 the fitness values of each solution were reported. The number of the fitness values *(MAD_i)* equals to the number of the solutions (S_i). It is important to mention that the number of *MAD* is a maximization optimizer. It means that the quality of the corresponding solution increases when the *MAD* score increases.

Table 4.2 Initial Population of Features, Each Row (S_i) is a Solution, and Each Column is a Feature

Solution	*Index*$_1$	*Index*$_2$	*Index*$_3$	...		*Index* $_n$	*Fitness*
S_1	0	1	1			1	MAD_1
S_2	1	0	1			1	MAD_2

S_3	1	1	1			0	MAD_3
							.
							.
S_n	0	0	0			1	MAD_n

The evaluation measures are important for observing the performance of each feature selection method and its effect on the clustering algorithm. Two types of measurements are used in the experiments: internal and external evaluation measures (Aggarwal and Reddy 2013). The F-macro and F-micro are used as external measures, and the internal measure is the Average-Document-Distance-to-the-Cluster-Centroid (ADDC) (Al-Jadir, Wong et al. 2017) as explained in chapter 3. It is noteworthy that the F-micro resembles the F-measure reported in the previous chapter. The next section reports a thorough analysis that was conducted by observing the maximization of the F-scores and the minimization of the ADDC measure using different feature selection methods. The relationship between the number of features and F-measure is also considered as these factors directly affects the performance. Thus, The Reduction Rate (RR) is also used as a measure to observe how many irrelevant features are dropped in relation to internal and external measurements. The RR can be calculated as is shown in equation (4.6)

$$RR = 1 - \frac{m}{n} \qquad \textit{Equation (4.6)}$$

where, *RR* is the reduction rate, *m* is the total number of features after applying the feature selection, and *n* is the number of original features.

The F-macro and F-micro (F-measure) measures, are calculated after computing the precision (P) and the recall(R) the two measurements that are extensively used in the Information Retrieval. P is the proportion of documents in group A and still in class B whereas R is the proportion of documents in class B and group A. The precision and recall measures are calculated as is shown in equations (4.7) and (4.8) and in equations (4.9) and (4.10), we obtain the F-micro and F-macro, respectively.

$$p(x, y) = \frac{n(A, B)}{n(A)} \qquad \text{Equation (4.7)}$$

$$R(x, y) = \frac{n(A, B)}{n(B)}$$ Equation (4.8)

$$F_{micro}(A, B) = \frac{2P(A, B)R(A, B)}{P(A, B) + R(A, B)}$$ Equation (4.9)

$$F_{macro} = \frac{\sum_{i=1}^{k} F_i}{k}$$ Equation (4.10)

Figure 4.2 shows the entire architecture of the proposed method for the unsupervised feature selection.

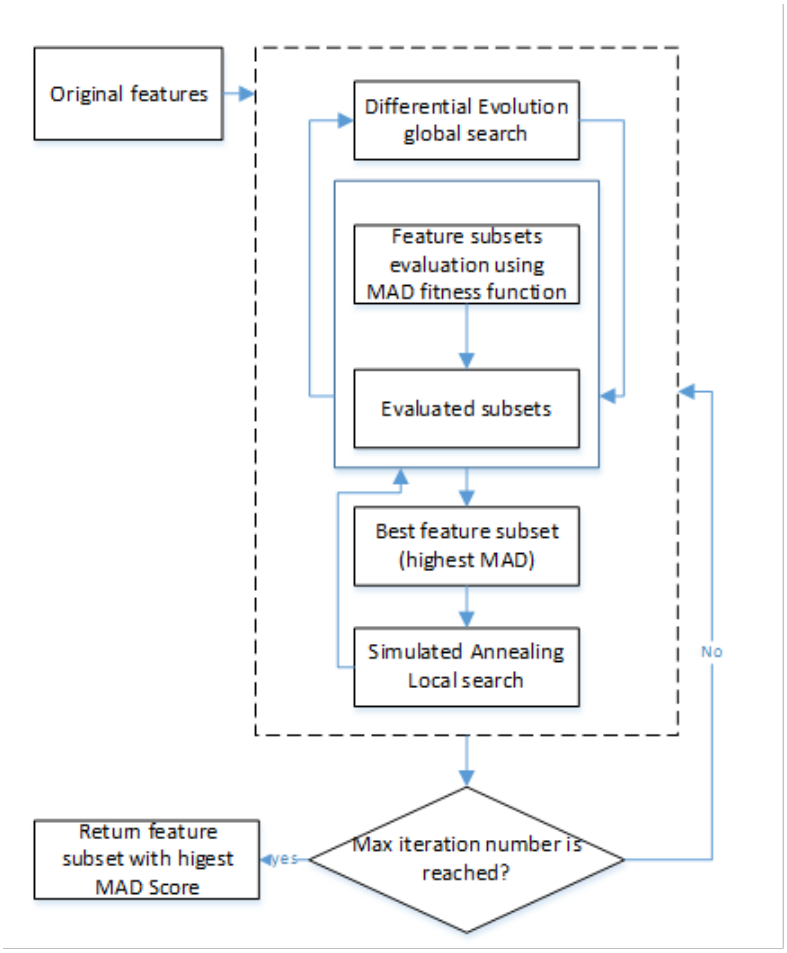

Figure 4.2 Memetic Unsupervised Feature Selection

An example of how the fitness function MAD is calculated for each solution and how the features are reduced is explained below. If there is a document set with eight documents and

seven features, the features are produced after using the Term Frequency. Inverse Document Frequency TF.IDF weight scheme (explained in chapter 2). Each weight is normalized between 0 and 1. The document corpus will be represented as shown in Table 4.3.

Table 4.3 Features Representation

	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**
Doc 1	0.1	0.2	1	0	0.87	0.21	0.3
Doc 2	0.9	0.9	0	0	0	1	0
Doc 3	0	0	0.8	0	0	0.98	0
Doc 4	0	0	0	0	0	0	0
Doc 5	0	0	0	0.1	0	0	0
Doc 6	0	1	0	0.98	1	0	0
Doc 7	1	0.98	0.8	1	0.9	0	0
Doc 8	0	1	0.55	0.8	0.3	0	0.56

As explained earlier, each solution is considered a binary string where each feature is identified by an *index.* When the index value equals 1, it means the feature is selected, whereas the 0 value means this feature is discarded. In the example shown in Table 4.2, the length of each solution would be 7 (the same size of features). Table 4.4 represents a random solution.

Table 4.4 An Example Solution

F1	F2	F3	F4	F5	F6	F7
0	1	1	0	1	0	1

The example solution in Table 4.4 shows that F1, F4, F6 feature are not selected while other features are selected as their index values equal 1. In order to measure the viability of the solution shown in Table 4.4, the MAD is calculated by considering the selected features (i.e. F2, F3, F5, and F7). Table 4.5 shows the values of the selected features throughout the 8 document.

Before calculating the MAD for the solution shown in Table 4.4, documents are discarded when the features equal 0 (Table 4.5). Thus, the average values of these features will be calculated according to the non-zero values. For instance, for F2 the values used to compute the mean are [0.2, 0.9, 1, 0.98, 1]; for F3 the values used to compute the mean are [1, 0.8, 0.8, 0.55]; for F5 the values used to compute the mean value are [0.87, 1, 0.9, 0.3]; and the values

used to compute the mean value of F7 are [0.3, 0.65]. Thus, the average value of each will be F2 = 0.77, F3 = 0.78, F5 = 0.76, F7 = 0.45.

Table 4.5 Features Selected in the Example Solution

	F2	F3	F5	F7
Doc 1	0.2	1	0.87	0.3
Doc 2	0.9	0	0	0
Doc 3	0	0.8	0	0
Doc 4	0	0	0	0
Doc 5	0	0	0	0
Doc 6	1	0	1	0
Doc 7	0.98	0.8	0.9	0
Doc 8	1	0.55	0.3	0.56

According to the MAD equation shown earlier, the values of the MAD scores can be calculated as follows:

For F2: |0.2-0.77|+|0.9-0.77|+|1-0.77|+|0.98-0.77|+|1-0.77|/5

= 0.75+0.13+0.23+0.21+0.23/5 = 0.275

For F3: |1-0.78|+|0.8-0.78|+|0.8-0.78|+|0.55-0.78|/4

= 0.22+0.02 +0.02+0.23/4 = 0.49/4 = 0.1225

For F5: |0.87-0.76|+|1-0.76|+|0.9-0.76|+|0.3-0.76|/4

= 0.11+0.24+0.14+0.46/4 = 0.2375

For F7: |0.3-0.45|+|0.65-0.45|/2 = 0.15+0.2/2 = 0.085

MAD= 0.275+0.1225+0.2375+0.085 = 0.75

Thus, the final MAD score of the solution shown in Table 4.4 is 0.75, which equals the sum of all the values resulting from the calculated values of features F2, F3, F5, and F7. It is

noteworthy to mention that the optimization of the MAD among features is a maximization problem: whenever the MAD score increases, the quality of the corresponding solution increases as well.

4.4 Test Results and Experimental Strategy

The tests were conducted by using the retrieved features for the *k*-means clustering and the same test strategy to test the method presented in chapter 3. The datasets presented in chapter 3 are used again for the test reported in this chapter, but the labels are not used. For the purpose of comparison, all features (ALL), FS Harmony Search Document Clustering (FSHSTC) (Abualigah, Khader et al. 2016), and the Feature Selection Genetic Algorithm Document Clustering (FSGATC) methods are used. The ALL represents the original feature space without reduction, and FSHSTC and FSGATC are two recently proposed unsupervised feature selection methods. Moreover, the native DE is also used as a FS method in the comparisons. Each of these are compared with the DESA and DDESA methods.

An indirect evaluation of features is conducted with the use of clustering evaluation measures after using the *k*-means, as seen in Table 4.6. The number of runs of the *k*-means is set to be more than one run for the same reasons mentioned in chapter 3. This number can be increased or decreased. Consequently, taking the average of all the runs is more reliable than dependence on only one run, because the *k*-means is highly sensitive to the initial centroid representation. On the other hand, a direct evaluation of feature subsets via *RR* also reported in Table 4.7.

Table 4.6 Internal and External Evaluation Measures

		Minimum			Maximum			Mean		
	Method	F-Macro	F-Micro	ADDC	F-Macro	F-Micro	ADDC	F-Macro	F-Micro	ADDC
6Events	All	0.346	0.371	0.503	0.750	0.779	0.573	0.623	0.655	0.546
	DE	0.238	0.259	0.539	0.781	0.803	0.584	0.626	0.665	0.561
	FSGATC	0.335	0.359	0.543	0.876	0.889	0.587	0.659	0.693	0.567
	FSHSTC	0.371	0.388	0.525	0.655	0.701	**0.558**	0.542	0.582	0.544
	DESA	0.541	0.568	0.539	0.860	0.870	0.596	0.710	0.732	0.577
	DDESA	**0.573**	**0.608**	0.554	**0.887**	**0.897**	0.629	**0.761**	**0.779**	0.491
Classic 3	All	0.416	0.484	0.524	0.735	0.766	0.627	0.617	0.661	0.595
	DE	0.462	0.510	0.509	0.691	0.732	0.610	0.553	0.587	0.583
	FSGATC	0.399	0.476	0.470	0.599	0.676	0.791	0.473	0.533	0.776

	FSHSTC	0.492	0.528	0.620	0.716	0.756	0.694	0.597	0.637	0.654
	DESA	0.416	0.484	0.484	0.760	**0.786**	0.552	0.639	0.691	**0.509**
	DDESA	**0.606**	**0.567**	**0.424**	**0.795**	0.767	**0.486**	**0.717**	**0.691**	**0.449**
Pair of 20news	All	0.515	0.673	0.539	0.520	0.675	0.605	0.519	0.675	0.591
	DE	0.538	0.684	0.649	0.543	0.686	0.670	0.542	0.686	0.666
	FSGATC	0.505	0.669	0.422	0.515	0.673	0.566	0.511	0.672	0.498
	FSHSTC	0.524	0.678	0.685	0.529	0.680	0.720	0.528	0.679	0.710
	DESA	0.524	**0.678**	0.428	0.524	**0.678**	0.428	0.524	**0.678**	0.428
	DDESA	**0.560**	0.675	**0.304**	**0.560**	0.675	**0.304**	**0.560**	0.675	**0.304**
Reuters	All	0.177	0.194	0.403	0.559	0.616	0.536	0.280	0.308	0.476
	DE	0.205	0.212	0.395	0.345	0.366	0.477	0.260	0.283	0.438
	FSGATC	0.181	0.196	0.450	**0.426**	**0.467**	0.493	0.290	0.320	0.468
	FSHSTC	0.179	0.232	0.416	0.280	0.309	0.504	0.238	0.267	0.474
	DESA	0.200	0.228	0.396	0.294	0.331	0.466	0.241	0.269	0.420
	DDESA	**0.308**	**0.310**	**0.157**	0.398	0.387	**0.181**	**0.355**	**0.334**	**0.163**
Ten Types	All	0.115	0.145	0.413	0.386	0.424	0.559	0.261	0.296	0.499
	DE	0.104	0.135	0.434	0.307	0.326	**0.526**	0.196	0.229	0.477
	FSGATC	0.104	0.135	0.363	0.383	0.410	0.606	0.220	0.241	0.454
	FSHSTC	0.143	**0.160**	0.501	0.334	0.349	0.609	0.215	0.237	0.549
	DESA	0.115	0.140	0.476	**0.425**	**0.453**	0.560	0.214	0.255	0.528
	DDESA	**0.235**	**0.225**	**0.308**	0.392	0.406	0.540	**0.340**	**0.324**	**0.386**

Table 4.7 Reduction Rate Table

Method	Dataset	Old Features	New Features	Reduction Rate
DE	6 Events	3863	1936	0.5
	Classic 3	362	141	0.61
	Pair of 20news	9496	4688	0.51
	Reuters	507	189	0.63
	Ten Types	15600	3697	0.76
FSGATC	6 Events	3863	1920	0.5
	Classic 3	362	183	0.49
	Pair of 20news	9496	4758	0.5
	Reuters	507	235	0.54
	Ten Types	15600	3670	0.76
FSHSTC	6 Events	3863	1924	0.5
	Classic 3	362	186	0.49
	Pair of 20news	9496	4770	0.5
	Reuters	507	250	0.51
	Ten Types	15600	3710	0.76
DESA	6 Events	3863	1910	**0.51**
	Classic 3	362	108	**0.7**
	Pair of 20news	9496	4600	**0.52**
	Reuters	**507**	174	**0.66**
	Ten Types	15600	3653	**0.77**
DDESA	6 Events	3863	**1892**	**0.51**
	Classic 3	362	**94**	**0.74**
	Pair of 20news	9496	**3780**	**0.6**
	Reuters	507	**160**	**0.68**
	Ten Types	15600	**3528**	**0.77**

The *RR*, the fitness convergence, and the internal and external clustering evaluation measures can give a complete view of the reduced features subset. Comparing the F-macro, F-micro, and ADDC results with the *RR* can demonstrate the effectiveness of feature selection methods used. Theoretically, if a particular method achieved a higher *RR* with higher F-macro and F-micro scores, it can be concluded that the method is more effective than one that might achieve comparable F-scores or ADDC score while, at the same time, achieving features with less *RR*.

4.5 F-Scores and ADDC Measure

In this subsection, the RR, the ADDC and the F-macro and F-micro (F-measure) results are given. It is important to mention that the F-macro and F-micro are referred to as F-scores. In

the tests, the highest F-scores after the clustering indicates the higher accuracy of the resulting features. On the other hand, the ADDC score is used to measure the compactness of the clusters, as used to evaluate the clusters in the tests reported in the previous chapter. In the tests reported in this chapter, we are looking for the features that shorten distances between documents with any particular cluster.

Ideally, the ADDC score should be minimized while the F-scores should be maximized. The relationship between the internal and external evaluation measures represented by the ADDC and the F-scores seems to have complex incremental and decrement trends. From experience, it can be said that the performance of these measures can be classified into three categories. First is the ideal case where the internal (ADDC) is minimized and the external (F-scores) is maximized with the same amount but in opposite directions. The second case occurs when the internal measure remains the same or slightly fluctuates while the F-scores move significantly. This case can be accepted because the F-scores variation can give a clue of the positive or negative algorithm's performance despite the stability (or the slight variation) of the internal measure.

The last case, which is the worst case, occurs when both criteria have similar trends. In other words, when the ADDC and the F-scores either increase or decrease both in one direction. This implies the method that exhibits such behavior could be considered an ill-performed method due to the instability of the internal measure. Further details about the relationship between internal and external evaluation measures are explained in chapter 3.

It follows that the internal and external measures have two different goals in data-driven problems such as document clustering, which means both the internal and external measures should be taken into account. The relationship between these two measures needs more in-depth research to understand their behavior and how results can be predicted for one measure by observing the performance of the other. Studying this relationship is beyond the scope of this present research and thesis. The explanation above is intended to make clear the nature of the performance of both F-scores and the ADDC in the text feature selection as reported in Table 4.6.

Table 4.6, lists the values of the minimum, maximum and average ADDC and F-scores for different runs. The 6-events crimes dataset has the minimum, maximum and the average scores of the F-scores and the ADDC. It can be seen that the results of the F-micro and F-macro values of all runs for the proposed DDESA method are 0.573 and 0.608 for the minimum, 0.886 and

0.897 for the maximum, and 0.761 and 0.778 for the average values. All these scores are higher than those obtained by other competent methods including the ALL. The ADDC measure of the average values of the six events crimes obtained by the DESA is slightly higher than the average ADDC obtained by the DESA method. The slight increase of the ADDC is acceptable as the corresponding F-scores are much higher than those achieved by other methods in relation to their ADDC scores.

The results for Classic 3 are similar to the 6-events crimes dataset. Again, it can be seen that the DDESA achieved higher results of clustering in terms of the F-scores in comparison to the other methods including the ALL method. The ADDC values of the DDESA method also show an improvement by obtaining the least values among other methods. However, the ADDC values are still insignificant when compared to the other methods. Therefore, the use of the external measures will be considered.

The Pair of the 20news group is the third dataset used. It is distinctive and different from other datasets because it has only two classes. Undoubtedly, the lower class number makes it much easier for the clustering algorithm to predict the right class for each document without the confusion of dividing features into multiple classes. Therefore, it can be seen that all the feature selection methods tested have compatible behaviors. There are no significant changes in the performance of the DESA method and the other methods in terms of the external and the internal measures. Due to the lower class number, the feature selection does not seem to play a notable role.

Reuters is one of the widely used benchmark datasets. In this dataset, the values achieved by the DDESA are comparable to those achieved by the FSGATC method for the maximum. Correspondingly, the ADDC values of the DDESA were less than the other methods including the ALL with the minimum, maximum and average values.

Finally, the Ten Types of crimes values of the DDESA are comparable to the other methods in terms of the F-scores and ADDC including the ALL method.

The results presented in Table 4.6 make it possible to conclude that using feature selection can improve the performance of the internal and external evaluation measures. However, due to the existence of some similarities between the results obtained using different feature selection methods in terms of the internal evaluation measure ADDC, it becomes necessary to use another measurement that can determine the effectiveness of each method in relation to the information shown in Table 4.6. Therefore, the use of the RR of features can be used in conjunction with the information provided in Table 4.7 to determine which method achieved the highest F-scores, the lowest ADDC, and the highest RR.

In this subsection, the RR of each feature selection method is explained. Table 4.7 lists the total number of the original features, the total number of the selected features and the relationship between them, which was explained earlier. Table 4.7 shows the RR of both DDESA and DESA methods exceed those achieved by other methods. The clustering performance after using the DESA FS in both versions remains at an equal or better performance state than using the ALL features or using other state-of-the-art methods as shown in Table 4.6. Furthermore, it can be noted in Table 4.7 that DESA and DDESA RRs are more than the half of the features ranging between 0.51 and 0.77. The F-scores and the ADDC achieved by the DDESA method are still comparable with the scores achieved by the other methods including the ALL method.

4.6 Summary

This chapter presented a feature selection method capable of detecting informative features by using the hybridization of a wrapper and filter methods in an unsupervised memetic feature selection manner. The proposed method combines SA to the global search using the DE. SA is used as a filter method to refine the best solution that resulted from the global search. Two versions of this hybridization were presented in this chapter. First, DESA, which uses the standard mutation of Differential Evolution. Second, DDESA, which uses the dichotomous mutation used in (Peng, Wu et al. 2016) to solve knapsack optimization problems. The DDESA outperformed the performance of DESA. However, both methods were compared against the DE wrapper, and also against two other state-of-the-art unsupervised feature selection methods, namely FSGATC and FSHSTC methods. The performance of the DDESA and other compared methods were evaluated indirectly via external and internal evaluation of the clustering. The RR was also taken into account; the RR measures the percentage of feature reduction for each method while MAD measures the convergence of each of the tested methods. The method that

achieves the highest RR and the highest MAD is considered the best. When it comes to the clustering results using feature subsets resulting from different methods, the minimum internal measure values using the ADDC and the maximum external measure values using the F-score indicate the superiority of that method. As mentioned earlier, the DDESA method achieved the highest F-scores in the majority of the datasets. It also achieved the least ADDC values with the majority of the datasets. The RR values also suggest that the DDESA outperformed other tested methods. Therefore, the DDESA method will be used in test for centroids allocation of document clustering of unlabeled document, reported in the next chapter.

Chapter 5

Document Clustering Using Memetic Optimization

5.1 Introduction

In this chapter, the same concept used for feature selection (chapter 4) is used, however the proposed memetic optimization techniques are intended to allocate clusters centroids optimally within the search space for document clustering instead of doing feature selection. In that case, the fitness function (e.g. distance measure) is used to find the distance between each cluster center and its surrounding documents. The aim of using memetic optimization here is to minimize distances between those cluster centers (centroids) and their relevant documents by positioning these centroids in the right location in the search space.

The usage of the memetic concept in feature selection differs from its usage in centroids allocation in many aspects. Both problems are considered NP-hard optimization problems, but the main distinction between them is in the calculation of their objective functions and local search methods. The main aim of this chapter is to present a method capable of allocating cluster centroids using memetic optimization, by using different combinations of global and local searchers in a memetic context. The first combination presented in this chapter is the use of the approximation methods via the CLS and the Discrete Differential Evolution (DDE) global search. The second combination is based on the gradient search using the *k*-means as a local search and the DHS as a global search. An extensive explanation is given regarding these two memetic combinations showing the advantages and the disadvantages of each in terms of the accuracy of the obtained results. The reason for this comparison is to find the most efficient method that provides optimal solutions to the problem of text document clustering.

The two combinations explained in this chapter are DEMC and MDHS. Later, two modified versions to the MDHS are presented; they are the CMDHS and the ACMDHS as there are two possible ways to enhance the MDHS. This chapter is constructed as follows.

Section 5.2. Intended to explain how the cluster centroids are generated and evaluated, which includes document corpus representation, solutions initialization, initial solution evaluation, centroids calculation and fitness evaluation.

Section 5.3. The clustering using HS is discussed as it is the base method of MDHS clustering methods presented in the next section.

Section 5.4. The main distinction between DEMC vs. Memetic Differential Evolution Harmony Search (MDHS) clustering is stated.

Sections 5.5 and 5.6. Both DEMC and document clustering using MDHS are detailed.

Section 5.7. An adaptive version of the Crossover MDHS is explained.

5.2 Documents, Solutions and Centroids Representation and Evaluation

This section is intended to highlight the important concepts used in optimizing cluster centroids for the proposed document clustering approaches presented in this chapter. This section will explain first the representation of the documents corpus. It also describes how the solutions are initialized, modified and evaluated. Furthermore, the centroids calculation process of each solution is explained.

5.2.1 Document Corpus Representation

The datasets are first uploaded in a text format. After pre-processing, datasets are transformed into a Term-Document Matrix format (TDM). It is also valid if the TDM matrix is expressed as a Document-Term Matrix (DTM). Table 5.1 shows an example of a dataset containing six documents, seven features (keywords) and three classes (these classes are only included in order to explain the concepts).

Table 5.1 Relationship Between Features, Documents, and Classes

		F1	F2	F3	F4	F5	F6	F7
Doc1	**c1**	0	0.1	0.2	0	0.2	1	0
Doc2	**c1**	0	1	2	0	0.2	1	0
Doc3	**c3**	0	3	0.1	0	0.4	0.5	0
Doc4	**c3**	0	3	1	0	0.4	0.5	0
Doc5	**c2**	7	0.1	2	0	0.1	0.2	0.1
Doc6	**c2**	7	0.9	2	0	0.1	0.2	0.1

As the main distinction between the clustering and classification problems is the availability of the class labels in classification problems, their existence is not necessary for any true clustering problems. However, with benchmark datasets the availability of these labels is important only for the purpose of system evaluation. This is essential to validate the system to be used later for unlabeled datasets. The actual representation of TDM used in the clustering systems is shown in Table 5.2.

Table 5.2 Small Dataset Without Class Labels

	F1	F2	F3	F4	F5	F6	F7
Doc1	0	0.1	0.2	0	0.2	1	0
Doc2	0	1	2	0	0.2	1	0
Doc3	0	3	0.1	0	0.4	0.5	0
Doc4	0	3	1	0	0.4	0.5	0
Doc5	7	0.1	2	0	0.1	0.2	0.1
Doc6	7	0.9	2	0	0.1	0.2	0.1

For text data produced from real world applications, a classified dataset such as the one presented in Table 5.1 is unlikely. Unsupervised learning problems are concerned with label-free data. For instance, the external evaluation measures are dependent on the availability of the original class labels. These measures match the new documents' configurations after clustering with the original ones. More researchers use this kind of evaluation because more accurate results can be achieved compared to internal measures such as the F-measure, to be explained later. The only limitation that restricts use of this measure is the unavailability of the previous categorization of the documents. Therefore, it can be used with benchmark datasets to validate the assessment performed by the internal measure. As the performance of these datasets is evaluated using the external measures and the internal measures, it becomes easier

to predict the accuracy of the clustering with other datasets when only the internal measures can be used, which assess only the generated clusters, depending on the intrinsic properties of them.

5.2.2 Solutions Initialization

The *'solution'* definition in optimization-based document clustering is a set of centroids that needs to be distributed accurately. To allocate the centroids efficiently, each centroid is supposed to be positioned at the nearest distance to all relevant documents. Such is the case of the example in Table 5.3, in which the number of documents is 7 (*n = 7)* and these documents need to be distributed among three clusters [c_1, c_2, c_3]. That can be stated as permutations of *n* documents allocated at a time to *r* clusters which can be represented in equation (5.1).

$$p(n,r) = \frac{n!}{(n-r)!} = \frac{7!}{(7-3)!} = \frac{5040}{24} = 210$$

Equation (5.1)

Thus, the number of possible solutions would be 210 for this small example, yet only one feasible solution should be considered from those solutions. This simple example is only intended to show the relationship between each document and its corresponding document. The increasing number of documents and centroids requires more intelligent methods to find the best centroids allocation. Consequently, the selection of the best solution will become more complicated. Therefore, intelligent methods such as the memetic algorithm could provide a faster convergence to the best solution (Neri and Cotta 2012). In addition, the problem is not only limited to the selection of the best solution in the search space, it is equally important to employ efficient techniques that are capable of modifying solutions on a local search basis. Such techniques should be capable of avoiding local optima where all solutions become non-productive. In typical optimization problems, a random initial population is first generated by using the *random number generator function*. For document clustering centroids allocation methods presented in this chapter, the initial population uses this technique. However, using other techniques to initialize the population could be more productive currently, but that exploration is beyond the scope of this present research. The size of the population and the initial assignments of documents are random and should be less than or equal to the desired number of clusters. The example in Table 5.3 shows only 10 sample solutions represented by

the columns *(sol1 . . . sol10)*. Each solution has the same number of documents in the original dataset.

5.2.3 Initial Solution Evaluation

A random initialization of solutions is first conducted (Table 5.3). The viability of each solution is calculated by using a fitness function.

Table 5.3 Solutions Matrix

	Doc1	**Doc 2**	**Doc 3**	**Doc 4**	**Doc 5**	**Doc 6**	**Doc 7**
sol1	2	1	1	3	3	1	3
sol2	2	3	3	3	2	1	1
sol3	1	1	1	1	2	2	2
sol4	1	2	3	2	1	1	3
sol5	2	1	2	1	3	3	3
sol6	3	2	3	3	2	3	3
sol7	3	1	1	2	2	2	1
sol8	1	2	2	3	2	2	1
sol9	2	3	1	1	2	2	2
so110	2	3	3	1	1	3	1
so111	1	2	1	1	1	2	2
so112	2	1	3	1	1	1	1

Each *sol* row in Table 5.3 represents a random solution that contains indices of clusters. For instance, the intersection of *(d1, sol4)* means that document *d1* belongs to *cluster index 1* while the intersection of *(d4, sol4)* means that document *d4* belongs to the second cluster and so on. The fitness function, which is used to evaluate the quality of each solution, aims to find the highest number of true positives by correctly allocating each document to its proper class. In the example in Table 5.3, if we assume that the right allocation of documents *is* $d_1 \in 2$, $d_2 \in 1$, $d_3 \in 3$, $d_4 \in 1$, $d_5 \in 6$, $d_6 \in 1$, that means the best solution vector is [2, 1, 3, 1, 6, 1], which obtains the least fitness function score (assuming that the least is the best). The fitness function, objective function or simply the cost are all interchangeable. The parameters required by the fitness function are the number of clusters, the original dataset shown in Table 5.2 and the initial solutions as reported in Table 5.3.

5.2.4 Centroids Calculation and Fitness Evaluation

This process uses the solutions existing in the solutions matrix sequentially to generate a specified number of centroids. For our example, each solution is used to create three centroid vectors. Each centroid vector has the same size of solutions. The use of optimization methods is vital to find the best solution that returns the best centroids. In order to perform the clustering of the documents, all document vectors stored in the TDM shown in Table 5.2 are compared to the centroids resulted from each solution. The comparison is based on distance measures. The comparison of *n* documents to *c* clusters (of one solution) is refered to as the fitness of score of that solution calculated by the ADDC. In other words, a good solution would generate good centroids; that solution minimizes the distance between each document to its corresponding centroids. We understand, then, that the locations of centroids are dynamic while documents are static in the search space.

The number of desired clusters will determine the size of the centroids matrix to be constructed. For instance, Table 5.4 shows what the centroid matrix would look like if three clusters were to be formed.

Table 5.4 Centroids Matrix

C1	3.5	2	1.5	0	2.5	2	3.5
C2	3.5	2	1.5	0	2.5	1.5	3.5
C3	0	1	3.5	0	2	2.5	1.5

Thus, the main idea behind using optimization methods is to find the best positions of centroids by adjusting their location in every iteration. The fitness function measures the effectiveness of each solution. The solution that can update the centroids with the highest fitness score is considered the most fitted, and it will be chosen for the next round. Algorithm 5.1 shows the centroid's calculation steps.

Algorithm 5.1. The centroid calculation

1. **Input** (*C, row, TDM, x*)
2. *C:* number of clusters,
3. *Nc:* number of documents
4. *TDM:* the documents matrix
5. *x:* the solution that is wanted to be evaluated.
6. **Output** (the centroids matrix)
7. Centroids (:,:) =0 // initialize the centroids matrix as an empty matrix.
8. **for** *i*=1to *C* //the number of clusters(classes)
9. *w*=0; //counter
10. **for** *j*=1:*Nc*
11. **if** *x(:,j)=i*
12. *w=w+1;*
13. Centroids *(i,:)*=TDM *(j,:)*+Centroids *(i,:)*; // the centroids matrix C is represented in the above table
14. **end**
15. **end**
16. Centroids *(i,:)=C(i,:)/w*
17. **end**
18. **end** //algorithm

In algorithm 5.1 and next algorithms, the (:) notion refers to the all rows or columns. For instance, *x(:,j)* means for all rows of *x* select the j^{th} column.

5.3 Differential Evolution Memetic Clustering vs. Memetic Differential Evolution Harmony Search

The first memetic-based document clustering method introduced in this chapter uses the DE global search with a CLS. It performs step-wise optimization moves via the logistic function. This method, the DEMC, has been introduced in the present researcher's published paper (Al-Jadir, Wong et al. 2017). The test results show that the DEMC outperformed other clustering methods such as the Chaotic Gradient Artificial Bee Colony (CGABC), Differential Evolution Simulated Annealing (DESA) and the Differential Evolution K-Means (DEKM). However, the obtained results are regarded as not yet satisfactory. Therefore, a second approach, MDHS is proposed. Two different experiments tested this method. The proposed MDHS was tested without the differential crossover operator (resembling the standard DHS, explained extensively in chapter 2).

The proposed method is then modified to add the crossover to MDHS, named as Crossover Memetic Differential Harmony Search (CMDHS), which is used later to observe the effect on

the exploitation aspect of the DHS global search. The memetic clustering using MDHS or CMDHS methods is based on the idea of cooperation between the global search and local search performed by the *k*-means. The ability of the *k*-means to search the vicinity of local areas makes it ideal for enhancing the solutions produced by the DHS global search where the *k*-means performs the clustering using the best solution produced by the global search. Following this, a greedy selection is conducted to check if the new solution outperforms the best solution in the HM.

One more enhancement of the CMDHS, the parameter adaptation property, has been added. The use of the adaptive parameter settings can also enhance the performance in comparison to the performance of static parameter settings (Wang, Li et al. 2017). These improvements can all be combined to produce an efficient method for document clustering. An Adaptive Crossover Memetic Differential Harmony Search (ACMDHS) method was developed for the purpose of optimizing document clustering.

5.4 Differential Evolution Memetic Clustering

The DEMC method incorporates a DE global search with a chaotic logistic local search. DE has a self-organization behavior, and performs well for many multi-modal optimization problems (Chunming, Yadong et al. 2017). The method that forms a part of this present research utilizes the DE combined with a CLS local search. The shrinking strategy is used to reduce CLS execution as the generation number increases to improve its efficiency (Jia, Zheng et al. 2011).

5.4.1 Differential Evolution Clustering Global Search

Like many other evolutionary methods of optimization, DE has two phases: population initialization and agents' evolution. However, the present research integrates those two phases with a local search, forming three phases instead of just two. The steps of the proposed DEMC are as follows.

A. The initialization phase, is conducted as described in chapter 4, section 4.2. However, the solutions boundary in this method will be set to match the number of clusters.

B. For the solutions evaluation, the same ADDC and cosine equations explained in previous chapters are used.

C. In order to update the solutions, the mutation and crossover steps will be used as was explained in section 4.2 in chapter 4. Figure 5.1 and Figure 5.2 show examples of the differential evolutions and mutations used for a clustering problem of three clusters. In Figure 5.1 it is clear that x_{r1}, xr_2 and x_{r3} are the input vectors. Vectors x_{r1} and x_{r2} are subtracted and the result is added to x_{r3}. The absolute values of the resulted vector is taken and finally the values were rounded.

D. In the selection step, the most fitted solutions resulting from the mutation, crossover or local search are substituted by the least fitted ones.

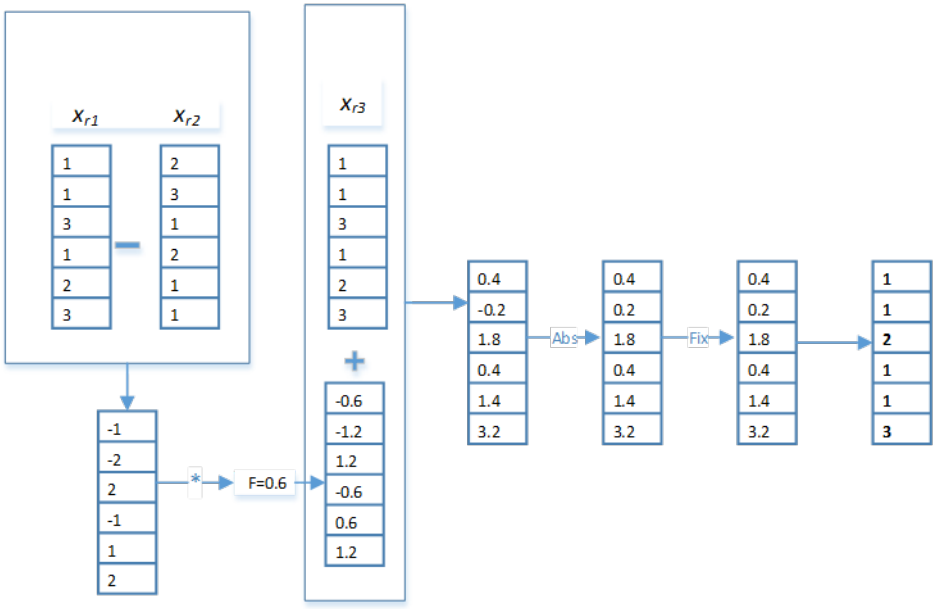

Figure 5.1 DE Mutation Example

5.4.2 Chaotic Logistic Search Shrinking Strategy

Chaos is typically a non-linear phenomenon that has two properties: randomicity and ergodicity. Due to randomicity and ergodicity, chaotic systems are updated randomly. This property of chaos systems could be used to optimize the distribution of the cluster centroids by optimizing the ADDC fitness function. However, chaos optimization performs efficiently with limited search spaces, whereas its performance could take a longer time with larger search spaces.

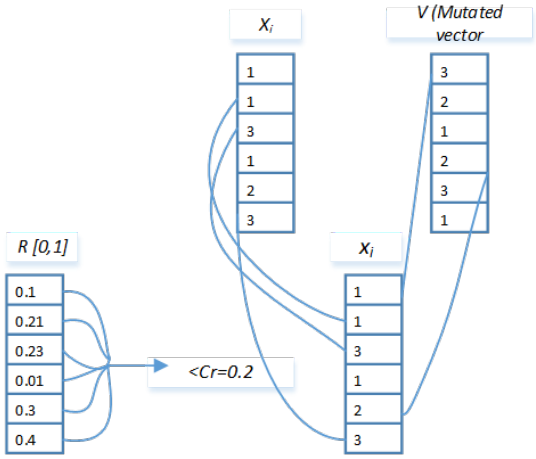

Figure 5.2 DE Crossover Example

Thus, chaotic optimization is usually combined with global search optimization methods to converge faster (Gwo-Ching and Ta-Peng 2006). In DEMC, the CLS is the final step of the clustering method. It is used to refine the best resulting solution(s). Algorithm 5.2 shows the main steps of applying a CLS local search while equation (5.2) represents the logistic function responsible for updating β, which is the chaotic variable required to update solutions.

$$\beta_j^{k+1} = \mu\beta_j^k(1-\beta_j^k) \qquad \textit{Equation (5.2)}$$

where β_j^k is randomly distributed number between [0,1] which is a particular chaotic variable in the j^{th} iteration, and μ is a control parameter used to update β_j^k.

The current research method being tested used the same shrinking strategy used in (Jia, Zheng et al. 2011) to avoid any premature convergence, in order to stabilize the algorithm within later generations. Shrinking the search to only local spaces can be useful and that is conducted by applying equation (5.3) that calculates the shrinking factor λ.

$$\lambda = 1 - \left|\frac{n-1}{n}\right|^q \qquad \textit{Equation (5.3)}$$

After β, λ have been calculated, the new solution x^{g+1} is generated by equation (5.4).

$$x^{g+1} = (1-\lambda)x^{g} + \lambda\beta^{k+1}$$

Equation(5.4)

where n is the current local search iteration number while q is the shrinking exponent that determines the speed of convergence. When q becomes larger, the convergence will go slower.

5.4.3 Differential Evolution Memetic Clustering Evaluation Metrics

For the evaluation of the experimental results, internal and external clustering evaluation measures are used (Forsati, Mahdavi et al. 2013). The internal evaluation measure is responsible for assessing the internal characteristics of resulting clusters, computed by the fitness function; the external measure computes the matching degree between classes and their corresponding clusters. The macro-F and micro-F measures (F-measure) are used as an external evaluation measure (Bharti and Singh 2016), and the ADDC is used as an internal measure. The values of the macro-F and micro-F range between [0,1] and the highest set of these values refers to the best set of the resulting clusters and vice versa, while the ADDC looks for the minimum, as seen in chapter 3.

Algorithm 5.2 Chaotic Logistic Search local search

1. Input: Objective function ADDC (x), Harmony Memory size HMS, Local search iteration number, the solutions size *Dim*, initialized random number $\beta^0(0,1)/0.25,0.5,0.75$
2. Begin
3. $i = 0$
4. while $i < iter$ do
5. Select the best solution in the population x_{ij}
6. for $j = 1$ to *Dim* do
7. $x`_{ij}{}^{(g)} = (1-\lambda)x_{ij}{}^{(g)} + \lambda\beta^{(g)}$
8. end for
9. compute λ^{g+1}
10. compute $\beta_j{}^{g+1}$
11. $i = i + +$
12. if $f(x_{ij}`) < f(x_{ij})$ then
13. $x_i = x`_{ij}$
14. $f(x_i) = f(x_{ij}`)$
15. end if

16. end while
17. the new *x* will be updated in the HMS along with its fitness function
18. end

5.4.4 Differential Evolution Memetic Clustering Experimental Results

The present experiments used the same datasets reported in chapter 3. Those datasets are diversified in terms of their number of classes, document lengths, topics and the number of documents. In order to verify the accuracy of the results using the DEMC method, a comparison was made with a number of other variants, namely the classical DE, the DESA (Saruhan 2014) and DEKM (Kwedlo 2011). DESA was used for a non-clustering purpose while DEKM was used for data clustering. However, these two methods were adapted for the present research as document clustering methods. Furthermore, traditional clustering using the *k*-means method is also used in the comparison.

Finally, DEMC is also compared to the method proposed in (Bharti and Singh 2016), the CGABC method. The CGABC parameters are the colony size (*CS*), set to 160; a_1,b_1 were limited to [−1.2, 1.2]; the number of food sources is $CS/2=80$; ϕ is ranged between [0,1.5] and φ is between [-1,1] while the G-cycle is 10. All these parameter values were set in the original paper. In the experiments, the algorithms were tested 20 times to reduce the effect of the random nature of the *k*-means and the random initial generations of the other methods.

The average results of all runs are taken in to account for both F-scores and ADDC. Tables 5.5, 5.6 and 5.7 report the values of the external and internal evaluation (fitness values) measures respectively after applying the algorithms on the five datasets. In Tables 5.5 and 5.6, it can be observed that DEMC provided better scores in comparison to other competent methods. Nonetheless, only in D4 can it be observed that the CGABC method achieved better results in terms of the evaluation measures used. This might be because of a small number of classes in the D4 dataset. Although the performance of CGABC was higher than the DEMC for that dataset, the difference is insignificant.

The proposed DEMC obtained better results for other datasets as shown in Table 5.5 and Table 5.6. On the other hand, Table 5.7 shows that the DESA, DE, CGABC methods, as well as the method being tested, have compatible ADDC results, whereas both DEKM and *k*-means methods obtained almost similar results. From the internal measure point of view, it is unclear which method performed better. However, by following the rules established in chapter 3

regarding the relationship between the internal and external evaluation measures, the external measure is considered a decisive measure in that situation. Indeed, the external measure uses the actual truth data (class labels) and that would give a more accurate description of the formed clusters than the internal measures. This also agrees with the use of external measures by other researchers in the field of document clustering such as in (Forsati, Mahdavi et al. 2013) and (Bharti and Singh 2016). In the next section, another method that aims to minimize the ADDC and maximize the F-scores will be discussed.

Table 5.5 Clustering Results Using F-macro Measure

	DESA	**DEKM**	**DEMC**	**DE**	**CGABC**	***k*-means**
D1	0.1764	0.5986	**0.8795**	0.7230	0.7927	0.6502
D2	0.8579	0.7295	**0.9470**	0.9454	0.9408	0.7211
D3	0.5976	0.5196	**0.9875**	0.5493	0.6330	0.0634
D4	0.6024	0.3294	0.9849	0.0036	**0.9894**	0.5196
D5	0.3333	0.0628	**0.5813**	0.5523	0.4938	0.0550

Table 5.6 Clustering Results Using F-micro Measure

	DESA	**DEKM**	**DEMC**	**DE**	**CGABC**	***k*-means**
D1	0.2266	0.6291	**0.8900**	0.7545	0.8265	0.6800
D2	0.8679	0.7711	**0.9480**	0.9465	0.9428	0.7509
D3	0.6024	0.6755	**0.9877**	0.5830	0.6617	0.0698
D4	0.6020	0.3635	0.9853	0.0036	**0.9896**	0.6755
D5	0.3333	0.0695	**0.6144**	0.5865	0.5341	0.0698

Table 5.7 ADDC Values

	DESA	DEKM	DEMC	DE	CGABC
D1	**0.5638**	0.5700	0.7217	0.7039	0.7222
D2	0.8591	0.7982	0.8605	0.8605	0.8606
D3	0.8326	0.6046	0.8445	0.8450	0.7827
D4	0.8265	0.6013	0.8304	0.8130	0.8450
D5	0.8562	0.4069	0.7827	0.7830	0.8306

5.5 Document clustering using Memetic Differential Harmony Search

The HS was proposed first by Geem (Geem, Kim et al. 2001) as an optimization method. In chapter 2 it was extensively explained in section 2.3.1.3. The population in a HS is represented as a set of harmonies stored in a data structure, such as a matrix, called Harmony Memory (HM), and each harmony represents one solution. The method uses the following parameters: HMS, that is the number of solutions in HM, HMCR that controls the selection of the solutions from HM, and PAR that resembles mutation in the GA. Furthermore, the BW parameter is used to modify harmonies. The optimal value of BW is not yet determined. Therefore, DE operators could eliminate the need to set up the BW parameters. In contrast to the traditional HS explained earlier, the proposed method substitutes DE mutation for the PAR phase. In contrast to the conventional PAR step that changes only one solution at a time with no interaction with other solutions, the use of DE operators could find the relationship between several solutions.

A modification to DHS, using the differential binomial crossover (Lin, Qing et al. 2011), could contribute to further exploitation than by utilizing the mutation alone. Thus, a modified version of DHS was tested. That modification was named Crossover CDHS (CDHS), and is shown in Algorithm 5.3. Both DHS and CDHS are global search methods that mean these two methods might have a limited ability in a local search. The memetic optimization could be used to enhance their local search capability for a clustering centroid distribution. That involves modifying each solution locally using *k*-means to rearrange the cluster centroids.

5.5.1 Memetic Differential Harmony Search vs. Crossover Memetic Differential Harmony Search

The combination of DHS and CDHS with the local search resulted in the MDHS and the CMDHS. The main steps are summarized below:

A. The first step begins by transforming the text documents into a numeric format using the same techniques explained in section 3.1.

B. The HM in both DHS and CDHS is initialized using the same technique as the initialization of the standard HS. The method sets a predefined number of clusters: *c*. HM contains a list of potential solutions, which is normally represented by a matrix as shown in Figure 5.3. Each row of the matrix is a solution that includes an assignment

of documents to cluster numbers. For instance, if there are six clusters, values in a solution will be between 1 and 6. The length of each solution is the same as the number of the documents while the number of solutions in HM or HMS can be set to any number, normally to twice the number of clusters.

$$\mathrm{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{\mathrm{HMS}-1} & x_2^{\mathrm{HMS}-1} & \cdots & x_{N-1}^{\mathrm{HMS}-1} & x_N^{\mathrm{HMS}-1} \\ x_1^{\mathrm{HMS}} & x_2^{\mathrm{HMS}} & \cdots & x_{N-1}^{\mathrm{HMS}} & x_N^{\mathrm{HMS}} \end{bmatrix}$$

Figure 5.3 Harmony Memory

The initialization of HM is performed using the following equation, which randomly sets the initial population in a particular range.

$$x_j^i = rand(0,1) \times (c-1) + 1,$$

$$i = 1, 2, 3, HMS \text{ and } j = 1, 2, 3, N \qquad \textit{Equation (5.5)}$$

where c is the predefined number of clusters and x_j^i represents the assignment of a cluster index.

Each row of the HM is a solution (harmony) that contains an assignment of documents to cluster numbers, and the length of each solution is fixed to the number of documents.

C. Every single solution is evaluated using an objective function. The objective function used for evaluation is the ADDC.

D. Harmony Improvising. This step is responsible for modifying the HM in the standard HS. The improvising step is shown in Algorithm 5.3 while the improvising of DHS and CDHS is shown in Algorithms 5.4 and 5.5 respectively.

E. Memetic Optimization. In order to apply the local search for the best solution in the harmony memory, the HM is ranked first to retrieve the best solution according to its ADDC fitness value. The best solution is then locally searched using the *k*-means method. Each time the local search is applied the new modified solution will be compared to its original version. If the resulting solution is given a better fitness score,

it will replace the original solution. For instance, if nine documents are to be distributed among three clusters, the length of the solution will be 9 while the range values of each bit in that solution is between [1,3] (assuming that the number of solutions is any number more than 3). The local search process will be conducted as is shown in Figure 5.4.

F. In Figure 5.4 it can be seen that in step 1 the local search is applied on the best solution (*sol*) while in steps 2 and 3 the evolution of the original solution and the modified solution is respectively completed. A comparison between both solutions is conducted according to the fitness values obtained by steps 2 and 3. In the method being tested for the present research, comparison is based on the minimization of the average distance of documents to clusters using the ADDC measure. Thus, if the modified solution obtained the minimum ADDC, it will replace the original solution.

G. The next step is to update HM. A comparison is done by checking the fitness value of an improvised solution with the older one. If the fitness value of the newly improvised solution is higher than the older one, the newly updated solution will replace the older one.

H. The termination condition is satisfied when the maximum number of iterations is reached, or no further improvement is observed.

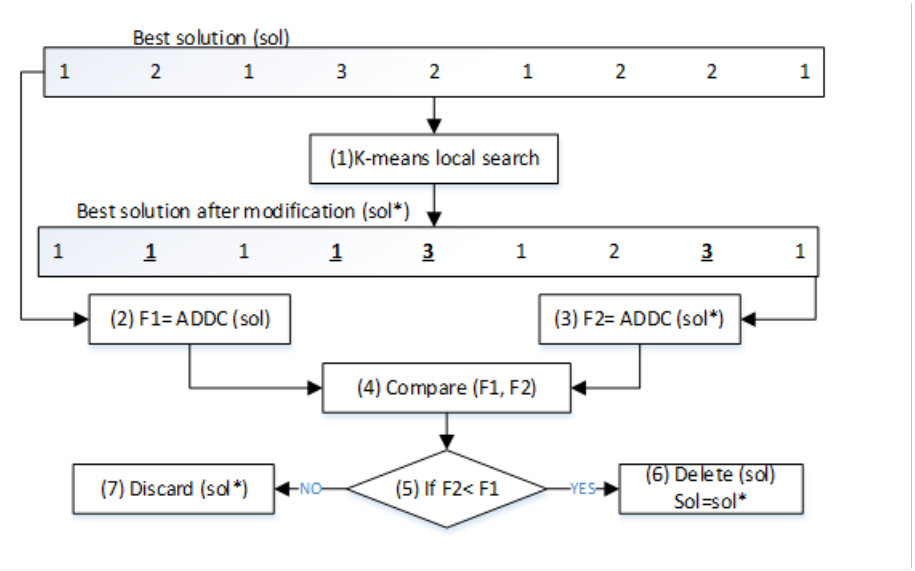

Figure 5.4 Local Search Modification

Algorithm 5.3 Improvising the population

1. Input Harmony Memory HM, HM size, PAR, HMCR and the modified solution X.
2. Begin
3. for $i = 1$ to HMS do
4. for $j = 1$ to N do
5. if $rand\ (0, 1) \leq HMCR$ then
6. $x = HM(i, j);$
7. if $rand\ (0, 1) \leq PAR$ then
8. $x = x + rand(0, 1) \times BW$
9. else
10. $x = x - rand(0, 1) \times BW$
11. end if
12. else
13. $x = rand(0,1)*(c-1) + 1;$
14. end if
15. $HM(i, j) = x$;
16. end for
17. end for
18. end

5.5.2 Test Strategy

In this section, the comparison results of the proposed MDHS and CMDHS methods are reported. The comparison involved the standard HS (Geem, Kim et al. 2001, Forsati, Mahdavi et al. 2013), *k*-means (Jain 2010), Memetic HS (Gao, Wang et al. 2015) and DHS (Abedinpourshotorban, Hasan et al. 2016) methods. The best performing method was compared with two state-of-the-art methods found in recent literature. The first is a KH method proposed by (Abualigah, Khader et al. 2016) while the second is an ABC-based method, named Chaotic Gradient ABC (CGABC) (Bharti and Singh 2016). In all tests, the external and internal evaluation measures of clustering were used. The F-measure is used as an external measure while the ADDC is used as an internal measure. The Friedman test for statistical significance is utilized to analyse the performance of the competent methods.

5.5.3 Parameter Tuning

The parameters used in these methods are a combination of the Differential Evolution (DE) parameters which are the mutation scaling factor F and crossover parameter Cr and the Harmony Search (HS) which were setup in the same way as they were set in (Forsati, Mahdavi et al. 2013) and (Abedinpourshotorban, Hasan et al. 2016). As these two methods were combined into one method therefore these parameters will be used together in one method. For

the sake of simplicity, we specifiy each methods' parameters in separate table where Table 5.8(a) shows the DE parameters. In Table 5.8(a) these parameters were tuned while in Table 5.8 (b) the HS parameters were used the same way as they were used in these two papers(Forsati, Mahdavi et al. 2013) and (Abedinpourshotorban, Hasan et al. 2016).

Three values were tested for the Cr and F parameters. For Cr, the three selected values were 0.2, 0.5 and 0.9 while for F the values tested were 0.8, 0.1 and 0.5. These two parameters should be between [0,1]. If a specific parameter generated the highest performance with the majority of datasets it is used for later tests. After the selection of the best value of the *Cr* parameter, the *F* parameter is tested. As can be seen in Table 5.8, the *Cr* value that helped to obtain the highest F-measure and lowest ADDC was 0.5. Therefore, the other two values were discarded. From Table 5.8 it becomes evident that the use of 0.5 for the *Cr* has 5 out of the six highest F-measure scores. The values of all runs for the ADDC measure are almost consistent, with minimal differences. Thus, depending on the F-measure values the best parameter will be chosen. Regarding the *F* parameter, 0.1 achieved the highest scores with 4 out of six datasets. In all of D1, D3, D4, and D5, the F-measure was the highest. For D2 and D5, in both datasets, the 0.1 F-measure score for the *F* parameter was higher after using 0.5, but both scores were lower than that using 0.8 for the *F*. The ADDC values were not greatly different in all parameter tests. Therefore, the F-measure values were considered to determine the best values of the *F* parameter in all cases.

Table 5.8 (a) *Cr* and *F* Parameter Tuning Table

	Cr=0.2		**Cr=0.5**		**Cr=0.9**	
Dataset	***F-Measure***	***ADDC***	***F-Measure***	***ADDC***	***F-Measure***	***ADDC***
D1	83.90945	0.718938	**85.81185**	**0.714983**	79.866677	0.714957
D2	94.540028	0.748749	**98.429393**	**0.74538**	96.452157	0.74342
D3	88.05235	0.711795	89.205819	0.711147	**92.794232**	**0.707603**
D4	94.953637	0.842767	**96.090954**	**0.843154**	94.857401	0.839821
D5	66.480562	0.847179	**99.547975**	**0.846073**	66.666667	0.848331
	F=0.1		**F=0.5**		**F=0.8**	
Dataset	***F-Measure***	***ADDC***	***F-Measure***	***ADDC***	***F-Measure***	***ADDC***
D1	**89.732655**	**0.721584**	77.772432	0.719301	85.81185	0.714983
D2	97.834008	0.749086	96.460589	0.747813	**98.429393**	**0.74538**
D3	**92.757559**	**0.711581**	90.263957	0.708986	89.205819	0.711147
D4	**98.230616**	**0.842119**	90.055755	0.823152	94.953637	0.842767
D5	**99.646309**	**0.846778**	99.176776	0.846605	99.547975	0.846073

Table 5.8 (b) HS Parameters used in the proposed MDHS and CMDHS methods

Parameter	Description
Harmony Memory (HM)	Population (Harmony vectors)
Harmony Memory Size(HMS)	Twice the number of clusters.
Bandwidth (BW)	Not used
Pitch Adjustment Rate (PAR)	0.99
Harmony Memory Consideration Rate (HMCR)	0.9

Algorithm 5.4: Improvising the population in DHS

1. *Input Harmony Memory HM, HM size, F, HMCR and the modified solution X.*
2. Begin
3. for $j = 1$ to N do
4. if *rand* $(0, 1) \leq HMCR$ then
5. *r1 =rand (1,HMS); r2 =rand (1,HMS); r3 =rand (1,HMS);*
6. while *(r1=r2 or r2=r3 or r1=r3)*
7. *r1 =rand (1,HMS); r2 =rand (1,HMS); r3 =rand (1,HMS);*
8. end
9. $x_1 = HM(r_1,:)$; $x_2 = HM(r_2,:)$; $x_3 = HM(r_3,:)$
10. $v(j)=x_1(j) + F*(x_2(j)- x_3(j))$
11. if $v(j) < 1$ *or* $v(j) > c$ then
12. *v(j) = rand(0,1)*(c-1) + 1;*
13. end if
14. else
15. *v(j) = rand(0,1)*(c -1) + 1;*
16. end if
17. end for
18. let k be the index of the solution with the best fitness value in the population
19. $x_k = HM(k,:)$
20. if *fitness(HMk,:))>fitness(v)* then
21. *HM(k,:)=v;*
22. end if
23. end

Algorithm 5.5: Improvising the population in CDHS

1. Input Harmony Memory HM, HM size, F, Cr, HMCR and the modified solution X.
2. Begin
3. for $j = 1$ to N do
4. if *rand* (0, 1) ≤ *HMCR* then
5. *r1 =rand (1,HMS); r2 =rand (1,HMS); r3 =rand (1,HMS);*
6. while *(r1=r2 or r2=r3 or r1=r3)*
7. *r1 =rand (1,HMS); r2 =rand (1,HMS); r3 =rand (1,HMS);*
8. end
9. $x_1 = HM(r_1,:);\ \ x_2 = HM(r_2,:);\ \ x_3 = HM(r_3,:);$
10. $v(j)=x_1(j) + F*(x_2(j)- x_3(j));$
11. if $v(j) < 1$ *or* $v(j) > c$ then
12. $v(j) = rand(0,1)*(c-1) + 1;$
13. end if
14. else
15. $v(j) = rand(0,1)*(c -1) + 1;$
16. end if
17. $x_k = HM(k,:)$
18. if *rand (0,1) <=Cr*
19. $u(j)= v(j);$
20. else
21. $u(j)=x_k(j);$
22. *end if*
23. end for
24. if *fitness(HM(k,:))>fitness(u)* then
25. *HM(k,:)=u;*
26. end if
27. end

5.5.4 Comparisons of MDHS and CMDHS

As the F-measure is used as an external evaluation measure, the original labels of documents are required. In that sense, Table 5.9 shows the F-measure results of each of the competent methods. Making a distinction between runs and iterations is important: each run might have hundreds of iterations. In the experiments, the maximum number of iterations was set at 100, and the number of runs set at 20. Table 5.9 shows that with all datasets the proposed CMDHS method achieved the best results in terms of the F-measure. On the other hand, MDHS performance was always less than CMDHS. Unsurprisingly, the results show a large gap between CMDHS and other methods. The effect of intensifying the exploitation through binomial crossover and local search can be seen from the results listed in Table 5.9. As for ADDC, Table 5.10 shows that despite the different results obtained by F-measures shown in Table 5.9, the ADDC differences are insignificant. The ADDC is a measurement that shows

the intra-cluster coherity of the clusters. If the differences of this measure are not significant it means that the algorithms are arranging the documents in a similar way within their classes. In that case, using the external measure will give us the understanding of how these clusters are different from each other.

The aim of observing ADDC was to see the stability of convergence through generations in relation to the F-measure values. Three observations can be drawn from the relationship between the external and internal measures as summarized by this researcher's previous study with colleagues (Al-Jadir, Wong et al. 2017). These observations were also detailed in chapter 3. The first observation is a decrease for both measures, which indicates a flaw in convergence as the external measure should always be maximized, otherwise, the algorithm might potentially be stagnated. The second observation is that the internal measure decreases while the external measure increases, which is a typical state. The last observation is that the internal measure does not change much, whereas the external measure increases, which might indicate the convergence is going toward optimality.

Among the plausible explanations for the above observations, the test results shown in Table 5.10 are categorized under the third observation, as the ADDC values of the CMDHS are almost similar with other compared methods, while the corresponding F-measure values are significantly higher for the CMDHS method than other methods. The single most striking observation to emerge from the comparsion conducted in Table 5.10 is that there is a remarkable improvement for the clustering results that maximized the external measure. On the other hand, other methods with a similar ADDC showed F-measure ranks deteriorated in comparison to the CMDHS shown in Table 5.9.

Table 5.9 F-measure Values

Method	**D1**	**D2**	**D3**	**D4**	**D5**
HS	0.70556	0.94054	0.96620	0.50888	0.91942
DHS	0.56283	0.88646	0.95662	0.6054	0.83843
MHS	0.76712	0.91839	0.97732	0.5353	0.88006
MDHS	0.66771	0.92625	0.96082	0.6379	0.88753
k-means	0.40273	0.44380	0.51477	0.18044	0.16367
CDMHS	**0.88500**	**0.98695**	0.100	**0.98429**	**0.99912**

Table 5.10 ADDC values

Method	D1	D2	D3	D4	D5
HS	0.72	0.74	0.7	0.84	0.73
DHS	0.71	0.74	0.7	0.84	0.73
MDHS	0.71	0.74	0.68	0.84	0.73
MHS	0.72	0.74	0.7	0.84	0.73
k-means	_	_	_	_	_
CMDHS	**0.72**	**0.74**	**0.7**	**0.84**	**0.72**

After comparing the MDHS and CMDHS methods with baseline methods represented by the HS, DHS, MHS, and *k*-means, the CMDHS method that obtained the highest F-measure was compared again with other counterparts. The CMDHS was compared with the KH method proposed by (Abualigah, Khader et al. 2016), and the ABC-based method, named Chaotic Gradient ABC (CGABC) proposed by (Bharti and Singh 2016). It was also compared with the DEMC method explained earlier in this chapter. In Table 5.11, the F-measure and ADDC values of the comparison results are listed. The method that achieves the highest F-measure and the lowest ADDC are considered the best results. Table 5.11(a) shows the proposed CMDHS obtained the highest F-measure scores for D2, D3, D4, and D5. However, for D1, the CMDHS provided a comparable rating when compared to the DEMC. On the other hand, the ADDC results show that the KH and CGABC methods have increased ADDC values when compared with the CMDHS for the D4 dataset.

This finding is evidence that despite the slight increase of the F-measure achieved by the KH method for D4 in Table 5.11(a), the alleviated ADDC means that there is a slow convergence for KH. However, with D2, D3, D5 it can be noticed that the ADDC values of the KH were slightly less than the CMDHS, but if one looks to the F-measure improvements achieved by the CMDHS, then these small ADDC differences can be considered insignificant.

In Table 5.11(b), for the DEMC, the ADDC score for D4 is the highest, but at the same time has the lowest F-measure for the same dataset as seen in Table 5.11(a). That means the DEMC cannot be efficient when the number of clusters increases as the number of clusters for D4 equals 10. Interestingly, the similarities of other values for the ADDC lead to the conclusion that the F-measure determines the superiority of all methods.

5.6 Statistical Significance Test

The results shown in Table 5.9 demonstrated that the CMDHS had higher F-measure values than other methods. To verify the reliability of the results, and to analyze these results statistically, it was necessary to run the competent algorithms more than once, and compare the average of the F-measure scores of the CMDHS against the average accuracy of the other competent methods.

5.6.1 Statistical SignificanceTest

The Friedman test is used for performance evaluation to evaluate the performance of the algorithms (Bharti and Singh 2016). The Friedman test is a non-parametric two–way analysis of ranks variance (Derrac, García et al. 2011). This thesis applies the Friedman test to the F-measure results, because it is a standardized evaluation measure used by other researchers in this domain.

The Friedman test performs *N*N* number of comparisons by ranking the tested algorithms. It then highlights the differences between the test algorithms and determines if their differences are statistically significant. The test first approves or rejects the null hypotheses. If the null hypothesis is denied, the highest ranked algorithms are considered the best. The null hypothesis states that all the tested algorithms are equally accurate. That means there are no significant differences between the tested methods.

From the mathematical point of view, the Friedman test can be formulated in equations(5.6),(5.7),(5.8) and (5.9):

$$T_f = \frac{(b - B_f - \frac{bk(k+1)^2}{4})}{A_f - B_f} \qquad \textit{Equation (5.6)}$$

$$A_f = \sum_{i=1}^{b} R_{ij}^2 \qquad \textit{Equation (5.7)}$$

$$B_f = \frac{1}{b}\sum_{j=1}^{k} R_j^2 \qquad \textit{Equation (5.8)}$$

$$R_j = \sum_{i=1}^{b} R_{ij} \qquad j=1,2,...k$$

Equation (5.9)

In the above, R_{ij} is the algorithms' rank while b is the number of *blocks* (runs number) and k is the number of *groups* (tested methods). To refute the first hypothesis that assumes all methods have the same accuracy, the p-values from the Friedman test should be equal to or higher than α, which represents the level of significance. The score should be equal to either 0.05 or 0.01. For these comparisons, α was set to 0.01. The first hypothesis (H_0) can be rejected using the p-values. Table 5.12 shows that all the p-values are less than the significance level of α value (p *values < 0.01)*. Thus, there is a significant difference between the competent methods in all runs. As all the values were less than the significance level of α, this gives a clear indication that the differences of the performance of the different methods are not similar.

After rejecting the null hypothesis, it is essential to know which algorithm achieved the highest ranks. In Table 5.13, Friedman ranks are listed according to the mean values of runs. The CMDHS achieved the highest rank among other methods with all datasets. For the other methods, the ranks were ordered from best to worst in the following order: HS, MHS, MDHS, DHS, and k-means. The test strategy was based on observing the maximization of the rank values of the average external F-measure presented in Table 5.9 with respect to their corresponding ADDC values presented in Table 5.10. By looking at the values depicted in Table 5.10, it can be noticed that nearly all methods converged to the same point. Even so, that result would not suggest the performance of these methods is equal, because the null hypothesis H_0 was rejected.

The statistical significance test of the state-of-the-art methods in comparison to CMDHS was also conducted, but this time the tests were based on F-measure values shown in Table 5.11. The ranks shown in Table 5.14 suggested the superiority of the CMDHS method over the other state-of-the-art methods. The null hypotheses can be rejected if the p-value is higher than 0.01. However, if the alpha value increased to 0.05 then the p-value becomes less, which indicates that the results are still significant. As is shown in Table 5.15, the Friedman test evaluated the differences in median values among the four tested methods, and the results were significant with X_2 (2, N = 6) = 7, p <.05. Where X_2 is the Chi-Square in Table 9 and N is the number of the Datasets, P is the significance level that equals 0.03 and finally, d_f is the test degree of freedom that equals 2 in Table 5.15.

Table 5.11 F-measure and ADDC Results of CMDHS and other State-of-the-Art Methods

5.11(a) F-measure

Criteria	Method	D1	D2	D3	D4	D5
F-measure	CMDHS	88.5003	**98.6959**	**100**	**98.4293**	**99.9127**
	KH	79.0762	85.4340	94.5528	98.71794	98.7804
	CGABC	84.6594	91.9599	95.4821	97.2078	99.1360
	DEMC	**89**	94.8054	81.9491	50.09021	98.4943

5.11(b) ADDC

Criteria	Method	D1	D2	D3	D4	D5
ADDC	CMDHS	0.7209	0.7092	0.8219	0.7228	0.8463
	KH	0.7212	0.6793	0.8043	0.7478	0.8448
	CGABC	0.7228	0.7233	0.8375	0.7489	0.8450
	DEMC	0.7217	0.8605	0.8032	0.8479	0.8304

Table 5.12 Friedman's P-Values

D#	P-Values < α
D1	1.3068E-16
D2	1.3371E-20
D3	3.4E-17
D4	1.6803E-18
D5	3.644E-21

Table 5.13 Ranks Table

Method	D1	D2	D3	D4	D5
HS	5.10	4.19	4.86	4.71	4.24
DHS	2.52	1.24	2.86	2.00	2.52
MHS	4.19	4.38	5.00	4.29	4.00

MDHS	2.38	2.48	2.29	3.00	3.43
k-means	1.14	2.76	1.00	1.00	1.05
CMDHS	5.67	5.95	5.00	6.00	5.76

Table 5.14 Mean Ranks

Method	Value
CMDHS	2.83
KH	1.33
CGABC	1.83
DEMC	1.14

Table 5.15 Test Statistics

Parameter	Value
N	6
Chi-Square	7.000
df	2
p	.030

Friedman Post Hoc test using Holm's method is used to test the statistical significance of tests among different methods. In this situation the adjusted alpha value was calculated as shown in equation 13.

$$adjusted\_\alpha = \frac{\alpha}{i} \tag{5.10}$$

Where α is the significance factor while i is the rank of that method.

It is clear from Table 9 that the Holm's method rejects all hypotheses. Thus, in relation to this procedure, the proposed method is statistically more significant than other methods in terms of clustering accuracy using the F-measure ranks.

Table 5.16 Post Hoc values of Friedman Test for the Base-line methods

D#	*i*	Algorithm	*p_*value	*α(0.05)/i*	*Hypothesis*
D1	5.71	HS	5.24E-23	0.008756567	Rejected
	2.14	DHS	5.24E-23	0.023364486	Rejected
	6.33	MHS	5.24E-23	0.007898894	Rejected
	3.24	MDHS	5.24E-23	0.015432099	Rejected
	5	K-means	5.24E-23	0.01	Rejected
	7.95	CDMHS	5.24E-23	0.006289308	Rejected
D2	6.48	HS	3.98E-26	0.007716049	Rejected
	3.57	DHS	3.98E-26	0.014005602	Rejected
	5.57	MHS	3.98E-26	0.008976661	Rejected
	3.43	MDHS	3.98E-26	0.014577259	Rejected
	6.29	K-means	3.98E-26	0.007949126	Rejected
	7.67	CDMHS	3.98E-26	0.006518905	Rejected
D3	6.14	HS	5.05E-29	0.008143322	Rejected
	3.71	DHS	5.05E-29	0.013477089	Rejected
	6	MHS	5.05E-29	0.008333333	Rejected
	3.05	MDHS	5.05E-29	0.016393443	Rejected
	7.9	K-means	5.05E-29	0.006329114	Rejected
	6	CDMHS	5.05E-29	0.008333333	Rejected
D4	5.24	HS	5.01E-24	0.009541985	Rejected
	3.52	DHS	5.01E-24	0.014204545	Rejected
	5.05	MHS	5.01E-24	0.00990099	Rejected
	4.43	MDHS	5.01E-24	0.011286682	Rejected
	7.71	K-means	5.01E-24	0.006485084	Rejected
	6.95	CDMHS	5.01E-24	0.007194245	Rejected
D5	6.31	HS	1.76E-27	0.00792393	Rejected
	3	DHS	1.76E-27	0.016666667	Rejected
	5.43	MHS	1.76E-27	0.009208103	Rejected
	4	MDHS	1.76E-27	0.0125	Rejected
	6.26	K-means	1.76E-27	0.00798722	Rejected
	8	CDMHS	1.76E-27	0.00625	Rejected
D6	6.82	HS	4.58E-28	0.007331378	Rejected
	3	DHS	4.58E-28	0.016666667	Rejected
	5.86	MHS	4.58E-28	0.008532423	Rejected
	4.27	MDHS	4.58E-28	0.011709602	Rejected
	4.86	K-means	4.58E-28	0.010288066	Rejected

The post hoc values using the Holm's method was also used to measure the statistical significance of the state-of-the-art method. The comparisons of Table 12 clearly shows that all the values are much smaller than the adjusted α values.

Table 5.17 Post Hoc values of Friedman Test for the state-of-the-art methods

D#	i	Algorithm	p_value	α(0.05)/i	Hypothesis
D1	1.05	KH	1.81741782409671E-09	0.047619048	Rejected
	2.45	CGABC	1.81741782409671E-09	0.020408163	Rejected
	3.70	DEMC	1.81741782409671E-09	0.013513514	Rejected
	2.80	CMDHS	1.81741782409671E-09	0.017857143	Rejected
D2	2.71	KH	8.8191106725875E-12	0.018450185	Rejected
	2.38	CGABC	8.8191106725875E-12	0.021008403	Rejected
	1.00	DEMC	8.8191106725875E-12	0.05	Rejected
	3.90	CMDHS	8.8191106725875E-12	0.012820513	Rejected
D3	3.45	KH	1.81741782409671E-09	0.014492754	Rejected
	3.40	CGABC	1.81741782409671E-09	0.014705882	Rejected
	1.25	DEMC	1.81741782409671E-09	0.04	Rejected
	1.90	CMDHS	1.81741782409671E-09	0.026315789	Rejected
D4	2.70	KH	8.10650901638872E-09	0.018518519	Rejected
	2.80	CGABC	8.10650901638872E-09	0.017857143	Rejected
	1.00	DEMC	8.10650901638872E-09	0.05	Rejected
	3.50	CMDHS	8.10650901638872E-09	0.014285714	Rejected
D5	1.69	KH	0	0.029585799	Rejected
	2.48	CGABC	0	0.02016129	Rejected
	2.10	DEMC	0	0.023809524	Rejected
	3.74	CMDHS	0	0.013368984	Rejected
D6	2.19	KH	0	0.02283105	Rejected
	2.86	CGABC	0	0.017482517	Rejected
	1.00	DEMC	0	0.05	Rejected
	3.95	CMDHS	0	0.012658228	Rejected

5.7 Adaptive CMDHS

The CMDHS was successful in all of the previous experiments. However, the proposed method still needs an accurate tuning of the DE parameters as the DE is sensitive to the settings of its control parameters. The adaptive version of the CMDHS is proposed to overcome the need to

statically tune the DE parameters using the same method of parameter adaptation proposed in (Cui, Li et al. 2016). This method updates the *F* and *Cr* parameters that control the mutation and crossover operators to their optimal selection.

In this section, a comparison between CMDHS and ACMDHS is conducted. The ADDC values need to be minimized while the F-measure values need to be maximized, as shown in the previous sections. The values of both measures are listed in Table 5.16 and Table 5.17. Table 5.16 depicts the external measure values using the F-measure while Table 5.17 shows the internal measure values using ADDC. It becomes clear that the statically-based parameter tuned version (CMDHS) outperformed the dynamically-based ACMDHS. The single most striking observation to emerge from that comparison is the tuned parameters. That is, *F* for the mutation and *Cr* for the crossover have only a minor effect on the performance of the centroids allocation. Table 5.17 shows the general trend of all results for both CMDHS and ACMDHS are compatible. The stability of the ADDC in comparison to

Table 5.16 F-measure Values

Dataset	CMDHS	ACMDHS
D1	**88.50**	85.97
D2	**98.69**	96.03
D3	96.94	**97.56**
D4	97.56	**98.84**
D5	**99.91**	98.92

the F-measure did not mean that both methods performed equally. That is because the F-measure values were changing when ADDC values were almost steady, but both methods are highly competitive nonetheless.

Table 5.17 F-measure Values

Dataset	CMDHS	ACMDHS
D1	**88.50**	85.97
D2	**98.69**	96.03
D3	96.94	**97.56**
D4	97.56	**98.84**
D5	**99.91**	98.92

Table 5.18 ADDC Values

Dataset	CMDHS	ACMDHS
D1	0.72	0.72
D2	0.74	0.71
D3	0.73	0.71
D4	0.82	0.75
D5	0.72	0.73

5.8 Summary

The clustering of the text documents is an important process for document categorization, archiving, summarization and retrieval. After the pre-processing of the text documents and feature selection using the supervised and unsupervised methods presented in chapters 3 and 4, the unsupervised feature selection method (DDESA) presented in chapter 4 was used to reduce the features for the text clustering. This chapter presented two different hybrid document clustering approaches, which are capable of distributing the cluster centroids using memetic optimization in the search space.

The first approach is the DEMC. This method combines the DE global search with the simulated anealing local search. The research found the DEMC to be superior to the *k*-means, DE, DEKM, DESA and CGABC methods in terms of the clustering internal and external evaluation measures.

Another memetic document clustering that fuses the global search using the DHS with the traditional clustering using the *k*-means was proposed. DHS was applied successfully as a global search and was successfully combined with the *k*-means to produce the MDHS method. However, this present study experimented with a combination of the binomial DE crossover

with the MDHS to produce the CMDHS method. It can be concluded from the results of the experimental study that the proposed CMDHS successfully outperformed other methods that were compared for document clustering. The test results using the F-measure, ADDC and the non-parametric statistical tests showed the superiority of the CMDHS over the baseline methods, namely the HS, DHS, *k*-means and the Memetic HS. The proposed CMDHS also outperformed two current state-of-the-art methods in most cases. In addition it was better than the Differential Evolution Memetic Clustering proposed earlier.

Finally, an enhancement was made to CMDHS, using the adaptive parameter tuning of the differential control parameters. The resultant method was named the Adaptive CMDHS (ACMDHS) that updates DE control parameters to their best values. The test results indicate that CMDHS and the ACMDHS are both highly competitive methods.

Chapter 6

Conclusion and Future Research

6.0 Introduction

This present research addressed the issues of centroids allocation and text feature selection for document clustering. Memetic optimization was proposed to manage these two issues and find a more efficient method to cluster the results of text documents than existing methods. This thesis first discussed the use of memetic optimization to resolve the problem of centroids distribution. As was observed in the literature review (chapter 2), the majority of optimization methods used for centroids allocation perform only a global search, using methods such as the EA, SI or HS. Despite the ability of these methods to perform a global search, they are not capable of performing the exploitation aspect in the local areas within the search space. Therefore, the memetic optimization was used to resolve this problem, because it combines the global and local searches. The research extensively explored memetic optimization in terms of the clustering centroids allocation of document clustering.

As reported in this thesis, the problem of document clustering was not limited to the distribution of the cluster centroids. The other focus of the research was text feature selection, because high text dimensionality affects the clustering system negatively. The thesis discussed supervised and unsupervised feature selection methods in terms of filter, wrapper, and hybrid techniques. Hybrid feature selection techniques were discussed extensively. In this context, the hybrid methods of feature selection are equivalent to memetic optimization with regard to centroids allocation. However, global and local searches have a different meaning in feature selection methods. The wrapper and filter methods are equivalent to global and local searches in optimization, respectively. The final aim of this thesis was to combine hybrid feature selection and hybrid centroids allocation for more efficient document clustering.

6.1 Research Summary and Contributions

The problems of missing labels of text features, local optima in feature selection, and poor centroids allocation were addressed to achieve the research aims and objectives (chapter 1,

section 1.3). This thesis contributed to solutions for each of these problems in the following ways.

6.1.1 Memetic Feature Selection for Text Document Clustering

A supervised memetic combination between filter and wrapper feature selection methods was applied to reduce extra text features. The traditional clustering methods, *k*-means and Spherical *k*-means, were utilized to examine the performance of the proposed feature selection methods. The research found these methods minimized the Average Document to Centroid Distance (ADDC), and in most cases the F-measure was maximized after using the experimental MAFS. Another major finding was that the memetic combination with MAFS performed better than the wrapper feature selection using GA.

Test results also revealed that using the proposed feature selection can improve the performance of traditional clustering. Comparisons showed that the proposed MAFS method performed better than the recently proposed Feature Selection Genetic Algorithm Text Clustering (FSGATC) method reported in the literature. The proposed MAFS method also performed better when it was compared to the results generated when using the ALL feature space. Moreover, the experiments also found a slight correlation between ADDC and F-measures. Finally, tuning the parameters has a positive impact on the accuracy of resulted clusters. Due to the fact that some of the datasets could not be labelled, it becomes necessary to find a method that deals with the unlabelled documents.

An unsupervised feature selection method was proposed to select informative features without using class labels and classifiers. The unsupervised method combined wrapper and filter methods, through the combination of a global search (using DE) and a local search (using SA). SA was utilized as a filter method to refine the best solution obtained by DE, because it is capable of modifying the solutions (feature subsets) without referencing class labels. The resultant method was named DESA. A variation of DESA was also presented, being the Dichotomous DESA (DDESA) that used the dichotomous mutation. This modification was necessary, for the purpose of demonstrating the impact of two different types of mutations on the feature selection results. The test results showed that DDESA outperformed DESA. DDESA was also compared to DE, FSGATC and FSHSTC methods, again showing a better performance.

6.1.2 Centroids Allocation for Document Clustering

The centroids allocation of document clustering was the second aim of the research. On that basis, two different methods were proposed to discover the prospects of using different memetic combinations for the centroids allocation process. The first method proposed for text document clustering was DEMC. It was found that DEMC was superior to the *k*-means, DE, DEKM, DESA and CGABC methods. Another memetic document clustering that combined the global search using DHS with the traditional clustering using *k*-means was also proposed and the results of experiments reported in this thesis. DHS was applied successfully as a global search, and was successfully combined with *k*-means to produce the MDHS method. Further experimentation on the use of the binomial crossover with MDHS produced the CMDHS method. It can be concluded that CMDHS outperformed other methods in the comparison tests. The results using F-measure, ADDC and non-parametric statistical measures demonstrated the superiority of CMDHS over the HS, DHS, *k*-means and the Memetic HS. The proposed CMDHS also outperformed two of the state-of-the-art methods as well as the DEMC method.

An enhancement was made to CMDHS by using the adaptive setting of the differential control parameters. The resultant method was named Adaptive CMDHS (ACMDHS). This method was capable of retaining the best values of these parameters, however, the test results indicated that CMDHS provided the best F-measure values in comparison to the ACMDHS method.

In conclusion, this thesis has made a significant contribution by identifying and demonstrating a way to distribute cluster centroids for text documents with a minimal number of labled or unlabled text features. This thesis met its aims by developing supervised and unsupervised text feature selection methods, and by using memetic optimization to find the optimal distribution of the clusters centroids.

6.2 Recommendations and Future Work

Several possible research directions could be pursued for future studies.

A. In this study the method used for the text features weighting is the Term-Frequency Inverse Document Frequency (TF.IDF). The TF.IDF could be considered if using other weighting schemes that work on a semantic level.

B. Although the main focus of this thesis was text data, the methods developed to perform centroids allocation and feature selection could be used with other data types.
C. For future research, it would be valuable to set up the number of clusters automatically. Also, the current feature selection and clustering methods initialize the population randomly. Finding a less random method could be worthy of research.
D. The use of the embeddings (doc2vec) that is based on neural networks could be used. This method is used to add more understandability to the text by the machine. TF.IDF is based on a word level while the doc2vec is based on a semantic level which is more specified.

Appendix A

F-measure Values of the MAFS method (A)

6 event crimes		*10 Types crimes*	
K-means	**spk**	**K-means**	**spk**
67.7	54.68	39.03	29.26
40.86	54.68	28.05	29.26
89.48	54.68	15.06	43.75
68.42	54.68	24.65	43.75
76.58	54.68	15.95	43.75
41.9	54.68	54.85	43.75
73.88	54.68	40.3	43.75
32.28	67.39	52.01	17.5
49.86	67.39	22.81	17.5
53.52	84.42	43.41	17.5
82.57	84.42	40.72	17.5
75.12	84.42	46.35	17.5
58.87	84.42	38.31	17.5
43.56	84.42	23.6	17.5
74.5	84.42	21.23	17.5
52.18	84.42	25.97	17.5
89.45	84.42	32.56	17.5
39.01	84.42	35.37	17.5
63.47	84.42	36.54	17.5
62.5885	72.308	33.51421	25.65
89.48	84.42	54.85	43.75
32.28	54.68	15.06	17.5

F-measure Values of the MAFS method (B)

Pair 20news		***Reuters***		***20 News Groups***	
K-means	**spk**	**K-means**	**spk**	**K-means**	**spk**
51.96	94.44	77.387553	46.24	27.51	95.804
51.96	94.44	68.307948	94.452	27.51	95.804
51.96	94.44	54.310501	94.452	31.94	95.804
51.96	99.49	70.916386	94.452	31.99	70.713
51.96	80.07	75.416908	89.25	25.13	70.713
51.48	79.08	77.332688	89.25	26.27	70.713
51.96	91.69	77.426602	89.25	30.52	70.713
51.96	91.69	77.274145	89.25	30.94	70.713
51.96	91.69	62.898253	89.25	30.47	70.713
51.96	99.49	42.879747	83.668	26.65	70.713
51.96	99.49	75.09245	83.668	31.59	70.713
51.96	99.49	62.532434	89.423	29.20	92.285
51.96	99.49	73.965092	89.423	31.67	92.285
51.96	99.49	43.900544	89.423	31.06	55.612
51.96	99.49	43.725643	89.423	31.98	55.612
51.96	99.49	62.474012	89.423	31.89	55.612
51.96	99.49	84.784785	89.423	33.92	80.623
51.96	99.49	62.493506	89.423	32.28	80.623
51.96	99.49	62.532434	89.423	26.96	80.623
51.93474	95.365	42.101043	89.423	26.23	80.623
51.96	99.49	84.784785	**94.452**	32.30	80.623
51.48	99.49	42.101043	46.24	33.92	95.804
				25.13	55.612

Appendix B

ADDC values for the MAFS (A)

6Event Crime		*10 types of crime*	
K-means	spk	spk	K-means
0.58	0.55	0.75	0.74
0.56	0.55	0.75	0.8
0.58	0.55	0.75	0.75
0.56	0.55	0.75	0.8
0.58	0.55	0.75	0.8
0.59	0.55	0.75	0.75
0.59	0.55	0.75	0.8
0.58	0.55	0.75	0.74
0.55	0.55	0.75	0.79
0.58	0.56	0.75	0.76
0.59	0.56	0.75	0.8
0.57	0.56	0.75	0.74
0.55	0.56	0.75	0.8
0.57	0.56	0.75	0.8
0.58	0.56	0.75	0.8
0.58	0.56	0.75	0.76
0.55	0.56	0.75	0.76
0.58	0.56	0.75	0.75
0.57	0.56	0.75	0.77
0.59	0.56	0.75	0.8

ADDC Values of the MAFS method (B)

Pair of 20 News Groups		*20News Groups*		*Reuters*	
K-means	spk	K-means	spk	K-means	spk
0.54	0.83	0.47	0.47	0.66	0.59
0.6	0.83	0.47	0.47	0.64	0.69
0.6	0.83	0.47	0.47	0.66	0.69
0.6	0.82	0.47	0.47	0.65	0.69
0.6	0.78	0.47	0.47	0.66	0.69
0.6	0.78	0.47	0.47	0.66	0.69
0.6	0.83	0.47	0.47	0.66	0.69
0.6	0.83	0.47	0.47	0.66	0.69
0.6	0.83	0.47	0.47	0.68	0.69
0.6	0.83	0.47	0.47	0.65	0.69

0.6	0.83	0.47	0.47	0.66	0.69
0.54	0.83	0.47	0.47	0.65	0.69
0.6	0.83	0.47	0.47	0.65	0.69
0.6	0.83	0.47	0.47	0.66	0.69
0.6	0.83	0.47	0.47	0.66	0.69
0.6	0.83	0.47	0.47	0.65	0.69
0.6	0.83	0.47	0.47	0.65	0.69
0.6	0.83	0.47	0.47	0.65	0.69
0.59	0.82	0.47	0.47	0.65	0.69

Appendix C

Convergence of the Memetic based Unsupervised feature selection and the Memetic based Unsupervised methods.

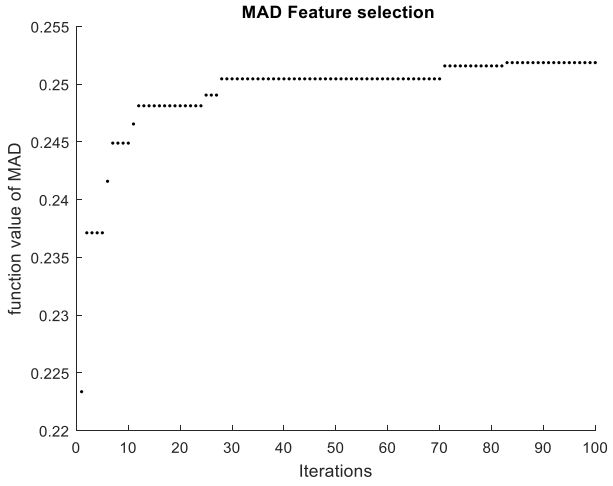

Fig1 a: 6 events crimes (no memetic)

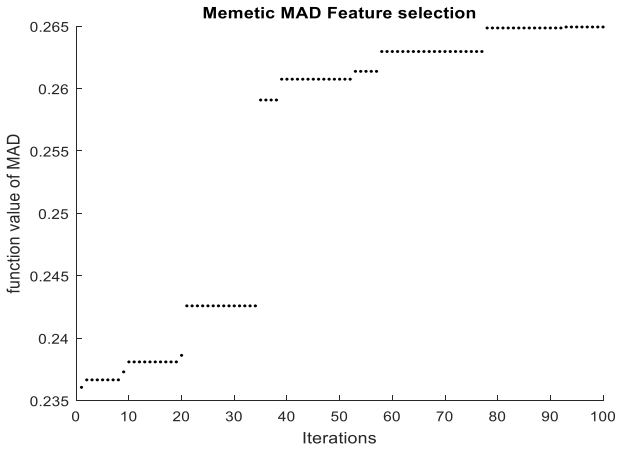

Fig1 b: 6 events crimes (Memetic)

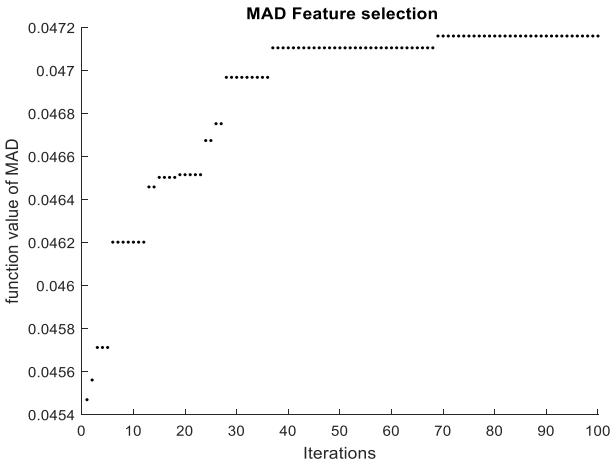

Fig2 a: 10 Types (no memetic)

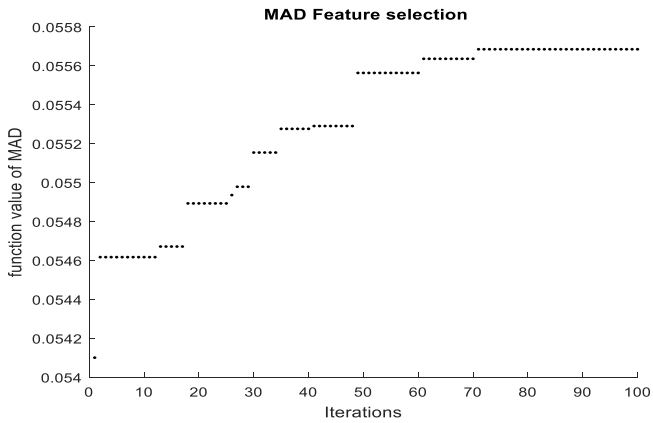

Fig2 b: 10 Types (Memetic)

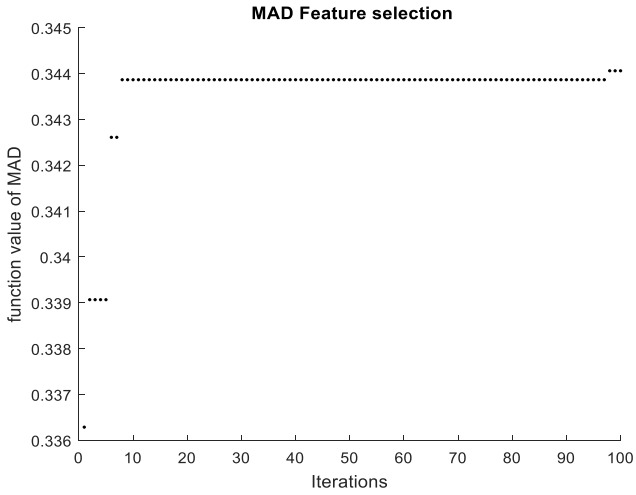

Fig3 a: Pair of 20 news (No Memetic)

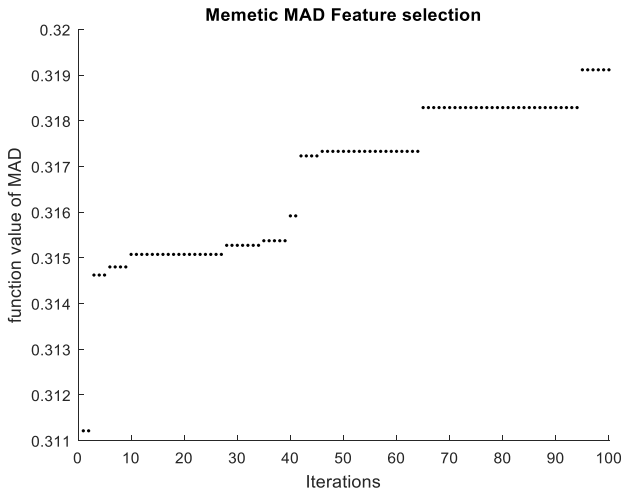

Fig3 b: Pair of 20 news (Memetic)

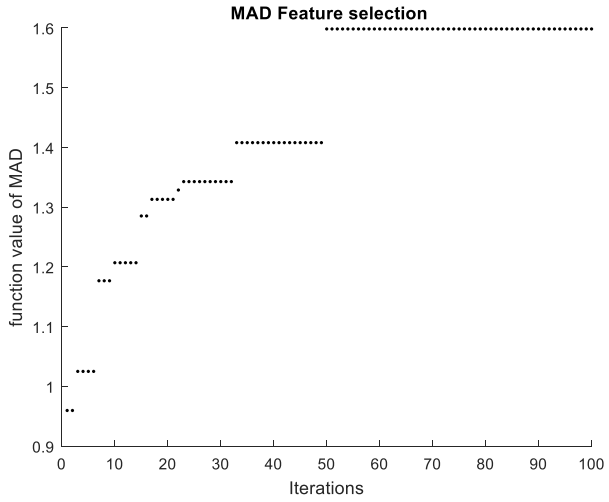

Fig4 a: Reuters (No Memetic)

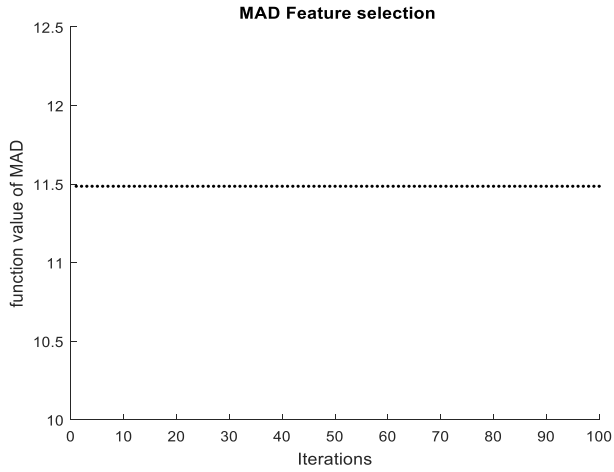

Fig4 b: Reuters (Memetic)

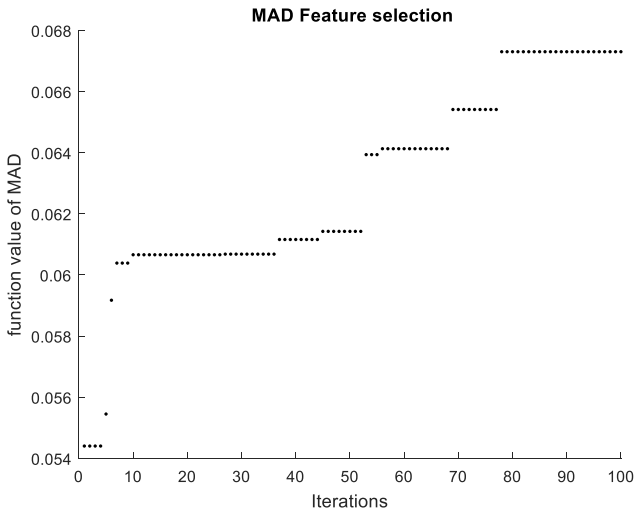

Fig5 a: 20 News Group (no memetic)

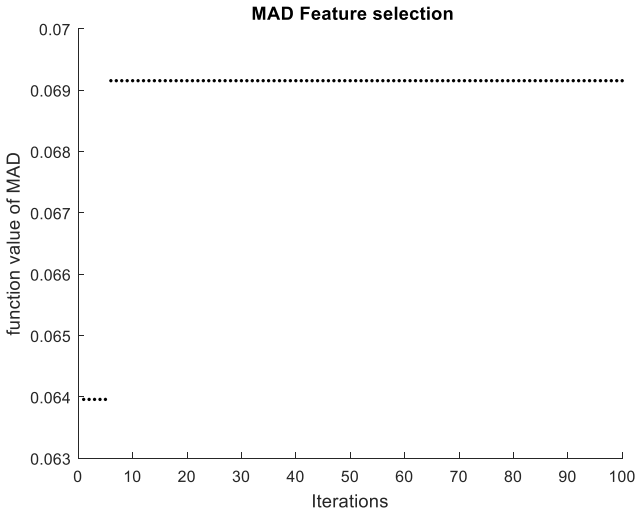

Fig5 b: 20 News Group (Memetic)

References

Aarts, E. and P. v. Laarhoven (1989). "Simulated annealing: an introduction." Statistica Neerlandica **43**(1): 31-52.

Abe, S. (2005). Modified backward feature selection by cross validation. ESANN, Citeseer.

Abedinpourshotorban, H., S. Hasan, S. M. Shamsuddin and N. F. As' Sahra (2016). "A differential-based harmony search algorithm for the optimization of continuous problems." Expert Systems with Applications **62**: 317-332.

Abraham, A., S. Das and A. Konar (2006). Document clustering using differential evolution. Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, IEEE.

Abualigah, L. M. and A. T. Khader (2017). "Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering." The Journal of Supercomputing: 1-23.

Abualigah, L. M., A. T. Khader and M. A. Al-Betar (2016). Unsupervised feature selection technique based on genetic algorithm for improving the Text Clustering. Computer Science and Information Technology (CSIT), 2016 7th International Conference on, IEEE.

Abualigah, L. M., A. T. Khader and M. A. Al-Betar (2016). Unsupervised feature selection technique based on harmony search algorithm for improving the text clustering. 2016 7th International Conference on Computer Science and Information Technology (CSIT).

Abualigah, L. M., A. T. Khader, M. A. Al-Betar and M. A. Awadallah (2016). A krill herd algorithm for efficient text documents clustering. Computer Applications & Industrial Electronics (ISCAIE), 2016 IEEE Symposium on, IEEE.

Abualigah, L. M., A. T. Khader, M. A. Al-Betar and A. H. Gandomi (2017). "A novel hybridization strategy for krill herd algorithm applied to clustering techniques." Applied Soft Computing.

Abualigah, L. M., A. T. Khader, M. A. Al-Betar and E. S. Hanandeh (2017). "A new hybridization strategy for krill herd algorithm and harmony search algorithm applied to improve the data clustering." management **9**: 11.

Abualigah, L. M., A. T. Khader and E. S. Hanandeh (2018). "Hybrid clustering analysis using improved krill herd algorithm." Applied Intelligence: 1-25.

Abualigah, L. M., A. T. Khader and E. S. Hanandeh (2018). "A new feature selection method to improve the document clustering using particle swarm optimization algorithm." Journal of Computational Science **25**: 456-466.

Aggarwal, C. C. and C. K. Reddy (2013). Data clustering: algorithms and applications, CRC Press.

Aggarwal, C. C. and C. Zhai (2012). A survey of text clustering algorithms. Mining Text Data, Springer**:** 77-128.

Aghdam, M. H., N. Ghasem-Aghaee and M. E. Basiri (2009). "Text feature selection using ant colony optimization." Expert systems with applications **36**(3): 6843-6853.

Aha, D. W. and R. L. Bankert (1996). A comparative evaluation of sequential feature selection algorithms. Learning from Data, Springer**:** 199-206.

Al-Jadir, I., K. W. Wong, C. C. Fung and H. Xie (2017). Differential Evolution Memetic Document Clustering Using Chaotic Logistic Local Search. Neural

Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part I. D. Liu, S. Xie, Y. Li, D. Zhao and E.-S. M. El-Alfy. Cham, Springer International Publishing**:** 213-221.
Al-Jadir, I., K. W. Wong, C. C. Fung and H. Xie (2017). Text Document Clustering Using Memetic Feature Selection. Proceedings of the 9th International Conference on Machine Learning and Computing. Singapore, Singapore, ACM**:** 415-420.
Alsaeedi, A., M. A. Fattah and K. Aloufi (2017). A hybrid feature selection model for text clustering. System Engineering and Technology (ICSET), 2017 7th IEEE International Conference on, IEEE.
Amaya, J. E., C. C. Porras and A. J. F. Leiva (2015). Memetic and Hybrid Evolutionary Algorithms. Springer Handbook of Computational Intelligence, Springer**:** 1047-1060.
Ang, J. C., A. Mirzal, H. Haron and H. N. A. Hamed (2016). "Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection." IEEE/ACM Transactions on Computational Biology and Bioinformatics **13**(5): 971-989.
Apolloni, J., G. Leguizamón and E. Alba (2016). "Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments." Applied Soft Computing **38**: 922-932.
Azmi, R., B. Pishgoo, N. Norozi, M. Koohzadi and F. Baesi (2010). A hybrid GA and SA algorithms for feature selection in recognition of hand-printed Farsi characters. Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on, IEEE.
Beni, G. (1988). The concept of cellular robotic system. Intelligent Control, 1988. Proceedings., IEEE International Symposium on, IEEE.
Beyer, H.-G. and H.-P. Schwefel (2002). "Evolution strategies–A comprehensive introduction." Natural computing **1**(1): 3-52.
Bezdek, J. C., S. Boggavarapu, L. O. Hall and A. Bensaid (1994). Genetic algorithm guided clustering. Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on, IEEE.
Bharti, K. K. and P. K. Singh (2014). "A three-stage unsupervised dimension reduction method for text clustering." Journal of Computational Science **5**(2): 156-169.
Bharti, K. K. and P. K. Singh (2016). "Chaotic gradient artificial bee colony for text clustering." Soft Computing **20**(3): 1113-1126.
Bharti, K. K. and P. K. Singh (2016). "Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering." Applied Soft Computing **43**: 20-34.
Boggs, P. T. and J. W. Tolle (1995). "Sequential quadratic programming." Acta numerica **4**: 1-51.
Bolaji, A. L. a., M. A. Al-Betar, M. A. Awadallah, A. T. Khader and L. M. Abualigah (2016). "A comprehensive review: Krill Herd algorithm (KH) and its applications." Applied Soft Computing **49**: 437-446.
Bolón-Canedo, V., N. Sánchez-Maroño and A. Alonso-Betanzos (2013). "A review of feature selection methods on synthetic data." Knowledge and information systems **34**(3): 483-519.
Bouras, C. and V. Tsogkas (2010). Assigning web news to clusters. Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on, IEEE.

Bui, D. T., K.-T. T. Bui, Q.-T. Bui, C. Van Doan and N.-D. Hoang (2017). Hybrid Intelligent Model Based on Least Squares Support Vector Regression and Artificial Bee Colony Optimization for Time-Series Modeling and Forecasting Horizontal Displacement of Hydropower Dam. Handbook of Neural Computation, Elsevier**:** 279-293.
Cagnina, L., M. Errecalde, D. Ingaramo and P. Rosso (2014). "An efficient particle swarm optimization approach to cluster short texts." Information Sciences **265**: 36-49.
Casillas, A., M. T. G. de Lena and R. Martínez (2003). Document Clustering into an Unknown Number of Clusters Using a Genetic Algorithm. Text, Speech and Dialogue, Berlin, Heidelberg, Springer Berlin Heidelberg.
Celebi, M. E. (2015). Partitional Clustering Algorithms, Springer.
Chandrashekar, G. and F. Sahin (2014). "A survey on feature selection methods." Computers & Electrical Engineering **40**(1): 16-28.
Chen, Q., K. K. F. Yuen and C. Guan (2017). Towards a Hybrid Approach of Self-Organizing Map and Density-Based Spatial Clustering of Applications with Noise for Image Segmentation. Developments in eSystems Engineering (DeSE), 2017 10th International Conference on, IEEE.
Chen, X.-w. (2003). "An improved branch and bound algorithm for feature selection." Pattern Recognition Letters **24**(12): 1925-1933.
Chen, Y.-P., Y. Li, G. Wang, Y.-F. Zheng, Q. Xu, J.-H. Fan and X.-T. Cui (2017). "A novel bacterial foraging optimization algorithm for feature selection." Expert Systems with Applications **83**: 1-17.
Choi, C. and J.-J. Lee (1998). "Chaotic local search algorithm." Artificial Life and Robotics **2**(1): 41-47.
Chunming, F., X. Yadong, C. JIANG, H. Xu and Z. HUANG (2017). "Improved Differential Evolution with Shrinking Space Technique for Constrained Optimization." Chinese Journal of Mechanical Engineering: 1-13.
Cobos, C., J. Andrade, W. Constain, M. Mendoza and E. León (2010). Web document clustering based on global-best harmony search, K-means, frequent term sets and Bayesian information criterion. Evolutionary Computation (CEC), 2010 IEEE Congress on, IEEE.
Cobos, C., M. Mendoza, E. Leon, M. Manic and E. Herrera-Viedma (2013). Clustering of web search results based on an Iterative Fuzzy C-means Algorithm and Bayesian Information Criterion. IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint, IEEE.
Cobos, C., C. Montealegre, M.-F. Mejía, M. Mendoza and E. León (2010). Web document clustering based on a new niching memetic algorithm, term-document matrix and Bayesian information criterion. Evolutionary Computation (CEC), 2010 IEEE Congress on, IEEE.
Cui, L., G. Li, Q. Lin, J. Chen and N. Lu (2016). "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations." Computers & Operations Research **67**: 155-173.
Cui, X., T. E. Potok and P. Palathingal (2005). Document clustering using particle swarm optimization. Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE, IEEE.
Dadaneh, B. Z., H. Y. Markid and A. Zakerolhosseini (2016). "Unsupervised probabilistic feature selection using ant colony optimization." Expert Systems with Applications **53**: 27-42.

Daoud, A. S., A. Sallam and M. E. Wheed (2017). Improving Arabic document clustering using K-means algorithm and Particle Swarm Optimization. Intelligent Systems Conference (IntelliSys), 2017, IEEE.
Das, S., A. Abraham and A. Konar (2008). "Automatic clustering using an improved differential evolution algorithm." Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **38**(1): 218-237.
Das, S., A. Abraham and A. Konar (2008). "Automatic clustering using an improved differential evolution algorithm." IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans **38**(1): 218-237.
Das, S., S. S. Mullick and P. N. Suganthan (2016). "Recent advances in differential evolution–an updated survey." Swarm and Evolutionary Computation **27**: 1-30.
Dasgupta, D. and Z. Michalewicz (2013). Evolutionary algorithms in engineering applications, Springer Science & Business Media.
Dasgupta, S. (2008). The hardness of k-means clustering, Department of Computer Science and Engineering, University of California ….
Debole, F. and F. Sebastiani (2005). "An analysis of the relative hardness of Reuters-21578 subsets." Journal of the American Society for Information Science and technology **56**(6): 584-596.
Deelers, S. and S. Auwatanamongkol (2007). "Enhancing K-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance." International Journal of Computer Science **2**(4): 247-252.
Derrac, J., S. García, D. Molina and F. Herrera (2011). "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms." Swarm and Evolutionary Computation **1**(1): 3-18.
Dhillon, I. S., J. Fan and Y. Guan (2001). Efficient clustering of very large document collections. Data mining for scientific and engineering applications, Springer**:** 357-381.
Dhillon, I. S., S. Mallela and D. S. Modha (2003). Information-theoretic co-clustering. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM.
Diaz-Valenzuela, I., V. Loia, M. J. Martin-Bautista, S. Senatore and M. A. Vila (2015). "Automatic constraints generation for semisupervised clustering: experiences with documents classification." Soft Computing.
El Akadi, A., A. Amine, A. El Ouardighi and D. Aboutajdine (2011). "A two-stage gene selection scheme utilizing MRMR filter and GA wrapper." Knowledge and Information Systems **26**(3): 487-500.
Farnstrom, F. and J. Lewis (2008). Fast, single-pass K-means algorithms.
Feng, A. (2007). Document clustering: an optimization problem. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM.
Ferreira, A. J., #225 and R. A. T. Figueiredo (2012). "Efficient feature selection filters for high-dimensional data." Pattern Recogn. Lett. **33**(13): 1794-1804.
Fodeh, S., B. Punch and P.-N. Tan (2011). "On ontology-driven document clustering using core semantic features." Knowledge and information systems **28**(2): 395-421.
Forsati, R., A. Keikha and M. Shamsfard (2015). "An improved bee colony optimization algorithm with an application to document clustering." Neurocomputing **159**: 9-26.

Forsati, R., M. Mahdavi, M. Kangavari and B. Safarkhani (2008). Web page clustering using harmony search optimization. Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on, IEEE.
Forsati, R., M. Mahdavi, M. Shamsfard and M. Reza Meybodi (2013). "Efficient stochastic algorithms for document clustering." Information Sciences **220**: 269-291.
Gao, X. Z., X. Wang and K. Zenger (2015). "A memetic-inspired harmony search method in optimal wind generator design." International Journal of Machine Learning and Cybernetics **6**(1): 43-58.
Geem, Z. W., J. H. Kim and G. Loganathan (2001). "A new heuristic optimization algorithm: harmony search." Simulation **76**(2): 60-68.
Geng, Y.-a., Q. Li, R. Zheng, F. Zhuang, R. He and N. Xiong (2018). "RECOME: a new density-based clustering algorithm using relative KNN kernel density." Information Sciences **436**: 13-30.
Ghareb, A. S., A. A. Bakar and A. R. Hamdan (2016). "Hybrid feature selection based on enhanced genetic algorithm for text categorization." Expert Systems with Applications **49**: 31-47.
Goldberg, D. E. and J. H. Holland (1988). "Genetic algorithms and machine learning." Machine learning **3**(2): 95-99.
Guan, S.-U., J. Liu and Y. Qi (2004). "An incremental approach to contribution-based feature selection." Journal of Intelligent Systems **13**(1): 15-42.
Gui, J., Z. Sun, S. Ji, D. Tao and T. Tan (2017). "Feature selection based on structured sparsity: A comprehensive study." IEEE transactions on neural networks and learning systems.
Günal, S. (2012). "Hybrid feature selection for text classification." Turkish Journal of Electrical Engineering & Computer Sciences **20**(Sup. 2): 1296-1311.
Guo, Z., H. Huang, C. Deng, X. Yue and Z. Wu (2015). "An enhanced differential evolution with elite chaotic local search." Computational intelligence and neuroscience **2015**: 6.
Gwo-Ching, L. and T. Ta-Peng (2006). "Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting." IEEE Transactions on Evolutionary Computation **10**(3): 330-340.
Han, M. and W. Ren (2015). "Global mutual information-based feature selection approach using single-objective and multi-objective optimization." Neurocomputing **168**: 47-54.
Hao, H., C.-L. Liu and H. Sako (2003). Comparison of genetic algorithm and sequential search methods for classifier subset selection. ICDAR, Citeseer.
He, Y., S. C. Hui and Y. Sim (2006). A novel ant-based clustering approach for document clustering. Asia Information Retrieval Symposium, Springer.
Holland, J. H. (1975). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, U Michigan Press.
Hong, S.-S., W. Lee and M.-M. Han (2015). "The Feature Selection Method based on Genetic Algorithm for Efficient of Text Clustering and Text Classification." International Journal of Advances in Soft Computing & Its Applications **7**(1).
Hruschka, E. R., R. J. Campello, A. Freitas and A. C. De Carvalho (2009). "A survey of evolutionary algorithms for clustering." Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **39**(2): 133-155.
Hsu, H.-H., C.-W. Hsieh and M.-D. Lu (2011). "Hybrid feature selection by combining filters and wrappers." Expert Systems with Applications **38**(7): 8144-8150.

Hua, J., W. D. Tembe and E. R. Dougherty (2009). "Performance of feature-selection methods in the classification of high-dimension data." Pattern Recognition **42**(3): 409-424.
Jain, A. K. (2010). "Data clustering: 50 years beyond K-means." Pattern recognition letters **31**(8): 651-666.
Jensi, R. and D. G. W. Jiji (2014). "A Survey on optimization approaches to text document clustering." arXiv preprint arXiv:1401.2229.
Jia, D., G. Zheng and M. K. Khan (2011). "An effective memetic differential evolution algorithm based on chaotic local search." Information Sciences **181**(15): 3175-3187.
Jo, T. (2009). Clustering news groups using inverted index based NTSO. Networked Digital Technologies, 2009. NDT'09. First International Conference on, IEEE.
Joyce, T. and J. M. Herrmann (2018). A Review of No Free Lunch Theorems, and Their Implications for Metaheuristic Optimisation. Nature-Inspired Algorithms and Applied Optimization, Springer**:** 27-51.
Jun, S., S.-S. Park and D.-S. Jang (2014). "Document clustering method using dimension reduction and support vector clustering to overcome sparseness." Expert Systems with Applications **41**(7): 3204-3212.
Kannan, S. S. and N. Ramaraj (2010). "A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm." Knowledge-Based Systems **23**(6): 580-585.
Karaa, W. B. A., A. S. Ashour, D. B. Sassi, P. Roy, N. Kausar and N. Dey (2016). MEDLINE text mining: an enhancement genetic algorithm based approach for document clustering. Applications of Intelligent Optimization in Biology and Medicine, Springer**:** 267-287.
Karaboga, D. and C. Ozturk (2011). "A novel clustering approach: Artificial Bee Colony (ABC) algorithm." Applied soft computing **11**(1): 652-657.
Karol, S. and V. Mangat (2013). "Evaluation of text document clustering approach based on particle swarm optimization." Open Computer Science **3**(2): 69-90.
Khalid, S., T. Khalil and S. Nasreen (2014). A survey of feature selection and feature extraction techniques in machine learning. Science and Information Conference (SAI), 2014, IEEE.
Khorsheed, M. S. and A. O. Al-Thubaity (2013). "Comparative evaluation of text classification techniques using a large diverse Arabic dataset." Language resources and evaluation **47**(2): 513-538.
Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi (1983). "Optimization by simulated annealing." science **220**(4598): 671-680.
Kohavi, R. and G. H. John (1997). "Wrappers for feature subset selection." Artificial intelligence **97**(1): 273-324.
Kohli, S. and S. Mehrotra (2016). "A clustering approach for optimization of search result." Journal of Images and Graphics **4**(1): 63-66.
Korde, V. and C. N. Mahender (2012). "Text classification and classifiers: A survey." International Journal of Artificial Intelligence & Applications **3**(2): 85.
Kramer, O., D. E. Ciaurri and S. Koziel (2011). Derivative-free optimization. Computational optimization, methods and algorithms, Springer**:** 61-83.
Krovi, R. (1992). Genetic algorithms for clustering: a preliminary investigation. System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on, IEEE.
Kudo, M. and J. Sklansky (2000). "Comparison of algorithms that select features for pattern classifiers." Pattern recognition **33**(1): 25-41.

Kumar, S., V. K. Sharma and R. Kumari (2014). Memetic search in Artificial Bee Colony algorithm with fitness based position update. Recent Advances and Innovations in Engineering (ICRAIE), 2014, IEEE.
Kumar, V., J. K. Chhabra and D. Kumar (2015). "Automatic unsupervised feature selection using gravitational search algorithm." IETE Journal of Research **61**(1): 22-31.
kumar, Y. and G. Sahoo (2017). "A two-step artificial bee colony algorithm for clustering." Neural Computing and Applications **28**(3): 537-551.
Kwedlo, W. (2011). "A clustering method combining differential evolution with the K-means algorithm." Pattern Recognition Letters **32**(12): 1613-1621.
Lamirel, J.-C., P. Cuxac, A. S. Chivukula and K. Hajlaoui (2015). "Optimizing text classification through efficient feature selection based on quality metric." Journal of Intelligent Information Systems **45**(3): 379-396.
Lazar, C., J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. De Schaetzen, R. Duque, H. Bersini and A. Nowe (2012). "A survey on filter techniques for feature selection in gene expression microarray analysis." IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **9**(4): 1106-1119.
Lee, C.-J., C.-C. Hsu and D.-R. Chen (2017). "A hierarchical document clustering approach with frequent itemsets." International Journal of Engineering and Technology **9**(2): 174.
Lee, J. and D.-W. Kim (2013). "Feature selection for multi-label classification using multivariate mutual information." Pattern Recognition Letters **34**(3): 349-357.
Lee, J. and D.-W. Kim (2015). "Memetic feature selection algorithm for multi-label classification." Information Sciences **293**: 80-96.
Lin, C., A. Qing and Q. Feng (2011). "A comparative study of crossover in differential evolution." Journal of Heuristics **17**(6): 675-703.
Liu, G., Y. Li, X. Nie and H. Zheng (2012). "A novel clustering-based differential evolution with 2 multi-parent crossovers for global optimization." Applied Soft Computing **12**(2): 663-681.
Liu, H. and H. Motoda (2012). Feature selection for knowledge discovery and data mining, Springer Science & Business Media.
Liu, H. and L. Yu (2005). "Toward integrating feature selection algorithms for classification and clustering." IEEE Transactions on knowledge and data engineering **17**(4): 491-502.
Liu, L., J. Kang, J. Yu and Z. Wang (2005). A comparative study on unsupervised feature selection methods for text clustering. Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on, IEEE.
Lu, Y., M. Liang, Z. Ye and L. Cao (2015). "Improved particle swarm optimization algorithm and its application in text feature selection." Applied Soft Computing **35**: 629-636.
Mafarja, M. and S. Abdullah (2013). "Investigating memetic algorithm in solving rough set attribute reduction." International Journal of Computer Applications in Technology **48**(3): 195-202.
Mafarja, M. M. and S. Mirjalili (2017). "Hybrid Whale Optimization Algorithm with simulated annealing for feature selection." Neurocomputing.
Mahdavi, M. and H. Abolhassani (2009). "Harmony K-means algorithm for document clustering." Data Mining and Knowledge Discovery **18**(3): 370-391.
Maldonado, S. and R. Weber (2009). "A wrapper method for feature selection using support vector machines." Information Sciences **179**(13): 2208-2217.

Manimala, K., K. Selvi and R. Ahila (2011). "Hybrid soft computing techniques for feature selection and parameter optimization in power quality data mining." Applied Soft Computing **11**(8): 5485-5497.
Mavrovouniotis, M., C. Li and S. Yang (2017). "A survey of swarm intelligence for dynamic optimization: Algorithms and applications." Swarm and Evolutionary Computation **33**: 1-17.
Mecca, G., S. Raunich and A. Pappalardo (2007). "A new algorithm for clustering search results." Data & Knowledge Engineering **62**(3): 504-522.
Merendino, S. and M. E. Celebi (2013). A Simulated Annealing Clustering Algorithm Based On Center Perturbation Using Gaussian Mutation. FLAIRS Conference.
Mohd, M., Q. W. Bsoul, N. M. Ali, S. A. M. Noah, S. Saad, N. Omar and M. J. A. AZIZ (2012). "OPTIMAL INITIAL CENTROID IN K-MEANS FOR CRIME TOPIC." Journal of Theoretical & Applied Information Technology **45**(1).
Montazeri, M., H. R. Naji and A. Faraahi (2013). A novel memetic feature selection algorithm. Information and Knowledge Technology (IKT), 2013 5th Conference on, IEEE.
Murtagh, F. and P. Contreras (2017). "Algorithms for hierarchical clustering: an overview, II." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **7**(6).
Nanda, S. J. and G. Panda (2014). "A survey on nature inspired metaheuristic algorithms for partitional clustering." Swarm and Evolutionary computation **16**: 1-18.
Narendra, P. M. and K. Fukunaga (1977). "A branch and bound algorithm for feature subset selection." IEEE Transactions on Computers **100**(9): 917-922.
Neri, F. and C. Cotta (2012). "Memetic algorithms and memetic computing optimization: A literature review." Swarm and Evolutionary Computation **2**: 1-14.
Neri, F. and E. Mininno (2010). "Memetic compact differential evolution for cartesian robot control." Computational Intelligence Magazine, IEEE **5**(2): 54-65.
Nguyen, Q. H., Y.-S. Ong and M. H. Lim (2009). "A probabilistic memetic framework." Evolutionary Computation, IEEE Transactions on **13**(3): 604-623.
Nie, P.-y. (2005). "A filter method for solving nonlinear complementarity problems." Applied mathematics and computation **167**(1): 677-694.
Ning, Z., Y. Ong, K. Wong and M. Lim (2003). "Choice of memes in memetic algorithm."
Olabiyisi Stephen, O., M. Fagbola Temitayo, O. Omidiora Elijah and C. Oyeleye Akin "Hybrid MetaHeuristic Feature Extraction Technique for Solving Timetabling Problem."
Omran, M. G. H. and M. Mahdavi (2008). "Global-best harmony search." Applied Mathematics and Computation **198**(2): 643-656.
Onan, A. and S. Korukoğlu (2015). "A feature selection model based on genetic rank aggregation for text sentiment classification." Journal of Information Science: 0165551515613226.
Ong, Y.-S., M.-H. Lim, N. Zhu and K.-W. Wong (2006). "Classification of adaptive memetic algorithms: a comparative study." Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **36**(1): 141-152.
Oreski, D., S. Oreski and B. Klicek (2016). "Effects of dataset characteristics on the performance of feature selection techniques." Applied Soft Computing.
Pal, J. and V. Bhattacherjee (2018). Analysis of Complete-Link Clustering for Identifying Multi-attributes Software Quality Data. International Conference on Advanced Machine Learning Technologies and Applications, Springer.

Patil, H. and R. Thakur (2016). "Document Clustering: A Summarized Survey." Pattern and Data Analysis in Healthcare Settings: 264.
Peng, H., Z. Wu, P. Shao and C. Deng (2016). "Dichotomous binary differential evolution for knapsack problems." Mathematical Problems in Engineering **2016**.
Peng, L., Y. Zhang, G. Dai and M. Wang (2017). "Memetic Differential Evolution with an Improved Contraction Criterion." Computational Intelligence and Neuroscience **2017**.
Pinheiro, R. H., G. D. Cavalcanti and T. I. Ren (2015). "Data-driven global-ranking local feature selection methods for text categorization." Expert Systems with Applications **42**(4): 1941-1949.
Poikolainen, I. and F. Neri (2013). Differential evolution with concurrent fitness based local search. Evolutionary Computation (CEC), 2013 IEEE Congress on, IEEE.
Premalatha, K. and A. Natarajan (2010). "Hybrid PSO and GA models for document clustering." Int. J. Advance. Soft Comput. Appl **2**(3): 302-320.
Pudil, P., J. Novovičová and J. Kittler (1994). "Floating search methods in feature selection." Pattern recognition letters **15**(11): 1119-1125.
Qian, W. and W. Shu (2015). "Mutual information criterion for feature selection from incomplete data." Neurocomputing **168**: 210-220.
Radcliffe, N. J. and P. D. Surry (1994). Formal memetic algorithms. Evolutionary Computing, Springer**:** 1-16.
Rafi, M., S. Shahid, J. Aftab, M. F. Uddin and M. S. Shaikh (2017). Towards A Soft Computing Approach to Document Clustering. Proceedings of the 2017 International Conference on Machine Learning and Soft Computing. Ho Chi Minh City, Vietnam, ACM**:** 74-81.
Reynoso-Meza, G., J. Sanchis, X. Blasco and J. M. Herrero (2011). Hybrid DE algorithm with adaptive crossover operator for solving real-world numerical optimization problems. Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE.
Rezaee Jordehi, A., J. Jasni, N. Abd Wahab, M. Z. Kadir and M. S. Javadi (2015). "Enhanced leader PSO (ELPSO): A new algorithm for allocating distributed TCSC's in power systems." International Journal of Electrical Power & Energy Systems **64**: 771-784.
Ripley, B. D. (2007). Pattern recognition and neural networks, Cambridge university press.
Saatchi, S. and C. C. Hung (2005). Hybridization of the ant colony optimization with the k-means algorithm for clustering. Image Analysis, Springer**:** 511-520.
Saeys, Y., I. Inza and P. Larrañaga (2007). "A review of feature selection techniques in bioinformatics." bioinformatics **23**(19): 2507-2517.
Saha, S. and R. Das (2017). "Exploring differential evolution and particle swarm optimization to develop some symmetry-based automatic clustering techniques: application to gene clustering." Neural Computing and Applications: 1-23.
Saiyad, N. Y., H. B. Prajapati and V. K. Dabhi (2016). A survey of document clustering using semantic approach. Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on, IEEE.
Saraç, E. and S. A. Özel (2014). "An ant colony optimization based feature selection for web page classification." The Scientific World Journal **2014**.
Saruhan, H. (2014). "Differential evolution and simulated annealing algorithms for mechanical systems design." Engineering Science and Technology, an International Journal **17**(3): 131-136.

Senthilnath, J., S. Kulkarni, D. Raghuram, M. Sudhindra, S. Omkar, V. Das and V. Mani (2016). "A novel harmony search-based approach for clustering problems." International Journal of Swarm Intelligence **2**(1): 66-86.
Sharp, H. (1968). "Cardinality of finite topologies." Journal of Combinatorial Theory **5**(1): 82-86.
Shreem, S. S., S. Abdullah and M. Z. A. Nazri (2016). "Hybrid feature selection algorithm using symmetrical uncertainty and a harmony search algorithm." International Journal of Systems Science **47**(6): 1312-1329.
Song, W. and S. C. Park (2009). "Genetic algorithm for text clustering based on latent semantic indexing." Computers & Mathematics with Applications **57**(11): 1901-1907.
Song, W., Y. Qiao, S. C. Park and X. Qian (2015). "A hybrid evolutionary computation approach with its application for optimizing text document clustering." Expert Systems with Applications **42**(5): 2517-2524.
Souza, J., N. Japkowicz and S. Matwin (2005). "Feature selection with a general hybrid algorithm." Feature Selection for Data Mining: 45.
Storn, R. and K. Price (1997). "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces." Journal of Global Optimization **11**(4): 341-359.
Subramanian, S. and D. Vora (2016). "Unsupervised Text Classification and Search using Word Embeddings on a Self-Organizing Map." International Journal of Computer Applications **156**(11).
Tabakhi, S. and P. Moradi (2015). "Relevance–redundancy feature selection based on ant colony optimization." Pattern recognition **48**(9): 2798-2811.
Tabakhi, S., P. Moradi and F. Akhlaghian (2014). "An unsupervised feature selection algorithm based on ant colony optimization." Engineering Applications of Artificial Intelligence **32**: 112-123.
Tang, B., H. He, P. M. Baggenstoss and S. Kay (2016). "A Bayesian classification approach using class-specific features for text categorization." IEEE Transactions on Knowledge and Data Engineering **28**(6): 1602-1606.
Tang, B., S. Kay and H. He (2016). "Toward optimal feature selection in naive Bayes for text categorization."
Tang, B., S. Kay and H. He (2016). "Toward optimal feature selection in naive Bayes for text categorization." IEEE Transactions on Knowledge and Data Engineering **28**(9): 2508-2521.
Tran, B., B. Xue, M. Zhang and S. Nguyen (2016). "Investigation on particle swarm optimisation for feature selection on high-dimensional data: Local search and selection bias." Connection Science **28**(3): 270-294.
Tran, T., B. Vo, T. T. N. Le and N. T. Nguyen (2017). "Text Clustering Using Frequent Weighted Utility Itemsets." Cybernetics and Systems **48**(3): 193-209.
Tutkan, M., M. C. Ganiz and S. Akyokuş (2016). "Helmholtz principle based supervised and unsupervised feature selection methods for text mining." Information Processing & Management **52**(5): 885-910.
Uğuz, H. (2011). "A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm." Knowledge-Based Systems **24**(7): 1024-1032.
Uysal, A. K. (2016). "An improved global feature selection scheme for text classification." Expert systems with Applications **43**: 82-92.
Uysal, A. K. and S. Gunal (2014). "The impact of preprocessing on text classification." Information Processing & Management **50**(1): 104-112.

Van der Merwe, D. and A. P. Engelbrecht (2003). Data clustering using particle swarm optimization. Evolutionary Computation, 2003. CEC'03. The 2003 Congress on, IEEE.
Vergara, J. R. and P. A. Estévez (2014). "A review of feature selection methods based on mutual information." Neural Computing and Applications **24**(1): 175-186.
Vesterstrom, J. and R. Thomsen (2004). A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. IEEE Congress on Evolutionary Computation.
Wang, F., D. Zhang and N. Bao (2009). "Fuzzy document clustering based on ant colony algorithm." Advances in Neural Networks–ISNN 2009: 709-716.
Wang, S., Y. Li and H. Yang (2017). "Self-adaptive differential evolution algorithm with improved mutation mode." Applied Intelligence: 1-15.
Xue, B., M. Zhang, W. N. Browne and X. Yao (2016). "A survey on evolutionary computation approaches to feature selection." IEEE Transactions on Evolutionary Computation **20**(4): 606-626.
Xue, Y., J. Jiang, B. Zhao and T. Ma (2017). "A self-adaptive artificial bee colony algorithm based on global best for global optimization." Soft Computing: 1-18.
Yaghoubyan, S. H., M. A. Maarof, A. Zainal and M. M. Oghaz (2016). "A SURVEY OF FEATURE EXTRACTION TECHNIQUES IN CONTENT-BASED ILLICIT IMAGE DETECTION." Journal of Theoretical and Applied Information Technology **87**(1): 110.
Yang, F., T. Sun and C. Zhang (2009). "An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization." Expert Systems with Applications **36**(6): 9847-9852.
Yang, X.-S. and S. Deb (2009). Cuckoo search via Lévy flights. Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, IEEE.
Yu, L. and H. Liu (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. Proceedings of the 20th international conference on machine learning (ICML-03).
Zahran, B. M. and G. Kanaan (2009). "Text Feature Selection using Particle Swarm Optimization Algorithm 1."
Zaw, M. M. and E. E. Mon (2015). Web Document Clustering by Using PSO-Based Cuckoo Search Clustering Algorithm. Recent Advances in Swarm Intelligence and Evolutionary Computation, Springer**:** 263-281.
Zhang, C., J. Chen and B. Xin (2013). "Distributed memetic differential evolution with the synergy of Lamarckian and Baldwinian learning." Applied Soft Computing **13**(5): 2947-2959.
Zhang, H. and G. Sun (2002). "Feature selection using tabu search method." Pattern recognition **35**(3): 701-711.
Zhang, J. and A. C. Sanderson (2009). "JADE: adaptive differential evolution with optional external archive." IEEE Transactions on evolutionary computation **13**(5): 945-958.
Zhang, Y., S. Wang, P. Phillips and G. Ji (2014). "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection." Knowledge-Based Systems **64**: 22-31.
Zhao, Y. and G. Karypis (2001). "Criterion functions for document clustering: Experiments and analysis."
Zhao, Y. and G. Karypis (2002). Evaluation of hierarchical clustering algorithms for document datasets. Proceedings of the eleventh international conference on Information and knowledge management, ACM.

Zheng, L., R. Diao and Q. Shen (2015). "Self-adjusting harmony search-based feature selection." Soft Computing **19**(6): 1567-1579.
Zhu, Z., Y.-S. Ong and M. Dash (2007). "Wrapper–filter feature selection algorithm using a memetic framework." Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **37**(1): 70-76.
Zong, W., F. Wu, L.-K. Chu and D. Sculli (2015). "A discriminative and semantic feature selection method for text categorization." International Journal of Production Economics **165**: 215-222.
Zorarpacı, E. and S. A. Özel (2016). "A hybrid approach of differential evolution and artificial bee colony for feature selection." Expert Systems with Applications **62**: 91-103.