

The copyright © of this thesis belongs to its rightful author and/or other copyright owner. Copies can be accessed and downloaded for non-commercial or learning purposes without any charge and permission. The thesis cannot be reproduced or quoted as a whole without the permission from its rightful owner. No alteration or changes in format is allowed without permission from its rightful owner.



**ACO-BASED FEATURE SELECTION ALGORITHM FOR  
CLASSIFICATION**



**DOCTOR OF PHILOSOPHY  
UNIVERSITI UTARA MALAYSIA  
2022**



Awang Had Salleh  
Graduate School  
of Arts And Sciences

Universiti Utara Malaysia

**PERAKUAN KERJA TESIS / DISERTASI**  
(*Certification of thesis / dissertation*)

Kami, yang bertandatangan, memperakukan bahawa  
(*We, the undersigned, certify that*)

**HASSAN FOUAD ABBAS AL-MAZINI**

calon untuk Ijazah **PhD**  
(*candidate for the degree of*)

telah mengemukakan tesis / disertasi yang bertajuk:  
(*has presented his/her thesis / dissertation of the following title*):

**"ACO-BASED FEATURE SELECTION ALGORITHM FOR CLASSIFICATION"**

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.  
(*as it appears on the title page and front cover of the thesis / dissertation*).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : **08 Mei 2022.**

*That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:*

**08 May 2022.**

Pengerusi Viva:  
(*Chairman for VIVA*)

**Prof. Ts. Dr. Suhaidi Hassan**

Tandatangan  
(*Signature*)

Pemeriksa Luar:  
(*External Examiner*)

**Prof. Dr. Rozaida Ghazali**

Tandatangan  
(*Signature*)

Pemeriksa Dalam:  
(*Internal Examiner*)

**Assoc. Prof. Dr. Yuhanis Yusof**

Tandatangan  
(*Signature*)

Nama Penyelia/Penyelia-penyelia:  
(*Name of Supervisor/Supervisors*)

**Prof. Dr. Ku Ruhana Ku Mahamud**

Tandatangan  
(*Signature*)

Tarikh:  
(*Date*) **08 May 2022**

## Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for a scholarly purpose may be granted by my supervisor or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain is not allowed without my written permission. It is also understood that due to recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use, which may be made of any material from my thesis.

Requests for permission to copy or to make other uses of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences  
UUM College of Arts and Sciences  
Universiti Utara Malaysia  
06010 UUM Sintok  

---

Universiti Utara Malaysia



## Abstrak

Set data dengan bilangan rekod yang kecil tetapi bilangan atribut yang besar mewakili fenomena yang dipanggil “kutukan dimensi”. Pengelasan set data jenis ini memerlukan kaedah pemilihan ciri (FS) untuk mengekstrakan maklumat berguna. Algoritma modified graph clustering ant colony optimisation (MGCACO) ialah kaedah pemilihan ciri yang berkesan yang dibangunkan berdasarkan pengelompokan ciri berkorelasi. Walau bagaimanapun, algoritma MGCACO mempunyai tiga kelemahan utama dalam menghasilkan subset ciri kerana kaedah pengelompokan, kepekaan parameter, dan penentuan subset akhirnya. Algoritma enhanced graf clustering ant colony optimisation (EGCACO) dicadangkan untuk menyelesaikan tiga (3) masalah algoritma MGCACO. Cadangan penambahbaikan termasuk: (i) kaedah pengelompokan ciri ACO untuk mendapatkan kelompok ciri berkorelasi tinggi; (ii) teknik pemilihan penyesuaian untuk pembinaan subset daripada kelompok ciri; dan, (iii) kaedah berasaskan genetik untuk menghasilkan subset akhir ciri. Kaedah pengelompokan ciri ACO menggunakan keupayaan pelbagai mekanisma seperti pengukuhan dan kepelbagaian untuk pengoptimuman tempatan dan global untuk menyediakan ciri berkorelasi tinggi. Teknik penyesuaian untuk pemilihan semut membolehkan parameter berubah secara adaptif berdasarkan maklum balas ruang carian. Kaedah genetik menentukan subset akhir secara automatik, berdasarkan pengiraan kualiti silang dan subset. Prestasi algoritma yang dicadangkan telah dinilai ke atas 18 set data penanda aras dari repositori University California Irvine (UCI) dan sembilan (9) set data mikroarray asid deoksiribonukleik (DNA) ke atas 15 algoritma metaheuristik penanda aras. Keputusan eksperimen algoritma EGCACO pada dataset UCI adalah lebih baik daripada algoritma pengoptimuman penanda aras lain dari segi bilangan ciri yang dipilih dan kedua terbaik untuk ketepatan pengelasan. Selanjutnya, eksperimen ke atas sembilan (9) set data microarray DNA menunjukkan bahawa algoritma EGCACO adalah lebih unggul daripada algoritma penanda aras dari segi ketepatan klasifikasi dan bilangan ciri yang dipilih. Algoritma EGCACO yang dicadangkan boleh digunakan untuk pemilihan ciri dalam tugas pengelasan microarray DNA yang melibatkan sebarang saiz set data dan dalam pelbagai domain aplikasi.

**Kata Kunci:** Pemilihan, Pengelompokan Ciri, Genetik, Pengoptimuman Koloni Semut, Microarray

## Abstract

Dataset with a small number of records but big number of attributes represents a phenomenon called “curse of dimensionality”. The classification of this type of dataset requires Feature Selection (FS) methods for the extraction of useful information. The modified graph clustering ant colony optimisation (MGCACO) algorithm is an effective FS method that was developed based on grouping the highly correlated features. However, the MGCACO algorithm has three main drawbacks in producing a features subset because of its clustering method, parameter sensitivity, and the final subset determination. An enhanced graph clustering ant colony optimisation (EGCACO) algorithm is proposed to solve the three (3) MGCACO algorithm problems. The proposed improvement includes: (i) an ACO feature clustering method to obtain clusters of highly correlated features; (ii) an adaptive selection technique for subset construction from the clusters of features; and (iii) a genetic-based method for producing the final subset of features. The ACO feature clustering method utilises the ability of various mechanisms such as intensification and diversification for local and global optimisation to provide highly correlated features. The adaptive technique for ant selection enables the parameter to adaptively change based on the feedback of the search space. The genetic method determines the final subset, automatically, based on the crossover and subset quality calculation. The performance of the proposed algorithm was evaluated on 18 benchmark datasets from the University California Irvine (UCI) repository and nine (9) deoxyribonucleic acid (DNA) microarray datasets against 15 benchmark metaheuristic algorithms. The experimental results of the EGCACO algorithm on the UCI dataset are superior to other benchmark optimisation algorithms in terms of the number of selected features for 16 out of the 18 UCI datasets (88.89%) and the best in eight (8) (44.47%) of the datasets for classification accuracy. Further, experiments on the nine (9) DNA microarray datasets showed that the EGCACO algorithm is superior than the benchmark algorithms in terms of classification accuracy (first rank) for seven (7) datasets (77.78%) and demonstrates the lowest number of selected features in six (6) datasets (66.67%). The proposed EGCACO algorithm can be utilised for FS in DNA microarray classification tasks that involve large dataset size in various application domains.

**Keywords:** Feature Selection, Feature Clustering, Genetic, Ant Colony Optimisation, Microarray

## Acknowledgement



And say, “My Lord, increase me in knowledge.”

“Writing this thesis has been fascinating and extremely rewarding. I would like to thank a number of people who have contributed to the final result in many different ways: To stay with, I pay my thanks to GOD, the almighty to have bestowed upon me good health, courage, inspiration, zeal, and the light. After GOD, I would like to express my gratitude to my supervisor Prof. Dr. Ku Ruhana Ku Mahamud for the useful comments, remarks, continuous support, generosity, and engagement through the learning process of this work. Also, I feel a deep sense of gratitude to my father Fouad Abbas Almazini , and my elder brother Aymen Almazini who formed part of my vision and taught me the good things that really matter and guided me in doing my work. There are not enough words to write down my feelings for my mother and my aunt, my sister and my sister-in-law and all my relatives for providing me constant encouragement, financial support, and helping me spiritually. Last, but not least, I express my gratitude from the core of my heart to all my colleagues, friends, and brothers Asst. Prof. Dr. Ayad Mohammed, Asst. Prof. Dr. Hayder Naser Khraibet, Dr. Hussein Almazini, Dr. Salah Mortada, Ammar, Jawad, and Hassan for extending their unstated support, timely motivation, sympathetic attitude, and unfailing assistance during the entire work.

## Table of Contents

Permission to Use .....	i
Abstrak .....	ii
Abstract .....	iii
Acknowledgement .....	iv
Table of Contents .....	v
List of Tables .....	viii
List of Figures .....	x
List of Abbreviations .....	xii
<b>CHAPTER ONE INTRODUCTION .....</b>	<b>1</b>
1.1 Problem Statement .....	8
1.2 Research Questions .....	11
1.3 Research Objectives .....	11
1.4 Significance of the Research .....	12
1.5 Scope of the Research .....	13
1.6 Thesis Organisation .....	14
<b>CHAPTER TWO LITERATURE REVIEW .....</b>	<b>15</b>
2.1 Introduction .....	15
2.2 Microarray Data Mining .....	15
2.2.1 Classification in Biomedical .....	17
2.2.2 Biomedical Clustering .....	19
2.2.3 Gene Selection Mining in Microarray Data .....	20
2.3 Feature Selection .....	22
2.3.1 Subset Generation .....	24
2.3.2 Subset Evaluation .....	25
2.3.3 Stopping Criteria .....	27
2.4 Feature Selection-based Clustering Method .....	29
2.4.1 Filter Method .....	29
2.4.2 Wrapper Method .....	34
2.4.3 Hybrid Method .....	39

2.4.4 Ant Colony Optimisation.....	42
2.4.5 Discussion on Feature Selection-based Clustering Method .....	44
2.5 Control Parameter .....	48
2.5.1 Deterministic Parameter Control Approach .....	51
2.5.2 Aggregated Parameter Control Approach.....	53
2.5.3 Adaptive Parameter Control Approach.....	55
2.5.4 Discussion on Methods for Parameter Control.....	62
2.6 Subset Determination.....	64
2.6.1 Predetermined Approach .....	64
2.6.2 Automatically Determined-based .....	69
2.6.3 Genetic Algorithm .....	71
2.6.4 Discussion of Methods for Subset Determination .....	73
2.7 Discussion .....	76
2.8 Summary .....	77
<b>CHAPTER THREE RESEARCH METHODOLOGY.....</b>	<b>78</b>
3.1 Introduction.....	78
3.2 Research Framework .....	78
3.3 Research Method .....	80
3.3.1 Ant Colony Optimisation for Features Clustering.....	81
3.3.2 Adaptive Technique for Parameter Control.....	83
3.3.3 Genetic-based Subset Determination.....	85
3.4 Performance Evaluation.....	86
3.4.1 Dataset .....	88
3.4.2 Classifier and Evaluation Criteria.....	91
3.5 Summary .....	92
<b>CHAPTER FOUR PROPOSED ENHANCED GRAPH CLUSTERING WITH ANT COLONY OPTIMISATION ALGORITHM FOR FEATURE SELECTION .....</b>	<b>93</b>
4.1 Introduction.....	93
4.2 ACO-based Features Clustering .....	94
4.3 Adaptive Feature Selection Technique .....	100

4.4 Genetic-based Subset Determination .....	109
4.5 The Proposed Enhanced GCACO Algorithm .....	115
4.6 Summary .....	121
<b>CHAPTER FIVE EXPERIMENTAL RESULTS.....</b>	<b>122</b>
5.1 Introduction.....	122
5.2 Experimental Design.....	122
5.3 Results and Analysis of the EC-GCACO Algorithm.....	126
5.3.1 Experiment on the UCI Datasets .....	127
5.3.2 Experiment on the Microarray Dataset.....	132
5.4 Results and Analysis of the A-GCACO Algorithm.....	136
5.4.1 Experiment on the UCI Datasets .....	136
5.4.2 Experiment on the Microarray Datasets .....	141
5.5 Results and Analysis of the G-GCACO Algorithm.....	145
5.5.1 Experiment on the UCI Datasets .....	146
5.5.2 Experiment on the Microarray Datasets .....	153
5.6 Results and Analysis of the EGCACO Algorithm.....	157
5.6.1 Performance of EGCACO on the UCI Datasets .....	158
5.6.2 Performance of EGCACO on the Microarray Datasets.....	166
5.7 Summary .....	174
<b>CHAPTER SIX CONCLUSION, LIMITATIONS AND FUTURE WORK ...</b>	<b>175</b>
6.1 Introduction.....	175
6.2 Research Contribution .....	175
6.2.1 Knowledge Contribution.....	175
6.2.2 Practical Contribution .....	177
6.3 Limitations and Future Work.....	178
<b>REFERENCES.....</b>	<b>180</b>

## List of Tables

Table 2.1 Advantages and disadvantages of FS methods .....	44
Table 2.2 Summary of FS based on clustering method .....	45
Table 2.3 Subset size determination approach for FS in ACO algorithm.....	74
Table 3.1 Main dataset features to be used in the experiment .....	89
Table 3.2 Confusion Matrices .....	92
Table 5.1 ACO-based features clustering experimental parameters.....	127
Table 5.2 Average classification accuracy using support vector machine classifier on UCI datasets .....	128
Table 5.3 Average classification accuracy using k-NN classifier on UCI datasets .	129
Table 5.4 Average classification accuracy using decision tree classifier on UCI datasets .....	130
Table 5.5 Average classification accuracy using random forest classifier on UCI datasets .....	131
Table 5.6 Results summary on UCI datasets .....	132
Table 5.7 Average classification accuracy using support vector machine classifier on microarray datasets .....	133
Table 5.8 Average classification accuracy using k-NN classifier on microarray datasets .....	133
Table 5.9 Average classification accuracy using decision tree classifier on microarray datasets .....	134
Table 5.10 Average classification accuracy using random forest classifier on microarray datasets .....	134
Table 5.11 Average classification accuracy, standard deviation and performance rank on microarray datasets .....	135
Table 5.12 ACO-based adaptive selection technique experimental parameters .....	136
Table 5.13 Average classification accuracy using support vector machine classifier on UCI datasets .....	137
Table 5.14 Average classification accuracy using k-NN classifier on UCI datasets	138

Table 5.15 Average classification accuracy using decision tree classifier on UCI datasets .....	139
Table 5.16 Average classification accuracy using random forest classifier on UCI dataset .....	140
Table 5.17 Result summary on average classification accuracy on UCI datasets ...	141
Table 5.18 Average classification accuracy using support vector machine classifier on microarray datasets .....	142
Table 5.19 Average classification accuracy using k-NN classifier on microarray datasets .....	142
Table 5.20 Average classification accuracy using decision tree classifier on microarray datasets .....	143
Table 5.21 Average classification accuracy using random forest classifier on microarray datasets .....	143
Table 5.22 Summary of results on microarray datasets .....	144
Table 5.23 Final subset determination experimental parameters .....	145
Table 5.24 Classification accuracy results on UCI datasets .....	147
Table 5.25 Number of selected features results on UCI datasets .....	148
Table 5.26 Performance rank on UCI datasets .....	152
Table 5.27 Classification accuracy result on microarray datasets .....	153
Table 5.28 Results of number of selected features on microarray datasets .....	154
Table 5.29 Performance rank on microarray datasets .....	156
Table 5.30 Classification accuracy results on UCI datasets .....	159
Table 5.31 Number of selected features on UCI datasets .....	160
Table 5.32 Average of precision, recall, and f-score of top four (4) FS algorithms	164
Table 5.33 Rank performance on UCI datasets .....	165
Table 5.34 Classification accuracy and standard deviation results on microarray datasets .....	168
Table 5.35 Results of number of selected features on microarray datasets .....	169
Table 5.36 Performance rank on microarray datasets .....	172

## List of Figures

Figure 1.1: Clustering-based FS process (Cai et al., 2018).....	5
Figure 2.1: Feature selection's three-step process (Liu & Yu, 2005).....	23
Figure 2.2: Taxonomy for parameter control.....	50
Figure 2.3: Distribution of control mechanisms .....	63
Figure 3.1: Research framework.....	79
Figure 3.2: Block diagram of the enhancement in MGCACO algorithm.....	80
Figure 3.3: Example representing a solution .....	81
Figure 3.4: Proposed ACO algorithm for feature clustering.....	83
Figure 3.5: Adaptation technique of a threshold value .....	84
Figure 3.6: Single point crossover .....	85
Figure 3.7: Process of GA-based subset optimisation .....	86
Figure 4.1: Solution representation.....	97
Figure 4.2: Pseudocode of the local search for ACO-based feature clustering method .....	98
Figure 4.3: Pseudocode of ACO-based feature clustering method.....	100
Figure 4.4: Ant selection process.....	106
Figure 4.5: Pseudocode of A-GCACO technique.....	108
Figure 4.6: Example of the final subset determination using user assign parameter.....	110
Figure 4.7: Initial population procedure of the G-GCACO .....	112
Figure 4.8: Crossover operator pseudocode.....	113
Figure 4.9: Example of crossover operation with single random-point and the minimum value as the middle section .....	114
Figure 4.10: Selection process of the best subset .....	115
Figure 4.11: Proposed EGCACO and MGCACO algorithm.....	117
Figure 4.12: Pseudocode of EGCACO algorithm.....	120
Figure 5.1: Experimental Design.....	123
Figure 5.2: Average classification accuracy for UCI datasets.....	132
Figure 5.3: Average classification accuracy of microarray data.....	135
Figure 5.4: Average classification accuracy of all UCI data .....	141
Figure 5.5: Average classification accuracy of microarray data.....	144
Figure 5.6: Graphical comparison for the average classification accuracy on UCI data .....	150
Figure 5.7: Graphical comparison for the average selected features on UCI data.....	150
Figure 5.8: Performance rank plot using UCI datasets .....	152

Figure 5.9: Graphical comparison for the average classification accuracy on microarray data .....	154
Figure 5.10: Graphical comparison for the average selected features on microarray data ..	155
Figure 5.11: Graphical comparison for the average classification accuracy on microarray data.....	155
Figure 5.12: Graphical comparison for the average classification accuracy on microarray data.....	156
Figure 5.13: Performance rank plot using microarray datasets .....	157
Figure 5.14: Graphical comparison for the average classification accuracy on UCI data...	161
Figure 5.15: Graphical comparison for the average selected features on UCI data.....	162
Figure 5.16: Performance rank plot using UCI datasets .....	165
Figure 5.17: Graphical comparison for the average classification accuracy on microarray data.....	171
Figure 5.18: Graphical comparison for the average selected features on microarray data ..	171
Figure 5.19: Rank plot for microarray datasets .....	173



## List of Abbreviations

ABC	Artificial Bee Colony
ACO	Ant Colony Optimisation
ALO	Ant Lion Optimiser
ASGW	Adaptive Switcher Grey Whale
ASO	Atom Search Optimisation
BA	Bat Algorithm
bALO-QR	Binary Ant Lion Optimiser with the Quickreduct
BGWOPSO	Grey Wolf Optimisation and Particle Swarm Optimisation
CP	Control Parameter
DE	Differential Evolution
DNA	Deoxyribonucleic Acid
ECWSA	Embedded Chaotic Whale Survival Algorithm
FS	Feature Selection
GA	Genetic Algorithm
GCACO	Graph Clustering-Based Ant Colony Optimisation
GCACOELM	Graph Clustering-Based ACO with Extreme Learning Machine
GOA	Grasshopper Optimisation Algorithm
GSA	Gravitational Search Algorithm
GWO	Grey Wolf Optimisation
HHO	Harris Hawks Optimiser
HSGW	Hybrid Serial Grey Whale
k-NN	k-Nearest Neighbor
MGCACO	Modified Graph Clustering-Based Ant Colony Optimisation
MGSACO	Microarray Gene Selection Based on Ant Colony Optimisation
MLACO	Multi-Label Ant Colony Optimisation
MDA	Mean Decrease in Accuracy
PSO	Particle Swarm Optimisation
RRFSACO	Relevance-Redundancy Feature Selection Based on ACO
RSGW	Random Switcher Grey Whale
SI	Swarm Intelligence
SSA	Salp Swarm Algorithm
TGA	Tree Growth Algorithm
UCI	University of California Irvine
UFSACO	Unsupervised Feature Selection-Based Ant Colony Optimisation
UPFS	Unsupervised Probabilistic Feature Selection
WOA	Whale Optimisation Algorithm
WOA-CM	Whale Optimisation Algorithm with Crossover and Mutation
WOASAT	Whale Optimisation Algorithm with Simulated Annealing

# CHAPTER ONE

## INTRODUCTION

The advancement of deoxyribonucleic acid (DNA) microarray technology has enabled biology researchers to gain the ability to simultaneously track thousands of gene expressions in an elementary examination useful for classifying or detecting a particular tumor gender. The classification of the DNA microarray data requires data mining and machine learning techniques for the extraction of worthy information by developing a model to analyze the samples into diverse categories. The natural structure of DNA microarray data is high-dimensional with a few records and many columns where they represent a well-known phenomenon called “curse of dimensionality” (Naseri & Hasheminejad, 2019). Many studies on tissue classification at the molecular level have indicated that genes with relevant information might significantly contribute to the enhancement of effective disease detection and classification platform. However, these studies agree that not all the genes include relevant information for the classification stage.

Therefore, to achieve reliable, accurate, and effective performance, important preprocessing data should be implemented in DNA microarray classification ( Yuan et al., 2019; Morovvat & Osareh, 2016; Liao et al., 2014; Mirzaei et al., 2014; Najafi et al., 2014; Bolón-Canedo et al., 2014; Lazar et al., 2012; Lee & Leu, 2011; Leung & Hung, 2010). One of the prevailing techniques in the pre-processing of DNA microarray data is gene selection which defines an informational gene subset from the whole gene dataset that reduces computational cost and enhances classification performance (Manbari et al., 2019; Tabakhi et al., 2014; Li et al., 2013).

The selection of the features (genes) can be classified as supervised, unsupervised, and semi-supervised (Cai et al., 2018). Supervised feature selection (FS) is employed when the class labels are available (Paniri et al., 2020; Rostami et al., 2020; Zhao et al., 2013; Xiang et al., 2012; Yang & Ong, 2011; Jenatton et al., 2011; Kim & Xing, 2008), while unsupervised feature selection is used in the absence of class labels. Many unsupervised feature selection methods have been proposed by various scholars (Parsa et al., 2019; Yan et al., 2019; Padungweang et al., 2012; Jiang & Ren, 2011; Yang et al., 2011; Zhao & Liu, 2007; He et al., 2005). Both unlabeled and labeled data are utilised in semi-supervised methods (Zeng et al., 2016; Xu et al., 2010).

Feature selection is important for the exploratory analysis of biological data (e.g., DNA microarray) and it provides an efficient way to discover the unknown meaningful insights into the classification of disease (Shukla et al., 2019). The key disadvantages of FS are that it neglects the potential correlation between different features and, therefore, the subsets generated may be suboptimal for the specific task of the discrimination. Feature selection also depends on some mathematical principles without ensuring that the principles are universally valid for all data (Yuan et al., 2019; Cai et al., 2010).

Feature selection methods are classified into three main groups based on their relationship with the learning model: filter, wrapper, and hybrid approaches (Agrawal et al., 2021; Manita & Korbaa, 2020; Tabakhi et al., 2014; Gheyas & Smith, 2010; Martínez Sotoca & Pla, 2010;). The filter method requires no learning algorithms for evaluation, and it is based on statistical feature information. Therefore, it is fast in computation and very efficient (Manbari et al., 2019a; Liao et al., 2014). Wrapper

methods utilise a learning algorithm in selecting feature subset and, generally, the learning efficiency is increased at the cost of higher computational complexity (Wei et al., 2017; Wan et al., 2016; Huang & Huang, 2009). The hybrid methods make use of the benefits from wrapper and filter methods to produce a balance combining the efficiency of learning and the execution time (Das et al., 2017). Methods based on filter-based solutions are worth indicating by utilizing the evaluation criteria with the class label, leading to enhanced correlation among features and reduced similarity among features (Sahu et al., 2018).

One of the important evaluation criteria in the correlation between features is relevance and redundancy analysis. The classical evaluation criterion that is based on the analysis of relevance and redundancy is Max-Relevance and Min-Redundancy (Rostami et al., 2020; Che et al., 2017; Tabakhi & Moradi, 2015; Herman et al., 2013; Peng et al., 2005). In the processing of selection features, several mechanisms are able to reduce both relevance and redundancy. Redundancy occurs when one feature provides information similar to another feature in the dataset. Therefore, choosing the redundant feature to represent the final subset leads to a negative effect on model accuracy. On the other hand, irrelevant features play a meaningless role and do not supply information to the clustering or classification of a set of data instances. In high-dimensional data, the correlation between the increasing numbers of features becomes more complicated. Consequently, those redundancy and irrelevant features should be analyzed more accurately. Several FS methods, such as max-relevance and min-redundancy, random subspace (Lai et al., 2006), fast correlation-based filter (Yu & Liu, 2003), and relevance-redundancy feature selection (A. J. Ferreira & Figueiredo, 2012), partly consider this issue, which leads to unsatisfactory results for the

correlation between features and the elimination of redundant features (Manbari et al., 2019b). Feature redundancy is one of the major challenges of these methods even though such methods take into account the correlation among features. These methods often have high computational complexity that is not effective for FS of high-dimensionality data.

Investigations on FS methods have been focused on the diversification between features (Sahoo & Chandra, 2017; Zhang et al., 2013). Such a method can express max relevance and max diversity as two optimisation problems (Liu et al., 2011; Ienco & Meo, 2008; Dhillon et al., 2003). The basic process for these methods, shown in Figure 1.1, consists of three stages. In the first stage, the correct distance measure is picked to form the feature space, while the second stage focuses on clustering of features. In the third stage, representative features from each cluster will be picked to form the desired subset of features. Additionally, the efficiency of the FS method increases with the utilisation of cluster information (Song et al., 2021; Rostami et al., 2021; García-Torres et al., 2016; Moradi & Rostami, 2015b; Song et al., 2013; Chuang, Tsai, et al., 2011; Jang et al., 2011). However, the selection of features from each cluster presents some challenges, as with many of these works, because the clustering approach is overly sensitive in the FS algorithm models (Manbari et al., 2019b).

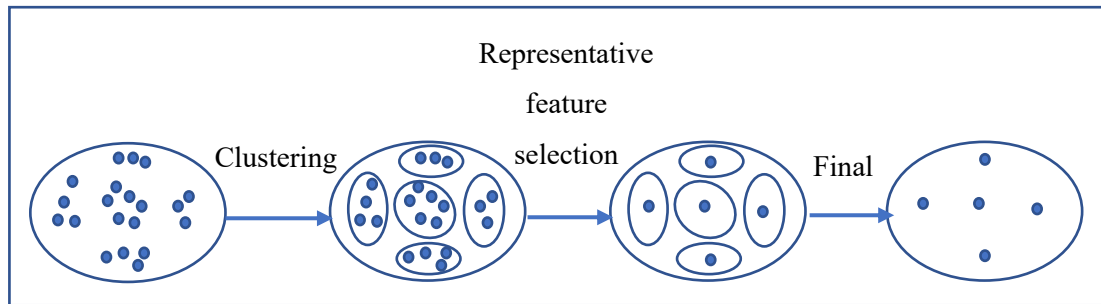


Figure 1.1: Clustering-based FS process (Cai et al., 2018)

One of the most effective FS algorithms to select subsets of candidate features is based on a search technique. The search technique breaks down FS methods into heuristic search, complete, and randomized. Complete search algorithms involve searching the total search space for the best feature subset. The complete search aims to select the best subset of features in a dataset with high dimensionality, which is practically impossible within a plausible time (Manbari et al., 2019b). A randomized search strategy investigates an entire search space through a finite space and the subspace scale is based on stopping criteria, such as the maximum number of iterations and the subset size. The algorithm of a randomized search still appears to get stuck in a local optimum, even when using the parameter setting to make a tradeoff between the optimality of the result and convergence speed. Such algorithms have lower computational complexity than complete search algorithms (Manbari et al., 2019b). In heuristic search-based FS algorithms, one feature is removed or added in each iteration from the selected feature set. Moreover, the computational complexity of the heuristic search-based algorithm is much less than that of complete search algorithms (Huang & Huang, 2009). Most algorithms have been constructed on the basis of heuristic search.

In recent years, more attention has been paid, in particular, to approaches based on Swarm Intelligence (SI) as a search technique for FS. Such approaches comprise Genetic Algorithm (GA) (Oh et al., 2004), Ant Colony Optimisation (ACO) (Wan et al., 2016), Forest Optimisation Algorithm (Ghaemi & Feizi-Derakhshi, 2016), Artificial Bee Colony (ABC) (Xu et al., 2021), Whale Optimisation Algorithm (WOA) (Mafarja & Mirjalili, 2018), and Particle Swarm Optimisation (PSO) (Chuang, Tsai, et al., 2011; Chuang, Yang, et al., 2011). Particularly in high-dimensionality data, metaheuristic swarm search methods are very useful with regard to their global search capability (Amoozegar & Minaei-Bidgoli, 2018). ACO has been widely used as a metaheuristic swarm search method for FS (Shukla et al., 2019). In comparison to other Swarm intelligence (SI) methods, ACO has several advantages including the capability of local and global search, long-term distributed memory, and learning reinforcement strategy (Moradi & Rostami, 2015b). In dealing with FS, ACO is appropriate for handling high dimensional, noise, irrelevant, and redundant datasets (Fahrudin et al., 2016).

Many real problems can be modeled into a graph form such as the Travel Salesman Problem (Cornuéjols et al., 1985), Graph Colouring Problem (Jensen & Toft, 1994), and FS for microarray (Cornuéjols et al., 1985; Moradi & Rostami, 2015b, 2015a; Swarnkar & Mitra, 2015; Yi et al., 2016). In FS for microarray, the graph-based step creates multiple views, each containing a specific number of automatically obtained genes. This is natural in genomic data where gene groups are important in deciding alternate definitions of the microarray data with regard to diversity, thereby facilitating

the collection of gene subsets which are informative genes with regard to different views.

Feature clustering is one of the FS methods that considers the correlation among features. Therefore, the feature clustering model can reduce the dimensionality in high-dimensionality data as a powerful tool in increasing the effectiveness search for the optimal feature subset. This is achieved by clustering the most associated features in the same groups. Furthermore, the performance of the FS process will be improved when it uses the information obtained from the clusters.

One of most popular methods that cluster the features is the Modified Graph Clustering-Based ACO (MGCACO), which is an extension from an algorithm called Graph Clustering-based ACO (GCACO) (Ghimatgar et al., 2018; Moradi & Rostami, 2015b).

The GCACO results have achieved the highest performance accuracy for microarray datasets compared to well-known filter-based methods. This is due to the fact that the selection process is driven in such a manner that at least one feature is picked from each cluster, along each ant search process, and comparatively less correlated features are injected in a large percentage with regard to more correlated features to the consecutive iteration.

Both the GCACO and MGCACO method are based on the Louvain community detection method (Blondel et al., 2008), which is used to detect the communities of the most correlated features by maximizing a modularity function and finding the local maximum. Such algorithms are easy and simple to proceed in determining the

communities in large networks due to the greedy search strategy that utilises modularity maximization as a goal to find the best community (Zhang et al., 2018), thus MGCACO is shown to outperform other well-known FS methods. Notwithstanding, there are several disadvantages such as the lack of clustering highly correlated features that could reduce the finding of the optimal subset, therefore decreasing the performance of the MGCACO algorithm. Moreover, the selection of the features is oversensitive to the feature clustering method due to the challenges of the predefined parameters and optimal subset determination.

### **1.1 Problem Statement**

The high dimensional data of the DNA microarray is characterized by a few samples in a thousand genes' sizes, which lead to significant problems such as complexity, noise, and irrelevant genes (i.e., features). This effect the effectiveness of the classification algorithm in DNA microarray (Rostami et al., 2020; Dhrif, 2019; Anuncia & Wiil, 2018; Morovvat & Osareh, 2016; Ahmad et al., 2008). High-dimensional data classification problems have been efficiently solved using the MGCACO algorithm. It is an efficient graph clustering FS algorithm based on ACO and has shown to be superior than other FS algorithms (Ghimatgar et al., 2018) due to using both feature clustering and ACO-based FS method. Thus, integrating the feature clustering approach based on the community detection algorithm with ACO-based search process for FS has shown improvement in the performance efficiency of the proposed algorithm (Manbari et al., 2019; Moradi & Rostami, 2015b). However, there are several limitations with the MGCACO that could lower the classification accuracy.

The feature clustering phase in the MGCACO algorithm using Louvain community detection method does not operate exhaustively on all possible solutions (i.e., clustering groups) (Zhe et al., 2019). The Louvain community detection method employs the greedy search strategy to create solutions based on modularity maximization. As a result, the communities are combine to create higher modularity. However, using greedy search strategy will restrict to only local search (Zhe et al., 2019). This is due to the mutual effect problem comes when some features in the search space are assigned into the same community with no possibility to reassigning them to other communities (Zhe et al., 2019; Good et al., 2010; Feo & Resende, 1995). This strategy leads to be suboptimal clustering groups and resulting from a low-quality subset of features in the MGCACO algorithm. Therefore, the Louvain community detection method could be replaced by a metaheuristic algorithm that utilises the ability of various mechanisms such as intensification and diversification for local and global optimisation to enhance the performance of the MGCACO algorithm (Sagban et al., 2017; Yang, 2010).

The MGCACO utilises a fixed threshold selection criteria to control the selection of one or more significant features from pre-established clusters (Ghimatgar et al., 2018; Moradi & Rostami, 2015b). The threshold value is considered very critical and highly data-dependent with various values that may lead to different classification results. Accordingly, user determination may not consider the relevance, dependency, and homogeneity among the features in the search space. Therefore, defining a high threshold value will make the algorithm remain in the current cluster to select the next feature. Consequently, it will ignore the dependence (maximizing redundancy

and minimizing relevance) of the feature on previously selected ones. On the other hand, low threshold value will propel the algorithm without any previous information to pick the next feature from the new cluster equivalent to a random walk. In this manner, the fixed threshold value will reduce the quality of the feature subset selection in MGCACO. Therefore, a more adaptive version should be developed, where the value of the threshold is automatically adapted rather than being predefined.

MGCACO has a predefined feature subset size determination (i.e., parameter) which is responsible to select features with high pheromone value. As a result, the classification performance will be affected by various values of this parameter. Although the MGCACO algorithm has a sensitivity analysis to determine the feature subset size within a certain range, there is no serious attempt in enhancing the setting of the parameter (Ghimatgar et al., 2018; Moradi & Rostami, 2015b). The offline-based analysis (non-optimised parameter setting) of recent works is built on a trial-and-error preliminary run mode. In addition, the subset determination is highly dependent on application domains. Therefore, this method will not guarantee optimal classification performance (Huang et al., 2019; Hamadi et al., 2013). Several existing studies have solved the problem of subset size determination (Ghimatgar et al., 2018; Moradi & Rostami, 2015b; Rasheeduddin & Rao, 2019; Manbari et al., 2019b). However, such algorithms that utilise the automatic approach still require a predefined parameter to proceed with the automatic selection mechanism. Another drawback is the construction of the automatic determining mechanism does not consider any information about the efficiency of features or the dependency of the dataset. Consequently, it will provide either fewer or more features that leads to lower

performance of the algorithm. Thus, by utilizing the metaheuristic population-based optimisation characteristic, the search technique for subset determination can be enhanced.

## 1.2 Research Questions

1. How can the clustering of features be improved in terms of grouping the highly correlated features in the MGCACO algorithm?
2. How to control the threshold value throughout the MGCACO algorithm runs to guide features subset construction for DNA microarray?
3. How to improve the search strategy for determining the final subset of the features in the MGCACO algorithm?
4. Will the proposed algorithm be able to reduce the high dimensionality and enhance the classification accuracy in the DNA microarray?

## 1.3 Research Objectives

The study's main objective is to develop an enhanced graph clustering ACO (EGCACO) algorithm for FS in the DNA microarray. To achieve the main objective, the following specific objectives need to be fulfilled:

1. To design an ACO based feature clustering method that enhances the grouping of highly correlated features in MGCACO.
2. To design an adaptive selection technique based on filter fitness function to control the threshold value in MGCACO algorithm.

3. To develop a genetic-based method that determines the final feature subset in MGCACO.
4. To evaluate the effectiveness of the proposed algorithm for classification.

#### **1.4 Significance of the Research**

The output of this study is an enhanced MGCACO algorithm for FS. This algorithm is a variant of ACO algorithm which can be used to reduce the dimensionality of the DNA microarray in classification task. In addition, the new algorithm has identified relevant and redundant features among source and target distributions and ensures better classification performance when target datasets are tested for models trained on source datasets. The clustering-based metaheuristic method produced highly correlated DNA microarray features that enhance the finding of the optimal subset. The automatic parameter selection technique has selected the most informative features that have a positive effect on the performance of the algorithm. Moreover, the genetic method has produced the appropriate final subset automatically rather than being predetermined by the user.

In particular, the proposed algorithm has selected the genes of the DNA microarray with consideration of relevance and redundancy among the genes, which was critical for good and reliable results. Therefore, it can be used in several domains (i.e., real-world application) such as to diagnose diseases, identify genes that affect illnesses, find genes for therapeutic targets, and select the best treatment.

## 1.5 Scope of the Research

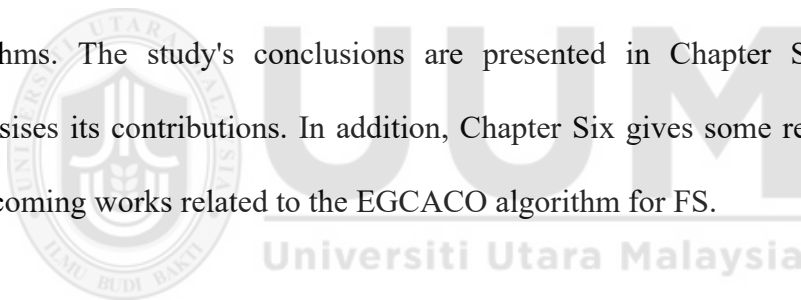
The study focuses on the MGCACO algorithm, which is specifically a variant of the GCACO algorithm for FS. The study proposes three directions to improve classification accuracy. The first is clustering highly correlated features to achieve max-relevance and min-redundancy while the second is the automatic parameter adaptation technique to choose suitable relevant features. The third is determining the optimal final subsets automatically.

The implementation of the algorithm has been applied for FS in the DNA microarray. To demonstrate the efficacy of the proposed method on different benchmark classification problems available in the literature, the experiments use several datasets from the University of California Irvine (UCI) Machine Learning Repository (Bache & Lichman, 2013) and several microarray datasets that have been used in a study by Li et al. (2017).

To highlight the generality of the proposed method, several well-known classical classifiers, available in the literature, including Random Forest, k-Nearest Neighbor algorithm (k-NN), Decision Tree, and Support Vector Machine, are used in testing the predictive classification ability of the selected features (Ghimatgar et al., 2018). Then, the performance of the proposed algorithm is compared with five algorithms (TGA, GWO, ASO, PSO, HHO, GA, DE, BAHSGW, RSGW, ASGW, BGWOPSO, WOA-CM, WOASAT, and bALO-QR, ) that are employed for dimensionality reduction using five evaluation metrics, namely F-measure, recall, precision, the number of selected features and classification accuracy.

## 1.6 Thesis Organisation

There are six chapters in this thesis. The first chapter provides background information on the study and the general problem that this research has addressed. The second chapter contains reviews of various FS tasks and their classification techniques and classes, as well as swarm intelligence algorithms and their variants. The third chapter discusses the research framework, methods, techniques, and experimental procedures utilised in this study. The fourth chapter presents three proposed modifications to overcome the limitations of the GCACO algorithm, as well as their integration with GCACO to form the proposed EGCACO for FS in DNA microarrays. The fifth chapter presents the results of performance evaluation of the proposed algorithm on different datasets and results comparison with other state-of-the-art swarm intelligence algorithms. The study's conclusions are presented in Chapter Six, which also emphasises its contributions. In addition, Chapter Six gives some recommendations for upcoming works related to the EGCACO algorithm for FS.



## **CHAPTER TWO**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

In recent years, the amount of acquisitions for biomedical data has increased drastically. Advancements of molecular genetics techniques, such as DNA microarrays, has authorized the access to a “global” outlook of cells. For instance, it is possible, now, to use DNA microarrays routinely to explore the biological molecular classification of a cell, weighing hundreds and thousands of gene expressions simultaneously. Therefore, the study in FS is deemed important.

This chapter presents the reviews of previous studies in FS-related activities and their application domains. Section 2.2 presents microarray data mining and its main challenges, which include gene selection. Section 2.3 describes FS and approaches in implementing FS. Section 2.4 presents several existing FS-based clustering methods. Section 2.5 discusses previous studies on automating the parameterization, while Section 2.6 explains the final subset determination for FS based on ACO. Section 2.7 summarizes this chapter.

#### **2.2 Microarray Data Mining**

The development of one of the first methodologies for DNA microarray analysis explains that data on expression can be utilised in a variety of class prediction or discovery for biomedical problems like tumor classification (Silwattananusarn et al., 2016). Data mining or statistical and machine learning techniques applied in data on gene expression are used to answer questions of finding molecular markers for disease,

tumor morphology, and predicting post-treatment outcomes. Today, in many instances, the microarray-based classification can be performed successfully in different cell histologies, lineages, and morphologies. In some cases, the findings of DNA microarray research have led to more specific clinical evaluation, such as a portion of clinical studies. This is the use of the results-specific, computationally elected gene markers like Protein Kinase C beta and associated lymphoma treatment inhibitor (Shipp et al., 2002). The testing in predicting the drug response or treatment result is limited. However, some of the findings are quite promising. Most results of microarray analysis using data mining lack further follow-up study and experimental validation. Furthermore, numerous existing efforts are being conducted in this area to address the challenges of microarray data mining.

Representative data mining applications in fields such as web or banking have a small number of sub-fields or attributes (at most several hundred), while the number of records is much larger (thousands and sometimes millions). Conversely, a typical sample of microarray data analysis may have a variety of sub-fields (e.g., number of genes), usually equivalent in thousands, though with a fewer number of records (smaller than a hundred). Due to the difficulty of obtaining microarray specimens in many interesting cases, the record count is likely to stay that way. However, the use of few records relative to many fields has produced a huge probability of “false positives” results, i.e., due to opportunity, both in determining the genes expressed differently and in the development of predictive models. There is a need to use robust methods and their applications in data mining to validate the models and determine their probability (Piatetsky-Shapiro & Tamayo, 2003).

Classification, clustering, and gene selection are the main tasks in the analysis of the biomedical applications (Allahyari et al., 2017; Piatetsky-Shapiro & Tamayo, 2003). Classification is predicting outcomes or classifying diseases that depend on patterns of gene expression and the possibility of determining the best genetic signature treatment (Charlesworth et al., 2019; Franks et al., 2018; Dam et al., 2018; Salem et al., 2017; Khan et al., 2005; Chang et al., 2004; Lapointe et al., 2004; Nutt et al., 2003; Ramaswamy et al., 2003, 2001; Tibshirani et al., 2002; Chinnaiyan & Rubin, 2002; Alizadeh et al., 2000; Bittner et al., 2000; Golub et al., 1999). Clustering is to find new biological classes or refine existing classes (Ramos et al., 2017; Roy et al., 2016; Chang et al., 2016; Pan et al., 2002; Hastie et al., 2001; Li & Hong, 2001; Tamayo et al., 1999; Eisen et al., 1998). In contrast, gene selection in microarray data mining is an attribute selection process that determines the genes most closely related to a specific class (Shukla et al., 2019; Sahu et al., 2018; Fahrudin et al., 2016; Tabakhi et al., 2015; Mandal & Mukhopadhyay, 2013; Lee & Leu, 2011; Leung & Hung, 2010; Storey & Tibshirani, 2003; Marchal et al., 2002; Dudoit et al., 2002; Baldi & Long, 2001; Ideker et al., 2001; Li & Hung, 2001; Tusher et al., 2001).

### **2.2.1 Classification in Biomedical**

The microarray dataset has fewer records than features (Piatetsky-Shapiro & Tamayo, 2003). Therefore, the common statistical and machine learning techniques, including global FS, may result in incorrect discoveries as a result of random chance. Simon (2003) addresses some of the familiar errors that can be solved in developing accurate classifiers, finding informative features, and demonstrating the appropriate approach for the classification of microarray data. O'Connell (2003) provides an overview of

the approaches used in Insightful S+ArrayAnalyzer, addressing the complete spectrum of microarray data analyses, including classification, clustering, differential expression analysis, gene selection, quality control, experimental design, and data pre-processing.

Bair and Tibshirani (2003) have introduced a “nearest shrunken centroid” approach that has been successfully used to detect clinically relevant divergences in cancer patients. The approach is applied in the privileged access management system, obtainable for researchers, and is successful in managing microarray-inherent noise. The study presents a powerful tool for diagnosing and treating cancer.

Dudoit et al. (2003), introducing a single loss-based method for selection, quality assurance estimation, and cross-validation, identified new theoretical findings that showed cross-validation choice could be utilised in intense searches of huge parameter spaces, including infinite sample situations. In addition, they introduced a new classification deletion/substitution/addition algorithm that allows the deletion, substitution, and addition of over-index set elements to determine the appropriate gene sets that minimize the estimate of errors. A number of problems in their study have traditionally been treated separately including multivariate outcome prediction and density estimation based on either censored or uncensored data.

Other issues in applying molecular classification models arise from the major technical challenge of addressing heterogeneity via the use of heterogeneous material sources, alternative techniques, and platforms. Various datasets reflecting the same biological system will be expected to show a certain amount of invariant biological characteristics

regardless of the idiosyncrasies or particulars of the sample sources, the preparation methods, and the technical platforms utilised to collect the data. Such invariant biological characteristics can provide the basis for the creation of more stable, general, and accurate classification models when properly captured and revealed. Hopefully, such models will give more focus on the reproducible biological activity as well as less focus on idiosyncrasy, technical data, and ideology. Fung and Ng (2003) address this problem by classifying heterogeneous variables based on impact factors. The impact factors provide a way to measure the differences in test and train samples between individual classes and can be integrated into generic classifiers such as k-NN or weighted voting, resulting in a major improvement in the accuracy for classification of heterogeneous samples.

### **2.2.2 Biomedical Clustering**

Clustering is an important task to classify the co-expressed gene groups that identify consistent expression patterns. However, the domain knowledge strongly relies on the understanding of co-expressed genes and consistent patterns, making it difficult to completely automate. Jiang et al. (2003) present an approach focusing on an interactive exploration of patterns of gene expression to solve this issue. The authors developed a coherent pattern index graph tool that provides users with visual feedback on the presence of coherent patterns and their strength. Their results showed that the approach is effective in mining real gene expression data and is scalable in mining large datasets.

Finding gene interactions and gene networks is another aspect of clustering. Wu et al. (2003) propose a graphical model-based interaction analysis using a Gaussian

graphical model to identify associations (i.e., the interactions) between the paired genes and a log-linear model to discover interactions between multiple genes. The study had many unresolved issues, such as sparse data, that need to be further explored.

Usually, scientific results can be enhanced by looking at the same events of gene expression from different perspectives. In the analysis of the gene expression study, there are several sources (i.e. gene regulation, tissue diversity, and protein composition) of biological information that could be used. Glenisson et al. (2003) explore how literature-extracted information and expression data can be integrated to find biologically relevant clusters not only identified from DNA microarray data. This is an example of integrative genomics as stated by He et al. (2005) and Zhang et al. (2005).

### **2.2.3 Gene Selection Mining in Microarray Data**

Gene selection aims to identify differential molecular activity or invariants relevant to a particular biological problem. This issue is constrained by the fact that, in many particular cases, little is known about the natural biological variability predictability of a biological state or specific tissue. A biological condition is usually presented along very coarse-grained phenotypic lines to complicate matters. Nadimpally and Zaki (2003) discuss this question of normal variation directly and also examine genes that exhibit normal variability among genetically identical mice to establish a collection that can be utilised in differential gene expression experiments. They aimed to identify false positives as well as gain insight into the underlying natural variation processes. They used six mouse tissues, which produced several genes displaying major

biological variations between similar mice and provided a helpful compendium of natural gene expression variations for mouse models.

Mukherjee et al. (2003) introduced a gene-ranked algorithm utilizing bootstrapped P-values, which takes another method to evaluate variance in small samples. This method is particularly beneficial if a small sample variability is taken into account in the observed test statistical values. The proposed algorithm demonstrated that this method exceeds a commonly utilised two-sample T-test on the artificial data. The authors applied their method on two real datasets. Experimental results showed that the algorithm was able to exploit the limited data available to infer biologically useful information.

Most gene selection approaches test each gene in isolation and disregard gene associations. However, from a biological point of view, the gene groups acting together as pathway concepts and reflecting cell states are specific atomic units or characteristics that would be able to determine the character or form of a given sample and its identical biological status. Such coherent gene expression information should be the information of the input on which the methods of sophisticated computational can work (Padilha & Carvalho, 2019; Xue et al., 2018; Buonvicino et al., 2018; Vengatesan et al., 2017; Brunet et al., 2004; Cunliffe et al., 2003; Mootha et al., 2003; Segal et al., 2002; Murali & Kasif, 2002; Cheng & Church, 2000; Alter et al., 2000; Califano et al., 2000). In this respect, Hanczar et al. (2003) suggest improving the precision of the DNA microarray classification by choosing suitable model genes representing a group of genes sharing a profile and describing the phenotypic class of

interest. They presented important findings on the advantages of adenocarcinoma classification utilizing prototype-based FS.

### **2.3 Feature Selection**

Feature selection and classification are the two major tasks in analyzing DNA microarray gene expression. The FS method is critical to perform with an appropriate level of accuracy prior to the classification activity. Gene expression data for DNA microarrays includes hundreds of thousands of features (genes). However, real-world datasets do not always have relevant information to solve problems with machine learning (Dhrif, 2019). Therefore, only a few feature subsets have effective correlation between each other (Hall, 1999). The FS problem is complicated because the search space is sparsely populated and the learning model is unable to determine between relevant data and noise especially if the number of instances is less than the number of features. A common rule of thumb is that the representation should include at least five training instances for each dimension (Chinnaswamy & Srinivasan, 2016). In the dataset, FS is a process that selects different expressed features and creates a new subset of features for classification (Gene et al., 2011). Moreover, FS can help maximize the relevancy of features while reducing redundancy. Additionally, simpler models have a lower risk of overfitting, i.e., when predicting new data points, they are more robust. As depicted in Figure 2.1, FS consists of a three-step process: i) subset generation, ii) subset evaluation, and iii) stopping criteria (Liu & Yu, 2005).

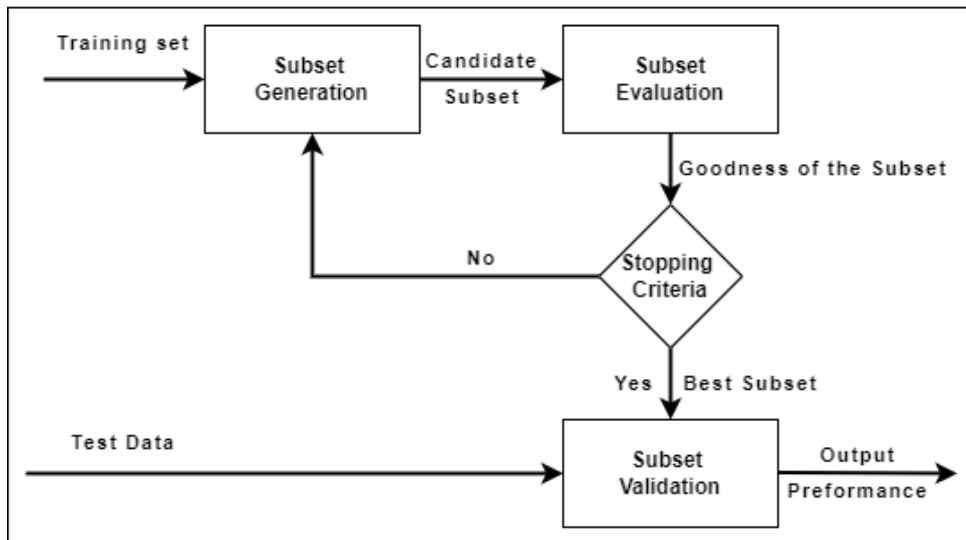


Figure 2.1: Feature selection's three-step process (Liu & Yu, 2005)

There are at least four challenges when designing any FS algorithm (Dhrif, 2019). The challenges comprise curse of dimensionality, small sample size, the functional interaction among features, and quality of data. With such a large number of features attempting to train a classifier (i.e. training samples based on the extent of each feature), it would have high computational costs and poor accuracy as all features must be present for future prediction as demonstrated by Löw et al. (2013), the accuracy depends on the amount of features utilised and FS was beneficial than using all features in terms of prediction accuracy. The relationship of instances and features in a small sample size causes FS to be a difficult problem (NP-hard) of non-deterministic polynomial time. The interaction among features will also introduce complex interdependency among them. There will also be some data-related issues such as

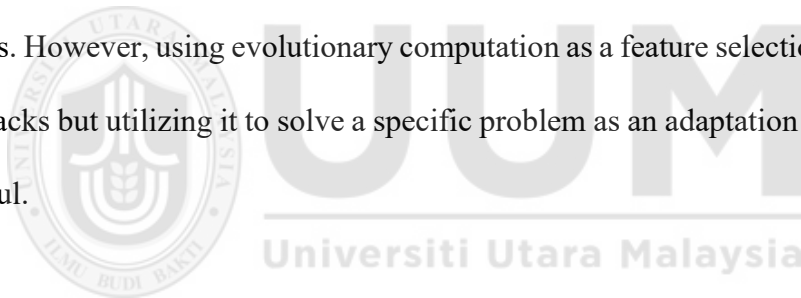
sparse, noisy, and imbalanced datasets, which will affect the performance of an FS algorithm.

### **2.3.1 Subset Generation**

The generation of subsets is a heuristic search in which each state defines a candidate subset for evaluation. The efficiency (e.g., the accuracy of classification) depends heavily on the search algorithm used. Search techniques can be divided into stochastic, heuristic, and exhaustive (Dhrif, 2019). Although the long computational time of exhaustive searching guarantees the best solutions, it is impractical for most applications in the real world. Sequential searches such as greedy stepwise backward selection (Caruana & Freitag, 1994) and linear forward selection (Pahikkala et al., 2010; Khan et al., 2007) are typical examples of heuristic search methods. These two methods are inspired, respectively, by variants of forwarding selection and backward sequential selection. Linear forward selection improves the efficiency and effectiveness of forwarding selection by the features' number reduction being taken into account at each stage. Although backward selection can find interactions of features better than forwarding selection, it cannot be implemented on large feature sets. Both forward and backward approaches tend to face local optima problems (Tang et al., 2013). A stochastic search will perfectly and randomly generate subsets using the Las Vegas algorithm (Ang et al., 2016) that is quite slow to converge in high-dimension search spaces.

Evolutionary computation is a family of stochastic methods applying evolutionary principles or SI to produce effective subsets of existing ones (Ashlock, 2006). Such

methods include Genetic Programming (Banzhaf et al., 1998), GA (Banzhaf et al., 1998), ACO (Dorigo & Caro, 1999), PSO (Kennedy & Eberhart, 1995), and Artificial Bee Colony (Karaboga, 2005). Evolutionary computation techniques do not make prior search space assumptions and do not require domain knowledge, whether it is linearly separable or differentiable. Another benefit of evolutionary computation is the population's ability to look for multiple solutions in one round, making it appropriate for finding non-dominated solutions. Nonetheless, evolutionary computation techniques have a major drawback as they have high computational costs caused by the huge number of evaluations for fitness functions. Another issue with evolutionary computation techniques is their lack of stability as the algorithms select different subsets of features at each run, which may require field experts in a further selection process. However, using evolutionary computation as a feature selection algorithm has drawbacks but utilizing it to solve a specific problem as an adaptation or hybridization is useful.



### **2.3.2 Subset Evaluation**

Subset evaluation is a process where evaluation criteria are applied to define the goodness of each generated subset. This will result in many evaluations that drive to high computational costs, particularly when there is a large number of instances. To overcome these shortcomings, methods such as evaluating the minimum redundancy, the total relevance based on mutual information (Unler et al., 2011; Zhang et al., 2005), or evaluating the distance between the whole search space and the selected subspace (Wang et al., 2015) have been proposed. To handle FS tasks for datasets with millions of features, sparse logistic regression (Tan et al., 2013) was introduced, where low

weights of irrelevant features have the effect of filtering them out. Typically, such statistical methods are generally very accurate, but they often make assumptions about the distribution of the data in terms of probability.

Among FS approaches, subset evaluation differs where there are two main approaches: wrapper and filter. Wrapper methods require a classifier or more than one classifier with a performance metric to determine the optimal subset as one that minimizes cardinality and maximizes efficiency. The performance metrics often used for wrapper approaches are best arithmetic means of true positive rate (TPR) and true negative rate (TNR), best geometric mean of TPR and TNR, area under the precision-recall curve, area under the receiver operating characteristic curve, and accuracy (number of correctly defined instances/total number of instances).

In contrast, filter methods, being autonomous of the two criteria of cardinality and performance, define the optimal subset of features as the one consisting of all the highly relevant features and weakly relevant but not redundant. Several criteria have been proposed to evaluate feature subsets or single features, such as consistency measures (Liu & Motoda, 2012), dependency measures (Hall, 2000), probability of error measure (Devijver & Kittler, 1982), information or uncertainty measures (Ben-Bassat, 1982), distance measure (Almuallim & Dietterich, 1994), fractal dimension (Mo & Huang, 2012), inference correlation (Mo & Huang, 2011), and classification error (Estévez & Caballero, 1998). The most widely used performance metrics are classification accuracy and number of obtained features.

### 2.3.3 Stopping Criteria

A stop criterion must be established to stop the selection process. Although many FS approaches do not have a stop criterion, several studies suggest some general stop criteria (Shukla et al., 2019) as follows:

- Deletion or addition to the subset of features does not make a significant difference.
- The minimum number of features or minimum rate of errors in the classification or the maximum number of iterations.
- The search completes by obtaining the most excellent subset according to an appropriate evaluation method.

Although it is not part of the selection process, the FS approach must be checked by performing various tests and comparisons with previously established findings or results of competing methods. In conducting the comparison, focus should be on using artificial datasets, real-world datasets, or both. The optimum size of the feature subset must be determined automatically (Dhrif, 2019).

Traditionally, the FS process not only minimizes data dimensionality, which facilitates inspiration and comprehension, it also produces more solid models with greater capabilities for generalization (Devijver & Kittler, 1982). All of these characteristics make the selection of features a fascinating area of study where various FS methods have been implemented over past decades.

Feature selection methods can be categorized into three types according to the class of the datasets; as supervised (Tang et al., 2014a, 2014b; Kotsiantis, 2011), unsupervised (Alelyani et al., 2013), and semi-supervised (Sheikhpour et al., 2017). The first needs a collection of labeled data (supervised dataset). This label may be a grade as a real value or ordered value (attach with a specific task) to select the recognizable and relevant features. The semi-supervised method requires only the labeling of some objects. On the other hand, the unsupervised method deals with the non-labeled data (Salem et al., 2013; Dy & Brodley, 2004; Fowlkes et al., 1988).

Several FS methods have been suggested over past decades. A majority of these methods focus on the supervised approach in the classification task. However, due to the technological developments that have taken place in recent years and multiple applications, many unlabeled data have been generated, for example, social media (Zafarani et al., 2014), bioinformatics (Saeys et al., 2007), image retrieval (Yasmin et al., 2014), text mining (Bharti & Singh, 2014), and intrusion detection (Ahmed et al., 2016; Agrawal & Agrawal, 2015; Ambusaidi et al., 2015).

On the other hand, relevance and diversity are required in FS high-dimensionality data (Zhang et al., 2013; Novovicová et al., 1996). This includes classifying features by a clustering method after choosing the appropriate measurement for correlation between features. The feature clustering-based method can be used to take the benefit of its relatively lower complexity on the FS high-dimensionality data. Furthermore, the efficiency of the FS method improves with the usage of cluster information.

## 2.4 Feature Selection-based Clustering Method

Feature selection methods are divided into three categories based on their interaction with the learning model: filter, wrapper, and hybrid (Tabakhi et al., 2014; Sotoca & Pla, 2010; Gheyas & Smith, 2010; Saeys, Inza, & Larrañaga, 2007; Liu & Yu, 2005). There are many techniques that have been used together with the feature selection methods for solving clustering problems. One of those popular technique is the ACO algorithm (Rostami et al., 2021; Song et al., 2021; Guha et al., 2020; Manbari et al., 2019a).

### 2.4.1 Filter Method

The selection of relevant features using the filter method is based on an evaluation criterion to measure how well features (subsets) differentiate between the two groups (classes) without using any learning algorithm. Their speed and scalability are the main characteristics of filter methods.

Xu et al. (2008) suggest an FS for genes using a filter-based evaluation method by using diffusion maps to tackle the multi-dimensional issue and utilizing Markov matrices as a coordinate framework on the initial dataset to gain an effective representation of geometric descriptions data. The optimal subset of features is then grouped with a fuzzy ART neural network that learns the arbitrary input patterns in a fast, settled, and self-organizing manner to shape cancer samples partitions.

Rostami et al. (2020) propose a PSO-based multi-objective method for FS which is divided into three major phases. The initial features are displayed as a graph

representation model in the first phase. For all nodes in the graph, the feature centralities are determined in the second phase. Finally, in the third phase, an improved PSO-based search method is utilised for the final FS. They applied their proposed method to five medical datasets. Experimental results showed that it outperforms previous related methods. However, the proposed method suffers from determining the suitable values for the parameters that have impacted the behaviours of the learning model.

Tabakhi et al. (2014) introduced one of the first bio-inspired swarm intelligence paradigms known as Unsupervised Feature Selection-Based ACO (UFSACO). The major objective of UFSACO is to choose feature subsets with minimum similarity (low redundancy) between features. In this algorithm, the nodes identify the features and the similarities between the features are the weights on the edges in a complete undirected graph. Features are redundant if there are two similar features. Each feature has a desirable amount of pheromone in the graph. The amount of pheromone is changed by agents (ants) depending on its current amount, the frequency, and a pre-specified rate of evaporation. Until a predetermined stopping criterion is reached (iteration number), the ants interactively traverse the graph preferring low similarities and high pheromone values. Finally, features with the largest pheromone value and low redundancy are chosen. The UFSACO algorithm was evaluated in terms of the number of selected features and classification accuracy to achieve the final feature subset by using several well-known classical classifiers i.e., Random Forest, Naïve Bayes, Decision Tree, and Support Vector Machine. The experimental findings on

many widely used datasets show the efficiency and efficacy of the UFSACO over other well-known algorithms.

Other methods that have the same concept as UFSACO are MGSACO (Tabakhi et al., 2015), relevance-redundancy feature selection-based ACO (RRFSACO) (Tabakhi & Moradi, 2015), and unsupervised probabilistic FS (UPFS) using ACO (Dadaneh et al., 2016) and are proposed for the optimal feature subset through an iterative process. For both RRFSACO and MGSACO, in addition to quantifying the redundancy feature as in UFSACO, each feature's relevance is further measured by its variance. Therefore, the aim of these algorithms is to choose attributes that reduce redundancy while maximizing relevance. The goal is to pick non-redundant attributes in UPFS by utilizing the similarity of cosine less than Pearson's correlation.

There are several limitations to the current ACO-based FS methods (Dadaneh et al., 2016; Tabakhi et al., 2015, 2014). In the search process, the possible dependence among the features is ignored. These ACO methods presume that the attributes are independent conditionally. Therefore, once the ant chooses the following attribute, the feature dependency on those already picked is overlooked in the computation. Consequently, the organized subset might include redundant attributes. To avoid this challenge, a novel FS method known as GCACO was proposed by Moradi and Rostami (2015) for classification problems. This method works on the attributes identified as a graph with features (nodes) and feature-similarities (edges) where a community detection method is applied to cluster features (nodes) (Blondel et al., 2008) and selects a minimum redundancy attribute subset between the attributes

collected from the clusters. In the GCACO algorithm, the community detection method is used to cluster and separate identically strong and correlated attributes together in the same group. The method detects communities of a network in two stages.

Every node in the first stage is assigned to a community selection based on the greedy operation to maximize the specific modularity of the network. The second stage simply creates a new network by merging those of the communities previously found. The process then iterates until a significant improvement of the modularity of the network is achieved and the features are then represented in a complete graph in each cluster. Then, ACO-based FS is applied to create the feature subset from the clusters. In every iteration, an agent randomly travels to a cluster and chooses an attribute from the cluster. The efficiency of every solution (i.e., a set of features) is estimated by implementing a separability index once the whole ants traverse completely on the graph. The value of pheromone of every node is adjusted by utilizing a particular ACO-based updating role. Finally, the feature with the highest pheromone will be a candidate in the final subset. The result of the GCACO outperforms other well-known algorithms in terms of classification accuracy. This is because every agent in the graph cluster will identify features with minimum similarity and maximum target class dependence.

Better performance than that of GCACO was obtained in its new version, called MGCACO, proposed by Ghimatgar et al. (2018). Three modifications were made in GCACO to produce MGCACO. First, the pheromone initialisation is improved by

performing the relevance of attributes to classes where higher priority is given to more relevant features. Secondly, depending on the multiple discriminant analysis, MGCACO uses an evaluation function that can effectively evaluate redundancy and the relevance of attributes. Finally, MGCACO utilises a better efficient cost mission, integrating redundancy and the relevance of attributes in a highly efficient manner, where features are sort based on redundancy and relevance analyses.

Ghimatgar et al. (2018) conducted several experiments to evaluate the MGCACO algorithm in terms of the number of selected features and classification accuracy to achieve the final feature subset. To demonstrate the generality of the MGCACO algorithm, the prediction capability of the classification for the selected features has been checked using various well-known classical classifiers i.e., Random Forest, Naïve Bayes, Decision Tree, and Support Vector Machine. Besides, many datasets with various properties were utilised to demonstrate the efficacy of the MGCACO method. These datasets are known as Arcene, Arrhythmia, Hepatitis, Ionosphere, Madelon, Sonar, Spambase, Wine, and Breast Cancer Wisconsin (Diagnostic) (WDBC.) These datasets represent different types such as physical, life, computer, artificial, and microarray which are often used in many machine learning studies including FS (Manbari et al., 2019b). The size of the datasets varies from small, to medium, and large. The method can address irrelevant as well as redundant features. The results of the MGCACO outperform other well-known algorithms in terms of classification accuracy. However, its performance depends on predefined parameters.

A novel hybrid FS method that depends on the Louvain community detection for modified binary ant system and clustering combination, known as FSCBAS, is presented by Manbari et al. (2019). This method deals with data processing issues in large dimension search spaces. This method offers global search as well as local search capabilities within and between clusters. This method, which is inspired by simulated annealing and GA, introduces a novel redundancy rule for reduction to evaluate the correlation among chosen attributes and a damped mutation technique that prevents falling into a local optimum. However, the performance of this method has low accuracy in terms of the number of chosen attributes from each dataset.

#### **2.4.2 Wrapper Method**

The fitness of a wrapper method is determined by the accuracy resulting from the feature subsets when a complete algorithm for classification or clustering is performed. However, the main limitations of wrapper methods stem from the use of a current learning algorithm in tandem and the very high computational cost.

Other work that uses PSO to perform an FS algorithm with a wrapper framework is by Chuang et al. (2008) and Shen et al. (2009), and a hybrid tabu search framework by Chuang et al. (2009). Chuang et al. (2008) improve binary PSO and k-NN in order to prevent being stuck in local optima and to look for superior classification outcomes in a search space with a lower gene number. By applying the modified PSO, Shen et al. (2009) recognize and eliminate the redundant genes and samples at the same time.

Babatunde et al. (2014) introduce an improvement for the performance of concerned classifiers by utilizing binary GA for dimensionality reduction. The method uses k-NN-based classification error as a fitness function to obtain a combinatorial set of features from images. The obtained results are better than other FS methods in terms of classification accuracy. Rostami et al. (2021) propose a GA based on community detection which consists of three stages. In the first stage, the feature similarities are calculated by the fisher score method. In the second stage, the features are grouped into clusters throughout the community detection algorithm. In the third stage, a GA algorithm selects features for a new community-based repair operation. In terms of the performance of the presented approach, six benchmark classification problems were analyzed by the authors. The accuracy of the proposed method outperforms the other FS algorithms. However, the proposed method suffers from determining appropriate values for the parameters that have influenced the learning model's behaviours.

Two different approaches of the Ant Lion Optimiser (ALO) are proposed by Emary et al. (2016a) for FS purposes in wrapper-mode. In the first approach, the native ALO is applied while its continuous steps contain thresholds using a suitable threshold function after squashing them. In the second approach, only the inspiration of ALO operators is used to create the corresponding binary operators. Three common optimisation algorithms were used by the authors to compare with the proposed ALO binary algorithm over 21 datasets from the UCI repository. The proposed algorithm achieve good results compared to common optimisation algorithms.

Taradeh et al. (2019) propose Gravitational Search Algorithm (GSA) to deal with FS tasks. Both k-NN and Decision Tree classifiers are utilised as evaluators in the proposed wrapper FS method. To evaluate the performance of the proposed approach, 18 UCI datasets are used. The results are compared to some popular algorithms to validate the efficacy of the proposed algorithm. The results of the proposed method show superiority in solving FS problems. Guha et al. (2020) introduce a method called clustering-based population in binary GSA to increase the inclusion of more promising features. The method utilises a clustering technique to distribute the initial population across the entire feature space. Twenty well-known UCI datasets are utilised to evaluate the performance of the proposed method and the findings are compared with some recent FS methods. The findings outperform other methods in terms of average classification accuracy. However, the drawback of the algorithm is the use of accuracies as the similarity measurement between the population, which usually requires classification using a learning algorithm, requires additional time.

Two new wrapper FS methods, introduced by Faris et al. (2018), use the Salp Swarm Algorithm (SSA) as the search strategy. In the first method, in addition to the transfer functions, the crossover operator is utilised to replace the average operator and improve the algorithm's exploratory behaviour. In the second method, to convert the continuous version of SSA to binary, eight transfer functions are utilised. The results, comparing five FS methods using common UCI datasets, show that the method is better than other methods.

Mafarja and Mirjalili (2018) propose two (2) binary variants of the wrapper FS method, namely the Whale Optimisation Algorithm with Crossover and Mutation (WOA-CM). In the first, the crossover and mutation operators are employed to improve the WOA algorithm's exploitation. The second method aims to investigate the impact of using the Roulette Wheel selection and Tournament mechanisms rather than a random operator in the search process. The two methods are compared to three algorithms and experimented on standard benchmark datasets, showing the efficiency in obtaining the best feature subsets. Mafarja and Mirjalili (2017) present two hybridization models for designing different WOA-based FS methods with simulated annealing called (WOASAT). The simulated annealing method is integrated into the WOA algorithm in the first model, while in the second model at each iteration of the WOA algorithm, simulated annealing is used to improve exploitation by locating the most promising regions (i.e., the best solution discovered). The performance of the methods is compared with three existing wrapper-based FS methods and is evaluated on 18 well-known UCI benchmark datasets. The experimental results prove the efficiency compared to other methods.

To solve FS problems, a binary version of the hybrid grey wolf optimisation (GWO) and PSO, namely BGWOPSO, is proposed by Al-tashi et al. (2019). The wrapper-based method k-NN classifier with Euclidean separation matrix is used. A total of 18 well-known benchmark datasets from the UCI repository are used to evaluate the performance of the proposed binary algorithm. The results show that BGWOPSO performs better than other algorithms in terms of selecting the most effective features and accuracy.

Mafarja et al. (2020) introduce a wrapper-based FS method with three enhanced hybrid metaheuristic algorithms utilizing GWO and WOA called hybrid serial (HSGW), the structure of adaptive switcher (ASGW), and the structure of random switcher (RSGW). They are compared with a number of well-known FS methods and validated on 18 UCI datasets. The results outperform other state-of-the-art methods.

Mafarja et al. (2019) also propose two mechanisms to design a Grasshopper Optimisation Algorithm (GOA) within a wrapper-based framework, to determine the best feature subset for classification purposes. The first mechanism is based on V-shaped and transfer functions. A strategy that combines the best solutions produced so far is used in the second mechanism. Furthermore, a mutation operator is utilised in the GOA method to improve the exploration phase. The proposed method is compared with eight well-known wrapper-based methods using different standard UCI datasets. The comparative results are superior to other, similar, methods.

One of the wrapper-based FS methods is Harris Hawks Optimiser (HHO), as proposed by Thaher et al. (2020), to deal with high dimensional real-world datasets. Evaluation of the proposed method utilises nine high-dimensional low sample cases and compares the efficacy with six well-known methods. The results show a promising method through obtaining good results compared to the other methods.

Too et al. (2018) introduce two FS methods for Electromyography signal classification based on a Tree Growth Algorithm (TGA). In the first method, in order to improve

exploration and exploitation behaviours, the crossover, mutation, and swap operators are incorporated in a modified binary TGA. In the second method, to transform the continuous TGA to a binary form, two transfer functions are performed. This method shows superiority in terms of both classification performance and feature reduction when it is implemented on a large available feature set.

A binary variant-based wrapper FS method with Atom Search Optimisation (ASO) is introduced by Too et al. (2018). To convert the continuous ASO into the binary form, the presented method employs eight transfer functions from the S-shaped and V-shaped families. The proposed ASO was tested on 22 UCI benchmark datasets and compared to five FS methods for evaluating its performance. The findings of the ASO are preferable in terms of high prediction accuracy and the smallest number of chosen features.

### **2.4.3 Hybrid Method**

The hybrid methods find a good quality balance among filter and wrapper methods to overcome computational cost and efficiency from the goodness of the associated objective task.

Boutsidis et al. (2008) introduce a hybrid framework by integrating the column subset selection problem and principal components analysis as an FS for genes. Kim and Gao (2006) obtain the original physical meaning of the gene subsets based on their reproductive capacity sample projections on main components by implementing the Least-Square-Estimation evaluation. To improve the quality of the partitions, they also

implement boost-expectation-maximization clustering. Therefore, FS utilises unlabeled data to pick the subset of the features.

Guha et al. (2020) propose a hybrid FS method, namely Embedded Chaotic Whale Survival Algorithm (ECWSA), where a wrapper method is utilised to obtain high classification accuracy, followed by a filter method, utilised to further trim the selected subset at a low computational cost. In the ECWSA, chaos has been incorporated to lead the picking of the style of movement followed by whales when seeking prey. A real-life scenario inspiration, which if whales are incapable of hunting the prey they will die, is introduced into the system as a fitness-dependent death mechanism. The authors have compared the proposed ECWSA method with popular FS methods as well as its predecessors. In addition, 18 well-known UCI datasets have been used to evaluate the proposed method. The proposed method's findings demonstrate a significant improvement in terms of FS.

A hybrid (i.e., filter-wrapper) method for FS is proposed by Mafarja and Mirjalili, (2019) where the binary ALO is hybridized with the QuickReduct (bALO-QR) as one incremental hill-climbing technique. A pool of solutions (ants) is randomly generated and then improved through two filter FS models; thus, the common informative features in the dataset are embedding. The binary ALO method then utilises the resulting population to find the best solution. The authors have compared the proposed method with the well-known methods and tested on 18 common datasets from the UCI repository. The experimental results of the proposed method prove superior in performance when seeking the best feature subset.

Song et al. (2021) propose a three-stage hybrid FS algorithm based on correlation-guided clustering and PSO. In the first and second stages, designing a filter FS method with a clustering-based method to decrease the search space is utilised through the third stage. The final stage involves employing an evolutionary algorithm with global searchability to determine an ideal feature subset. The authors have compared the proposed algorithm with nine FS algorithms and validated it on 18 real-world datasets. The results show the ability of the proposed algorithm in obtaining a significant feature subset. However, the feature clustering method requires setting a threshold value manually which impacts the sophisticated clustering method.

A hybrid filter and wrapper approach with the concept of the minimum redundancy maximum relevancy with a modified Bat Algorithm (BA) is proposed by Al-betar et al. (2019). Minimum redundancy and maximum relevancy is utilised as a filter to pick the most informative genes while the modified bat algorithm is used to determine a small set of promising genes (i.e., search engine in wrapper method). The performance of the proposed, compared against ten FS methods and tested on ten gene expression datasets, produces better results than the compared FS methods.

Xu et al. (2021) propose a filters and wrappers FS method based on an improved ABC algorithm. Three various variable ranking algorithms are used in the filter to pre-rank the candidate features, and the population of the ABC is then initialised based on the importance of the re-rank features. In the wrapping, the ABC method algorithm individuals (feature subsets) are based on the classifier's classification accuracy in

order to produce the best feature subset. The method, validated on Parkinson's disease datasets, achieved better results than other compared algorithms.

A combination of ACO algorithm with wrapper and filter method was introduced by Ghosh and Sarkar, (2020). At each step to determine the classification capabilities of the new feature, the heuristic desirability is determined using a classifier (i.e., wrapper based ACO). Then, to reduce computational complexity of subset evaluation, the filter method is utilised rather than wrapper method. The proposed method has been evaluated on different datasets (i.e., facial emotion recognition, UCI, microarray, and NIPS2003 FS challenge). The experimental results outperform other compared FS methods.

#### **2.4.4 Ant Colony Optimisation**

The ACO, a nature-inspired metaheuristic approach was developed to address challenging combinatorial optimisation problems. The ACO inspiration is stigmergy, i.e., the indirect communication between ants. For example, ant in the real condition, uses indirect interaction with other ants. The probability that other ants in the colony will build solutions to the problem is represented by the pheromone deposited in the environment (Falaghi & Haghifam, 2007; Dorigo et al., 2006). The most common use of ACO is in NP-hard combinatorial optimisation problems including scheduling, network routing, and data mining (Chang et al., 2009; Dorigo et al., 2006). In the context of FS, ants can find solutions to problems with complicated and large search spaces. Exploration and exploitation are the strategies for searching into space. (Sagban, 2016).

Ant system is the first variant of ACO algorithm to be proposed to solve travelling salesman problems. It was able to match other general-purpose heuristics' performance, such as evolutionary computation (Colormi et al., 1992; Dorigo et al., 1996). Despite these initial impressive results, ant system failed to match the results of other state-of-the-art algorithms such as integer linear programming formulations, branch-and-bound algorithm, and composite algorithm (Laporte, 1992). As a result, a significant amount of ACO research has concentrated on other variants of ACO that demonstrate superior performance to ant system. Other improvements were designed based on Ant System such as the Ant Colony System (Dorigo & Gambardella, 1997b), and MAX-MIN Ant System (Stützle & Hoos, 2000).

The family of ACO algorithms has been successfully used to solve a wide range of combinatorial optimisation problems, including routing (López-Ibáñez & Blum, 2010), scheduling (Morin et al., 2009), feature selection (Ortega et al., 2022) (Sivagaminathan & Ramakrishnan, 2007), and bioinformatics (YiMing et al., 2020). Since there are no known polynomial-time algorithms for these problems, heuristic methods like ACO are frequently utilised to produce high-quality solutions in manageable computation times (Stützle et al., 2011).

Tabakhi et al., (2014) introduced a FS method called UFSACO that aim to select subsets of the feature with a low similarity between features (low redundancy). Other later methods based on the same idea are UPFS (Dadaneh et al., 2016), RR-FSACO (Tabakhi & Moradi, 2015), and MGSACO (Tabakhi et al., 2015). In both MGSACO and RR FSACO, they also calculate the relevance of each feature by its variance in

addition to quantifying the feature. Another interesting method of FS is the MGCACO which is extinction from an algorithm called GCACO (Ghimatgar et al., 2018; Moradi & Rostami, 2015b) that is integrating the feature clustering approach with ACO-based search process for FS to improve the performance efficiency of selecting the feature with a low similarity.

#### 2.4.5 Discussion on Feature Selection-based Clustering Method

The overall benefits and drawbacks of FS algorithms associated with the wrapper, hybrid, and filter methods are summarized in Table 2.1.

Table 2.1

*Advantages and disadvantages of FS methods*

Method	Disadvantages	Advantages
Filter	<ul style="list-style-type: none"> <li>• Ignores interaction with clustering algorithms</li> </ul>	<ul style="list-style-type: none"> <li>• Fast</li> <li>• Scalable</li> <li>• Independent of the clustering algorithm</li> <li>• Parallelizable</li> </ul>
Wrapper	<ul style="list-style-type: none"> <li>• An overfitting risk</li> <li>• High computational cost</li> <li>• The selection is particular to the clustering algorithm used</li> </ul>	<ul style="list-style-type: none"> <li>• Interacts with the algorithm used for clustering</li> <li>• Can model feature dependencies</li> </ul>
Hybrid	<ul style="list-style-type: none"> <li>• The selection is particular to the clustering algorithm used</li> </ul>	<ul style="list-style-type: none"> <li>• Can model feature dependencies</li> <li>• Interacts with the algorithm used for clustering</li> <li>• Requires less time than wrapper</li> </ul>

Feature selection algorithms, which depend on the filter approach, focus on general data characteristics and evaluate features without any classification algorithms. Therefore, these methods do not focus on any specific learning models. Furthermore, filter methods are simple to construct, easy to understand, and are typically very fast to implement, making them suitable for high-dimensionality data (Zhao, 2010). Nevertheless, there is an inclination for FS filter methods to be improved based on feature clustering. Such methods obtain good results with regard to the quality of the selected features due to the useful information that can be obtained from the clusters to the FS search algorithms. Table 2.2 provides a review summary of the feature clustering approaches implemented in the FS methods. In this table, the datasets are also highlighted.

Table 2.2

*Summary of FS based on clustering method*

No.	Reference	Clustering concept	Dataset
1	Xu et al. (2008)	-	Gene Expression
2	Moradi and Rostami (2015b)	√	Microarray & UCI
3	Ghimatgar et al. (2018)	√	Electroencephalogram & UCI
4	Manbari et al. (2019b)	√	Microarray & UCI
5	Rostami et al. (2020)	-	Microarray
6	Babatunde et al. (2014)	-	UCI
7	Tabakhi et al. (2014)	-	UCI
8	Tabakhi and Moradi (2015)	-	Microarray & UCI
9	Tabakhi et al. (2015)	-	Microarray & UCI
10	Dadaneh et al. (2016)	-	UCI
11	Faris et al. (2018)	-	Microarray & UCI
12	Al-tashi et al. (2019)	-	UCI
13	Guha et al. (2020)	-	Microarray & UCI
14	Mafarja et al. (2019)	-	UCI
15	Mafarja and Mirjalili, (2017)	-	UCI
16	Mafarja and Mirjalili, (2019)	-	UCI
17	Thaher et al. (2020)	-	Microarray
18	Rostami et al. (2021)	√	Microarray & UCI
19	Emary et al. (2016a)	-	UCI

20	Taradeh et al. (2019)	-	UCI
21	Song et al. (2021)	√	Microarray & UCI
22	Al-betar et al. (2019)	-	Microarray & UCI
23	Xu et al. (2021)	-	Parkinson's disease
24	Ghosh and Sarkar, (2020)	-	UCI
25	Too et al. (2018)	-	Electromyography
26	Too and Rahim Abdullah, (2020)	-	Microarray & UCI
27	Guha et al. (2020)	√	UCI
28	Mafarja and Mirjalili, (2018)	-	UCI
29	Mafarja et al. (2020)	-	UCI

The data listed in Table 2.2 provide a distribution of 29 studies from 2008 to 2021. The usage of the microarray and UCI datasets is prominent because of the availability of both irrelevant and redundant features as compared to other datasets. In addition, there are seven (7) studies where the clustering concept has been employed in the task of FS. Generally, clustering of features or grouping of features by enhancing the stability of the feature set and reducing the model's complexity is a useful learning technique for high-dimensionality data (Shen & Huang, 2010; Jörnsten & Yu, 2003). In recent studies, the clustering of features has emerged as a forceful tool for more effective search and reducing dimensionality in high-dimensionality data. However, most existing feature clustering methods have some limitations (Moradi & Rostami, 2015b). Firstly, the required number of clusters must be pre-specified. In general, setting the right number of clusters requires exhaustive trial-and-error. Secondly, the current algorithms do not consider the variance, and the data division in a cluster is a significant consideration in the dependent cluster's similarity calculation. Thirdly, all attributes in a cluster participate in the resulting extracted feature in the same degree. To overcome this issue, an FS method based on the feature clustering algorithm to group the attributes could be possible.

Several bio-inspired algorithms based on the paradigm of SI have been proposed for the FS method based on the feature clustering algorithm (Rostami et al., 2021; Song et al., 2021; Guha et al., 2020; Manbari et al., 2019a). SI involves a group of artificial agents that attempt to mimic the natural behaviours of a population of animals. Every agent does a search mission individually and, together, they determine a complex issue. The most popular swarm-based algorithms are PSO (Song et al., 2021), GSA (Guha, Ghosh, Chakrabarti, et al., 2020), and ACO (Manbari et al., 2019a; Ghimatgar et al., 2018; Moradi & Rostami, 2015b). However, ACO is more flexible in the scope of FS than other algorithms (Manbari et al., 2019b; Fahrudin et al., 2016). ACO has many advantages, such as a similar function to the reinforcement learning scheme, distributed long-term memory, positive feedback, parallel nature implementation, and good exploration and exploitation capabilities due to greedy and stochastic algorithm components (Dorigo & Caro, 1999; Dorigo & Stützle, 2019; Tabakhi et al., 2014; Dorigo & Gambardella, 1997b; 1997a; Dorigo et al., 1996).

Based on the analysis of relevance and redundancy, GCACO, MGCACO, and FSCBAS methods show a major advantage over other methods due to the use of both feature clustering and ACO-based FS filter method. Thus, integrating the feature clustering approach based on the community detection algorithm with ACO-based search process for FS has shown improvement in the performance efficiency of the proposed algorithms (Manbari et al., 2019; Moradi & Rostami, 2015b). However, these algorithms still suffer from the shortcoming of community detection techniques to cluster highly correlated features, where the algorithms utilise a greedy search basis that usually leads to limited search space. Visible solution occurs in higher modularity

when the utilisation of modularity maximization as a goal in these algorithms produces merging communities that should be independent. In comparison to metaheuristic-based clustering methods, these algorithms have lower efficiency because they are easily stuck in local optima, which affects performance on production of a low-quality subset. Therefore, this study will use the ACO-based clustering method to cluster highly correlated features within the same group.

## **2.5 Control Parameter**

The control parameters (CPs) have a considerable influence on the performance of most algorithms. Correct setting of the values for the parameters is a crucial aspect in the search for an optimal solution. However, the values of the parameters are problem-dependent. Therefore, value setting is not an easy task. The purpose of CPs is to maintain the balance between exploration and exploitation of the optimisation algorithms (Feoktistov, 2006). Exploration is linked to finding new solutions, whereas exploitation is linked to searching for good solutions. Exploration and exploitation are both interwoven in the evolutionary search (Fister et al., 2011).

The MGACO algorithm shows the sensitivity of determining the parameters due to the use of manual CP setting to set the appropriate threshold value to guide the ACO subset construction before execution (Ghimatgar et al., 2018; Moradi & Rostami, 2015b). To explore the existing techniques on the replacement tuning of parameters (also referred to as offline parameter tuning) with parameter control (also known as online parameter tuning) for metaheuristic algorithms, this section presents the reviews on works in parameter control.

Metaheuristics are general algorithmic templates or high-level methodologies that, typically, do not deeply acclimate particular problems (Boussaïd et al., 2013). Therefore, they can usually solve a variety of problems (Talbi, 2009). The prefix “meta” is defined as “higher level methodology”, therefore the metaheuristic algorithms are regarded as “higher level” heuristics algorithms. Various metaheuristic algorithms have been developed and widely implemented including Evolutionary Algorithms (Davis et al., 2012), Gene Expression Programming (C. Ferreira, 2001), Tabu Search (Glover, 1986), Differential Evolution (DE) (Engelbrecht & Pampará, 2007), Simulated Annealing (Kirkpatrick et al., 1983), PSO (Kennedy & Eberhart, 1995), and ACO (Dorigo & Birattari, 2010). Some metaheuristics are inspired by nature (inspired by some concepts, e.g., in physics or biology), which include stochastic procedures, as well as having several free parameters that users can set manually or modify according to the problem (Boussaïd et al., 2013). The adjustment of parameters has a strong impact on the metaheuristic’s performance or effectiveness since the parameters regulate the demeanor of the heuristic mechanisms of the algorithm (Hamadi et al., 2013). As a result, to achieve high efficiency, it is important to clarify which configuration ought to be the most promising. An optimisation problem arises within the problem at the point of “how to adjust these parameters?” Eiben et al. (1999) illustrate two ways to adjust parameters, while Zhang et al. (2012) adapt by gathering some specifics about the adaptation mechanisms as shown in Figure 2.3.

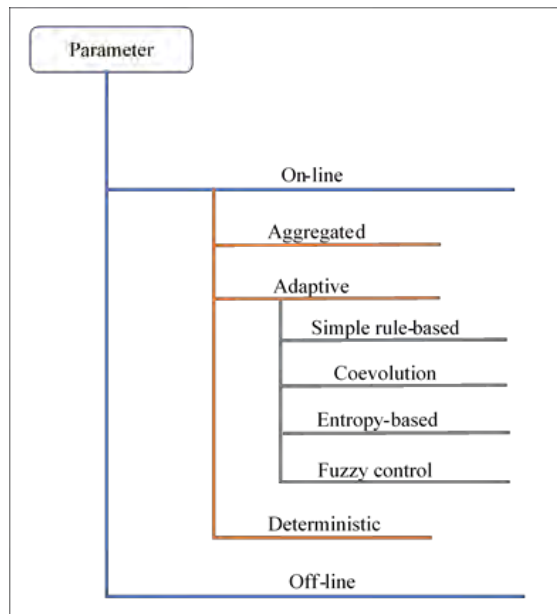


Figure 2.2: Taxonomy for parameter control

The first is to classify offline parameter control before the algorithm is implemented, in which several checks are performed on different values. The aim is to obtain a set of parameters with top values. With this introductory review, the standard parameter values for possible executions can be proposed. However, such guidelines should not be generalized for all types of problems. Due of the high percentage of possible numbers that can be expected for each parameter, an offline controlling parameter has proved to be a complex combinatorial problem (Aleti & Moser, 2013a).

An online parameter setting is also known as the second type of parameter adjustment; when the algorithm is run, the parameter values change. The online parameter control removes the earlier stage of parameter analysis so that the change takes place through the optimisation progression. Consequently, the designer or developer of the optimisation algorithm disclaims the liability of modifying these parameters in the

online parameter control. An essential feature of parameter online controlling is that a parameter's value may vary depending on the search stage (Hinterding et al., 1997). The results obtained can be significantly improved by changing parameter values during the optimisation process and this has inspired work for methods to self-adjust parameters from multiple metaheuristics (Aleti & Moser, 2013b; Kamrath et al., 2013; Leung et al., 2012; Marques & Gomide, 2011; Kramer, 2010; Simons & Parmee, 2010). Approaches for changing the value of a parameter during the optimisation process can be classified into deterministic, aggregated, and adaptive (Brest et al., 2006; Liu & Lampinen, 2005).

### **2.5.1 Deterministic Parameter Control Approach**

The deterministic parameter control method is applied when a determinist rule adjusts the importance of the parameter without utilizing any information about the fitness function. Typically, the values of the parameter are adjusted over time using a predefined function. For example, over the optimisation process, the parameter value starts with a large value, then gradually decreases. It is common to use fitness tests or generation number (iteration) to control the value. Due to its simplicity and low computational effort, this method of control is used in many applications. During optimisation, this control strategy barely considers the algorithm's bio-inspired conduct (Zhang et al., 2012).

Huang (2013) applies the DE algorithm by introducing a method of adjusting the crossover likelihood and the deterministic mutational factor using sine and cosine

functions. Similarly, Zhang et al. (2013) provide a technique to modify the probability of crossover to change the value of the parameter using the number of iterations.

Boeringer and Werner (2002) propose a simple control method in GA for the possibility of mutation through a uniform distribution. Li and Chang (2006) present a method to adjust the probability of crossover and mutation that slowly lowers the values' relativeness to the generation number. Fernandez-Prieto et al. (2011) introduce different deterministic approaches for adjusting the probability of mutation. The first approach defines a linear function that reduces the value of mutation by the generation number following the idea suggested by Bäck and Schiitz (1996). The second approach involves the usage of random values in each generation for the mutation parameter. Likewise, Vajda et al. (2008) offer multiple methods in adjusting the tournament size based on the generation number. In this case, the parameter value is tuned linearly.

In utilizing the ACO algorithm, Merkle et al. (2002) suggest two methods under deterministic CP for the amount of pheromone used to choose the direction and rate of pheromone evaporation. The pheromone value decreases; nonetheless, the pheromone evaporation rate increases as the number of iterations increases. Meyer (2004) also provides a mechanism to control the parameters deterministically using the annealing method.

The online deterministic control technique has been implemented in PSO algorithms by many authors. Yasuda et al. (2010) present a basic parameter adaptation method using linear function for the parameters' cognitive and social components. Similarly, another control method, suggested by Khadhraoui et al. (2016), is based on the

iteration number of the social and cognitive parameters that will change. On the other hand, a linear function is used by Niknam and Farsani (2010) to adapt the parameter of inertia weight. Shi and Eberhart (1999) also present a linear decrease in the inertia weight of particles with a deterministic approach and based on the iteration number. Conversely, Chatterjee and Siarry (2006) present a deterministic method in adapting the weight of inertia with a non-linear function. Alfi (2012) and Jiao et al. (2008) pursue the same concept as Chatterjee and Siarry's (2006) work through the use of non-linear functions. Eberhart and Shi (2001) suggest a simple and random method to change the weight of inertia. Wang (2011) introduces deterministic CP where the parameter of weight decreases, and social parameter increases via a linear function dependent on the number of iterations. The cognitive parameter is adjusted utilizing the personal best (pbest) and global best (gbest) particle information. A different parameter control method is suggested by Montalvo et al. (2010). The parameter of inertia weight is regulated by a deterministic method, which reduces the parameter value over iteration. Both social and cognitive parameters are controlled utilizing a simple rule. Setyawan and Jazidie (2017) employ a Gaussian function-based approach for tuning inertia weight, cognitive, and social parameters. The cognitive and social parameters are controlled according to the values of their maximum, minimum, and iteration numbers.

### **2.5.2 Aggregated Parameter Control Approach**

The parameters are coded in aggregated control as extra dimensions directly in the solution vector and optimised during the optimisation process. The optimisation of parameters could be achieved within the same procedures utilised to optimise the

population or via unique routines (Zhang et al., 2012). Due to the natural selection analogy used in this approach throughout evolutionary computation, there are more chances of best-coded parameter values that drive to better solutions for distributing the values to coming generations (Eiben et al., 1999). SI algorithms can use a similar approach to optimise their parameter values.

Brest et al. (2006) introduce a technique that controls the individually encoded parameters of crossover and mutation using simple rules. Teo (2006) also present two aggregated control models for changing the population size. An aggregated system of control for crossover and mutation parameters is implemented in these two models. The population size for the first model is encoded in each parameter. The rate of population growth in the second model is coded in individual parameters at which the positive value of the population size rises and the negative value decreases. Similarly, Dragoi et al. (2013) propose an aggregated CP approach for crossover and mutation that is updated through the same individual operators.

In Băutu et al.'s (2007) work, Gene Expression Programming is used to adapt an organism's genes' number. The adaptation process takes place at chromosome level, enabling chromosomes to develop with diverse genes in the population. On the other hand, Smętek and Trawiński (2011) conducted research comprising two control models. One of the control models individually encodes the crossover and mutation probabilities. Another way to adapt the probability of mutation is introduced by Fernandez-Prieto et al. (2011), where a specific parameter factor is applied and a new value is generated that is utilised to apply a mutation or transformation to individual parameters.

### **2.5.3 Adaptive Parameter Control Approach**

The adaptive parameter approach utilises information from the optimisation process as feedback to define the parameter value adjustment. Generally speaking, the assessment is provided from the search process (i.e., fitness values) to guide the modification of the parameters (Qin et al., 2009). According to Zhang et al. (2012), there are four types of adaptation method.

#### **1. Simple Rule-based Adaptation Method**

The method occurs when the CP is gained via simple rules (Zhang et al., 2012) and determined from the observation of the algorithm's characteristics and behaviour through its run. This approach uses optimisation knowledge to modify the parameter values. The fitness or objective function, population diversity measures, and others may be the assessment information.

Ding and Wang (2008) introduce an adaptive control method for crossover opportunity by taking into account the population maturity, which is calculated by the population's average fitness. Similarly, Osaba et al. (2013) presented a population-islands-based GA with a crossover probability control method adjusted by simple rules-based with the best fitness value in individual parameters. Likewise, using DNA sequence evolution statistical models, Vafae and Nelson (2009) implement a control model for the probability of mutation by adjusting the optimal distribution of individual genes. Yang et al.

(2001) also present a CP method for the probability of mutation that is tuned depending on the average fitness of the population by a linear function.

Yang et al. (2013) suggest two simple probabilities of rule mutation, one for each population-based average fitness. A method for predicting crossover and mutation parameters, proposed by Aleti and Moser (2011), uses the average fitness of the population together with the success rate of parameters to adjust the possibility for the next generations. The crossover and mutation probabilities are modified by Huang et al. (2016) depending on a function of linear scaling that compares the average fitness of the population with the fitness of individual parameters. On the other hand, Rajappan and Rangasamy (2017) propose a technique of adaptive control based on fitness knowledge for the parameters of mutation and crossover. The population is split into low and high fitness sub-populations. Local exploitation is extended to a higher fitness sub-population. Therefore, only a procedure of crossover is utilised and adjusted to simple rules. However, a sub-population with lower fitness exploration through the mutation method is used with adjustment to simple rules. Conversely, Algethami and Landa-Silva (2017) present a CP approach that can adjust the crossover, mutation, and population parameters depending on a genotypical diversity calculation.

The use of CPs in ACO by Zhaoquan and Han (2008) introduces a method in assigning the value of each individual parameter for evaporation based on the quality of the solution. Chusanapiputt et al. (2006) suggest two distinct

populations for ACO parameters where individuals move between different populations and the parameters are updated based on the diversity control approach. Therefore, heuristic and pheromone parameters are balanced by the population size. On the other hand, in the study of Yuan et al. (2017), the control procedure of heuristic and evaporation parameters is used only if the optimisation method will not enhance the solution within a specified number of iterations.

Recently, Al-Behadili et al. (2020) have presented a new hybrid Ant-Miner classification algorithm that integrates the ant colony system algorithm. The Ant-Miner algorithm presents a new parameter, named importance rate, which is a pre-pruning criterion based on the probability of the pheromone and heuristic amount. This criterion is liable for only including important terms to each rule such that noisy data are discarded. The ant colony system algorithm is implemented to enhance the importance rate parameter during the Ant-Miner algorithm learning process. Experimental findings reveal that the proposed algorithm significantly outperforms the other algorithms.

Yang et al. (2007) present a method using CP to change the inertia weight parameter of the PSO algorithm. This method utilises a function of best fitness of individual parameters and the average fitness of the population to control the parameters. Similarly, Kumar and Rao (2017) suggest an adjustment method for modifying the parameters according to the fitness of the particles. Likewise, a variant of PSO with sub-swarms is proposed by Zhang and Ding

(2011) using an adaptive dynamic strategy based on all sub-swarms' fitness values for inertia weight. Zhang (2018) also suggests adjusting the parameters based on the diversity of the population. The adaptive modification is made according to the degree of particulate aggregation. The inertia weight parameter is controlled in a study of Tang et al. (2015) based on a diversity measurement of the solutions. Likewise, Zielinski and Laur (2007) introduce a CP for social, cognitive and inertia weight parameters where the values of different combinations are utilised in each generation and are adjusted depending on their performance in producing better solutions. Conversely, Harrison et al. (2018) suggest a method in tuning inertia weight, social, and cognitive parameters. If the best individual location of the particles for a predefined number of iterations has stagnated, and a known region contains promising parameter configurations, new parameter values are sampled randomly.

## **2. Coevolution Method**

Coevolution methods use some form of parameter generation mechanism and place those parameters into the process of evolution (Spears, 1995; Back, 1992). In every generation, the parameters coevolve with the population. Various parameters are usually generated for each individual. The parameters of the better individuals will survive in the process of evolution and are used to produce new parameters.

Normally, the self-adaptive methods encode the evolutionary computation algorithm parameters directly into the evolution of the procedure itself. The parameters are encoded directly in the original individuals and pass through the process of evolution as population genes using the same methods as the original genes (Zhang & Sanderson, 2009; Hinterding, 1995). Different approaches are also introduced by Qin et al. (2009) and Huang et al. (2006), where the parameters pass a different process of evolution from the original population. In the study of Qin et al. (2009), the parameters are influenced by Gaussian random disturbance, while in the study of Huang et al. (2006), the parameters of the ACO algorithm are adjusted automatically by PSO. Zhan and Zhang (2010) also apply the PSO learning technique to the DE algorithm for the development of DE parameters of DE throughout the evolution process.

### **3. Fuzzy Control Method**

Fuzzy controllers use relevant degrees to adjust the parameter values (Zhang et al., 2012). Zadeh introduced Fuzzy Logic in 1965 with the intent to add degrees of ambiguity in the conventional Boolean logic, making inferences of human beings in ideas and knowledge not well known to machines. A fuzzy controller's core components are knowledge base, fuzzy law, and inference mechanism (Jantzen, 2007). The inference mechanism is used in the form of parameter control to determine the current state of the system. The rules are provided as described in the state of identification and are applied to verify parameter values.

Liu and Lampinen (2005) present a crossover and mutation parameter factor adaptation control with a DE algorithm using fuzzy logic. The difference between the objective function values and those of the population individuals represent the input of the method. Similarly, Kotinis (2014) introduces a parameter mutation and crossover factor using the fuzzy control method. The control is executed based on the number of solutions not explored and the differential distribution of population among the two iterations.

Peng et al. (2014) present a hybrid GA with simulated annealing. A fuzzy control framework is utilised to adjust the crossover and mutation probabilities based on individual and population fitness. Likewise, Tarokh and Zhang (2014) introduce a fuzzy control method for the likelihood of mutation and crossover parameters based on population fitness diversity for each generation. A fuzzy controller is also proposed by Gudino-Penaloza et al. (2013) to control crossover and mutation parameters in GA. In this study, the GA is run several times while the fuzzy engine is operating. On the other hand, Pereira et al. (2013) suggest a fuzzy controller that adjusts the probability of mutation using iteration number as the input and mutation probability as the output.

Castillo et al. (2013) present a fuzzy control method to adjust the evaporation rate in ACO. The fuzzy control uses error measure to tune the parameter values. Melin et al. (2013) propose PSO with a fuzzy control method for social and cognitive parameters. This control uses the iteration number, diversity measurement, and fitness-based error measure to change the values of the

parameters. Similarly, Olivas et al. (2013) introduce a control that uses a measure of the iteration number and population diversity to adjust the parameters using a fuzzy control method for social and cognitive parameters. In the same work, the fuzzy control adjustment is suggested in which the input variables, diversity measurement, and number of generations are considered fuzzy. Likewise, Neshat (2013) provides a fuzzy control technique for social and cognitive parameters. The technique makes use of population fitness values to modify the parameters.

#### **4. Entropy-Based Control Method**

Entropy-based control methods have been used to control parameters. In information theory, the entropy concept is introduced by Shannon (2001). Entropy is known as a measure of predictable information or a probability distribution ambiguity. It is also defined as a point of disorder or uncertainty in a partition system (Okafor, 2005). The generation of new possible solutions in bio-inspired algorithms generally involves probabilistic and random factors. Entropy can be utilised to evaluate population characteristics and status in such a way as to adjust the parameters accordingly. In GA, Aleti and Moser (2013a) present a control method consisting of a k-means clustering algorithm based on the performance of crossover and mutation values. The entropy is determined for each cluster which is then applied to adjust the parameter values. On the other hand, Li (2009) proposes an entropy-based control method to control the parameters of the ACO algorithm. The method calculates

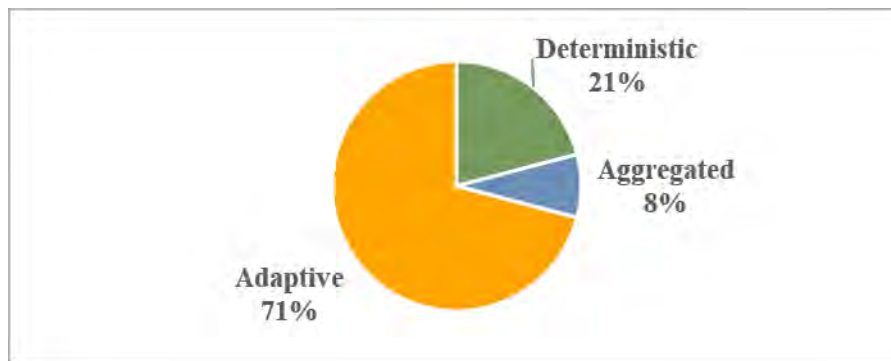
the mean entropy and the parameters of heuristic and pheromone are balanced by a simple rule. Li and Li (2013) follow the same concept of the study by Li (2009). However, they propose a different rule for adapting the heuristic and pheromone parameters.

#### **2.5.4 Discussion on Methods for Parameter Control**

The CPs have a major influence on the quality of algorithms since the appropriate setting is a crucial aspect in the search for optimal solutions. The initial method of trial-and-error (manual control/offline control) is time and resource-consuming and, at the same time, does not guarantee optimal results. Therefore, new approaches have been proposed whereby one of the best solutions developed by researchers is automatic control (online control) (Dragoi & Dafinescu, 2016). One important advantage of online parameter adjustment is that algorithms can better adapt to specific characteristics. It is highly desirable to automate this task, i.e., to implement an algorithmic technique to solve parameter control tasks. In the search stage, adapting the parameters online can also be effective in obtaining better performance and, thus, the best results. Explorative and exploitative search phases of an algorithm are often identifiable and good parameter settings in these phases can, again, be very different. Finally, if algorithms are applied in scenarios very distinct from the situation in which they are designed or offline tuned, authorizing parameters to adjust online may increase the robustness of an algorithm (Hamadi et al., 2013).

Parpinelli et al. (2019) present the distribution of online control parameter methods used in bio-inspired algorithms (refer Figure 2.3). The aggregated method is the least

used online method for CPs due to the drawback of complexity and time. The search space increases as the size of the environment becomes bigger, which leads to an increase in complexity and time in the process of achieving the optimum values for the parameters (Fialho et al., 2010).



*Figure 2.3: Distribution of control mechanisms*

The use of the deterministic method is motivated by the simplicity to execute. Even without the use of information as assessed through the optimisation process, the outcomes achieved with the deterministic method are better than that of static values. Therefore, considering its easiness, this method encourages better intensification and diversification of the selection balance as compared to strategies utilizing static value parameters. Nevertheless, the term pre-scheduled parameter variation is preferred since the “deterministic” adjective does not accurately express this method of adjusting the parameters, as the schedule can be authorized to make randomized controlled choices. If the pre-scheduled parameter diversity is an online tuning process, it will not allow the issue of offline tuning to disappear, because the parameters that determine the schedule must always be set appropriately (Hamadi et al., 2013).

The adaptive method is most commonly used for CPs. This indicates the scientific community's concern with information used as an assessment from the optimisation method to adapt the values of the parameter. Depending on the search stage, there can also be a different set of optimum parameter values. The use of the online parameter adaptation strategy for the threshold value that guides the ACO subset construction can be a focus in future studies.

## **2.6 Subset Determination**

One of the main stopping criteria for the FS process is the optimum size of the feature subset, which is usually predetermined (fixed number of features). However, the best possible subset size of the features must be determined automatically (Dhrif, 2019) because the prevaricated over-fitting with FS method is more capable of generating optimal feature subsets with less computational load. This would reflect an appropriate stop criterion. The decisions taken in the preceding stages may impact the preference for the criterion to terminate (Shukla et al., 2019). Therefore, designing an algorithm with an ineffective subset determination method will result in complex feature subsets and overfitting issues. The following sub-sections discuss the subset determination strategies in ACO-based FS variants as well as GA for subset selection.

### **2.6.1 Predetermined Approach**

The majority of ACO variants for FS utilise a predetermined, fixed number of features for the selection procedure to terminate. This method aims to determine the optimum subset. A new approach based on ACO, namely UFSACO, proposed by Tabakhi et al. (2014) for the unsupervised FS problem, seeks to find the optimal solution. The

method consists of a group of agents, called ants, which collaborate by exchanging pheromone information. Every agent iteratively selects several features using heuristics and knowledge from prior stages. As the method in nature is parallel and iterative, the ant seeks to choose the feature subset in the bigger space that would be near to the optimum global solution. The suggested method is a multivariate methodology that considers potential interactions between features to minimize redundancy between the selected features. The stop criterion is determined as the node number that must be chosen by each ant. The state transition rule attempts to pick features with minimum similarities and maximum pheromone values to previously chosen features. The number of times each ant selects a specified feature will be held in a counter list of features. Then, until a certain number of iterations is reached, the procedure is repeated. Consequently, the features in the subset are listed in decreasing order, depending on the pheromone values. Lastly, the final subset size (as specified by user) with the maximum pheromone value is chosen as the final subset of the feature.

Following the UFSACO approach, Tabakhi et al. (2015) suggest an unsupervised gene selection process known as MGSACO that consolidates the filter approach with the ACO algorithm. The relevance and redundancy analysis are utilised in MGSACO. The amounts of pheromone are correlated with the graph edges. In each step, an ant builds its candidate solution by iteratively selecting the edge with the lowest iterative similarity and the greatest pheromone value. This approach produces a significant and effective analysis of feature redundancy. Then, MGSACO measures the accuracy of the identified solutions via a particular fitness function and also provides fitness values

in the pheromone updating rule. In the next iteration, the fitness function tends to raise the level of the solution pheromone to make the solution components more appealing to the ants. Accordingly, in each iteration, MGSACO memorizes the best gene. Ultimately, across all iterations, the global best subset size number will be considered as the optimally selected subset.

Tabakhi and Moradi (2015) propose two methods based on ACO. The first considers the relevance-redundancy among the selected features based on ACO, while the second uses a new value of the heuristic information to the unvisited features. For the first stage in both methods, each attribute's relevance is calculated by utilizing a specified criterion. Subsequently, the similarity values are computed between each pair of features and correlated with the graph edges. The initial pheromone is then computed for every trait. The second stage is to measure the probability of increasing features during iterative procedures. In this stage, a feature counter array is utilised to specify and compute the number of times the ants pick a given feature. The initial feature counter values are set to zero during each iteration. Then, each ant iteratively picks the following features through a "state transition rule" until a specified number of features is selected by the ant. Thereafter, each time the ant selects a feature, it increases the corresponding feature counter. Finally, the pheromone values are updated at the end of each iteration for all features in compliance with the "Global Pheromone Updating Rule". In this rule, pheromone fraction will evaporate on all nodes. As a result, the frequently chosen features by the ants will gain higher feature counter values (amount of pheromone). The learning process will terminate when the maximum number of iterations is obtained. In the third stage, the features are ranked

according to the highest pheromone values in descending order and features with the maximum pheromone value will be selected.

A UPFS based on the ACO algorithm was introduced by Dadaneh et al. (2016) to iteratively find the optimal subset of features. This algorithm uses inter-feature knowledge that detects the similarities among the features. This will guide the search of the UPFS algorithm to reduce feature redundancy in the final result. In the ACO algorithm, during each stage, the calculation process implemented for the quantities of redundancy among the current feature and all those chosen so far is used to pick the next potential feature. Furthermore, UPFS uses an ant-related pheromone matrix to indicate the extent of the co-presence of each pair or two of the features in the solutions. Then, the features are sorted depending on the probability function of the extracted matrix. The final solution (final subset) is returned based on the top number of features.

A new FS algorithm based on the improved binary ant system and mutation (FSBACOM), introduced by Manbari et al. (2019), aims to enhance FS by growing the search space in a short time and finding the optimal solution by reducing redundancy. In the FSBACOM algorithm, the features are ordered in a circuitous graph wherein each of the select/deselect edges are connected to each of the following features. Continuously, the ant moves through the circular graph in the direction of motion until it returns to its original location. The ant chooses or discards any feature dependent on the state transition rule probability as it goes through the circular graph. The aim of the “state transition rule” is to select features with the highest values of term variance (features that have more information) and features with minimum

similarity with the previously chosen features. The number of times the ants select/deselect specific nodes is used to determine the feature counter value. The pheromone of each node is changed using an updating rule via the feature state counter. Once the stop criteria are satisfied, those features with the lowest reported pheromone values are chosen.

Recently, an unsupervised gene selection method focused on multi-objective ACO was suggested by Naseri and Hasheminejad (2019). This approach is used to identify the most discriminatory powerful genes with the lowest rates of redundancy and similarity by proposing a new fitness function. The approach includes two main components, namely the initialisation and the selection of genes. The search of the problem space is modeled as a completely connected weighted graph. The edges indicate the relationship between the genes while the nodes represent the major set of genes. The values of similarity are determined among the genes and added to the graph edges. Consequently, the pheromone's initial concentration quantity shall be determined by a fixed value on the edges. Finally, each gene's relevance is simply measured by utilizing a novel objective function. The number of times each ant uses a particular edge is recorded in each iteration of the gene selection component. Additionally, the ant will iteratively attach the gene to the gene subset currently chosen to build a candidate solution through the "state transition rule". This process continues until each ant selects a specified number of genes. Subsets of genes for candidates are evaluated using a specified fitness method. The highest result in the current iteration is the subset of genes with a better fitness score. The concentration of pheromone values will be updated through the "pheromone updating rule" and a proportion from the amounts of

pheromone are evaporated on each edge. Then, until the predefined number of iterations is exceeded, this process is repeated. Finally, the global optimal subset of the number of genes in all iterations is selected as the final gene's subset.

Paniri et al. (2020) propose a novel multi-label FS method using ACO (MLACO) to deal with both supervised and unsupervised FS. A graph is used to model the problem where edges represent the similarity between features and nodes represent the features. The ant randomly passes through the features and places some pheromones. Then, in every iteration, each ant updates the pheromone trail that demonstrates intensity and efficiency of the solution. There are two supervised and unsupervised heuristic functions to detect the redundancy and relevance of the feature, with the similarity among features as the redundancy feature and the highest similarity among features and class labels as the relevant features. Features will be sorted based on their pheromone values and features with high values will be chosen.

### **2.6.2 Automatically Determined-based**

The ultimate aim of automatically determining the final subset is to determine an optimal subset of features from the original set to improve the learning algorithm performance. Moradi and Rostami (2015b) propose a novel FS method called GCACO that is based on graph clustering and the ACO algorithm. In this method, the size of each ant's subset construction process and the final subset size of the features are also automatically specified. There are three stages in the GCACO approach. The problem to be solved is modeled as a graph whereby each feature indicates a node while the similarities among the features are represented by the edge weights. Then, a

community detection algorithm is utilised to split the features into diverse clusters. In the last stage, the ant will start picking the final feature subset. In this process, the ant randomly selects features from the clusters based on a threshold value that decides either to pick the next node (feature) from the current cluster or explore a new cluster. Therefore, if the ant decides to stay in the current cluster, the probability values for the features are determined for the current cluster. Otherwise, the probability values for the features of the next explored cluster are determined. After all ants have completed their route through the clustering graph, each feature subset efficiency (i.e., solution) will be measured by applying the pheromone quantity depending on the separability index. The process is replicated after a certain iteration number. Then, the features will be listed in decreasing order and based on the pheromone values for each feature. Finally, in representing the final feature subset, first, the number of features that will be chosen is determined from multiplying a predetermined number of features with the number of clusters. Then, the feature with the maximum pheromone value will be selected.

Ghimatgar et al. (2018) and Rasheeduddin and Rao (2019) propose methods known as MGCACO and extreme learning machine with GCACO (GCACOELM), respectively. In the work of MGCACO, modification in the ACO component (i.e., pheromone strategy, exploration, and exploitation) has been suggested to enhance the efficacy of the feature subset selection process. The GCACOELM algorithm is developed to predict thyroid disease by classifying the thyroid nodules utilizing unsupervised FS features. Both methods follow the same process of determining the final subset size, where all the features of the dataset will be sorted depending on the highest pheromone

values. Then, the final subset will be returned based on their first number of features with the highest pheromone value.

Manbari et al. (2019) introduce a hybrid filter-based FS method based on the combination of clustering and enhanced BAS. The approach uses linear BAS, a new reduction redundancy policy, mutation injection, and clustering to enhance the learning techniques by minimizing redundancy to the limit and providing an effective result to the search area in a limited time. In the first step, feature clustering will be implemented. Then, the features are distributed in an annular graph in each cluster, and the proposed BAS-based FS, named BASM, is applied. In the second step, based on the pheromone's values, the highest features of the corresponding classes will be set on a circle graph to compete with the others. Following this policy, the lower-ranking feature in a cluster has the opportunity of selecting the final set of features relative to the higher-ranking feature in the other cluster. The search space representation in BASM significantly decreases "computational time", especially for high-dimensionality datasets.

### **2.6.3 Genetic Algorithm**

Darwin's biological evolution theory introduced the basic principles of GA (natural selection, recombination, mutation, and reproduction) (Srinivas & Patnaik, 1994). This population-based metaheuristic approach is used to address NP-hard problems. The GA begins by generating several initial solutions at random and then iteratively improves them. Generating an initial population known as chromosomes, which stands for the solutions to the optimisation problem, is the fundamental concept of the GA

algorithm. The produced chromosomes are then assessed using a fitness function after being merged from two chromosomes (i.e. parents) using a crossover operator. In order to produce better solutions, namely offspring, the crossover operator exchanges critical information across solutions. Small changes in the offspring are made at random using a mutation operator. The mutations solution could differ from the one that was previously produced. Lastly, the selection operator chooses solutions from the parents and offspring based on the fitness function to maintain a constant population size (i.e. the highest fitness will survive). The full cycle of these three operations (crossover, mutation, and selection) is referred to as a generation. The method will settle on the optimal solutions after a number of generation cycles, thus indicate the best solutions to the optimisation problem.

There are some simple variants that are sufficiently effective. Thus, improving the efficiency and robustness of the GA (i.e., adaptive GA, hybrid GA, and Elitism). In adaptive GA, the parameters of such as population size, crossing-over probability, or mutation probability, are changed as the GA is operating. In addition, when additional auxiliary information, such as derivatives or other special knowledge about the goal function, is available, hybrid Gas is used. Elitism is a very effective element that realizes the “best must survive” principle. Thus, elitism is a commonly used element but it should be utilised with caution, as it can lead to premature convergence (Huang & Yang, 2019; Sharapov, 2007).

A wide range of combinatorial optimisation problems, including path planning (Hu & Yang, 2004), image processing (Hashemi et al., 2010), scheduling (Chen et al., 2020), and real-time systems (Abdelhady et al., 2020) has been successfully solved using GA

techniques. Thus, GA has been approved it extremely applicable in different artificial intelligence approaches as well as different basic approaches (Al-Behadili et al., 2021; Al-behadili et al., 2020b; Demestichas et al., 2000; Podgorelec et al., 2000).

Rostami et al. (2021) and Babatunde et al. (2014) introduce an FS methods to obtain a combinatorial set of features. In terms of the performance of the presented methods, different benchmark classification problems were analyzed by the authors. Both methods used k-NN-based classification as a fitness function. The obtained results are better than other FS methods in terms of classification accuracy.

#### **2.6.4 Discussion of Methods for Subset Determination**

Subset size determination is important in selecting the subset of features from the original data features. The process of determining the subset can produce top informative features, reduce overfitting, and improve classification accuracy. Table 2.5 provides a breakdown of the strategies for determining subsets that were employed by ACO-based FS. The review includes 11 studies from 2014 to 2020.

Table 2.3

*Subset size determination approach for FS in ACO algorithm*

No.	Reference	Algorithm Name	Subset Determination Approach		FS using Clustering
			predetermined	Automatic	
1	Tabakhi et al. (2014)	UFSACO	√	-	-
2	Tabakhi et al. (2015)	MGSACO	√	-	-
3	Tabakhi and Moradi (2015)	RRFSACO	√	-	-
4	Moradi and Rostami (2015b)	GCACO	-	√	√
5	Dadaneh et al. (2016)	UPFS	√	-	-
6	Ghimatgar et al. (2018)	MGCACO	-	√	√
7	Manbari et al. (2019a)	FSBACOM	√	-	-
8	Rasheeduddin and Rao (2019)	GCACOEL M	-	√	√
9	Manbari et al. (2019b)	FSCBAS	-	√	√
10	Naseri and Hasheminejad (2019)	MOACO	√	-	-
11	Paniri et al. (2020)	MLACO	√	-	-

The usage of the predetermined approach is more popular when compared to the automatically defined approach. All experiments in these studies were conducted on datasets from the UCI repository. It is shown that using the predetermined approach produces lower classification accuracy and uninformative feature (Manbari et al., 2019b; Ghimatgar et al., 2018; Moradi & Rostami, 2015b). This is due to the fact that the predetermined approach seeks offline tuning that is generally constructed in a trial-and-error process. This process is error-prone, human-intensive, time-consuming, and

highly dependent on application domains that often lead to irregular tuning of the algorithm (Hamadi et al., 2013). In addition, the value that determines the size of the final subset is dataset-dependent and crucial. A user-determined value may produce a large subset or small, which is not regarded as an intelligent method of solving the problems associated with optimal subset determination. Choosing the optimal final subset is a challenging task. Very few features will be selected if the final subset size value is set too small. Insufficient information will be available for the classification task. On the other hand, when the final subset size value is high, several more features will be picked and, in the data, the noise will blur the natural patterns. There is a high probability that redundant features will be chosen (Moradi & Rostami, 2015b).

The automatic approach performs superior to the predetermined approach in terms of feature number and classification accuracy. However, such algorithms that utilise the automatic approach still require a predefined parameter to proceed with the automatic selection mechanism. Another drawback is the construction of the automatic determining mechanism does not consider any information about the efficiency of features or the dependency of the dataset. Consequently, it will provide either fewer or more features that leads to lower performance of the algorithm.

To overcome the drawbacks of the automatic approach, a new method for the automatic determination of the final subset can be proposed. The final subset can be evaluated based on the accuracy of the picked features (i.e., number of features, classification accuracy, and information or importance). This method will involve a subset determination to select an appropriate subset, wherein the subset can be instantaneously adopted automatically rather than being fixed or chosen by the user.

The metaheuristic population-based optimisation features discovered in GA can be used to enhance the search strategy for subset determination.

## **2.7 Discussion**

Data reduction in high dimension DNA microarrays is a crucial issue in the domain of machine learning, especially in the task of classification. Therefore, improving the abilities to determine the most accurate genes in the datasets will reduce the computational costs. The FS-based filter method has been widely used for datasets with high dimensions because it requires less computational costs than wrapper and hybrid approaches. The studies on FS-based filter methods showed ways to determine the accurate subset of features. Nevertheless, one of the main drawbacks of the FS-based filter method is that it fails to detect the importance of features in the dataset. Thus it does not take into account the redundancy of the features. To overcome this issue, an FS method based on a community detection algorithm to cluster the features is proposed. However, the community detection algorithm showed no possibilities of seeking the global optima. Another challenge that was observed is that most methods demand the specification of hyper-parameters in their models such as threshold value that guides the ACO subset construction or subset determination. Therefore, an automatic technique can be developed utilising the learning process' feedback to change the value of the parameter or the chosen subset in accordance with the datasets being used and the search stage.

## 2.8 Summary

Reviews highlighted that in several datasets, the MGCACO algorithm generates a set of features that perform well and outperform other FS techniques. This algorithm can obtain the correlation among features, which makes it one of the competitive FS algorithms due to its ability to determine the most accurate features. However, the feature clustering algorithm (i.e. community detection algorithm) is unable with the global search procedure and thus faces a local optimisation problem that impacts the MGCACO performance. Additionally, the MGCACO algorithm's proposed threshold value is constant. Selecting such a value is critical for selecting the appropriate features to be present in the subset that these values be used. The original MGACACO final subset determination has a limitation with no possibilities for automation, making the existing final subset determination approach in the literature ineffective. Therefore, this method produces a poor quality subset. Thus, developing metaheuristic-based feature clustering algorithm will have the ability to explore and exploit the search space and obtain good feature groups. Additionally, utilising the feedback of the learning process' to change the threshold value in accordance with the dataset and search stage, can improve the proposed threshold in the literature. A new approach with the ability to automate the final subset determination can be developed. The methodology of this study is presented in the next chapter.

## **CHAPTER THREE**

### **RESEARCH METHODOLOGY**

#### **3.1 Introduction**

The selection of an appropriate research methodology is essential as it has a significant influence on the validity of the outcome. A methodology enables the formulation of the path to be used in conducting the study and reporting the findings.

This chapter presents the methodology of the study to produce the proposed EGCACO algorithm. The methodology that is involved in this study is organized as follows. Section 3.2 describes the research framework. The methodology, techniques, and proposed algorithms are explained in Section 3.3. The datasets and the classifiers together with the evaluation criteria are discussed in Section 3.4. Finally, Section 3.5 presents a summary of the chapter.

#### **3.2 Research Framework**

The research framework aims to provide a direction in achieving the research objectives. It represents the guideline of the proposed study. The research framework as shown in Figure 3.1 includes four stages. The first stage is enhancing the clustering method in the MGCACO algorithm. This includes grouping highly correlated features using the ACO algorithm. The second stage focuses on developing an adaptive threshold value selection technique that makes use of an adaptation criterion as well as ACO ideas. The third stage focuses on developing an information-based method in

automatically identifying the final subset using GA principles. Finally, the fourth stage focuses on the evaluation of the proposed algorithm.

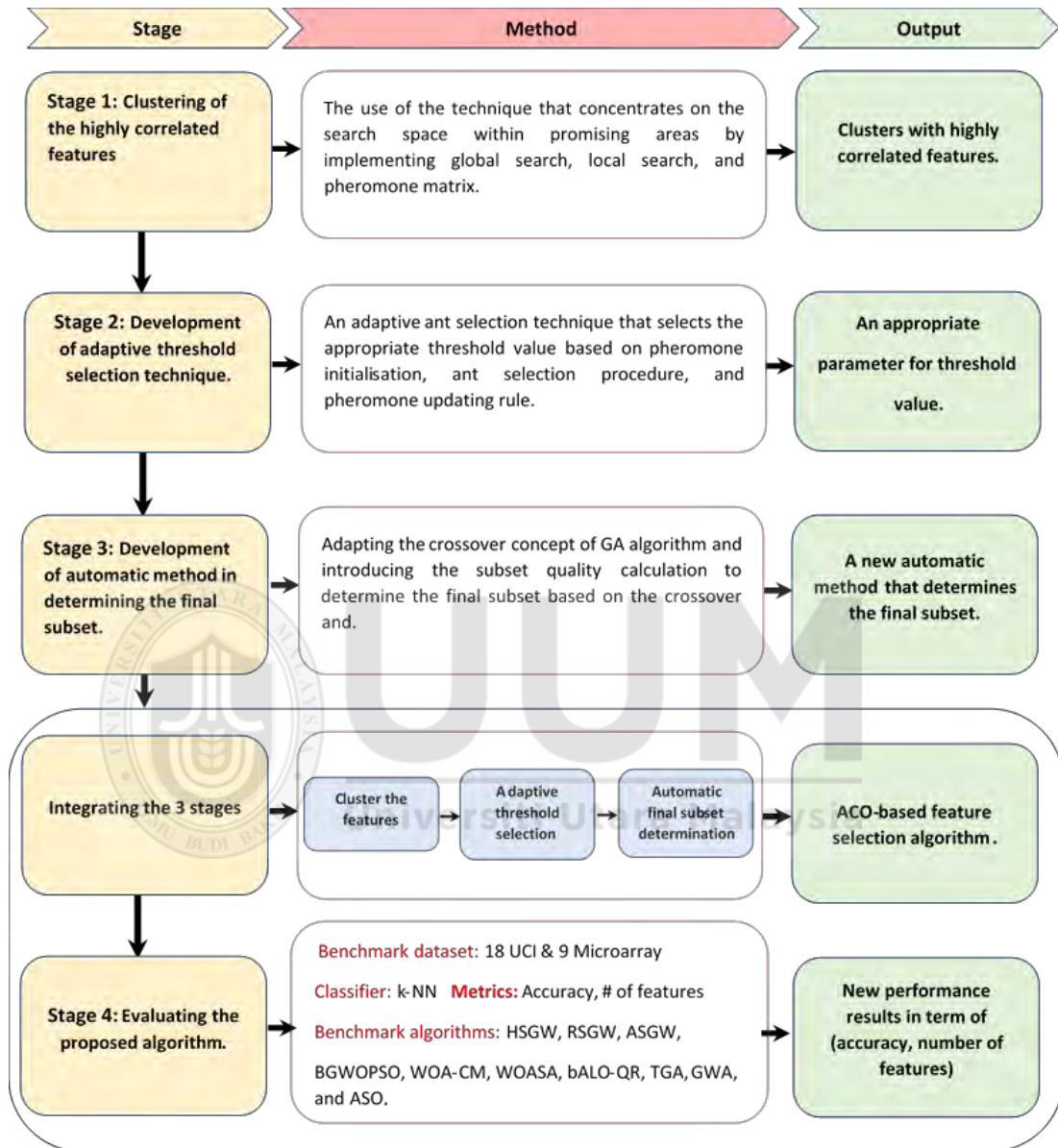


Figure 3.1: Research framework

### 3.3 Research Method

In the area of FS algorithm development, the common methodology that is used is the experimental methodology (Kabir et al., 2012; Nemati et al., 2009; Wang et al., 2015). In this study, there are three (3) enhancements in MGCACO that have been made as shown in Figure 3.2. The first stage deals with the enhancing and clustering of highly correlated features. The ACO algorithm has been customized for this purpose. In the second stage, an adaptive threshold technique that guides the ACO subset construction has been designed. The final stage focuses on developing a fully automatic technique in determining the final subset. This has been based on GA principles and information on the features.

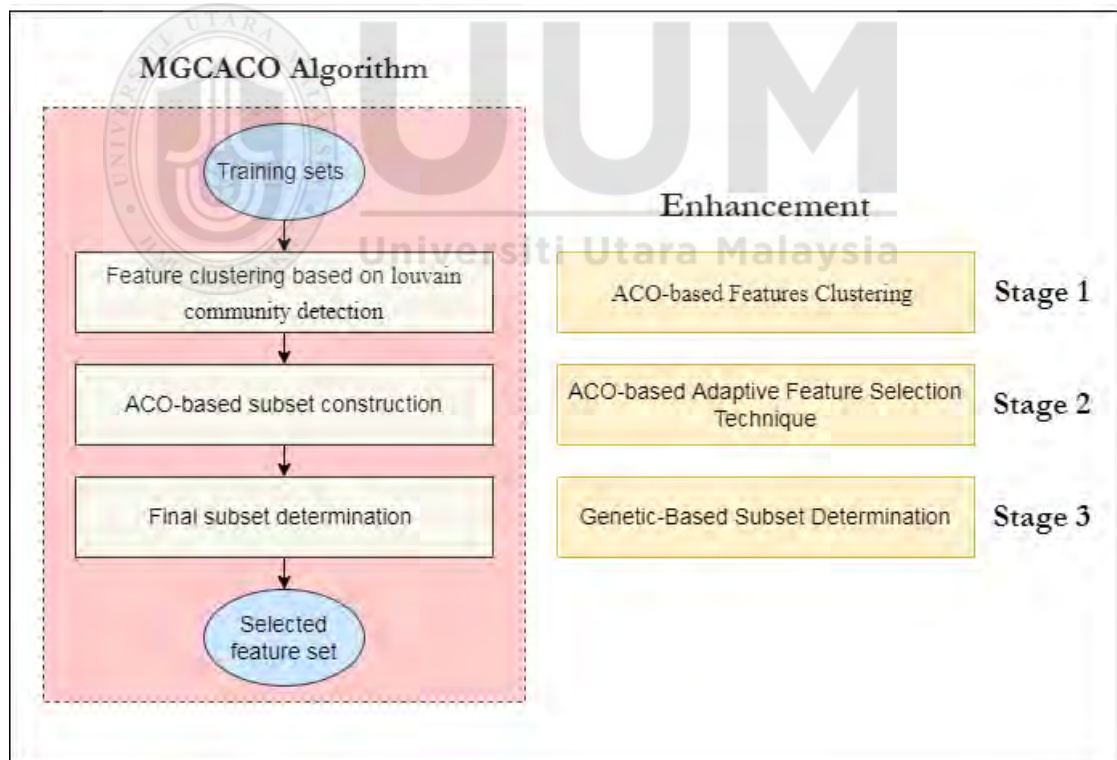


Figure 3.2: Block diagram of the enhancement in MGCACO algorithm

The stages in the proposed MGCACO are presented in the following sections.

### 3.3.1 Ant Colony Optimisation for Features Clustering

This section discusses the ACO algorithm to solve the issue of feature clustering. The main difference between the proposed ACO-based feature clustering and the existing ACO-based clustering is that the proposed method adapts the ACO to construct clusters of features depending on the data features (i.e. an ant starts with an empty solution string  $S$  of length  $N$  where each element of the string corresponds to one of the test features). In the existing ACO, the data samples are used to construct a cluster of samples (i.e. an ant starts with an empty solution string  $S$  of length  $N$  where each element of the string corresponds to one of the test samples). Thus, the goal is to achieve the optimal distribution of  $N$  features into one of the  $K$  clusters to minimize the sum of squared Euclidean distances among each feature and the cluster centre. The algorithm deals with a number of ants,  $A$ , to construct solutions. An  $S$  represents an empty solution string of  $N$  length where each element of string corresponds to one of the test features. The value assigned to a solution string element  $S$  reflects the number of clusters to which the test sample is allocated in  $S$ . For example, Figure 3.3 shows  $S_1$  constructed for  $N \text{ features} = 10$  and  $K \text{ clusters} = 3$  as a representative of a solution string.

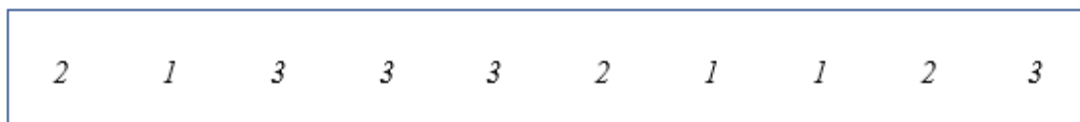


Figure 3.3: Example representing a solution

String cluster number 2 is assigned to the first feature and cluster number 1 assigned to the second feature and so forth. The ant assigns a cluster number to each feature in string  $S$  in one of the following ways:

The first process is referred to as exploitation where the cluster with the highest pheromone concentration is selected using probability. The second process is referred to as biased exploration where one of the  $K$  clusters utilises a stochastic distribution and a probability. Then, a local search procedure is applied since the heuristic information of the features is not available. For this purpose, the best 20% of the total solutions will be enhanced by the local search procedure. After the local search procedure is completed, the ant utilises the pheromone trail information to construct a solution that assigns each feature in string  $S$  to an appropriate cluster label. At the beginning of the process, the matrix of the pheromone is initialised to a small value,  $\tau_0$ . The position  $(i, j)$  is assigned with a trail value of  $\tau_{ij}$ , which represents the feature pheromone concentration  $i$  associated with cluster  $j$ . The size of the pheromone matrix is  $N \times K$  for the problem of distributing  $N$  features into  $K$  clusters. Therefore, each feature is correlated with the concentration of pheromone  $K$ . The trail matrix of pheromone improves as it iterates. At any iteration level, each ant will improve the test solutions using a pheromone-mediated communication process with a view to obtain a near-optimal partition of the  $N$  test features into  $K$  groups that satisfy the given objective. The matrix of the pheromone is then updated according to the quality of the solutions provided by the ants. Therefore, the updating of the pheromone matrix will guide the ants to construct improved solutions and repeat the above steps for a particular number of iterations. This type of metaheuristic feature clustering algorithm

is as shown in Figure 3.4 which enhanced the performance of the MGCACO algorithm, that concentrates on the search space within promising areas by implementing global search, local search, and pheromone matrix.

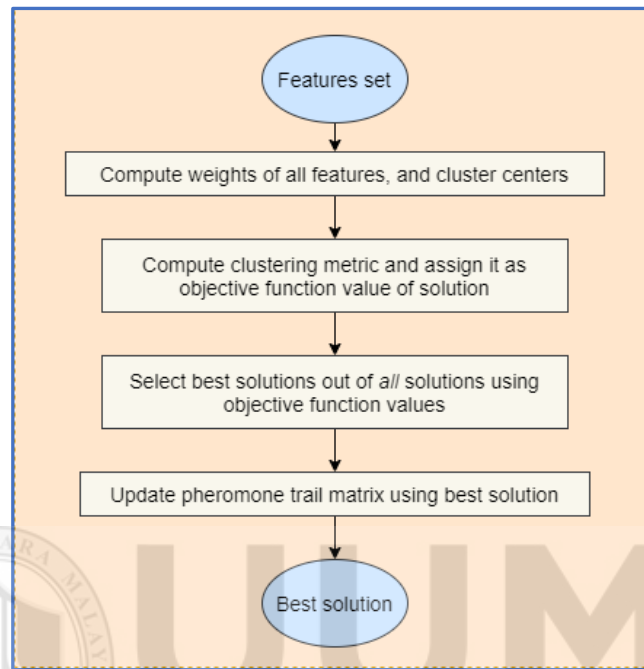
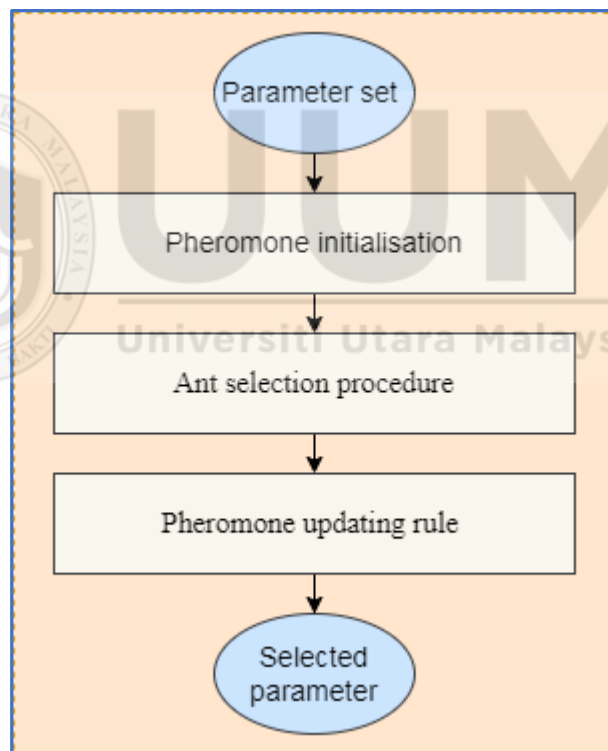


Figure 3.4: Proposed ACO algorithm for feature clustering

### 3.3.2 Adaptive Technique for Parameter Control

The adaptive technique is presented to determine the suitable threshold value based on the data. The proposed enhancement is related to the work of Ghimatgar et al. (2018) in the MGCACO algorithm that uses a fixed threshold value. Therefore, an ACO algorithm is used in this study to optimise the value of the parameter,  $\epsilon$ . Thus, developing an ant selection technique adaptively based on the heuristic and pheromone information. As shown in Figure 3.5, the ACO algorithm follows the procedures of pheromone initialisation, ant selection procedure, and pheromone updating rule to control the selection of  $\epsilon$  value. The ant selection procedure is used to simultaneously

select the  $\varepsilon$  value based on either the amount of pheromone or randomization. Stochastically, the selection of a single parameter value has been done by ACO. Subsequently, during the learning process of the proposed algorithm, the quality of each discovered feature subset is gathered (i.e., to be used as feedback) as a result of the search behaviour. Then, to guide the search process, these feedback indicators are used for pheromone updating. The transition probability of ACO is utilised to decide whether or not the current  $\varepsilon$  value has to be dominated. The process is then repeated in the learning process of the proposed algorithm.



*Figure 3.5: Adaptation technique of a threshold value*

### 3.3.3 Genetic-based Subset Determination

In general, the method aimed at increasing classification accuracy while ensuring the selected feature subset size is kept as low as possible, which requires the determination of the optimum subset size in any dataset. With the different sizes of datasets, the automatic finding of the appropriate subset calculated using the quality is more effective than the parameter specified by the user. Therefore, this study adopts the GA algorithm (i.e., initial population and crossover) to determine the final subset based on the crossover and subset quality calculation. The subset selection processes entail crossover (i.e., exchanges) among a generated temporary array and the current original features array. Specifically, a single random-point crossover is utilised to swap the features among the arrays (the generated temporary array and the feature array), as illustrated in Figure 3.6.

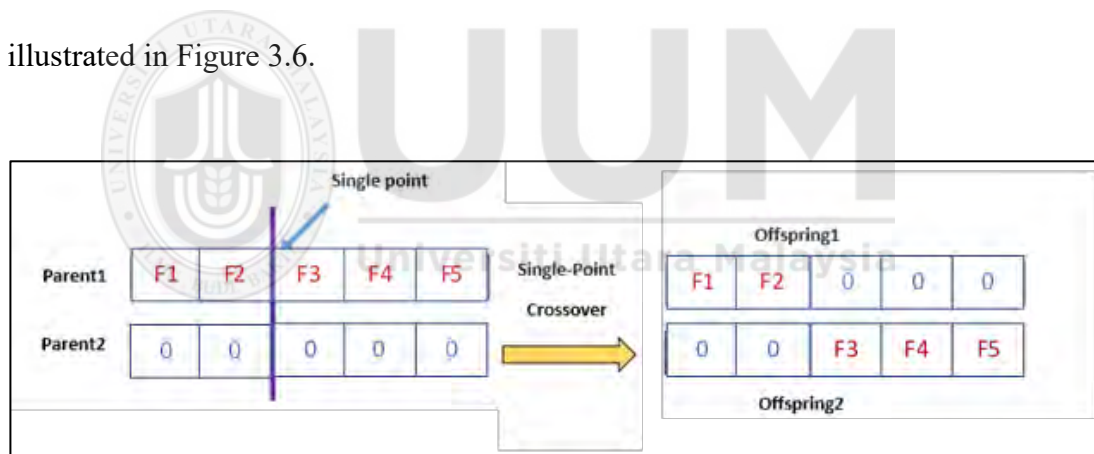
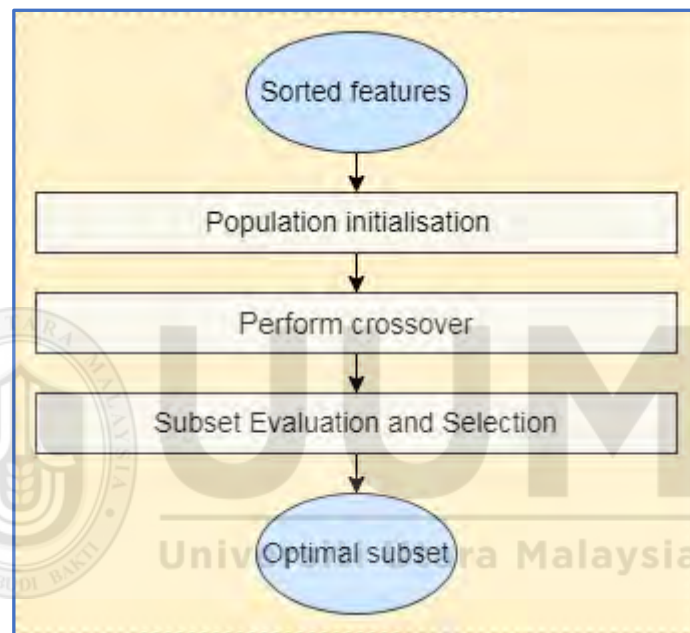


Figure 3.6: Single point crossover

In this process, a random point is generated. All the features after the generated point will be crossed from the first parent to the second parent which creates two offspring. The first offspring contains features that have the highest amount of pheromone while the second offspring contains the rest of the features that have the lowest amount of pheromone. The quality of the new offspring (i.e., feature subset) is evaluated and

compare with the quality of the previous feature subset (i.e., parent). If the new subset's quality is higher than the previous subset's, the former will take the place of the latter. Then, this procedure is repeated until the termination condition is satisfied. To optimise the optimal feature subset using the GA algorithm, a block diagram schema is employed as shown in Figure 3.7. Notably, the proposed method requires predefined size of target subsets in any dataset that will be used.



*Figure 3.7: Process of GA-based subset optimisation*

### **3.4 Performance Evaluation**

Evaluation of the proposed algorithm EGCACO is carried on 18 UCI and nine (9) microarray datasets with different features size. Several experiments that measure the classification accuracy are conducted to achieve a subset that represents the final features. Generally, in evaluating the efficiency of the FS algorithms, the measurement of the classification accuracy is utilised.

To achieve stable estimates and relative accuracy, 10 independent runs are implemented to obtain the classification accuracy for the UCI datasets and, in each experiment, holdout method with 80% of the normalized datasets is used for the training set and 20% of the datasets are for the testing set. The training set is utilised to determine the final feature subset, while the test set is to evaluate the accuracy of the selected features. The proposed EGCACO algorithm is compared with HSGW (Mafarja et al., 2020), RSGW (Mafarja et al., 2020), ASGW (Mafarja et al., 2020), BGWOPSO, WOA-CM (Mafarja & Mirjalili, 2018), bALO-QR and WOASA (Mafarja et al., 2020; Mafarja & Mirjalili, 2018).

In the experiment for the microarray datasets, the 10-fold cross-validation approach is used as adopted in the benchmarked algorithms (Manita & Korbaa, 2020) while 100 independent runs are employed. In the 10-fold cross-validation approach, the dataset is partitioned into ten (10) subgroups of the same size. The learning stage employs (9) nine groups, whereas the testing stage employs the remaining group. To ensure that all groups are utilised in both stages, the learning and testing stages are employed ten (10) times with a different group for testing. Additionally, the proposed EGCACO algorithm is compared with TGA (Too et al., 2018), GWO (Emary et al., 2016b), ASO (Too & Abdullah, 2020), PSO, HHO, GA, DE (Engelbrecht & Pampará, 2007), and BA (Al-betar et al., 2019).

The performance of all algorithms is also evaluated in terms of the number of selected features. In general, the least number of features for the same accuracy indicates the best algorithm. The nonparametric Friedman test is used in conjunction with the Holm post-hoc test to identify the best combination between the number of selected features

and the classification accuracy (Al-behadili et al., 2020a; Demsar, 2006; Liang et al., 2016; Salama & Freitas, 2012; Lei Yang et al., 2015, 2016). The average classification accuracy rank versus the average number of selected features rank for all algorithms is then obtained using the outcome of the nonparametric Friedman test with Holm's post-hoc test. The test is conducted to rank which algorithms performed best for each dataset. Good algorithm performance is indicated by a small rank value (Yang et al., 2016). The test is meant to determine the best algorithm that achieves a balance between the number of selected features and classification accuracy. As an illustration, algorithm A outperforms algorithm B if and only if the two criteria are met: first, A must not perform worse than B in terms of both metrics, i.e., the number of selected features and classification accuracy. The second criterion, A completely outperforms B in at least one objective (Otero et al., 2013a). As a result, algorithm A will only be the best algorithm if no other algorithm dominates it (Ghosh & Nath, 2004).

### 3.4.1 Dataset

Experiments are conducted on different benchmark classification problems (i.e., UCI and microarray datasets) available in the literature (Agrawal et al., 2021; Ghimatgar et al., 2018; Jaddi & Abdullah, 2021; Mafarja et al., 2020; Manbari et al., 2019b; Moradi & Rostami, 2015b) to demonstrate the efficacy of the proposed method. Table 3.1 summarizes the characteristics of the datasets. Detailed descriptions are available in the UCI machine learning repository (Bache & Lichman, 2013) while the microarray datasets are accessible in Li et al. (2017).

Missing values for each attribute are replaced with the mean of the available data (Theodoridis & Koutroumbas, 2009). A non-linear method of normalization, i.e., SoftMax Scaling (Theodoridis & Koutroumbas, 2009), was utilised to scale the datasets into the range [0 1].

Table 3.1

*Main dataset features to be used in the experiment*

No.	Dataset	No. of features	Sample Size	No. of classes	Domain of dataset	Size of datasets			
						Small	Medium	Large	Very large
1	Breastcancer	9	699	2	Biological	√	-	-	-
2	Tic-tac-toe	9	958	2	Game	√	-	-	-
3	Exactly	13	1000	2	Biological	√	-	-	-
4	Exactly2	13	1000	2	Biological	√	-	-	-
5	HeartEW	13	270	2	Biological	√	-	-	-
6	M-of-n	13	1000	2	Biological	√	-	-	-
7	WineEW	13	178	3	Chemistry	√	-	-	-
8	CongressEW	16	435	2	Politics	√	-	-	-
9	Vote	16	300	2	Politics	√	-	-	-
10	Zoo	16	101	6	Artificial	√	-	-	-
11	Lymphography	18	148	2	Biological	√	-	-	-
12	SpectEW	22	267	2	Biological	-	√	-	-
13	BreastEW	30	569	2	Biological	-	√	-	-
14	Ionosphere	34	351	2	Electromagnetic	-	√	-	-
15	KrVsKpEW	36	3196	2	Game	-	√	-	-
16	WaveformEW	40	5000	3	Physics	-	√	-	-
17	Sonar	60	208	2	Biological	-	-	√	-

	18	PenglungEW	325	73	2	Biological	-	-	-	√
	19	CLL_SUB_111	11340	111	3	Biological	-	-	-	√
	20	Colon	2000	62	2	Biological	-	-	-	√
	21	Leukemia	7070	72	2	Biological	-	-	-	√
Microarray	22	Lung	3312	203	5	Biological	-	-	-	√
	23	Lung_discrete	325	73	7	Biological	-	-	-	√
	24	Lymphoma	4026	96	9	Biological	-	-	-	√
	25	nci9	9712	60	9	Biological	-	-	-	√
	26	Prostate_GE	5966	102	2	Biological	-	-	-	√
	27	SMK_CAN_187	19993	187	2	Biological	-	-	-	√

The listed datasets vary in the instance numbers (from 60–5000), attribute numbers (from 9–19993), and class labels (from 2-9). The datasets are chosen from five (5) application domains to solve several challenging issues in different application domains effectively. The biggest size (70.37%) belongs to the biological domain, where (52.94%) from these datasets are related to the microarray application domains.

The size of the dataset can be divided into four categories based on the number of attribute: small, medium, high, and very high (Kudo & Sklansky, 2000). The least number of datasets (3.7%) are classified as having high number of attributes (between 50 and 100). While 40.74 % percent of the datasets are classified as small, with attributes ranging from 0 to 20, 18.51 % of the datasets have medium number of attributes (in the range of 20–50). The remaining datasets with the highest percentage (37.03%) are datasets that have 100 and more attributes.

### 3.4.2 Classifier and Evaluation Criteria

To show the generality of the proposed method, several well-known benchmark classifiers are used in the evaluation process (Agrawal et al., 2020; Ghimatgar et al., 2018; Guha et al., 2020; Long et al., 2021; Manbari et al., 2019b; Mohammadzadeh & Gharehchopogh, 2021; Moradi & Rostami, 2015b). The classifiers comprise Random Forest, k-NN, Decision Tree, and Support Vector Machine. These classifiers are implemented in the WEKA software package (Hall et al., 2009).

Accuracy is an effective way of validating FS methods using each classifier's performance (Aarabi et al., 2006). Therefore, it is utilised to evaluate the generated feature subsets and it is computed using equation (3.1) as follows:

$$AC = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

where  $TN$ ,  $TP$ ,  $FN$ , and  $FP$  are the confusion matrices that describe the classification results (true or false).

- $TN$ : The model predicted a negative value and the actual value is negative
- $FN$ : The model predicted a negative and result is false
- $TP$ : The model predicted a positive value and the actual value is positive
- $FP$ : The model predicted a positive, and result is false

Table 3.2 shows the possible cases in identifying the cases.

Table 3.2

*Confusion Matrices*

Predicted Values	Actual Values		
	Positive	Positive	Negative
Positive	TP	TP	FP
Negative	FN	FN	TN

However, accuracy does not give relevant information to determine the durability rate of the findings. Other performance measures that have been utilised to evaluate the proposed algorithm are precision, F-measure, and recall (Manbari et al., 2019b). F-measure is calculated using equation (3.2) as follows:

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3.2)$$

Recall is the number of TP divided by the sum of FN and TP. Precision is the number of TP separated by the number of FP and TP.

### 3.5 Summary

A framework has been introduced in this chapter to fulfill the main objectives. The ACO for clustering and automatic parameter adaptation technique are proposed as a new mechanism to enhance the performance of the MGCACO algorithm for dimensional reduction of several types of datasets including the microarray data. Additionally, different classifiers with evaluation criteria have been suggested to evaluate the proposed algorithm.

# CHAPTER FOUR

## PROPOSED ENHANCED GRAPH CLUSTERING WITH ANT COLONY OPTIMISATION ALGORITHM FOR FEATURE SELECTION

### 4.1 Introduction

The proposed EGCACO is based on the MGCACO algorithm which has three (3) main drawbacks. The first drawback is the feature clustering stage which is easily trapped into local optimum. Such a disadvantage leads to losing the diversification in the search (global search) which impacts the MGCACO algorithm and results in the production of a low-quality subset of features. To guarantee the precision of the feature's subset, a metaheuristic algorithm is utilised to provide intensification and diversification in the search process. The second drawback is the lack of setting (predefined) the appropriate threshold value for a parameter that is responsible for guiding the ACO selection of the features from the clusters. This drawback leads to not consider the relevance, dependency, and homogeneity among the features in the search spaces. Thus, an ACO-based adaptive feature selection technique is redesigned to maximize dependency and minimize redundancy among the features. Lastly, the MGCACO algorithm lacks the attempt to optimise the subset determination that is responsible for selecting the final subset of features where it still requires a predefined parameter. Therefore, this will not guarantee optimal performance in determining the accurate subset. Consequently, the automated selection of the final subset is carried out based on a characteristic of metaheuristic population-based optimisation, where the subset can be generated automatically rather than being static or specified by the user.

The proposed EGCACO algorithm is constructed based on the MGCACO algorithm which is a variant of the GCACO algorithm. The three drawbacks of MGCACO algorithm are also the drawbacks of GCACO (Ghimatgar et al., 2018; Moradi & Rostami, 2015b). Section 4.2 presents the enhanced clustering of the GCACO algorithm (EC-GCACO) based on the concepts and components of the ACO-based clustering method to improve the feature grouping quality. Section 4.3 introduces the adaptive GCACO algorithm (A-GCACO) as an FS algorithm with an adaptive technique for ant selection to construct the appropriate subset of features through the process of learning. Section 4.4 provides the genetic GCACO algorithm (G-GCACO) that determines the final subset automatically and overcomes the limitation of the existing technique. Section 4.5 illustrates on the combination of the aforementioned components to construct the proposed EGCACO algorithm. Finally, Section 4.6 summarizes the chapter.

## 4.2 ACO-based Features Clustering

The ACO-based features clustering reports the application of the ant algorithm concept that aims to achieve the optimal distribution of  $N$  features into one of the  $K$  clusters thus grouping highly correlated features into the same group. The ants' concept starts with strings representing an empty solution  $S$ , the length  $N$  (number of dataset features) where each element of string corresponds to one of the test features. Before the first iteration, the components of the pheromone matrix are all initialised to the same amounts. The elements of the pheromone matrix are updated as iterations proceed, based on the goodness of solutions produced. To construct solution  $S$ , the ant chooses a cluster number for each feature of string  $S$  in one of the following ways:

The cluster with the highest pheromone concentration is selected using probability  $q_0$ , where  $q_0$  is a predefined constant number in the range [0 1], and/or one of the  $K$  clusters utilizing a stochastic distribution and a probability of  $(1 - q_0)$ , indicated as  $p_{ij}$ .

The latter is referred to as biased exploration, while the first process is referred to as exploitation (Gambardella & Dorigo, 2000). The clusters with the highest pheromone concentration will be chosen in the first procedure if the random numbers  $q$  (i.e.,  $q$  is equal to the length of the solution string and is generated at random from a uniform distribution in the range [0 1] ) corresponding to these features are less than  $q_0$ . On the other hand, if the features' corresponding random numbers  $q$  are greater than the  $q_0$ , the features will be allocated to one of the clusters using the second procedure with a normalized pheromone probability to 1 given by (Shelokar et al., 2004):

$$p_{ij} = \frac{\tau_{ij}}{\sum_{k=1}^k \tau_{ik}}, j = 1, \dots, k \quad (1)$$

where  $p_{ij}$  denotes the normalized pheromone probability for feature  $i$  within cluster  $j$ . The value of the objective function for a given feature clustering is used to measure the quality of the solution constructed. The sum of squared Euclidean distances among each feature and the centre of the belonging cluster is utilised to determine this objective function. In  $\mathcal{R}^n$  dimensional space consider a given dataset of  $N$  features  $\{x_1, x_2, \dots, x_n\}$  to be partitioned into a number,  $K$  comprises groups or clusters. The mathematical formulation of the feature clustering can be defined as (Shelokar et al., 2004):


$$\text{Min } F(w, m) = \sum_{j=1}^k \sum_{i=1}^N \sum_{v=1}^n w_{ij} \|x_{iv} - m_{jv}\|^2 \quad (2)$$

in such a manner that

$$\sum_{j=1}^k w_{ij} = 1, i = 1, \dots, N \quad (3)$$

$$\sum_{i=1}^N w_{ij} \geq 1, j = 1, \dots, K \quad (4)$$

where  $x_{iv}$  denotes as the value of  $v$ th attribute of  $i$ th feature,  $m$  is the centre of cluster matrix of size  $K \times n$ ,  $m_{jv}$  indicates the mean of the  $v$ th attribute values of all features within cluster  $j$ ,  $w$  is the weight matrix of size  $N \times K$ ,  $w_{ij}$  denotes as correlated weight of feature  $x_i$  with cluster  $j$  which can be assigned as:



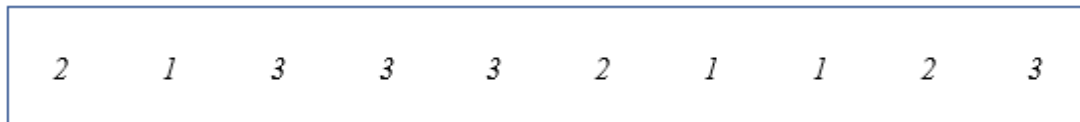
$$w_{ij} = \begin{cases} 1 & \text{if feature } i \text{ is within in cluster } j \\ 0 & \text{otherwise} \end{cases}$$

$i = 1, \dots, N, j = 1, \dots, K$

After getting  $w_{ij}$ 's, each cluster centre  $m_j$  can be obtained as (Shelokar et al., 2004):

$$m_{jv} = \frac{\sum_{i=1}^N w_{ij} x_{iv}}{\sum_{i=1}^N w_{iv}}, j = 1, \dots, K, v = 1, \dots, n \quad (5)$$

Thus, Equation (2) can be used to calculate the fitness value (objective function) of a given solution string  $S$  knowing the centre of the cluster matrix  $m$  and weight matrix  $w$ . For example, Figure 4.1 shows  $S1$  constructed for  $N = 10$  for and  $K = 3$  as a representative of a solution string.



*Figure 4.1: Solution representation*

String cluster number 2 is assigned to the first feature and cluster number 1 is assigned to the second feature and so forth. Since the heuristic information of the features is not available, the local search method will enhance some of the obtained solutions. For this purpose, on a few percent of  $L$  solutions (best 20% percent of the total solutions), a local search procedure is applied. Thus, to conduct local search, the fitness values of the selected subset of the population are sorted in ascending order. Then, on the highest  $L$  solutions in terms of fitness values (lowest values in terms of objective function), a simple local search procedure is implemented. During the local search procedure, the cluster number of each feature in the solution string is changed with a certain threshold probability in the range  $[0, 1]$ . Thus, a random number is generated between  $(0,1)$  for each feature, then if the random number corresponding to the feature is less than the threshold probability, the feature is assigned a different cluster number with similar probability to the rest of the clusters through generating a random number. For illustration Figure 4.1 is referred. For the threshold  $p = 0.01$  and using the topmost solution string, the generated random value that corresponds to the sixth feature should

be less than the threshold which indicate cluster number 2. Therefore, it has to be assigned to either cluster number 1 or 3 with equal probability by generating a random number. After the local search is completed, the fitness values for the newly created solutions are obtained using Equation (2), where only certain solutions that have an improvement in fitness will be accepted and replaced. Figure 4.2. displays the procedure of the local search method for ACO-based feature clustering.

With the probability threshold of the local search  $p_{ls}$  in  $[0, 1]$ , a neighbor of  $S_k$ ,  $k = 1, \dots, L$  is produced as:

- 1  $k = 1$
- 2 Generate temporary solution  $S_t$ ,  $S_t(i) = S_k(i)$ ,  $i = 1, \dots, N$ .
- 3 For each feature  $i$  of  $S_t$
- 4 if  $r \leq p_{ls}$  //  $r$  a random number in  $(0, 1)$
- 5 Randomly select  $j$  in the range  $(1, \text{number of clusters})$ ,  $S_k(i) \neq j$ ,  $S_t(i) = j$
- 6 Calculate cluster centres of  $S_t(i)$ , Eq. (16)
- 7 Calculate Fitness of  $S_t(i)$ , Eq. (13) as  $F_t$
- 8 If  $F_t < F_k$
- 9  $S_k(i) = S_t(i)$  and  $F_k < F_t$
- 10  $k = k + 1$ ; If  $k \leq L$  go to step (2), else stop
- 11 If  $k \leq L$  go to step (2), else stop

Figure 4.2: Pseudocode of the local search for ACO-based feature clustering method

The pheromone matrix is updated after the local search procedure is completed. The usefulness of dynamic information generated by the ants' search mechanism is reflected in this pheromone updating rule. As a result, the pheromone updating rule is an adaptive memory, storing information obtained by previously discovered superior solutions and updating it at the end of each iteration. Thus, at iteration level  $t$ , the pheromone updating rule is implemented to the best  $L$  solutions out of all solutions found by the ants based on the given criteria (Equation. (2)). By assigning some real numbers  $\tau_{ij}$  correlated with solution attributes, these  $L$  ants simulate the pheromone

trail deposition of real ants. The pheromone trail is updated utilizing the following rule (Shelokar et al., 2004):

$$\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \sum_{l=1}^L \Delta\tau_{ij}^l$$

$$i = 1, \dots, N, j = 1, \dots, K \quad (6)$$

where the  $\rho$  value is between [0 1],  $(1 - \rho)$  indicates the pheromone decay parameter in order to avoid infinite aggregation of a specific parameter value,  $\Delta\tau_{ij}^l$  is the additional pheromone boost and it is equal to  $1/F_l$ , if cluster  $j$  is allocated to  $i$ th feature of the solution produced by ant one and zero otherwise. An optimal solution is one that minimizes the value of the objective function. In each iteration, if the solution produced as "current iteration best solution" has a lower objective function value than the best solution in memory, the value of the best solution in memory is updated. Consequently, the algorithm basically executes three steps at every iteration stage. (1) The ants construct new solutions by using the updated pheromone trail information obtainable from the beforehand iteration, (2) executing a local search on the newly constructed solutions, and (3) performing the pheromone updating rule. Until a specified number of iterations, these three steps are repeatedly carried out. Then, the solution with the highest fitness value (lowest objective function value) takes the place of the optimal separating of features in a particular dataset into various clusters. Until a specified number of iterations, these three steps are repeatedly carried out. Then, the solution with the highest fitness value (lowest objective function value) takes the place of the optimal separating of features in a particular dataset into various clusters. The pseudocode depicted in Figure 4.3 is the ACO-based feature clustering method.

In this pseudocode, the number of clusters,  $K_{clusters}$ , is fixed and assigned to 3, and the size of the population,  $Population_{size}$ , is 50.

```
1 Input:  $Algorithm_{parameters}$ ,  $K_{clusters}$ ,  $Data_{features}$ 
2 Output:  $S_{bestClustering}$ 
3 Initialise  $Population$ ,  $Pheromone$  matrix
4 While  $\neq StopCondition()$  do
5   For  $i = 1$  to  $Population_{size}$ 
6     Construct solution using pheromone trail
7     Compute weights of all test features, and cluster centres
8     Compute clustering metric and assign it as objective function value of
       solution
9   End For
10  Select best solution out of all solutions using objective function values
11  Compute the local search on the selected best solutions //(Figure 4.2)
12  Update pheromone trail matrix using best solution
13 End While
```

Figure 4.3: Pseudocode of ACO-based feature clustering method

### 4.3 Adaptive Feature Selection Technique

The adaptive selection technique reports the application of ACO concept which collects the feedback from the subset selection process to adapt the threshold of picking the features from the clusters in the MGCACO. Thus, the threshold value called  $\varepsilon$ , can be automatically determined rather than being determined by the user. A state transition probability function is used to select the  $\varepsilon$  value based on the amount of pheromone and heuristic. Stochastically, the selection of a single parameter value is conducted by ACO. Subsequently, during the learning process of the proposed algorithm, the quality collected for each discovered feature subset and the quality of

the significant feature subset are determined as feedback collection. Then, to guide the search process, these feedback indicators are transformed into reward assignments.

To determine whether the current  $\varepsilon$  value is dominated or not, the transition probability of ACO is utilised. Consequently, significant features that can maximize dependency and minimize redundancy among the features are chosen by the ants based on the value of a parameter,  $\varepsilon$ , to be included in the feature subset. Three steps are comprised in the ACO-based selection process to enhance the selection of the appropriate  $\varepsilon$  value to guide the subset construction, which is the pheromone initialisation, ant selection, and pheromone updating rule as described below.

**Pheromone initialisation:** To establish the feature selection process, the pheromone initialisation requires two vectors to implement the feedback collection of EGCACO. The first vector represents the quality of the threshold parameter values, referred to as  $\tau p_n = \{ \tau p_1, \tau p_2, \tau p_3, \dots, \tau p_n \}$ , which reflects the different  $\varepsilon$  values initiated using the proposed equation.

$$\tau p_n = \emptyset \quad (7)$$

where  $\emptyset$  is a constant number at time  $t = 0$  that can be set to any value in the range [0 1]. The second vector denotes the quality amount assigned to each feature and is represented as  $\tau f_n = \{ \tau f_1, \tau f_2, \tau f_3, \dots, \tau f_n \}$ . At time  $t = 0$ , the pheromone initialisation is determined by performing the relevance of attributes to classes. Higher priority is given to more relevant features. Therefore, the MI among any feature ( $f_a$ ) and classes is computed for this purpose as in (Ghimatgar et al., 2018):

$$MI(d) = \sum_m \sum_{k=1}^c p(F_d[m].k) \log \frac{p(F_d[m].p(k))}{p(F_d[m].k)} \quad (8)$$

where  $F_d[m]$  is the discretizing value of the  $d$ th feature for bin  $m = [1, 2, \dots, M]$ , in which  $M$  (number of bins) is set to 10 in the MGCACO algorithm. The probability of the  $k$ th class is  $p(k)$  and the number of classes is  $k$ .

**Ant selection:** When the ACO algorithm starts, two selection stages will be implemented. In the first stage, each ant selects a parameter (node) that represents its direction in time  $t$ , which is determined by the proposed probability Equations (9) and (10).

$$P_{(an)} = \begin{cases} \arg \max_{an \in P} \text{if } q < q_0 \\ 0, \text{ otherwise} \end{cases} \quad (9)$$

where  $P_{(an)}$  is denoted as the probability for each parameter value and  $P_{(an)} = \{a_1, a_2, a_3, \dots, a_n\}$  determines the selection probability of each  $\varepsilon$  value.  $q$  is the parameter in the range [0 1]. The predefined constant,  $q_0$ , is a random number. In the probabilistic method, the ant selects the next parameter with a new probability of  $P_n$  calculated as follows:

$$P_n = \begin{cases} \frac{[\tau p_{n(t)}]^\alpha [\pi_n]^\beta}{\sum_{n=1}^L [\tau p_{n(t)}]^\alpha [\pi_n]^\beta} \text{ if } q > q_0 \\ 0, \text{ otherwise} \end{cases} \quad (10)$$

where  $\alpha$  and  $\beta$  represent the significant predefined constant parameter value in the range [0 1] of both the pheromone and the heuristic information, respectively.  $[\tau p_{n(t)}]$

is the quantity of quality correlated with each threshold value, while  $[\pi_n]$  indicates the heuristic information that is calculated for this purpose as:

$$\pi_n = \frac{1}{P_{(v_n)}} \quad (11)$$

where  $P_{(v_n)}$  indicates the threshold value for each parameter and  $P_{(v_n)} = \{v_1, v_2, v_3, \dots, v_n\}$  determines the initialisation of each  $\varepsilon$  value between [0 1]. Thus, according to Equation (10), one of the parameters is selected corresponding to its probability value by using roulette wheel selection.

In the second stage, each ant begins to construct the feature subset from a group of clusters that are generated based on the Louvain algorithm for community detection. As a result, in each iteration of a completely connected undirected graph, the  $j$ th ant picks at random at least one attribute from the cluster in its route. Then, the ant chooses the next feature from an unique cluster by using probabilistic decision rules based on both probability and greedy process with the dependency on the chosen parameter  $\varepsilon$  value (produced in the first stage); otherwise, the ant remains in the current cluster. If  $q_0$  is less than  $q$ , then the  $j$ th ant chooses the next feature by following the greedy method as (Moradi & Rostami, 2015b).

$$F_{next} = arg \max_{F_d \in UF_m^j} \{ \tau f^i(F_d)^\alpha \cdot \pi(F_d, VF_j)^\beta \} \quad (12)$$

where  $UF_m^j$  is the set of features that has not yet been reached by the  $j$ th ant in the existing cluster ( $m$ th cluster),  $\tau f^i$  is the pheromone quantity value correlated with feature ( $F_d$ ),  $VF_j$  indicates the previously selected features (visited features), and

$\pi(F_d, VF_j)$  denotes the heuristic information function which is defined as follows (Ghimatgar et al., 2018) :

$$\pi(F_d, VF_j) = F - Score(F_d) - \frac{1}{N_{VF_j}} \sum_{m=1}^{N_{VF_j}} w(F_d, F_m) \quad (13)$$

where  $N_{VF_j}$  is the size of  $VF$ ,  $w(F_d, F_m)$  indicates the values of the Pearson correlation (Saidi et al., 2019) among feature  $F_d$  and feature  $F_m$  that are obtained from all the features ( $VF_j$ ) visited by the  $j$ th ant from all priors clusters computed as:

$$w(F_d, F_m) = \left| \frac{(F_d - \bar{F}_d) - (F_m - \bar{F}_m)}{\sqrt{(F_d - \bar{F}_d)^2 (F_m - \bar{F}_m)^2}} \right| \quad (14)$$

Where  $\bar{F}_d$  and  $\bar{F}_m$  denote the vectors of features,  $F_d$  and  $F_m$  are completely uncorrelated if  $w(F_d, F_m)$  has a value equal to 0 or correlated if  $w(F_d, F_m)$  has a value equal to 1. The similarity values among features are identical to each other in most cases. Thus, to reduce the influence of outliers without excluding them from the distribution of weights a nonlinear normalization method named SoftMax Scaling is utilised to scale the weight  $w(F_d, F_m)$  of the edge to the range [0 1] as follows (Theodoridis & Koutroumbas, 2009):

$$\hat{w}_{dm} = \frac{1}{1 + \exp\left(\frac{w_{dm} - \bar{w}}{\sigma}\right)} \quad (15)$$

Where  $\hat{w}_{dm}$  denotes the normalized value of  $w_{dm}$  between  $F_d$  and  $F_m$  features,  $\bar{w}$  and  $\sigma$  are the mean and variance, respectively, of all  $w_{dm}$  values.

In addition,  $F - Score(F_d)$  in Equation (13) denotes the score of feature  $F_d$  based on its relevance to the classes (Gu et al., 2011), calculated as follows:

$$F - Score(F_d) = \frac{\sum_{k=1}^K n_k \cdot (\mu_k^d - \mu^d)^2}{\sum_{k=1}^K n_k \cdot (\sigma_k^d)^2} \quad (16)$$

where  $K$  indicate the classes number of the dataset,  $\mu_k^d$  and  $\sigma_k^d$  are the variance and the mean of the  $k$ th class with  $n_k$  samples ( $n_k$  is the samples number in class  $k$ ), respectively,  $\mu^d$  denotes the mean of the patterns in the  $d$ th feature vector. The F-Score values are normalized by the Softmax scaling (Theodoridis & Koutroumbas, 2009) in ranges 0 and 1 as in Equation (15). Features with larger F-Score have a higher ability of discrimination power.

Therefore, in the greedy method, the features picked by the ants are those with maximum dependence and minimum parity to the features already selected on the target class.

On the other hand, If  $q_0$  is larger than  $q$ , a probabilistic method for each of the features not visited yet in the current cluster ( $F_d \in UF_m^j$ ) is defined as follows (Moradi & Rostami, 2015b):

$$P(F_d) = \frac{[\tau f^i(F_d)]^\alpha [\pi(F_d, VF_j)]^\beta}{\sum_{F_d \in UF_m^j} [\tau f^i(F_d)]^\alpha [\pi(F_d, VF_j)]^\beta} \text{ For } F_d \in UF_m^j \quad (17)$$

where  $\alpha$  and  $\beta$  define the significant predefined parameter value in the range [0 1] of both the pheromone and the heuristic information, respectively.  $\tau f^i(F_d)$  is the quantity

of quality correlated with each feature  $F_d$ , while  $\pi(F_d, VF_j)$  indicates the heuristic information that is calculated by Equation (13). Thus, according to the roulette wheel rule, the next feature would be chosen. Figure 4.4 shows the search for the best feature subset from the clusters using ACO.

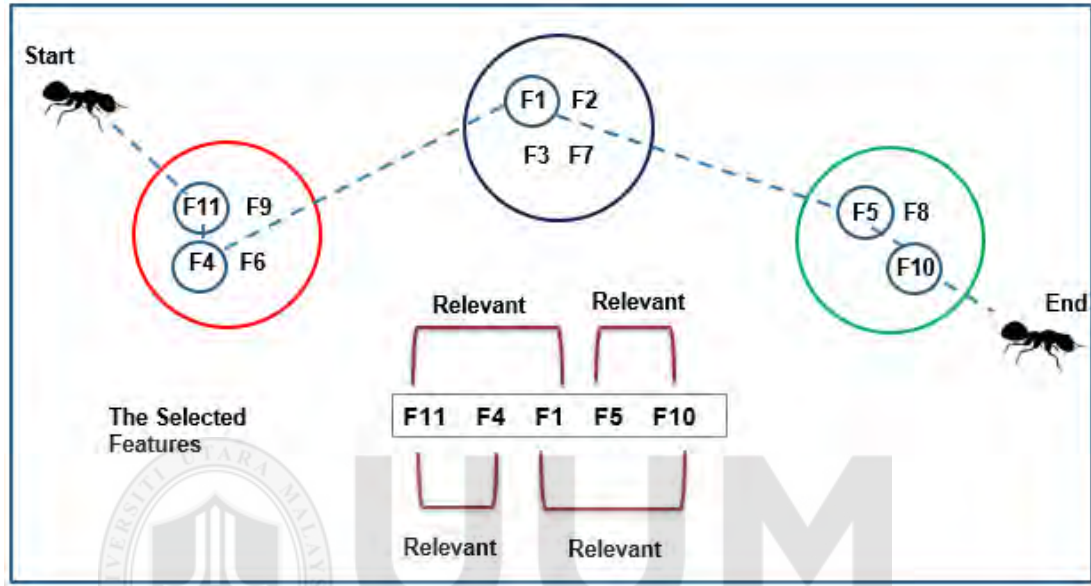


Figure 4.4: Ant selection process.

**Pheromone updating rule:** Once all the ants have executed their routes on the graph, and before ending each iteration  $i$ , the quantity of pheromone of each selected parameter and feature (*i. e.*,  $\tau p_n(t), \tau f^i(f_d)$ ) is updated in two stages. In the first stage, the proposed updating rule of the selected parameters is calculated as:

$$\tau p_{x(t+1)} = (1 - \rho) \cdot \tau p_{x(t)} + \sum_{k=1}^A \Delta_k^i(t) \quad (18)$$

where the  $\rho$  value indicates the pheromone decay parameter to escape infinite accumulation with a particular parameter value,  $\tau p_{x(t)}$  and  $\tau p_{x(t+1)}$  demonstrate the

pheromone amounts on parameter  $P_{(n)}$  at times  $(t)$  and  $(t + 1)$ , respectively.  $A$  is the number of ants,  $\Delta_k^i \tau p_{x(t)}$  is the additional pheromone boost to parameter  $P_{(n)}$  by ant  $k$ , which is computed from the use of MDA as follows:

$$\Delta_k^i(t) = \frac{\operatorname{argmax}_{t \in MDA_{k=1}^A}}{MDA_n} \quad (19)$$

In the second stage, the quantity of pheromone of each feature is updated by implementing the updating rule below (Moradi & Rostami, 2015b):

$$\tau f^{i+1}(F_d) = (1 - \rho) \cdot \tau f^i(F_d) + \sum_{j=1}^A \Delta_j^i(F_d) \quad (20)$$

where the  $\rho$  value indicates the pheromone decay parameter to escape infinite accumulation with a particular parameter value,  $\tau f^i(F_d)$  and  $\tau f^{i+1}(F_d)$  demonstrate the pheromone amounts on feature  $F_i$  at times  $(t)$  and  $(t + 1)$ , respectively,  $A$  is the number of ants,  $\Delta_j^i(F_d)$  is the additional pheromone boost to feature  $F_i$  by ant  $j$ , which is computed from the use of MDA (Bihl et al., 2016) as follows:

$$\Delta_j^i(F_d) = \begin{cases} \gamma_j^i, F_d \in FS_j^i \\ 0, F_d \notin FS_j^i \end{cases} \quad (21)$$

where  $FS_j^i$  is the feature chosen in the  $i$ th iteration by the  $j$ th ant,  $\gamma_j^i$  is denoted as MDA (Bihl et al., 2016) corresponding in the  $i$ th iteration to the  $j$ th chosen subset. Suppose  $N_{f_s}$  features should be selected. At each iteration, all the features are sorted using Equation (22) (Ghimatgar et al., 2018).  $N_{f_s}$  features are chosen and the MDA values identical to these features are computed using Equation (23) (Ghimatgar et al.,

2018). If the MDA values are enhanced relative to the previous iteration, the pheromone values are permitted to be updated; otherwise, no change is made.

$$Sf(d) = \tau f(d).Fscore(f_d) \quad d = 1, 2, \dots, D \quad (22)$$

$$Fitness = MDA(Feature(1, \dots, N_{fs})) \quad (23)$$

The processes (ant selection, pheromone updating rule) are repeated until the iterations are terminated. Lastly, the features are ordered in descending order based on their pheromone values in order to determine the final subset. The number of features that will be selected is calculated by multiplying a fixed number of features,  $\omega$ , by the number of clusters,  $K$ . Figure 4.5 provides a low-level description of the A-GCACO technique.

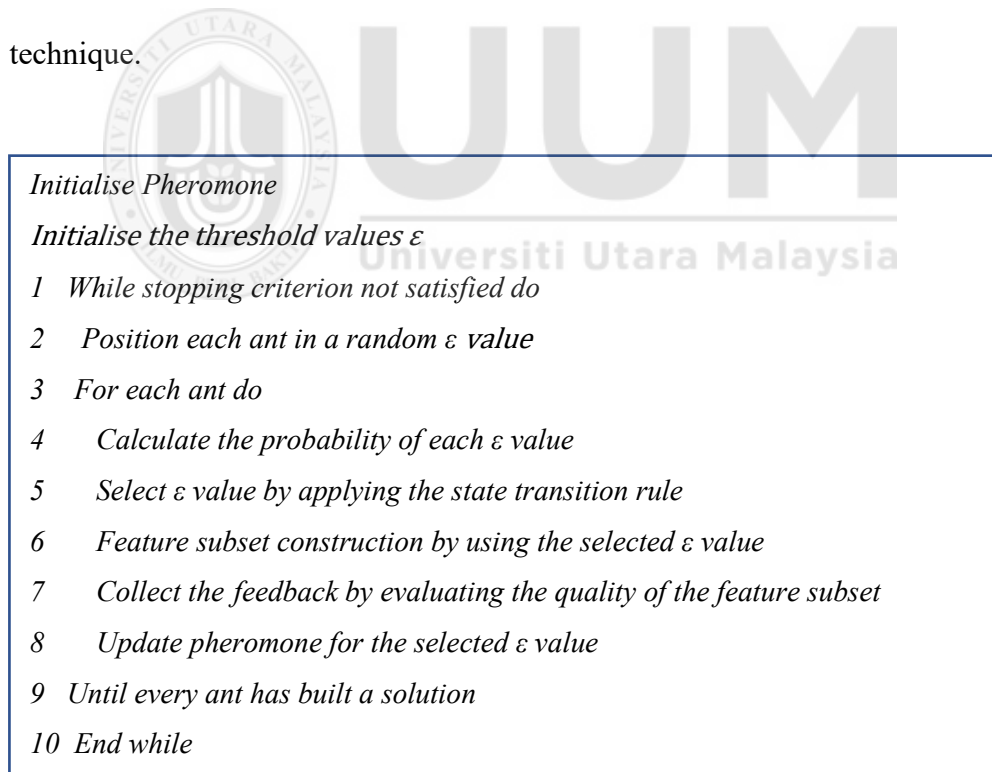


Figure 4.5: Pseudocode of A-GCACO technique

#### 4.4 Genetic-based Subset Determination

The main aim of this section is to define a new method that uses the GA principle to automatically determine the final subset. Genetic algorithm is because this algorithm does not require an initial guess, instead it requires only prior knowledge of the physical properties of the system, which is not bias to the final result (Ganatra et al., 2011). The proposed method uses two algorithmic components (initial population and crossover). The aim of the enhancement is to obtain relevant features that will reduce the number of selected features in the final subset while maintaining classification accuracy. All of the components of the G-GCACO remain the same (as discussed in the previous section), except for determining the final subset, which would incorporate population initialisation, crossover, feature list updating, subset quality calculation, and stopping criteria.

The MGCACO processes feature selection on a one-dimensional array that has a size equal to the number of features in the dataset, where it will divide these features into several clusters. Then in each iteration, each ant generates a subset as a sub-one-dimensional array consisting of at least one feature from each cluster until all the clusters are visited as a stopping criterion. Each ant carries a different number of features. So the size of the ants is different. Once all of the ants have completed their travels on the graph, and before the completion of each iteration  $i$ , each generated subset is evaluated using MDA and the quantity of pheromone of each subset of feature is updated. Subsequently, the whole features in the one-dimensional array are listed in decreasing order, depending on the pheromone values. Lastly, the MGCACO generates the final subset as an integer obtaining it by multiplying the number of

clusters,  $K$ , by a fixed number of features,  $\omega$ . Figure 4.6 shows an example of the original procedure of the final subset determination on a wine dataset that consists of 13 features.

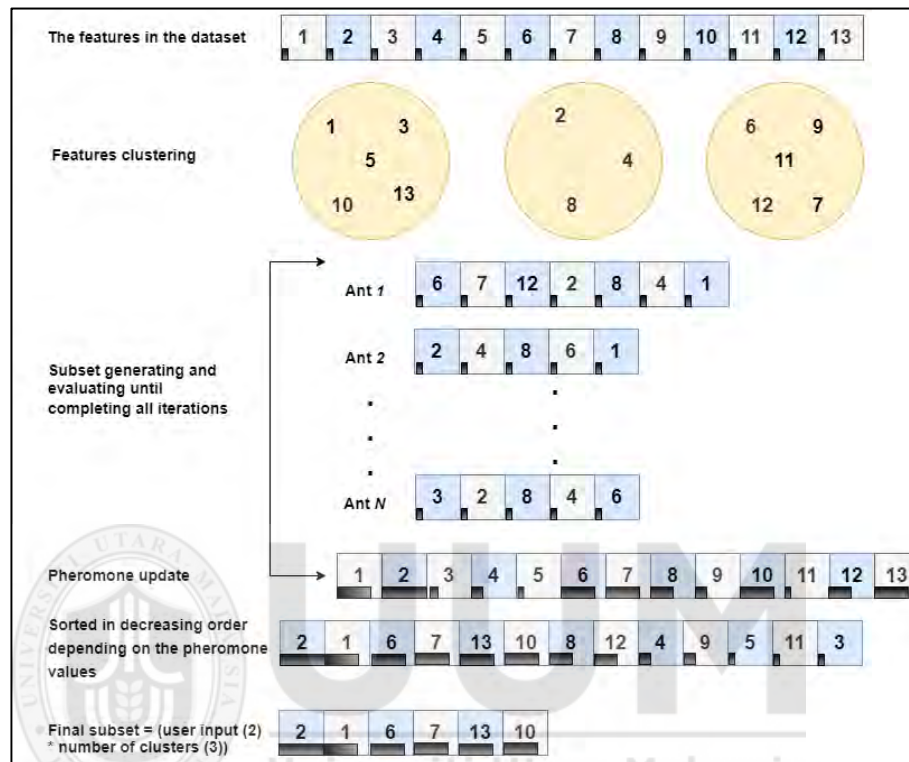


Figure 4.6: Example of the final subset determination using user assign parameter

The steps in the method for subset determination are i) determination of initial population, ii) crossover and iii) subset selection and evaluation.

## Initial population

The aim of the initial population procedure in the proposed genetic-based subset determination is to construct a new temporary one-dimensional array (features) with the same size as the dataset's original features array, as shown in Figure 4.7. (e.g., wine dataset consists of 13 features), thus, the additional array size=13. The population size is fixed (i.e. two arrays), and the process for determining the final subset selects the sorting features in these two arrays to enhance their performance. This represents the exchanges (crossover) between the same features in the original features array (first parent) and another temporary array (second parent). The temporary array contains 0 value which indicates the parent (i.e., temporary array) is not assign to any feature index. In this way, the initial population of the genetic-based subset determination technique is initialised.

To produce offspring chromosomes, two arrays (i.e., two parent chromosomes) are combined. The procedure of selection is repeated until the termination condition is satisfied (number of iterations). The subset is determined implicitly by the crossover between the two chromosomes in this loop, where the features with the highest pheromone quantity is selected. The crossover operator refers to the process through which parent chromosomes (features) exchange genetic information in order to generate the optimal feature subset.

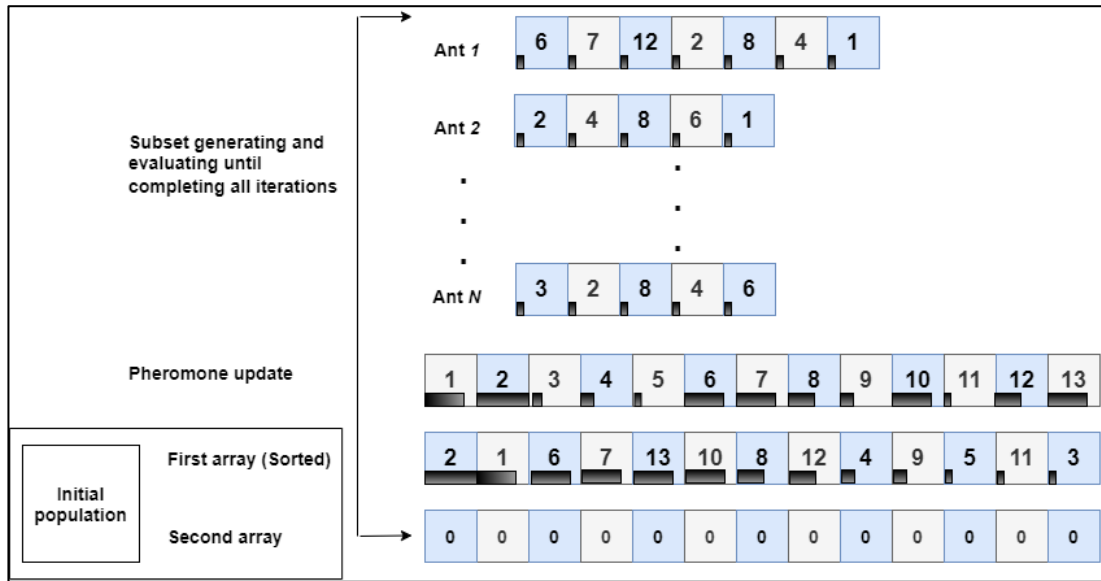


Figure 4.7: Initial population procedure of the G-GCACO

### Crossover

Figure 4.8 shows the proposed crossover pseudocode. In addition, different methods are utilised to exchange genetic information between two individuals. In this study, single random-point crossover procedure (Kora & Yadlapalli, 2017) were utilised on the basis of Equation (24):

$$CP_{i...n} = Rand(L_{i..n}/2) + 1 \quad (24)$$

where  $CP_{i...n}$  shows the crossover point,  $Rand(L_{i..n}/2)$  is a random value in the range  $[1, L_{i..n}/2]$ ,  $L_{i..n}$  indicates the size of the all features in dataset.

```

1 For each listed features set
2   FirstValue = GetFirstPoint Equation (24);
3   SecondValue = GetSecondPoint Equation (24);
4   MiddleSection = Min (FirstValue, SecondValue);
5   CrossedChild1 = Parent1(1: MiddleSection);
6   CrossedChild2 = Parent2(MiddleSection: end);
7 End For

```

Figure 4.8: Crossover operator pseudocode

The difference between this proposed pseudocode and the basic crossover is that two random values are generated according to Equation (24) instead of one value randomly generated in the basic crossover (Leloup et al., 1997). Then, the minimum value among these two values is obtained which will represent the middle section for exchange. The advantage of the minimum value is to decrease the number of the selected features in the final subset. Thus, all the features after the middle section are crossed from the first parent (first array) to the second parent (second array) which produce the offspring, i.e., child 1 with features that have the highest pheromone and child 2 with the rest of the features that have the lowest pheromone. An example of a single random-point crossover operator used in the subset determination technique on the wine dataset is shown in Figure 4.9. In this figure, one of the parent's array contains the values zero. This concept is proposed in Al-Behadili et al. (2021), Al-behadili et al. (2020b), Demestichas et al. (2000) and Podgorelec et al. (2000). Assume that the random values are 5 and 7, respectively, and the minimum value among these two random values will be 5 which represents the middle section for the cross (i.e., single random-point).

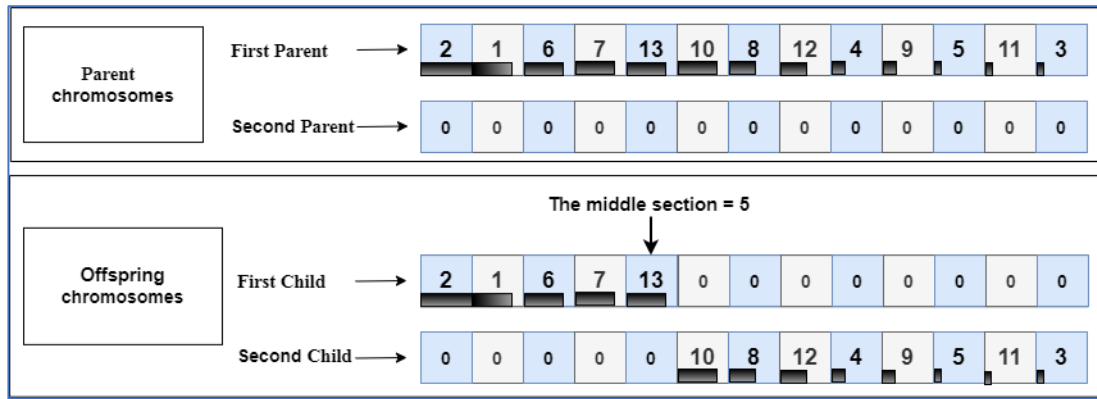


Figure 4.9: Example of crossover operation with single random-point and the minimum value as the middle section

### Subset Evaluation and Selection

In the subset evaluation, a multi-objective function known as the similarity mean fitness function is proposed. This function attempts to reduce the number of selected features while maintaining high classification accuracy. The accuracy and number of features provided in a solution are used to determine the fitness of a candidate solution. The accuracy is calculated using the separability index/Mean Decrease in Accuracy (MDA) as in Equation (21), while the fitness value for the proposed multi-function is computed utilizing the formula:

$$\text{similarity mean fitness function} = \text{Accuracy} \times \left( \frac{1}{N_F} \times \text{Mean}(\text{sim}(F_1, \dots, F_n)) \right) \quad (25)$$

where  $N_F$  is the size of the subset (number of selected features),  $\text{sim}(F_1, \dots, F_n)$  indicates the values of the Pearson correlation among all features in the subset. Each subset of feature (child) contains a list of features. Thus, according to the similarity mean fitness function, the proposed selection process will follow one of two rules as

shown in Figure 4.10. The first rule states that “if the new accuracy of the subset is higher than the previous subset and the number of features is less than the previous subset”, then the subset is the best. Otherwise, the previous subset will remain as the best subset. In the second rule, if two subsets have the same accuracy but different number of features, the subset with the fewer features will be chosen. This proposed selection process differs from the selection process of Leardi et al. (1992) where instead of considering only accuracy in selecting the best subset, the proposed process takes into account both the number of selected features and the accuracy according to Equation (24).

```

1  For each Child
2    IF Quality (New Child) >Quality (Old Child)
3      best = New Child
4    Else IF Quality (New Child) = Quality (Old Child)
5      best = Child with the fewer features
6    Else
7      best = Old Child
8    End IF
9  End IF
10 End For

```

Figure 4.10: Selection process of the best subset

#### 4.5 The Proposed Enhanced GCACO Algorithm

This section explains the EGCACO for FS based on the integration of the three enhancements as explained in Sections 4.2 to 4.4. The fundamental objective of this combination is to obtain the feature subset that balances the classification accuracy and the number of selected features.

The proposed EGCACO algorithm is presented in Figure 4.11, where the improvements or the differences between the EGCACO and MGCACO algorithms are highlighted. The first enhancement proposes the ACO-based features clustering as a significant component to group highly correlated features into the same cluster. The ants' concept starts with initialising the pheromone matrix for all clusters in the same amounts. Then, the pheromone matrix is updated on the basis of the quality of solutions generated. Finally, a local search method is implemented on the part of the best solutions to assist in locating good results. After the local search method is achieved, the pheromone matrix is updated to the best solutions obtained. Section 4.2 explains these procedures in detail.

Thereafter, the second enhancement is the ACO algorithm that is used as an adaptive strategy to determine the threshold value  $\epsilon$  dynamically rather than being determined by the user. This strategy aims to determine the threshold value with respect to searching complexity, selection process stages, and quality of the obtained solution. This control strategy uses the ACO principles and consists of three procedures comprising pheromone initialisation, ant selection, and pheromone updating rule (feedback collection).

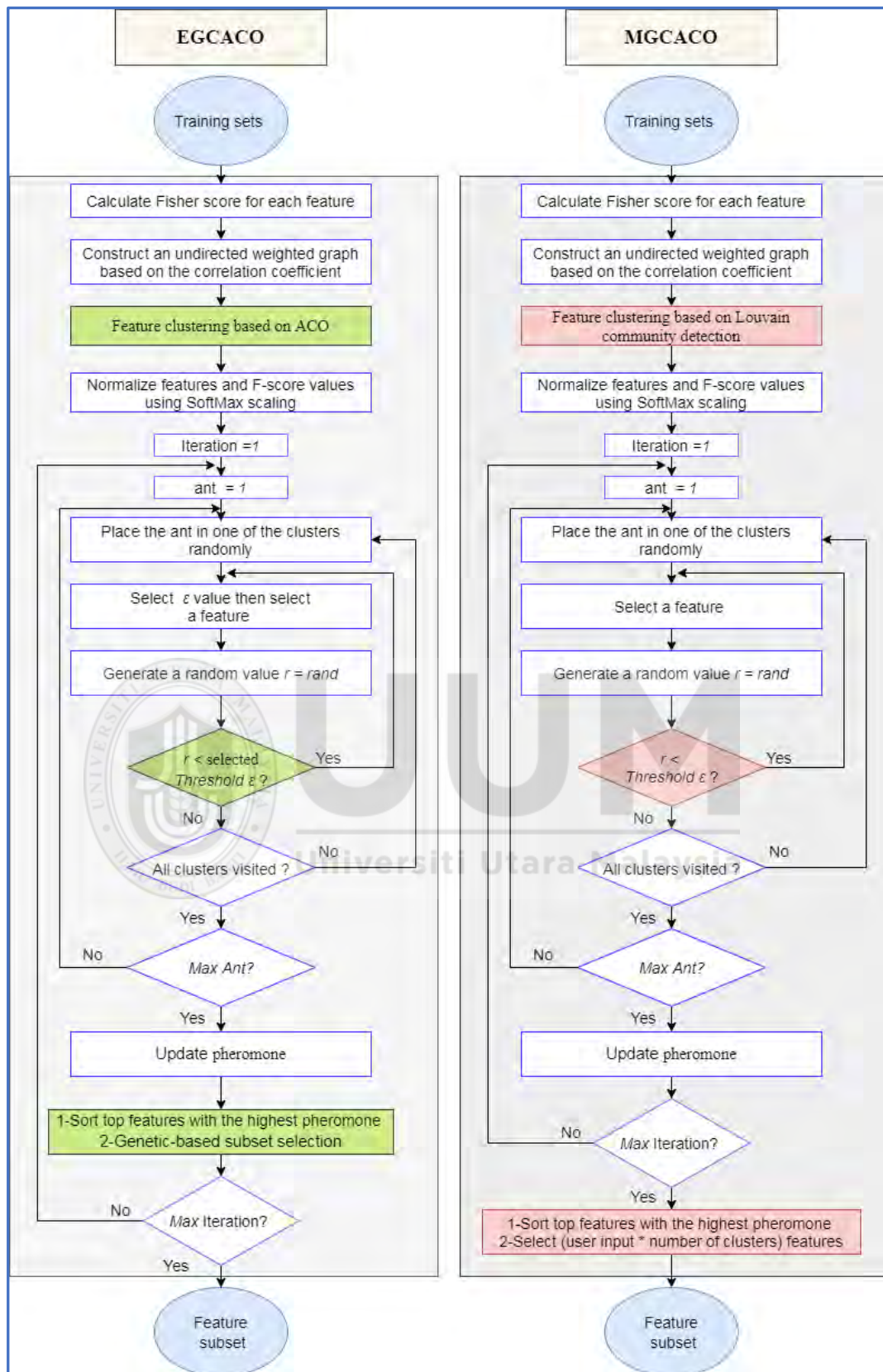


Figure 4.11: Proposed EGCACO and MGCACO algorithm

The last enhancement is done by applying the concept of the GA to automatically determine the final subset rather than being statically adopted. This proposed method utilises three operational steps comprising initialisation, crossover, and subset evaluation and selection. A new proposal introduced in this technique aim to obtain high classification accuracy while reducing the number of selected features, as discussed in the G-GCACO section. Figure 4.12 shows the pseudocode of the enhancement's integration in the MGCACO algorithm for FS.



**1 Input** Training dataset  
*I*: Number of iteration that algorithm repeated  
*A*: Number of ant  
*γ*: The initial amount of pheromone for each feature  
*LB*: lower bound to the threshold  $\varepsilon$  for remain in current cluster  
*UB* : upper bound to the threshold  $\varepsilon$  for remain in current cluster  
*ω*: Parameter to determine size of final feature subset  
*θ*: Parameter to determine the minimum edge weights of graph  
**2 Output** Feature subset =  $\{f_1, \dots, f_n\}$   
**3 Normalize** dataset using softmax scaling method  
**4 Calculate** edge weights  
**5 Remove** edges which their associated weighs are less than  $\theta$   
**6 Function:** ACO-based feature clustering //C-GCACO starts here  
**7 Input:** Algorithm<sub>parameters</sub> ,  $K_{clusters}$ , Data<sub>features</sub>  
**8 Output:** SbestClustering  
**9 Initialize** Population, Pheromone matrix  
**10 While**  $\neq$ StopCondition ( ) do  
**11 For**  $i = 1$  to Population<sub>size</sub>  
**12 Construct** solution using pheromone trail  
**13 Compute** weights of all test features, and cluster centres  
**14 Compute** clustering metric and assign it as objective function value of solution  
**15 End For**  
**16 Select** best solutions out of all solutions using objective function values  
**17 Compute** the local search on the selected best solutions Figure (4.2)  
**18 Update** pheromone trail matrix using best solution  
**19 End While** //C-GCACO ends here  
**20**  $\tau p_n(1) = \gamma, \tau f_n(1) = MI(d), \forall i, j = 1, \dots, n$   
**21 For**  $i = 1$  to  $I$   
**22 For**  $k = 1$  to  $A$   
**23 Position** ant in a random  $\varepsilon$  value //A-GCACO starts here  
**23 Calculate** the probability of each  $\varepsilon$  value

```

24 Select  $\varepsilon$  value by applying the state transition rule
25  $Traced\_Cluster = \emptyset$  ,  $Not\_Traced\_Cluster = \{Cluster\ 1, Cluster\ 2, \dots, Cluster\ k\}$ 
26 While ( $|Traced\_Cluster| < k$ ) do
27 Place the ant  $k$  randomly in one of the  $Not\_Traced\_Cluster$ 
28 Select one of the feature in  $Current\_Cluster$  according to probabilistic decision rule
29 Move the  $k$ -th ant to the new selected feature  $f$ 
30 Rand= Generate random value between  $[0,1]$ 
31 IF ( $Rand \geq \varepsilon$ ) then
32 While ( $Rand > \varepsilon$ ) do
33 Select one of the feature in the  $Current\_Cluster$ 
34 Move the  $k$ -th ant to the new selected feature  $f$ 
35 Rand= Generate Random Value between  $[0,1]$ 
36 End While
37 Else
38  $Traced\_Cluster = Traced\_Cluster \cup Current\_Cluster$ 
39  $Not\_Traced\_Cluster = Not\_Traced\_Cluster - Current\_Cluster$ 
40 End IF
41 End While
42 End For
43 Evaluate each constructed subset separability index
44 Update pheromone for all features  $\tau_{f_{1..n}}$ 
45 Update pheromone for the selected  $\varepsilon$  value  $\tau_{p_{1..n}}$  //A-GCACO ends here
46 Sorted features based on their pheromone values  $\tau_{f_{1..n}}$  in decreasing order
47 Population Initialisation //G-GCACO starts here
48 Crossover procedures Figure (4.8)
49 Evaluate the subsets
50 Select the best subset Figure (4.10) //G-GCACO ends here
51 End For
52 The Final subset

```

Figure 4.12: Pseudocode of EGCACO algorithm

#### 4.6 Summary

The proposed ACO-based FS algorithm is to improve classification accuracy while reducing the number of features. The characteristics of feature selection, feature clustering, and subset determination are integrated to form the proposed EGCACO algorithm for this purpose. The proposed FS algorithm involves of three main enhancements to the MGCACO algorithm. The first enhancement simplifies the selection process in dealing with both redundant and irrelevant features while the second enhancement maximizes the dependency and minimizes redundancy among features. The third enhancement is to enable the automatic determination of the final subset of the features. All three (3) enhancements proposed in EGCACO will be able to perform the FS process in a DNA microarray. The next chapter shows the evaluation of the proposed FS algorithms.



## **CHAPTER FIVE**

### **EXPERIMENTAL RESULTS**

#### **5.1 Introduction**

This chapter provides an empirical evaluation of the proposed EGCACO algorithm, which is based on the MGCACO algorithm. Apart from testing the proposed EGCACO algorithm, three components of the MGCACO were enhanced, and experiments were performed to test these components. The performance of the EGCACO algorithm was compared to several benchmark FS algorithms.

Section 5.2 presents the experimental design that used to guide the evaluation of the three modifications and the proposed EGCACO algorithm. Section 5.3 shows the results of the enhanced feature clustering method while Section 5.4 presents the experimental results of the adaptive feature selection technique). Section 5.5 presents the experimental outcomes of the proposed genetic method in determining the feature subset. The experimental findings of the EGCACO algorithm are presented in Section 5.6, followed by the chapter summary in Section 5.7.

#### **5.2 Experimental Design**

Figure 5.1 displays the experimental design for evaluating the proposed modifications and the EGCACO algorithm.

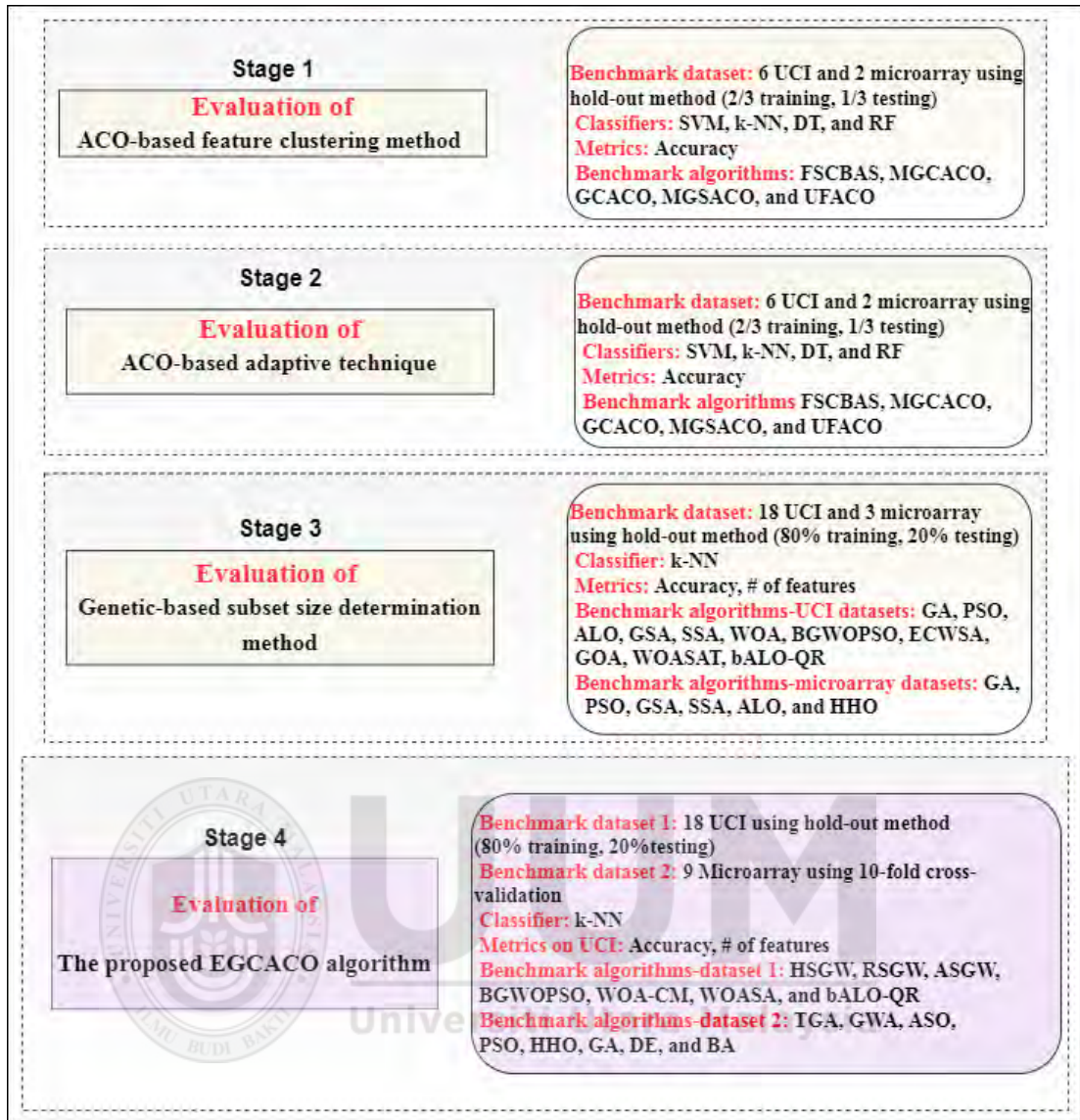


Figure 5.1: Experimental Design

The first, second, and third stages show what is involved in evaluating the three modifications. These include the datasets, algorithms for comparison, the performance metrics and the classifiers used in performing the classification task. The fourth stage focuses on the performance evaluation of the proposed algorithm. Missing values in all the datasets are replaced with the mean values of each respective feature (Moradi & Rostami, 2015b). All datasets were then normalized.

Similar configurations are adopted in stages 1 and 2 to evaluate the proposed ACO-based feature clustering method and ACO-based adaptive technique. The benchmark datasets are from the UCI machine learning repository and two (2) other microarray datasets from the Bioinformatics Research Group at the Universidad Pablo de Olavide (Bolón-Canedo et al., 2014) that are commonly used in FS studies (Ghimatgar et al., 2018; Manbari et al., 2019b; Moradi & Rostami, 2015b). In addition, the hold-out method (Press et al., 2009) is used in all the experiments of the first and second stages. In the studies that involved FS algorithms, the hold-out validation method is often utilised. (Ghimatgar et al., 2018; Manbari et al., 2019b; Moradi & Rostami, 2015b). In the first and second stages, each dataset is divided into two (2) parts. The ratio between the parts is 66% is for the training and 34% of the datasets is for testing. The training set is used to determine the feature subset, while the testing set is used to evaluate the accuracy of the selected features. In the evaluation of the first two modifications (stages 1 and 2), the classifiers used are the random forest, k-NN, decision tree, and support vector machine while the performance metrics are the classification accuracy and the standard deviation. The classifiers and performance metrics are commonly used for classification in FS algorithms (Ghimatgar et al., 2018; Manbari et al., 2019b; Moradi & Rostami, 2015b). The benchmark algorithms used for comparison in the first two stages are FSCBAS (Manbari et al., 2019b), MGCACO (Ghimatgar et al., 2018), GCACO (Moradi & Rostami, 2015b), MGSACO (Tabakhi et al., 2015), and UFACO (Tabakhi et al., 2014).

The evaluation of the proposed method based on genetic algorithm to determine the subset is performed in stage 3. In the experiment, the classifier used is k-NN while the

performance metrics are the classification accuracy and the number of selected features (Agrawal et al., 2020; Guha et al., 2020; Long et al., 2021; Mohammadzadeh & Gharehchopogh, 2021). The evaluation process utilises 18 UCI datasets and three (3) microarray datasets. These UCI benchmark datasets are commonly used in FS tasks (Agrawal et al., 2021; Jaddi & Abdullah, 2021; Mafarja et al., 2020). The datasets are divided in the ratio 80:20 for training and testing using the holdout method. A total of 11 benchmark algorithms are utilised (for the case of UCI datasets) namely GA (Ghamisi & Benediktsson, 2015), PSO (Chuang et al., 2008), ALO (Emary et al., 2016a), GSA (Taradeh et al., 2019), SSA (Faris et al., 2018), WOA (Mafarja & Mirjalili, 2018), BGWOPSO (Al-tashi et al., 2019), ECWSA (Guha, Ghosh, Mutsuddi, et al., 2020), GOA (Mafarja et al., 2019), WOASAT (Mafarja & Mirjalili, 2017), and bALO-QR (Mafarja & Mirjalili, 2019). Among these algorithms, BGWOPSO, WOASAT, and bALO-QR are classified as hybridized algorithms. For the microarray datasets, the GA, PSO, GSA, SSA, ALO, and HHO (Thaher et al., 2020) are used as benchmark algorithms.

The evaluation of the proposed algorithm is conducted in stage 4. The datasets employed are 18 UCI and nine (9) microarray datasets. The k-NN classifier is used for the FS task. The common benchmark algorithms used for the case of UCI datasets are HSGW (Mafarja et al., 2020), RSGW (Mafarja et al., 2020), ASGW (Mafarja et al., 2020), BGWOPSO, WOA-CM (Mafarja & Mirjalili, 2018), bALO-QR and WOASA (Mafarja et al., 2020; Mafarja & Mirjalili, 2018). The benchmark algorithms used for the case of microarray datasets are TGA (Too et al., 2018), GWO (Emary et al., 2016b), ASO (Too & Abdullah, 2020), PSO, HHO, GA, DE (Engelbrecht & Pampará,

2007), and BA (Al-betar et al., 2019). All datasets are partitioned into two parts based on the ratio 80:20 for training and testing processes using the holdout method. The performance metrics used in this stage are the classification accuracy and the number of selected features. Experiments are repeated ten (10) times for both the proposed algorithm and benchmark algorithms.

The microarray datasets in our empirical evaluation stage are addressed utilising the 10-fold cross-validation approach as adopted in the benchmarked algorithms (Manita & Korbaa, 2020). There are ten (10) subsets in the datasets, and the subsets are all the same size. The learning process is divided into nine (9) subsets, with the remaining subset used for testing. This process is carried out ten (10) times with a new subset for testing and a new combination of subsets for learning, to guarantee that all subsets are used in both stages. To obtain statistically significant findings, the experiment of the fourth stage on microarray datasets is performed 100 times independently.

In general, the use of the different datasets and classifiers is to match what has been used by all the benchmark algorithms. This is for fair comparison. The k-NN was chosen since all the benchmark classifier algorithms have used it as it is the best classifier in their contexts. Thus, for comparison purpose, k-NN is also used in this study.

### **5.3 Results and Analysis of the EC-GCACO Algorithm**

The implementation of the proposed ACO-based features clustering method in the MGCACO algorithm is called EC-GCACO. The initialisation parameters that are used

to evaluate the ACO-based features clustering are adopted from Shelokar et al. (2004) for standard ACO parameter values, and Ghimatgar et al. (2018) that introduce the ACO-based FS parameters. These parameters are listed in Table 5.1.

Table 5.1

*ACO-based features clustering experimental parameters*

	Parameter	Description	Value
ACO-based clustering	$A$	Ant number	50
	$I$	Iterations number	1000
	$q_0$	Exploration/exploitation	0.98
	$\rho$	Evaporation rate	0.01
	$P_{ls}$	Local search probability	0.01
ACO-based FS	$A$	Ant number	100
	$I$	Iterations number	50
	$\alpha$	Importance of pheromone	1
	$\beta$	Importance of heuristic	1
	$q_0$	Exploration/exploitation	0.7
	$\rho$	Evaporation rate	0.9
	$\varepsilon$	Threshold to remain in current cluster	0.4
	$Run$	Number of runs	10

### 5.3.1 Experiment on the UCI Datasets

Tables 5.2-5.5 show the experimental results of the average classification accuracy (Acc) and the standard deviation (Std) of EC-GCACO and five other FS algorithms with support vector machine, decision tree, k-NN, and random forest classifiers, respectively. The best result is highlighted while the figures in parentheses indicate the rank of the algorithms.

Table 5.2 shows that for the case of the support vector machine classifier, the proposed EC-GCACO is able to obtain the highest classification accuracy in all the datasets except for the lonosphere dataset, which obtains the third rank. In Table 5.3, the proposed EC-GCACO obtains the best classification accuracy results in four (4) out of six (6) datasets when the k-NN classifier is employed.

Table 5.2

*Average classification accuracy using support vector machine classifier on UCI datasets*

Dataset	# Selected features		EC-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Wine	6	Acc	<b>98.03 (1)</b>	97.2 (3)	98 (2)	95.1 (5)	94.61 (6)	96.02 (4)
		Std	$\pm 1.22$	$\pm 0$	$\pm 1.72$	$\pm 2.6$	$\pm 2.52$	$\pm 2.04$
Hepatitis	6	Acc	<b>83.86 (1)</b>	81.55 (4)	83.85 (2)	83.64 (3)	81.1 (5)	80.78 (6)
		Std	$\pm 1.59$	$\pm 3.74$	$\pm 4.82$	$\pm 2.83$	$\pm 1.13$	$\pm 0.4$
lonosphere	15	Acc	85.79 (3)	85.62 (4)	86.24 (2)	<b>86.79 (1)</b>	81.6 (6)	84.96 (5)
		Std	$\pm 2.32$	$\pm 3.57$	$\pm 2.83$	$\pm 2.41$	$\pm 2.09$	$\pm 2.12$
SpamBase	24	Acc	<b>91.78 (1)</b>	85.78 (3)	90.98 (2)	84.51 (4)	81.86 (6)	83 (5)
		Std	$\pm 0.80$	$\pm 0.22$	$\pm 0.90$	$\pm 2.12$	$\pm 2.67$	$\pm 2.67$
Arrhythmia	20	Acc	<b>86.03 (1)</b>	68 (3)	70.84 (2)	62 (4)	56.55 (5)	54.39 (6)
		Std	$\pm 1.31$	$\pm 3.63$	$\pm 3.83$	$\pm 5.2$	$\pm 1.4$	$\pm 0.29$
Madelon	40	Acc	<b>66.51 (1)</b>	61.2 (3)	58.98 (6)	64.61 (2)	60.98 (4)	60.75 (5)
		Std	$\pm 0.27$	$\pm 2.07$	$\pm 2.83$	$\pm 5.58$	$\pm 0.27$	$\pm 0.24$

Table 5.3

*Average classification accuracy using k-NN classifier on UCI datasets*

Dataset	# Selected features		EC-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Wine	6	Acc	<b>98.36 (1)</b>	97.2 (3)	98 (2)	95.68 (5)	94.61 (6)	96.02 (4)
		Std	$\pm 1.63$	$\pm 0$	$\pm 2.02$	$\pm 2.78$	$\pm 2.52$	$\pm 2.04$
Hepatitis	6	Acc	81.73 (2)	79.36 (6)	<b>82.12 (1)</b>	81.36 (3)	81.1 (4)	80.78 (5)
		Std	$\pm 1.01$	$\pm 3.74$	$\pm 6.97$	$\pm 1.33$	$\pm 1.13$	$\pm 0.41$
Ionosphere	15	Acc	<b>86.89 (1)</b>	85.62 (3)	86.5 (2)	85.36 (4)	81.6 (6)	84.96 (5)
		Std	$\pm 2.31$	$\pm 3.75$	$\pm 2.24$	$\pm 4.14$	$\pm 2.1$	$\pm 2.12$
SpamBase	24	Acc	<b>90.29 (1)</b>	85.78 (5)	89.6 (2)	87.04 (4)	81.86 (6)	88.33 (3)
		Std	$\pm 0.76$	$\pm 0.22$	$\pm 1.26$	$\pm 1.32$	$\pm 2.68$	$\pm 0.68$
Arrhythmia	20	Acc	<b>86.03 (1)</b>	57.31 (3)	64.07 (2)	53.75 (5)	55.25 (4)	50.87 (6)
		Std	$\pm 0.43$	$\pm 1.33$	$\pm 3.14$	$\pm 5.38$	$\pm 1.84$	$\pm 1.2$
Madelon	40	Acc	75.45 (3)	59.8 (6)	60.94 (5)	<b>76.21 (1)</b>	75.85 (2)	75.13 (4)
		Std	$\pm 1.04$	$\pm 1.34$	$\pm 2.56$	$\pm 1.74$	$\pm 0.87$	$\pm 0.4$

Table 5.4 displays the classification accuracy for the algorithms using the decision tree classifier. The proposed EC-GCACO algorithm is able to achieve the highest classification accuracy in all six (6) datasets.

Table 5.4

*Average classification accuracy using decision tree classifier on UCI datasets*

Dataset	# Selected features		EC-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Wine	6	Acc	<b>96.22 (1)</b>	93.83 (2)	93.17 (4)	93.57 (3)	92.93 (6)	93.09 (5)
		Std	$\pm 1.84$	$\pm 0$	$\pm 3.37$	$\pm 2.98$	$\pm 3.47$	$\pm 3.02$
Hepatitis	6	Acc	<b>80.79 (1)</b>	79.36 (4)	78.08 (5)	80.31 (2)	80.26 (3)	77.04 (6)
		Std	$\pm 0.74$	$\pm 0$	$\pm 6.71$	$\pm 1.14$	$\pm 0.92$	$\pm 1.22$
Ionosphere	15	Acc	<b>91.42 (1)</b>	88.32 (5)	90.86 (2)	90.54 (3)	86.7 (6)	89.06 (4)
		Std	$\pm 1.62$	$\pm 1.41$	$\pm 4.21$	$\pm 1.98$	$\pm 2.14$	$\pm 1.51$
SpamBase	24	Acc	<b>91.64 (1)</b>	89.68 (4)	90.88 (2)	89.12 (5)	89.83 (3)	89.08 (6)
		Std	$\pm 0.65$	$\pm 0.22$	$\pm 0.66$	$\pm 1.12$	$\pm 1.52$	$\pm 0.51$
Arrhythmia	20	Acc	<b>86.10 (1)</b>	59.76 (3)	66.71 (2)	56.62 (4)	49.94 (6)	50.89 (5)
		Std	$\pm 0.97$	$\pm 1.37$	$\pm 3.22$	$\pm 4.13$	$\pm 2.05$	$\pm 2.32$
Madelon	40	Acc	<b>84.17 (1)</b>	67.4 (6)	71.42 (5)	81.77 (2)	80.12 (3)	79.49 (4)
		Std	$\pm 1.11$	$\pm 1.25$	$\pm 3.50$	$\pm 2.38$	$\pm 0.57$	$\pm 0.67$

The results in Table 5.5 show that for the random forest classifier, the proposed EC-GCACO obtains the best results in four (4) out of six (6) datasets, wherein the SpamBase and Madelon datasets, the algorithm obtains the second rank with very small differences from the best rank algorithms.

Table 5.5

*Average classification accuracy using random forest classifier on UCI datasets*

Dataset	# Selected features		EC-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Wine	6	Acc	<b>98.19 (1)</b>	97.2 (3)	97.5 (2)	96.11 (4)	94.61 (6)	96.02 (5)
		Std	$\pm 1.71$	$\pm 0$	$\pm 2.53$	$\pm 2.32$	$\pm 2.53$	$\pm 2.04$
Hepatitis	6	Acc	<b>85.29 (1)</b>	81.39 (3)	84.43 (2)	81.3 (4)	81.1 (5)	80.78 (6)
		Std	$\pm 0.16$	$\pm 0$	$\pm 4.30$	$\pm 1.76$	$\pm 1.14$	$\pm 0.41$
Ionosphere	15	Acc	<b>94.62 (1)</b>	85.77 (3)	93.59 (2)	85.1 (4)	81.6 (6)	84.95 (5)
		Std	$\pm 1.43$	$\pm 3.57$	$\pm 1.77$	$\pm 2.41$	$\pm 2.1$	$\pm 2.12$
SpamBase	24	Acc	94.16 (2)	85.78 (5)	<b>94.17 (1)</b>	89.02 (4)	81.86 (6)	89.96 (3)
		Std	$\pm 0.62$	$\pm 0.23$	$\pm 0.75$	$\pm 1.98$	$\pm 2.68$	$\pm 0.54$
Arrhythmia	20	Acc	<b>84.21 (1)</b>	66.38 (3)	75.36 (2)	61.18 (4)	57.22 (5)	50.05 (6)
		Std	$\pm 0.42$	$\pm 1.33$	$\pm 2.24$	$\pm 2.22$	$\pm 1.6$	$\pm 2.91$
Madelon	40	Acc	83.26 (2)	61.19 (6)	76.51 (4)	72.27 (5)	<b>83.29 (1)</b>	83.24 (3)
		Std	$\pm 0.85$	$\pm 1.63$	$\pm 2.75$	$\pm 1.72$	$\pm 0.48$	$\pm 0.49$

Table 5.6 depicts the summary of the averages for the classification accuracy (Acc) and standard deviation (Std) algorithms with respect to the classifiers. The best result for each algorithm is highlighted. Overall, it can be seen that EC-GCACO shows superior performance compared to other algorithms in all classifiers in terms of classification accuracy. In terms of standard deviation, the FSCBAS obtained the best value for the decision tree classifier. This is due to the parallel process of the FSCBAS algorithm which may decrease the randomization of the solution. However, the proposed EC-GCACO shows very competitive small values when compared to the other algorithms, indicating the stability of its performance. The results of the accuracy in Table 5.6 are, again, displayed in Figure 5.2.

Table 5.6

*Results summary on UCI datasets*

Classifiers		EC-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
SVM	Acc	<b>85.33</b>	79.89	69.67	79.44	76.11	76.65
	Std	1.22	2.20	2.82	3.45	1.68	1.29
k-NN	Acc	<b>86.46</b>	77.51	80.20	79.90	78.37	79.34
	Std	1.19	1.64	1.73	3.03	1.85	1.14
DT	Acc	<b>88.39</b>	79.72	81.85	81.98	79.96	79.77
	Std	1.15	0.70	3.61	2.28	1.77	1.54
RF	Acc	<b>89.95</b>	79.61	86.92	80.83	79.94	80.83
	Std	0.86	1.15	2.39	2.06	1.75	1.14

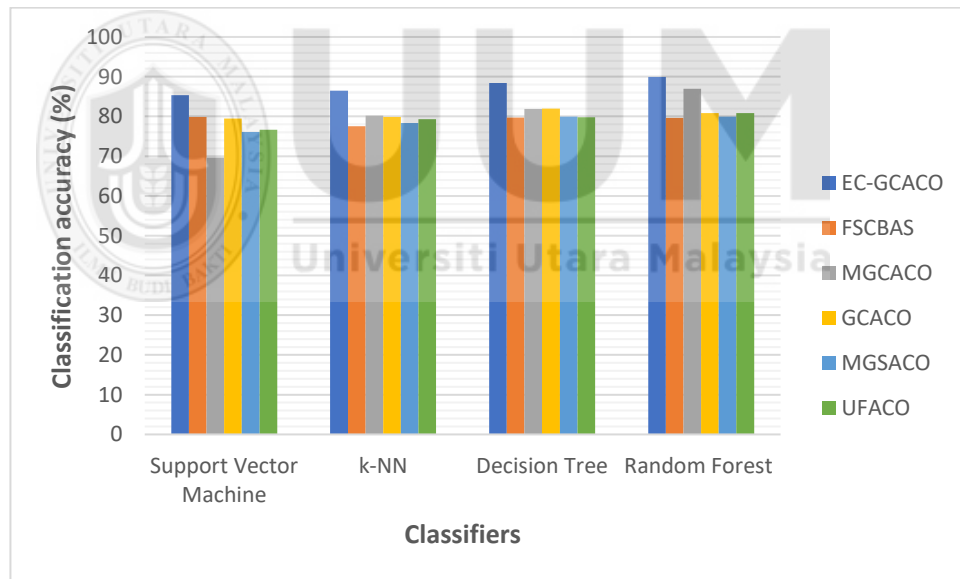


Figure 5.2: Average classification accuracy for UCI datasets

### 5.3.2 Experiment on the Microarray Dataset

Tables 5.7 and 5.10 show the experimental results of the average classification accuracy (Acc) and the standard deviation (Std) of the proposed EC-GCACO and five other FS algorithms. The classifiers used for FS in the microarray comprise support

vector machine, decision tree, k-NN, and random forest classifiers. The colon and leukemia datasets are classified as microarray datasets. The best result is highlighted for each dataset while the figures in parentheses indicate the rank of the algorithms. Tables 5.7, 5.8, 5.9, and 5.10 show that the proposed EC-GCACO achieves the best average classification accuracy in all cases except for the case of the k-NN classifier where the FSCBAS has the best result on the Colon dataset.

Table 5.7

*Average classification accuracy using support vector machine classifier on microarray datasets*

Dataset	# Selected features		EC-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Colon	40	Acc	<b>90.00 (1)</b>	87.59 (2)	84.76 (3)	81.99 (6)	83.88 (4)	82.1 (5)
		Std	± 1.51	± 1.31	± 6.31	± 2.23	± 2.94	± 3.26
Leukemia	40	Acc	<b>98.33 (1)</b>	95.42 (3)	95.83 (2)	93.67 (4)	90.14 (5)	89.45 (6)
		Std	± 0.90	± 0.67	± 3.22	± 2.35	± 5.17	± 3.28

Table 5.8

*Average classification accuracy using k-NN classifier on microarray datasets*

Dataset	# Selected features		EC-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Colon	40	Acc	87.14 (2)	<b>87.59 (1)</b>	81.43 (4)	81.13 (5)	82.1 (3)	80.65 (6)
		Std	± 2.04	± <b>2.52</b>	± 5.01	± 4.32	± 2.79	± 2.28
Leukemia	40	Acc	<b>98.75 (1)</b>	95.42 (2)	93.75 (3)	88.02 (5)	88.06 (4)	87.92 (6)
		Std	± 1.66	± 1.64	± 9.63	± 3.11	± 2.27	± 1.45

Table 5.9

*Average classification accuracy using decision tree classifier on microarray datasets*

Dataset	# Selected features		EC-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Colon	40	Acc	<b>87.61 (1)</b>	86.13 (2)	78.09 (6)	84.23 (4)	85.17 (3)	82.91 (5)
		Std	$\pm 2.80$	$\pm 0.83$	$\pm 7.73$	$\pm 2.39$	$\pm 2.82$	$\pm 4.18$
Leukemia	40	Acc	<b>93.75 (1)</b>	93.62 (2)	90.83 (3)	85.74 (5)	90.28 (4)	77.92 (6)
		Std	$\pm 0.08$	$\pm 0$	$\pm 3.63$	$\pm 1.52$	$\pm 4.29$	$\pm 2.96$

Table 5.10

*Average classification accuracy using random forest classifier on microarray datasets*

Dataset	# Selected features		EC-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Colon	40	Acc	<b>88.57 (1)</b>	87.59 (2)	84.29 (3)	81.22 (6)	82.91 (4)	81.78 (5)
		Std	$\pm 2.33$	$\pm 2.53$	$\pm 4.28$	$\pm 3.95$	$\pm 2.18$	$\pm 4.93$
Leukemia	40	Acc	<b>99.76 (1)</b>	95.42 (3)	98.75 (2)	89.69 (4)	89.36 (5)	86.12 (6)
		Std	$\pm 0.46$	$\pm 0.67$	$\pm 1.90$	$\pm 3.61$	$\pm 5.16$	$\pm 3.53$

Table 5.11 summarizes the results presented in Tables 5.7-5.10 where the best result for each classifier is highlighted. Overall, the EC-GCACO algorithm performs higher and outperforms the other algorithms in terms of average classification accuracy. When compared to the other algorithms, the proposed EC-GCACO shows very competitive small values in terms of standard deviation, indicating the stability of its performance. Figure 5.3 shows the average classification accuracy for all classifiers. In general, the proposed algorithm outperforms the other algorithms in all classifiers.

Table 5.11

*Average classification accuracy, standard deviation and performance rank on microarray datasets*

Classifiers		EC-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
SVM	Acc	<b>94.16</b>	91.50	90.29	87.83	87.01	85.77
	Std	1.20	0.99	4.765	2.29	4.05	3.27
k-NN	Acc	<b>92.94</b>	91.50	87.58	84.57	85.08	84.28
	Std	1.85	2.08	7.23	3.71	2.53	1.86
DT	Acc	<b>90.68</b>	89.87	84.46	84.98	85.20	78.70
	Std	1.44	0.41	5.68	3.91	3.55	3.57
RF	Acc	<b>94.16</b>	91.50	91.51	85.45	86.13	83.95
	Std	1.39	1.6	3.09	6.78	3.67	4.23

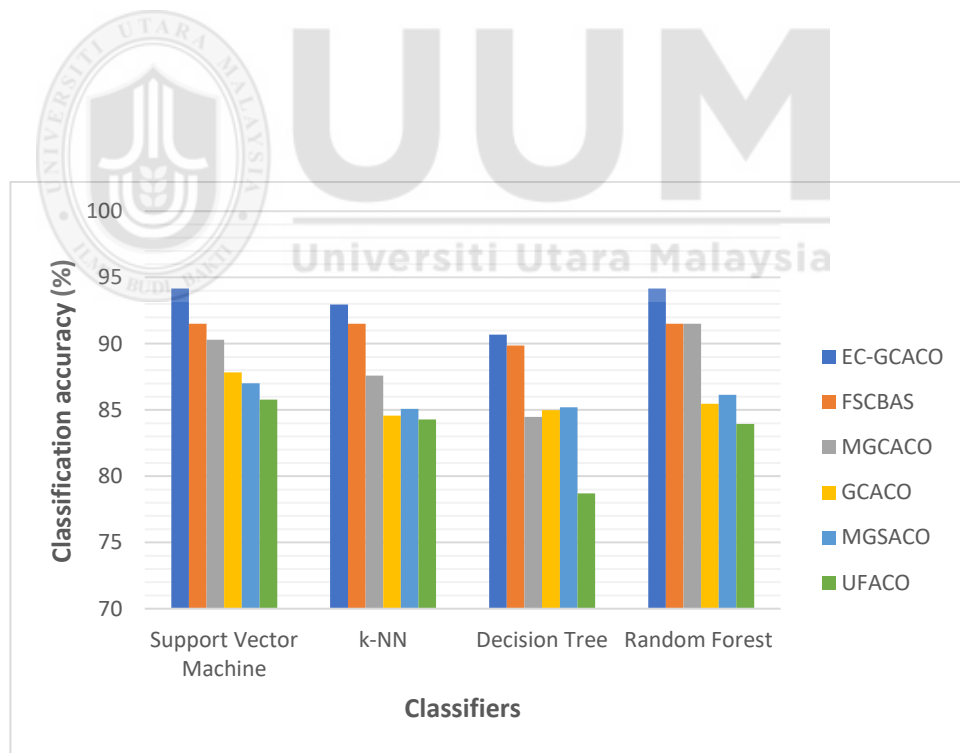


Figure 5.3: Average classification accuracy of microarray data

In summary, the EC-GCACO obtains the best classification accuracy. This good performance is due to the effectiveness of the enhancement on the clustering method

which is based on the ACO algorithm concept. Wherein the proposed ACO intensification and diversification are changed iteratively through pheromone update, enabling the successful implementation of both a global and a local search. Thus, the method has managed to escape from local optima in grouping the highly correlated features during the exploration process, thus improving the classification accuracy.

#### 5.4 Results and Analysis of the A-GCACO Algorithm

The implementation of the adaptive selection technique based on ACO in the MGCACO is called A-GCACO. The parameters and values used in evaluating the proposed A-GCACO algorithm are listed in Table 5.12. The parameter settings are adopted from the ACO-based FS algorithm (Ghimatgar et al., 2018).

Table 5.12

*ACO-based adaptive selection technique experimental parameters*

Parameter	Description	Value
$A$	Ant number	100
$I$	Iterations number	50
$\varphi$	Initial amount of pheromone	0.06
$\alpha$	Importance of pheromone	1
$\beta$	Importance of heuristic	1
$q_0$	Exploration/exploitation	0.7
$\rho$	Evaporation coefficient	0.9
$Run$	Number of runs	10

##### 5.4.1 Experiment on the UCI Datasets

Tables 5.13–5.16 show the average classification accuracy (Acc) and the standard deviation (Std) of the A-GCACO algorithm and five other FS algorithms with support

vector machine, decision tree, k-NN, and random forest classifiers for FS, respectively.

The best result is highlighted for each dataset while the figures in parentheses indicate the rank of the algorithms.

Table 5.13 shows that the proposed A-GCACO obtains the best result in four (4) out of six (6) datasets when the support vector machine classifier is used. The proposed algorithm is unable to obtain the best classification accuracy with datasets that had missing values that were replaced and datasets that contained a combination of feature value types (i.e. Hepatitis and SpamBase).

Table 5.13

*Average classification accuracy using support vector machine classifier on UCI datasets*

Dataset	# Selected features		A-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Wine	6	Acc	<b>98.69 (1)</b>	97.2 (3)	98 (2)	95.1 (5)	94.61 (6)	96.02 (4)
		Std	± 1.40	± 0	± 1.72	± 2.6	± 2.52	± 2.04
Hepatitis	6	Acc	78.85 (6)	81.55 (3)	<b>83.85 (1)</b>	83.64 (2)	81.1 (4)	80.78 (5)
		Std	± 1.10	± 3.74	± 4.82	± 2.83	± 1.13	± 0.4
lonosphere	15	Acc	<b>89.15 (1)</b>	85.62 (4)	86.24 (3)	86.79 (2)	81.6 (6)	84.96 (5)
		Std	± 2.10	± 3.57	± 2.83	± 2.41	± 2.09	± 2.12
SpamBase	24	Acc	88.22 (2)	85.78 (3)	<b>90.98 (1)</b>	84.51 (4)	81.86 (6)	83 (5)
		Std	± 0.80	± 0.22	± 0.90	± 2.12	± 2.67	± 2.67
Arrhythmia	20	Acc	<b>86.29 (1)</b>	68 (3)	70.84 (2)	62 (4)	56.55 (5)	54.39 (6)
		Std	± 1.10	± 3.63	± 3.83	± 5.2	± 1.4	± 0.29
Madelon	40	Acc	<b>64.73 (1)</b>	61.2 (3)	58.98 (6)	64.61 (2)	60.98 (4)	60.75 (5)
		Std	± 1.32	± 2.07	± 2.83	± 5.58	± 0.27	± 0.24

In Table 5.14, A-GCACO obtains the best classification accuracy results on two (2) datasets when the k-NN classifier is employed. k-NN classifier is known to work well with numerical data. This is the case for the lonosphere and Arrhythmia datasets. In spite of this, A-GCACO is able to select significant features in the Arrhythmia dataset, which contains both types of data. However, the proposed algorithm is strongly better in the average classification accuracy over all other algorithms except for MGCACO, where the results obtained compete.

Table 5.14

Average classification accuracy using k-NN classifier on UCI datasets

Dataset	# Selected features		A-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Wine	6	Acc	97.38 (2)	97.2 (3)	<b>98 (1)</b>	95.68 (5)	94.61 (6)	96.02 (4)
		Std	± 0.82	± 0	± 2.02	± 2.78	± 2.52	± 2.04
Hepatitis	6	Acc	76.35 (6)	79.36 (5)	<b>82.12 (1)</b>	81.36 (2)	81.1 (3)	80.78 (4)
		Std	± 1.65	± 3.74	± 6.97	± 1.33	± 1.13	± 0.41
lonosphere	15	Acc	<b>90.84 (1)</b>	85.62 (3)	86.5 (2)	85.36 (4)	81.6 (6)	84.96 (5)
		Std	± 2.00	± 3.75	± 2.24	± 4.14	± 2.1	± 2.12
SpamBase	24	Acc	89.37 (2)	85.78 (5)	<b>89.6 (1)</b>	87.04 (4)	81.86 (6)	88.33 (3)
		Std	± 0.70	± 0.22	± 1.26	± 1.32	± 2.68	± 0.68
Arrhythmia	20	Acc	<b>81.56 (1)</b>	57.31 (3)	64.07 (2)	53.75 (5)	55.25 (4)	50.87 (6)
		Std	± 1.11	± 1.33	± 3.14	± 5.38	± 1.84	± 1.2
Madelon	40	Acc	65.54 (4)	59.8 (6)	60.94 (5)	<b>76.21 (1)</b>	75.85 (2)	75.13 (3)
		Std	± 1.50	± 1.34	± 2.56	± 1.74	± 0.87	± 0.4

Table 5.15 displays the classification accuracy for the algorithms using decision tree classifier. The proposed algorithm obtains the highest classification accuracy in four (4) out of six (6) datasets. Decision tree classifier can work with both types of feature

values. The proposed algorithm is able to select significant features to produce good classification accuracy even for databases with only continuous feature values such as the SpamBase database.

Table 5.15

*Average classification accuracy using decision tree classifier on UCI datasets*

Dataset	# Selected features		A-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Wine	6	Acc	<b>95.74 (1)</b>	93.83 (2)	93.17 (4)	93.57 (3)	92.93 (6)	93.09 (5)
		Std	± 2.22	± 0	± 3.37	± 2.98	± 3.47	± 3.02
Hepatitis	6	Acc	75.19 (6)	79.36 (3)	78.08 (4)	80.31 (2)	<b>80.26 (1)</b>	77.04 (5)
		Std	± 0.95	± 0	± 6.71	± 1.14	± 0.92	± 1.22
Ionosphere	15	Acc	<b>91.51 (1)</b>	88.32 (5)	90.86 (2)	90.54 (3)	86.7 (6)	89.06 (4)
		Std	± 1.94	± 1.41	± 4.21	± 1.98	± 2.14	± 1.51
SpamBase	24	Acc	<b>92.51 (1)</b>	89.68 (4)	90.88 (2)	89.12 (5)	89.83 (3)	89.08 (6)
		Std	± 0.71	± 0.22	± 0.66	± 1.12	± 1.52	± 0.51
Arrhythmia	20	Acc	<b>87.14 (1)</b>	59.76 (3)	66.71 (2)	56.62 (4)	49.94 (6)	50.89 (5)
		Std	± 1.44	± 1.37	± 3.22	± 4.13	± 2.05	± 2.32
Madelon	40	Acc	69.45 (5)	67.4 (6)	71.42 (4)	<b>81.77 (1)</b>	80.12 (2)	79.49 (3)
		Std	± 1.40	± 1.25	± 3.50	± 2.38	± 0.57	± 0.67

The results in Table 5.16 show that for the case of the random forest classifier, A-GCACO is able to gain the highest accuracy in all the datasets except for the wine dataset, which obtains the second rank with a very small difference. Random forest classifier is known to operate well with datasets that have categorical data. Nevertheless, in this case, the proposed algorithm is able to select significant features for classification even for datasets that had both categorical and numerical data types.

Table 5.16

*Average classification accuracy using random forest classifier on UCI dataset*

Dataset	# Selected features		A-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Wine	6	Acc	97.35 (2)	97.2 (3)	<b>97.5 (1)</b>	96.11 (4)	94.61 (6)	96.02 (5)
		Std	± 0.50	± 0	± 2.53	± 2.32	± 2.53	± 2.04
Hepatitis	6	Acc	<b>86.57 (1)</b>	81.39 (3)	84.43 (2)	81.3 (4)	81.1 (5)	80.78 (6)
		Std	± 0.40	± 0	± 4.30	± 1.76	± 1.14	± 0.41
Ionosphere	15	Acc	<b>94.03 (1)</b>	85.77 (3)	93.59 (2)	85.1 (4)	81.6 (6)	84.95 (5)
		Std	± 1.85	± 3.57	± 1.77	± 2.41	± 2.1	± 2.12
SpamBase	24	Acc	<b>94.74 (1)</b>	85.78 (5)	94.17 (2)	89.02 (4)	81.86 (6)	89.96 (3)
		Std	± 0.75	± 0.23	± 0.75	± 1.98	± 2.68	± 0.54
Arrhythmia	20	Acc	<b>83.57 (1)</b>	66.38 (3)	75.36 (2)	61.18 (4)	57.22 (5)	50.05 (6)
		Std	± 1.32	± 1.33	± 2.24	± 2.22	± 1.6	± 2.91
Madelon	40	Acc	<b>83.31 (1)</b>	61.19 (6)	76.51 (4)	72.27 (5)	83.29 (2)	83.24 (3)
		Std	± 0.28	± 1.63	± 2.75	± 1.72	± 0.48	± 0.49

The experimental results in Table 5.17 displays the summary of results on UCI datasets with respect to the classifiers. The accuracy (Acc) and standard deviation (Std) are the averages of the accuracy and standard deviation values from Tables 5.15-5.16. The best result for each classifier is highlighted. The proposed A-GCACO manages to obtain the best results in all classifiers. Furthermore, the A-GCACO has the smallest value for standard deviation which reflects the stability of the algorithm. The average classification accuracies for all classifiers is displayed in Figure 5.4. In all classifiers, the proposed algorithm outperforms the compared algorithms.

Table 5.17

*Result summary on average classification accuracy on UCI datasets*

Classifiers		A-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
SVM	Acc	<b>84.32</b>	79.89	69.67	79.44	76.11	76.65
	Std	1.30	2.20	2.82	3.45	1.68	1.29
k-NN	Acc	<b>83.50</b>	77.51	80.20	79.90	78.37	79.34
	Std	1.30	1.64	1.73	3.03	1.85	1.14
DT	Acc	<b>85.25</b>	79.72	81.85	81.98	79.96	79.77
	Std	1.44	0.70	3.61	2.28	1.77	1.54
RF	Acc	<b>89.92</b>	79.61	86.92	80.83	79.94	80.83
	Std	0.85	1.15	2.39	2.06	1.75	1.14

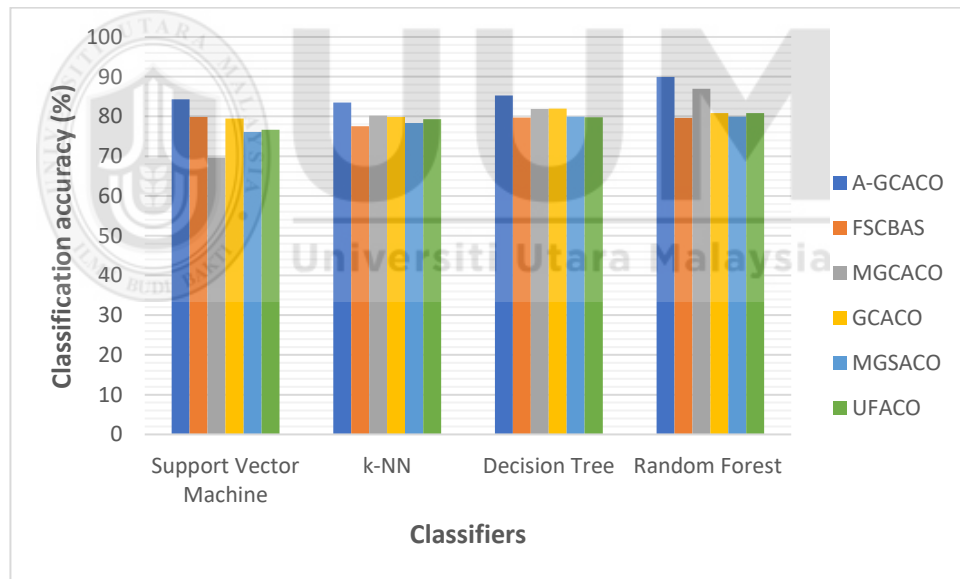


Figure 5.4: Average classification accuracy of all UCI data

#### 5.4.2 Experiment on the Microarray Datasets

Tables 5.18-5.21 show the experimental results of the average classification accuracy (Acc) and the standard deviation (Std) of selected features using the proposed A-GCACO algorithm with other common algorithms. The support vector machine, decision tree, k-NN, and random forest classifiers are used in the experiments on FS

classification. The best result is highlighted for each dataset while the figures in parentheses indicate the rank of the algorithms based on the classification accuracy. A-GCACO outperforms other algorithms in all datasets in term of classification accuracy, which is the relation between the numbers of features classified correctly with the dataset size. Thus, in all the experiments, A-GCACO obtains the best classification accuracy for microarray datasets (i.e., Colon and Leukemia) when all the classifiers are used. These datasets are considered as large size datasets (Kudo & Sklansky, 2000).

Table 5.18

*Average classification accuracy using support vector machine classifier on microarray datasets*

Dataset	# Selected features		A-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Colon	40	Acc	<b>92.38 (1)</b>	87.59 (2)	84.76 (3)	81.99 (6)	83.88 (4)	82.1 (5)
		Std	± 1.85	± 1.31	± 6.31	± 2.23	± 2.94	± 3.26
Leukemia	40	Acc	<b>98.75 (1)</b>	95.42 (3)	95.83 (2)	93.67 (4)	90.14 (5)	89.45 (6)
		Std	± 1.90	± 0.67	± 3.22	± 2.35	± 5.17	± 3.28

Table 5.19

*Average classification accuracy using k-NN classifier on microarray datasets*

Dataset	# Selected features		A-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Colon	40	Acc	<b>91.42 (1)</b>	87.59 (2)	81.43 (4)	81.13 (5)	82.1 (3)	80.65 (6)
		Std	± 2.01	± 2.52	± 5.01	± 4.32	± 2.79	± 2.28
Leukemia	40	Acc	<b>97.08 (1)</b>	95.42 (2)	93.75 (3)	88.02 (5)	88.06 (4)	87.92 (6)
		Std	± 1.90	± 1.64	± 9.63	± 3.11	± 2.27	± 1.45

Table 5.20

*Average classification accuracy using decision tree classifier on microarray datasets*

Dataset	# Selected features		A-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Colon	40	Acc	<b>90.48 (1)</b>	86.13 (2)	78.09 (6)	84.23 (4)	85.17 (3)	82.91 (5)
		Std	$\pm 0.32$	$\pm 0.83$	$\pm 7.73$	$\pm 2.39$	$\pm 2.82$	$\pm 4.18$
Leukemia	40	Acc	<b>97.08 (1)</b>	93.62 (2)	90.83 (3)	85.74 (5)	90.28 (4)	77.92 (6)
		Std	$\pm 0.76$	$\pm 0$	$\pm 3.63$	$\pm 1.52$	$\pm 4.29$	$\pm 2.96$

Table 5.21

*Average classification accuracy using random forest classifier on microarray dataset*

Dataset	# Selected features		A-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
Colon	40	Acc	<b>88.09 (1)</b>	87.59 (2)	84.29 (3)	81.22 (6)	82.91 (4)	81.78 (5)
		Std	$\pm 2.87$	$\pm 2.53$	$\pm 4.28$	$\pm 3.95$	$\pm 2.18$	$\pm 4.93$
Leukemia	40	Acc	<b>98.75 (1)</b>	95.42 (3)	98.75 (2)	89.69 (4)	89.36 (5)	86.12 (6)
		Std	$\pm 0.66$	$\pm 0.67$	$\pm 1.90$	$\pm 3.61$	$\pm 5.16$	$\pm 3.53$

Table 5.22 summarizes the averages of classification accuracy (Acc) and the standard deviations (Std) of the algorithms with respect to the classifiers. The best result for each algorithm is highlighted. Overall, the A-GCACO algorithm performs higher and outperforms the other algorithms in terms of the average classification accuracy. The proposed A-GCACO shows very competitive small values in terms of standard deviation, indicating the stability of its performance. Figure 5.5 shows the average

classification accuracy for all classifiers. In general, the proposed algorithm outperforms the other algorithms in all classifiers.

Table 5.22

*Summary of results on microarray datasets*

Classifiers		A-GCACO	FSCBAS	MGCACO	GCACO	MGSACO	UFACO
<b>SVM</b>	Acc	<b>95.56</b>	91.50	90.29	87.83	87.01	85.77
	Std	1.87	0.99	4.765	2.29	4.05	3.27
<b>k-NN</b>	Acc	<b>94.25</b>	91.50	87.58	84.57	85.08	84.28
	Std	1.95	2.08	7.23	3.71	2.53	1.86
<b>DT</b>	Acc	<b>93.77</b>	89.87	84.46	84.98	85.20	78.70
	Std	0.54	0.41	5.68	3.91	3.55	3.57
<b>RF</b>	Acc	<b>93.42</b>	91.50	91.51	85.45	86.13	83.95
	Std	1.76	1.6	3.09	6.78	3.67	4.23

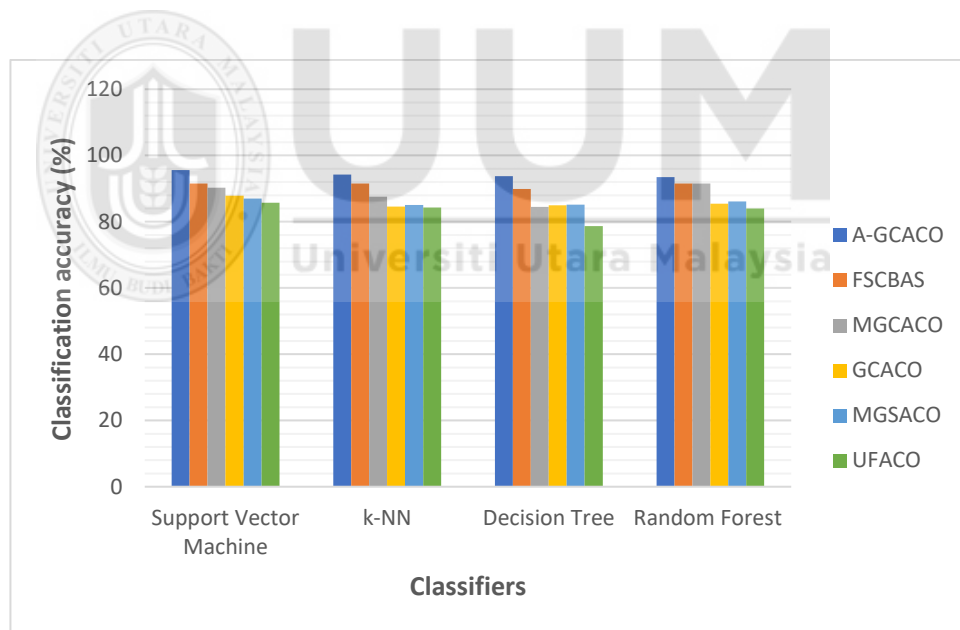


Figure 5.5: Average classification accuracy of microarray data

In conclusion, the A-GCACO has the highest classification accuracy and the smallest standard deviations when compared to the other algorithms, indicating that its performance is stable. The adaptive ACO technique (A-GCACO) has the ability to

determine the suitable value to be utilised in selecting significant features that can maximize dependency and reduce redundancy among the features, hence improving classification accuracy.

### 5.5 Results and Analysis of the G-GCACO Algorithm

The implementation of the new method that can determine the final subset based on genetic algorithm is called G-GCACO. The parameters and values used in evaluating the proposed final subset size determination are listed in Table 5.23. The parameters and values, as used by Raymer et al. (2000) in their study on genetic parameter values and Ghimatgar et al. (2018), introduce the ACO-based FS parameters.

Table 5.23

*Final subset determination experimental parameters*

	Parameter	Description	Value
<b>Genetic algorithm</b>	$PS$	Population size	2
	$G$	Generations	50
<b>ACO-based FS</b>	$A$	Ant number	100
	$I$	Iterations number	50
	$\alpha$	Importance of pheromone	1
	$\beta$	Importance of heuristic	1
	$q_0$	Exploration/exploitation	0.7
	$\rho$	Evaporation rate	0.9
	$\varepsilon$	Threshold to remain in current cluster	0.4
	$Run$	Number of runs	10

### 5.5.1 Experiment on the UCI Datasets

In this section, experiments for the FS utilise 11 benchmark algorithms for the purpose of evaluating the performance of G-GCACO algorithm on 18 datasets from UCI. Tables 5.24 and 5.25 show the experimental results of the average classification accuracy and the average number of selected features respectively, using the k-NN classifier. The best result is highlighted for each dataset while the figures in parentheses indicate the rank of the algorithms in terms of the classification accuracy for each dataset.

The G-GCACO algorithm performs better than other algorithms where it is able to obtain the best result in nine (9) out of 18 datasets. For the remaining nine (9) datasets, the G-GCACO has competitive results with the rest of the algorithms. Only two (2) of the 18 datasets have the worse rank, wherein the proposed algorithm obtains the eleventh rank in KrVsKpEW and WavefonnEW datasets. The proposed G-GCACO is better than other algorithms in terms of the average classification accuracy except for the BGWOPSO algorithm, where the G-GCACO shows a competitive result and achieves the second rank with a very small difference.

Table 5.25 shows that the proposed G-GCACO obtains the least number of features for 12 of the 18 datasets (i.e., 66.67% of the UCI datasets). For the remaining eight (8) datasets, the proposed algorithm shows competitive results. Moreover, with respect to nine (9) datasets, the G-GCACO algorithm fails to achieve the highest accuracy (as displayed in Table 5.24) but, in terms of the number of selected features, it surpasses other algorithms.

Table 5.24

*Classification accuracy results on UCI datasets*

Dataset	G- GCACO	GA	PSO	ALO	GSA	SSA	WOA	BGWOPSO	ECWSA	GOA	WOASAT	bALO- QR
Breastcancer	<b>99</b> (1)	97.4 (4)	96.3 (7)	97.4 (4)	96.86 (6)	97.68 (3)	95.71 (8)	98 (2)	95.18 (9)	97.37 (5)	97 (6)	97.4 (4)
Tic-tac-toe	79.1 (6)	79.96 (5)	79.96 (5)	78.3 (9)	77.66 (10)	<b>82.05</b> (1)	75.11 (11)	81 (2)	78.78 (8)	80.38 (3)	79 (7)	80 (4)
Exactly	95 (6)	<b>100</b> (1)	<b>100</b> (1)	96.5 (5)	99.4 (3)	98.03 (4)	75 (9)	<b>100</b> (1)	78.11 (8)	99.99 (2)	<b>100</b> (1)	91.2 (7)
Exactly2	76 (5)	77 (2)	76.8 (3)	76.2 (4)	77 (2)	75.82 (6)	69 (9)	76 (5)	<b>79.12</b> (1)	75.65 (7)	75 (8)	76 (5)
HeartEW	<b>88.5</b> (1)	87.41 (3)	83.7 (9)	83.8 (8)	82.96 (10)	86.05 (5)	76 (11)	85 (7)	85.56 (6)	86.35 (4)	85 (7)	88.4 (2)
M-of-n	99 (4)	<b>100</b> (1)	<b>100</b> (1)	96.7 (5)	99.4 (2)	99.18 (3)	85 (7)	<b>100</b> (1)	92.13 (6)	<b>100</b> (1)	<b>100</b> (1)	<b>100</b> (1)
WineEW	98.3 (6)	98.88 (5)	97.75 (8)	97.2 (9)	97.75 (8)	99.33 (3)	92.8 (10)	<b>100</b> (1)	98.02 (7)	99.85 (2)	99 (4)	<b>100</b> (1)
CongressEW	<b>99</b> (1)	96.79 (6)	96.33 (7)	98.1 (2)	96.33 (7)	96.28 (8)	92.9 (10)	98 (3)	96.19 (9)	97.72 (4)	98 (3)	97.2 (5)
Vote	<b>99</b> (1)	97.33 (2)	96 (5)	97.2 (3)	96 (5)	95.11 (6)	93.87 (10)	97 (4)	95 (7)	94.84 (8)	97 (4)	94.8 (9)
Zoo	<b>100</b> (1)	90.2 (9)	96.08 (8)	98 (3)	98.04 (2)	<b>100</b> (1)	96.47 (6)	<b>100</b> (1)	98 (3)	97.78 (4)	97 (5)	96.1 (7)
Lymphography	<b>96</b> (1)	83.78 (10)	89.19 (4)	91.7 (3)	86.49 (8)	89 (5)	78.5 (11)	92 (2)	87.39 (7)	85.86 (9)	89 (5)	88.6 (6)
SpectEW	86 (6)	89.55 (3)	88.81 (4)	89.9 (2)	84.33 (8)	83.61 (9)	78.7 (11)	88 (5)	79.88 (10)	85.65 (7)	88 (5)	<b>90</b> (1)
BreastEW	<b>98.141</b> (1)	97.54 (3)	97.19 (6)	97.4 (4)	95.44 (12)	94.84 (11)	95.53 (10)	97 (7)	97.33 (5)	96.06 (9)	98 (2)	96.2 (8)

Ionosphere	95.9 (2)	94.89 (3)	94.89 (3)	90.4 (7)	94.32 (4)	91.82 (6)	89 (8)	95 (2)	86.72 (10)	92.2 (5)	<b>96</b> <b>(1)</b>	86.9 (9)
KrVsKpEW	92.8 (11)	<b>98.5</b> <b>(1)</b>	97.31 (4)	97.3 (5)	95.49 (9)	96.44 (8)	91.51 (12)	97 (7)	93.92 (10)	97.08 (6)	98 (2)	97.5 (3)
WaveformEW	72.6 (11)	78.36 (5)	75.6 (8)	79.7 (4)	73.44 (9)	73.35 (10)	71.2 (12)	80 (2)	79.85 (3)	75.62 (7)	76 (6)	<b>89.4</b> <b>(1)</b>
Sonar	<b>100</b> <b>(1)</b>	99.04 (2)	94.23 (6)	84.5 (10)	91.35 (8)	93.72 (7)	85.43 (9)	96 (4)	76.38 (12)	94.97 (5)	97 (3)	84 (11)
PcnglungEW	<b>100</b> <b>(1)</b>	91.89 (4)	91.89 (4)	82.7 (8)	83.33 (7)	87.75 (5)	72.9 (10)	96 (2)	87.66 (6)	77.12 (9)	94 (3)	66.5 (11)
Average Accuracy	92.9	91.8	91.4	90.3	89.8	90.7	83.7	<b>93.2</b>	87.7	90.3	92.3	89.7

Table 5.25

Number of selected features results on UCI datasets

Dataset	G- GCACO	GA	PSO	ALO	GSA	SSA	WOA	BGWOPSO	ECWSA	GOA	WOASAT	bALO- QR
Breastcancer	4 (2)	4 (2)	4 (2)	4.7 (7)	4 (2)	<b>3.8</b> <b>(1)</b>	5.4 (8)	4.4 (5)	4.5 (6)	4.2 (4)	4.2 (4)	4.05 (3)
Tic-tac-toe	5 (2)	5 (2)	6 (4)	5 (2)	4 (1)	6 (4)	10.8 (7)	5.2 (3)	7.74 (6)	6 (4)	5.2 (3)	6.48 (5)
Exactly	<b>4</b> <b>(1)</b>	6 (4)	6 (4)	5.75 (2)	4 (1)	7.2 (6)	6 (4)	6 (4)	7.1 (5)	6 (4)	6 (4)	5.85 (3)
Exactly2	1 (1)	<b>1</b> <b>(1)</b>	<b>1</b> <b>(1)</b>	1.5 (2)	<b>1</b> <b>(1)</b>	2.7 (4)	5.7 (8)	1.6 (3)	9 (10)	5.4 (7)	2.8 (5)	5.85 (9)
HeartEW	<b>3</b> <b>(1)</b>	5 (2)	<b>3</b> <b>(1)</b>	8.6 (6)	<b>3</b> <b>(1)</b>	5.8 (4)	8.6 (6)	5.8 (4)	9.4 (7)	7.4 (5)	5.4 (3)	5.8 (4)

M-of-n	<b>4.6</b> (1)	6 (4)	6 (4)	6 (4)	5 (2)	7.1 (5)	9.7 (6)	6 (4)	5 (2)	6 (4)	6 (4)	5.85 (3)
WineEW	<b>3.5</b> (1)	4 (2)	5 (3)	5.4 (4)	4 (2)	6.3 (7)	8.8 (10)	6 (6)	6 (6)	6.8 (9)	6.4 (8)	5.46 (5)
CongressEW	4 (3)	<b>2</b> (1)	3 (2)	6.6 (10)	4 (3)	5.7 (8)	10.3 (11)	4.4 (6)	5.6 (7)	4.1 (4)	6.4 (9)	4.32 (5)
Vote	3.1 (2)	5 (7)	<b>3</b> (1)	6.6 (11)	4 (4)	4.8 (5)	7.6 (12)	3.4 (3)	6 (10)	4.9 (6)	5.2 (8)	5.6 (9)
Zoo	4.4 (2)	<b>4</b> (1)	5 (3)	5.7 (6)	6 (8)	6.7 (9)	9.9 (12)	6.8 (10)	8 (11)	5.4 (4)	5.6 (5)	5.92 (7)
Lymphography	<b>5</b> (1)	<b>5</b> (1)	<b>5</b> (1)	7.3 (4)	6 (2)	10.3 (9)	10.5 (10)	9.2 (8)	7.7 (6)	7.4 (5)	7.2 (3)	9 (7)
SpectEW	5.5 (2)	<b>5</b> (1)	6 (3)	7.6 (4)	5 (1)	10.9 (10)	11.2 (11)	8.4 (7)	7.8 (6)	8.7 (8)	9.4 (9)	7.7 (5)
BreastEW	<b>8</b> (1)	<b>8</b> (1)	9 (2)	13.8 (8)	10 (3)	16.7 (9)	21 (10)	13.6 (7)	15 (8)	13.2 (6)	11.6 (4)	12.6 (5)
Ionosphere	<b>4.7</b> (1)	7 (2)	7 (2)	11.7 (6)	9 (3)	15.8 (10)	21.4 (11)	13 (8)	9.52 (5)	9.1 (4)	12.8 (7)	13.6 (9)
KrVsKpEW	<b>10</b> (1)	<b>11</b> (2)	12 (3)	16.1 (7)	14 (4)	20.4 (10)	28 (11)	15.8 (6)	13 (3)	16.9 (8)	18.4 (9)	14.4 (5)
WavefonnEW	<b>9.7</b> (1)	15 (4)	15 (4)	20.5 (5)	14 (2)	22.9 (9)	33.2 (10)	14.2 (3)	14 (2)	21.5 (8)	20.6 (6)	20.8 (7)
Sonar	<b>11.9</b> (1)	19 (2)	22 (4)	26.6 (7)	24 (5)	33.4 (10)	43.4 (11)	31.2 (9)	20 (3)	30.5 (8)	26.4 (6)	24 (5)
PcnglungEW	<b>30.6</b> (1)	84 (3)	130 (6)	133.1 (8)	140 (9)	171.6 (11)	144.3 (10)	130.8 (7)	65 (2)	95.7 (4)	127.4 (5)	130 (6)
Average # of features	<b>6.778</b>	10.889	13.778	16.252	14.5	19.895	21.989	15.878	12.243	14.4	15.945	15.96

Figures 5.6 and 5.7 depict the graphical comparison of the average classification accuracies and the average number of selected features obtained by the G-GCACO algorithm and 11 benchmark algorithms on the UCI datasets. As compared to other benchmark algorithm, it clearly shows from Figures 5.6 and 5.7, that the G-GCACO algorithm attains a competitively average classification accuracy and the lowest number of features.

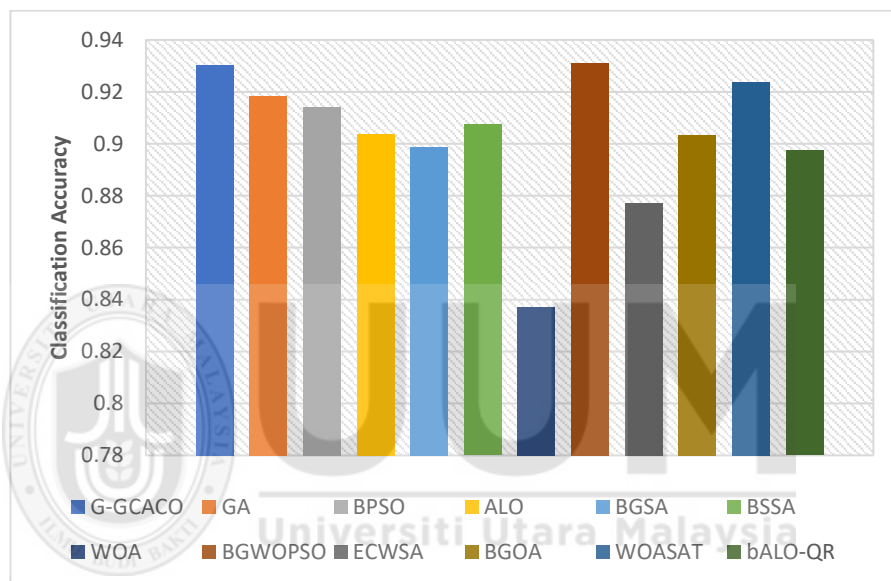


Figure 5.6: Graphical comparison for the average classification accuracy on UCI data

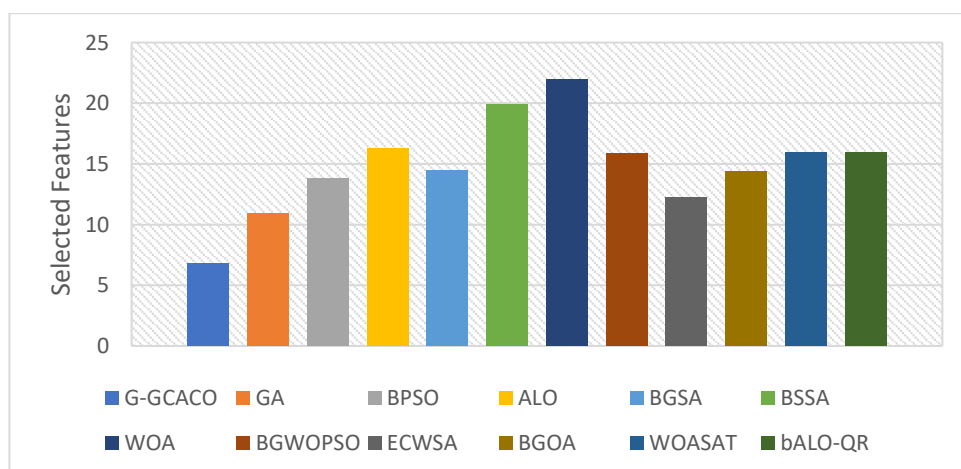


Figure 5.7: Graphical comparison for the average selected features on UCI data.

The proposed G-GCACO algorithm's performance is also evaluated using the Friedman test with Holm's post-hoc test rank. In this test, the performance of the best algorithm is indicated by smallest values for the accuracy and number of selected features. Thus, it can be seen in Table 5.26, the best algorithm for the classification accuracy and the number of selected features are the BGWOPSO and G-GCACO algorithms respectively as highlighted.

Results in Table 5.26 are reproduced in Figure 5.8, where the performance metrics feature is plotted against accuracy. The best algorithm is the one that is the nearest to the origin which indicates that it has achieved a balance between classification accuracy and the number of selected features. In this figure, it can be observed that the proposed G-GCACO algorithm is capable of increasing classification accuracy while at the same time able to reduce the number of features.

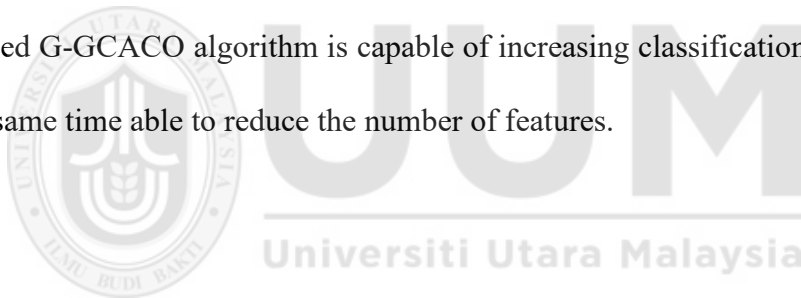


Table 5.26

*Performance rank on UCI datasets*

	<b>G-GCACO</b>	<b>GA</b>	<b>PSO</b>	<b>ALO</b>	<b>GSA</b>	<b>SSA</b>	<b>WOA</b>	<b>BGWOPSO</b>	<b>ECWSA</b>	<b>GOA</b>	<b>WOASAT</b>	<b>bALO-QR</b>
<b>Accuracy</b>	3.667	3.834	5.167	5.278	6.667	5.612	9.667	<b>3.223</b>	7.056	5.389	4.056	5.278
<b>Selected features</b>	<b>1.389</b>	2.334	2.778	5.723	3	7.278	9.334	5.723	5.834	5.667	5.667	5.667

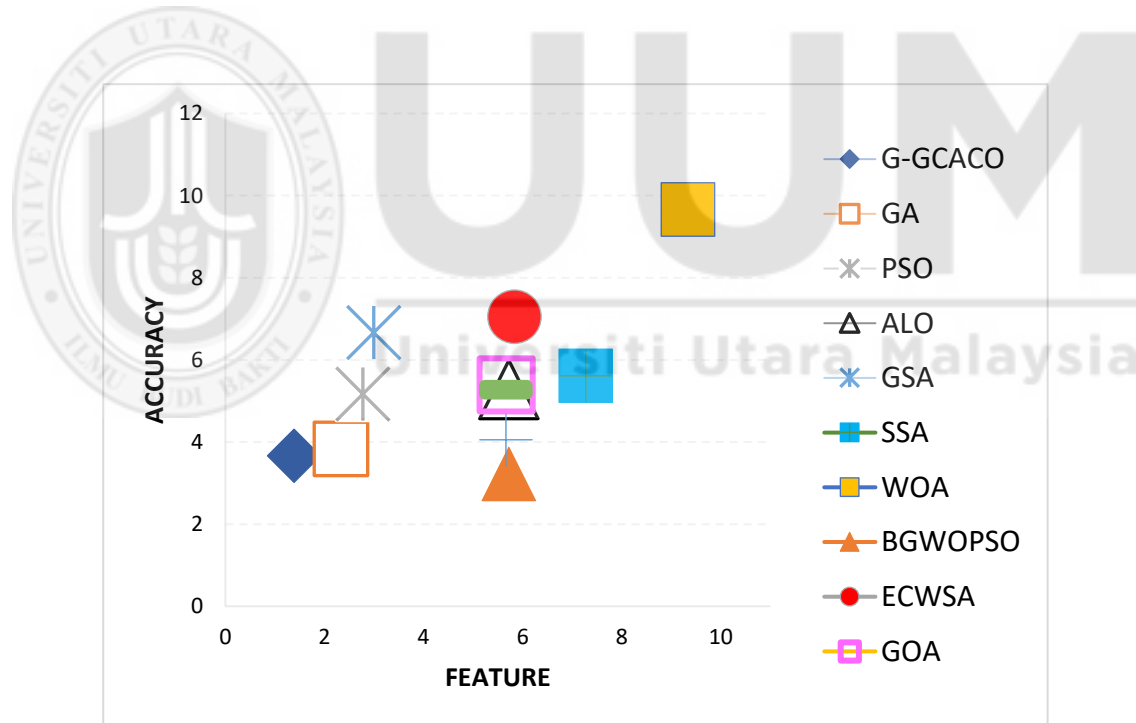


Figure 5.8: Performance rank plot using UCI datasets

### 5.5.2 Experiment on the Microarray Datasets

Tables 5.27 and 5.28 show the experimental results of the average classification accuracy and the average number of selected features respectively, using the k-NN classifier on three (3) microarray datasets. In both tables, the best result for each dataset is highlighted while the figures in parentheses indicate the rank of the algorithms.

Tables 5.27 shows that the G-GCACO obtains the best classification accuracy for large-sized datasets such as microarray datasets (i.e., DLBCL, SRBCT, and Leukemia). Thus, the G-GCACO outperforms other algorithms in all datasets in terms of classification accuracy. Furthermore, the proposed algorithm achieves the best results compared with the other algorithms in terms of the average classification accuracy for all the three (3) datasets.

Table 5.27

*Classification accuracy result on microarray datasets*

Dataset	G-GCACO	GA	PSO	GSA	SSA	ALO	HHO
Leukaemia	<b>100 (1)</b>	93.52 (2)	86.05 (5)	76.11 (6)	<b>100 (1)</b>	86.23 (4)	92.23 (3)
DLBCL	<b>100 (1)</b>	93 (3)	82 (5)	87 (4)	77 (6)	87 (4)	94.7 (2)
SRBCT	<b>100 (1)</b>	88 (6)	96.52 (3)	97 (2)	88.5 (5)	94 (4)	<b>100 (1)</b>
Average Accuracy	<b>100</b>	91.5	881	86.7	88.5	89	95.6

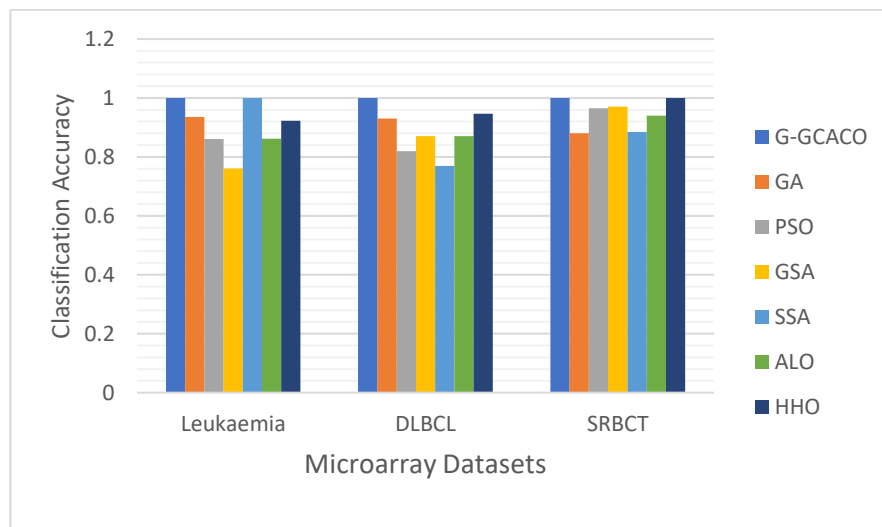
From Table 5.28 it can be seen that the G-GCACO algorithm obtains the best results in all datasets for the number of selected features.

Table 5.28

*Results of number of selected features on microarray datasets*

Dataset	G-GCACO	GA	PSO	GSA	SSA	ALO	HHO
Leukaemia	<b>614.9 (1)</b>	5111 (5)	5198 (6)	1472 (2)	5228 (7)	5073 (4)	3979 (3)
DLBCL	<b>68.1 (1)</b>	2452 (2)	2505 (3)	2558 (4)	2802 (5)	3476 (6)	1741 (2)
SRBCT	<b>53.3 (1)</b>	949 (4)	865 (3)	3872 (7)	1042 (5)	1071 (6)	798 (2)
Average # of features	<b>245.43</b>	2837.33	2856	2634	3024	3206.66	2172.66

The results in Table 5.27 and 5.28 are again shown in Figures 5.9 and 5.10, where the average classification accuracy and the average number of selected features are plotted against the microarray datasets.



*Figure 5.9: Graphical comparison for the average classification accuracy on microarray data*

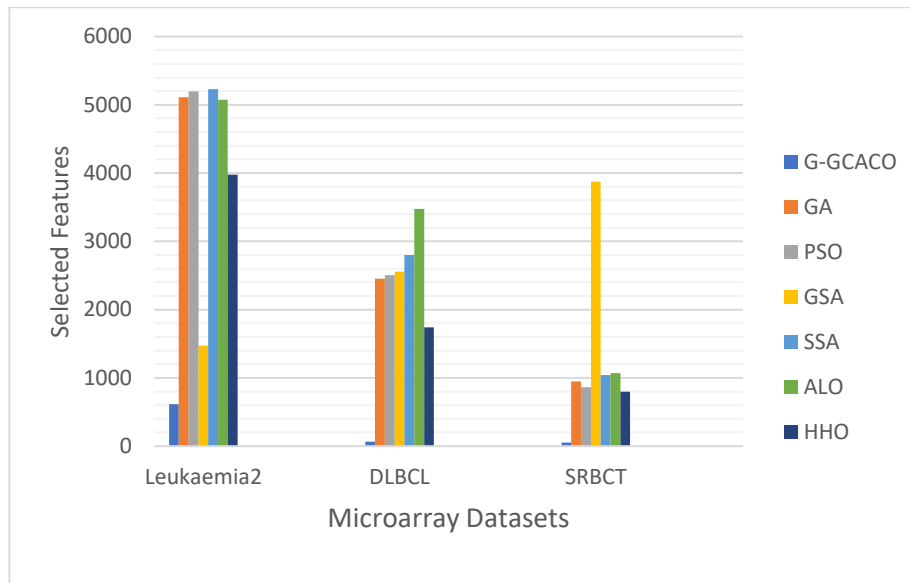


Figure 5.10: Graphical comparison for the average selected features on microarray data

Figures 5.11 and 5.12 show the graphical comparison of the average classification accuracies and the average number of selected features achieved by the G-GCACO algorithm and ten (10) benchmark algorithms on the microarray datasets. As compared to other benchmark algorithms, it clearly shows from Figures 5.11 and 5.12, that the G-GCACO algorithm attains the highest average classification accuracy and the lowest number of features.

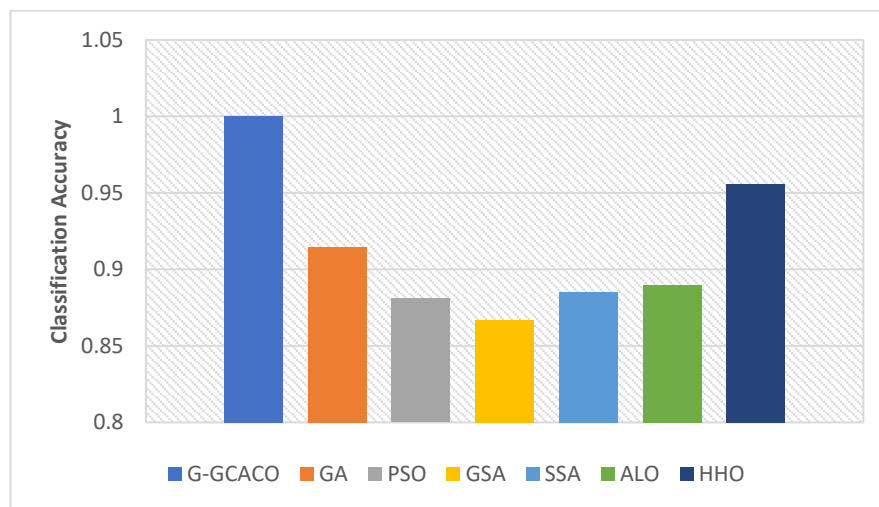


Figure 5.11: Graphical comparison for the average classification accuracy on microarray data

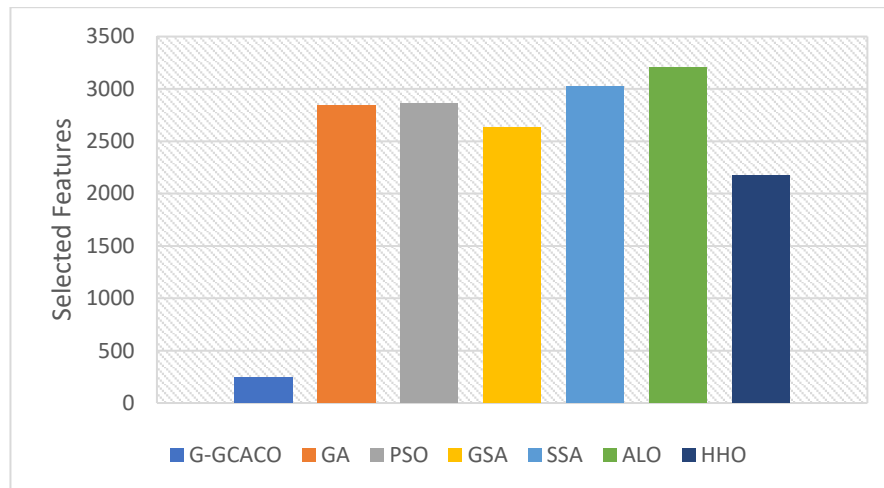


Figure 5.12: Graphical comparison for the average classification accuracy on microarray data

The performance of the proposed G-GCACO algorithm is also compared using the Friedman test with Holm's post-hoc test rank. Table 5.29 displays the results of the ranking. In this test, the smallest value indicates the highest rank (highlighted).

Table 5.29

*Performance rank on microarray datasets*

	<b>G-GCACO</b>	<b>GA</b>	<b>PSO</b>	<b>GSA</b>	<b>SSA</b>	<b>ALO</b>	<b>HHO</b>
<b>Accuracy</b>	<b>1</b>	3.66	4.33	4	4	4	2
<b>Selected features</b>	<b>1</b>	3.66	4	4.33	5.66	5.33	2.33

Table 5.29 findings are reproduced in Figure 5.13, where the performance metric feature is plotted against accuracy. The best algorithm is the one that is nearest to the origin which indicates that the algorithm achieves a balance between classification accuracy and the number of selected features.

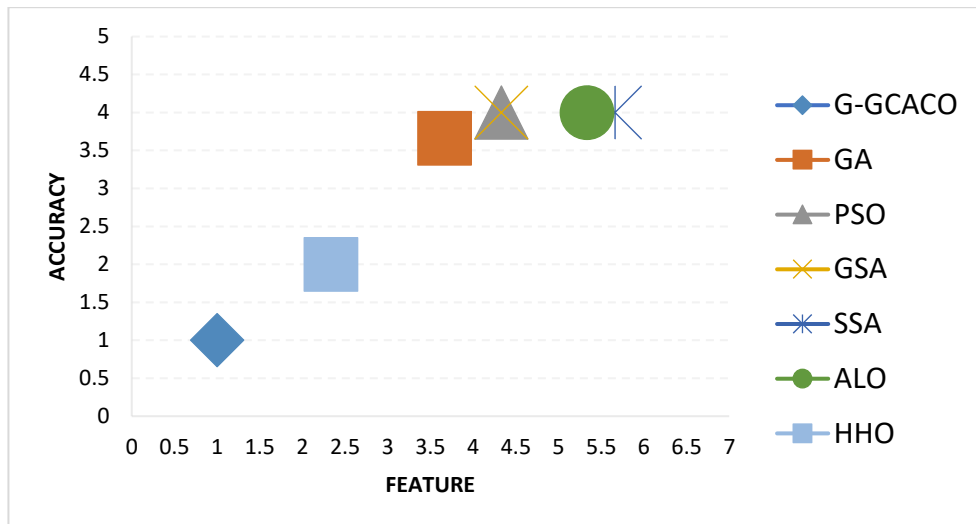


Figure 5.13: Performance rank plot using microarray datasets

In Figure 5.13 the G-GCACO is the best algorithm that balances the classification accuracy and the number of selected features. This reflects the advantage of the proposed genetic method (G-GCACO), which incorporates an appropriate selecting method that reduces the probability of selecting redundant features, thus the natural patterns in the data will not be blurred by noise. On the other hand, the genetic method can automatically determine the number of selected features in the final subset which contains the significant features. Thus, the achieved experimental results prove that the proposed G-GCACO has the ability to determine a suitable subset for the several application domains (i.e., microarray and the other type of UCI datasets) by considering the most informative features to enhance classification accuracy.

## 5.6 Results and Analysis of the EGCACO Algorithm

This section displays the results of the EGCACO algorithm's performance evaluation. The experiments being carried out in accordance with the experimental design as outlined. Since the EGCACO algorithm combined EC-GCACO, A-GCACO, and G-GCACO algorithms, it utilised the same parameters in each algorithm (i.e. Table 5.1,

Table 5.12, and Table 5.23). Experiments, repeated ten (10) times, give each dataset a turn as the testing dataset, generating ten particular sets of performance statistics for the classification accuracy and number of features selected. The standard deviations for each of the performance statistics are also calculated.

### **5.6.1 Performance of EGCACO on the UCI Datasets**

Tables 5.30 and 5.31 show the EGCACO performance evaluated based on average classification accuracy and the average number of selected features from the subset. The UCI datasets are used in the experiments.

In Table 5.30, the best result for the classification in each dataset is highlighted while the figures in parentheses indicate the rank of the algorithms. The EGCACO algorithm performs the best in eight (8) (44.47%) of the datasets. For the remaining ten (10) datasets, the proposed algorithm shows competitive results. Furthermore, the average classification accuracy over all the datasets for each algorithm is also calculated. For this metric, the proposed EGCACO outperforms other algorithms (i.e., first rank) with the exception of the HSGW algorithm. However, the difference in results between the EGCACO and HSGW is very small.

Table 5.30

*Classification accuracy results on UCI datasets*

Dataset	EGCACO	HSGW	RSGW	ASGW	BGWOPSO	WOA-CM	WOASAT	bALO-QR
Breastcancer	98.1 (3)	<b>98.6</b> <b>(1)</b>	97.1 (6)	98.5 (2)	98 (4)	96.8 (8)	97 (7)	97.4 (5)
BreastEW	<b>100</b> <b>(1)</b>	98.1 (4)	98.2 (3)	<b>100</b> <b>(1)</b>	97 (7)	97.1 (6)	98 (5)	96.2 (8)
CongressEW	98 (2)	97.5 (3)	96.1 (5)	<b>99.4</b> <b>(1)</b>	98 (2)	95.6 (6)	98 (2)	97.2 (4)
Exactly	99.7 (3)	<b>100</b> <b>(1)</b>	99.7 (3)	99.9 (2)	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	91.2 (4)
Exactly2	75.3 (6)	<b>81.5</b> <b>(1)</b>	77.9 (2)	77.7 (3)	76 (4)	742 (8)	75 (7)	76 (5)
HeartEW	84.16 (5)	<b>92.3</b> <b>(1)</b>	84.8 (4)	83.1 (6)	85 (3)	80.7 (7)	85 (3)	88.4 (2)
ionosphereEW	<b>97.9</b> <b>(1)</b>	94.4 (6)	97.8 (2)	97.2 (3)	95 (5)	92.6 (7)	96 (4)	86.9 (8)
KrvskpEW	97 (6)	97.3 (3)	97.2 (4)	97.1 (5)	97 (6)	97.2 (4)	<b>98</b> <b>(1)</b>	97.5 (2)
Lymphography	88.9 (5)	<b>93.4</b> <b>(1)</b>	89.3 (3)	88.4 (7)	92 (2)	852 (8)	89 (4)	88.6 (6)
M-of-n	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	99.1 (2)	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>
PenglungEW	<b>100</b> <b>(1)</b>	94.2 (3)	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	96 (2)	79.2 (5)	94 (4)	66.5 (6)
SonarEW	<b>98.11</b> <b>(1)</b>	96.4 (4)	97.9 (2)	94.8 (6)	96 (5)	91.9 (7)	97 (3)	84 (8)
SpectEW	<b>90</b> <b>(1)</b>	86.2 (6)	81.5 (7)	87 (4)	88 (3)	86.6 (5)	88 (3)	<b>90</b> <b>(1)</b>
Tic-tac-toe	82 (4)	82.8 (3)	85.9 (2)	<b>86.5</b> <b>(1)</b>	81 (6)	78.5 (8)	79 (7)	80 (5)
Vote	98.5 (2)	98.3 (4)	<b>99.6</b> <b>(1)</b>	98.4 (3)	97 (5)	93.9 (7)	97 (5)	94.8 (6)
WavefonnEW	76 (3)	74.8 (6)	75.7 (4)	74.6 (7)	80 (2)	75.3 (5)	76 (3)	<b>89.4</b> <b>(1)</b>
WineEW	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	95.9 (3)	99 (2)	<b>100</b> <b>(1)</b>
Zoo	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	<b>100</b> <b>(1)</b>	98 (2)	97 (4)	96.1 (5)
Average Accuracy	93.5	<b>93.6</b>	93.2	93.4	93.2	89.8	92.3	90.1

According to Table 5.31, the EGCACO algorithm outperforms the other seven (7) benchmark FS algorithms in this classification task in terms of the number of selected features. The proposed EGCACO obtains the best result for 16 of the 18 UCI datasets

(88.89%). It is worth mentioning that the EGCACO obtains competitive results for the remaining two (2) datasets (i.e., SpectEW and Vote). In the case of some datasets with respect to other FS algorithms, the EGCACO algorithm fails to achieve the highest accuracy but, in terms of the number of selected features, it surpasses all these algorithms by a huge margin and is the first ranked. The average number of selected features for all the datasets is calculated for each of the algorithms as displayed in the last row which shows that the proposed EGCACO has the least number of selected features. This shows that, even with a minimal number of features, the proposed EGCACO manages to attain the highest classification accuracy.

Table 5.31

*Number of selected features on UCI datasets*

Dataset	EGCACO	HSGW	RSGW	ASGW	BGWOPSO	WOA-CM	WOASAT	bALO-QR
Breastcancer	4 (1)	5 (7)	5.933 (8)	4.867 (6)	4.4 (5)	4.302 (4)	4.2 (3)	4.05 (2)
BreastEW	11 (1)	16.667 (6)	17.5 (7)	15.833 (5)	13.6 (4)	15.81 (4)	11.6 (2)	12.6 (3)
CongressEW	3.38 (1)	8.867 (7)	9.7 (8)	8.833 (6)	4.4 (3)	6.448 (5)	6.4 (4)	4.32 (2)
Exactly	4 (1)	6.7 (5)	7.1 (7)	6.867 (6)	6 (3)	6.045 (4)	6 (3)	5.85 (2)
Exactly2	1.5 (1)	9.033 (7)	9.2 (8)	7.933 (6)	1.6 (2)	5.252 (4)	2.8 (2)	5.85 (5)
HeartEW	3 (1)	8.767 (7)	6.133 (4)	6.367 (5)	5.8 (3)	6.995 (6)	5.4 (2)	5.8 (3)
ionosphereEW	5 (1)	18.167 (7)	20.5 (8)	17.3 (6)	13 (3)	14.416 (5)	12.8 (2)	13.6 (4)
KrvskpEW	12.83 (1)	24.8 (7)	24.8 (7)	24.5 (6)	15.8 (3)	18.54 (5)	18.4 (4)	14.4 (2)
Lymphography	5.9 (1)	10.567 (6)	10.567 (6)	11.2 (7)	9.2 (5)	8.208 (3)	7.2 (2)	9 (4)
M-of-n	5 (1)	6.8 (6)	7.1 (8)	6.867 (7)	6 (3)	6.006 (4)	6.4 (5)	5.46 (2)
PenglungEW	37 (1)	165.333 (6)	181.2 (8)	170.3 (7)	130.8 (5)	128.05 (3)	127.4 (2)	130 (4)
SonarEW	12 (1)	34.3 (5)	36.433 (8)	35.3 (6)	31.2 (4)	35.64 (7)	26.4 (3)	24 (2)

SpectEW	8.50 (4)	10.233 (7)	13.3 (8)	10.167 (6)	8.4 (3)	<b>8.052</b> <b>(2)</b>	9.4 (5)	7.7 (1)
Tic-tac-toe	<b>5.2</b> <b>(1)</b>	7 (5)	7 (5)	7 (5)	<b>5.2</b> <b>(1)</b>	6.903 (4)	<b>5.2</b> <b>(1)</b>	6.48 (3)
Vote	4.17 (2)	7.567 (6)	8.8 (7)	8.967 (8)	<b>3.4</b> <b>(1)</b>	7.408 (5)	5.2 (3)	5.6 (4)
WavefonnEW	<b>11.33</b> <b>(1)</b>	26.933 (7)	27.533 (8)	25.833 (6)	14.2 (2)	25.4 (5)	20.6 (3)	20.8 (4)
WineEW	<b>3</b> <b>(1)</b>	4.533 (2)	5.867 (4)	5.933 (5)	6 (6)	6.4 (7)	6.4 (7)	5.46 (3)
Zoo	<b>5</b> <b>(1)</b>	5.533 (3)	5.3 (2)	7.6 (8)	6.8 (7)	6 (6)	5.6 (4)	5.92 (5)
Average # of features	<b>7.878</b>	20.934	22.442	21.203	15.878	17.548	15.967	15.938

Figures 5.14 and 5.15 provide a graphical comparison of the average classification accuracy and number of selected features achieved by the EGCACO algorithm and seven (7) benchmark algorithms on the UCI datasets. Figures 5.14 and 5.15 clearly illustrate superior performances of the EGCACO algorithm where it is able to achieve the second highest average classification accuracy and the least number of selected features. Observing both Figure 5.14 and Figure 5.15, the top four (4) FS algorithms are the RSGW, EGCACO, ASGW, and HSGW.

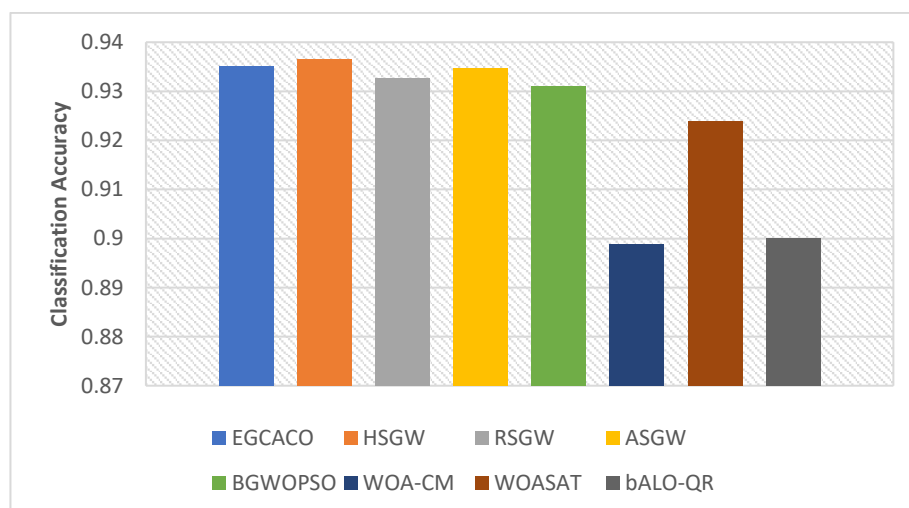
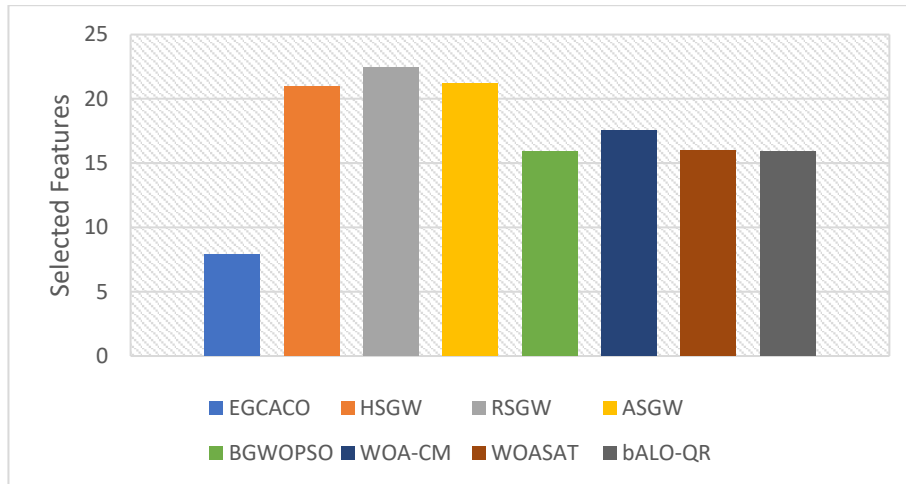


Figure 5.14: Graphical comparison for the average classification accuracy on UCI data



*Figure 5.15:* Graphical comparison for the average selected features on UCI data

Table 5.32 shows the performance results obtained by the top four (4) algorithms in terms of three (3) metrics (i.e., precision, recall, and f-score) on 18 UCI datasets. It can be seen that the proposed EGCACO algorithm obtains the best performance for 11 of the 18 datasets in Precision and ten (10) datasets for both F-measure and Recall.

Table 5.33 displays the performance of the proposed EGCACO algorithm and the benchmark algorithms using the Friedman test with Holm's post-hoc test rank. These tests calculate the rank for the algorithm in terms of the accuracy and the number of features. The smallest value for the accuracy and the number of selected features indicates the highest rank. Thus, for this case, the best algorithm for both the accuracy and the number of selected features is the proposed EGCACO algorithm.

Figure 5.16 shows the performance metric feature plotted against accuracy, which is similar to the findings in Table 5.33. The best algorithm is the one that is closest to the origin and has found a fair balance between classification accuracy and the set of

features selected. The proposed EGCACO algorithm obtains the best balance between accuracy and number of features among the algorithms.



Table 5.32

*Average of precision, recall, and f-score of top four (4) FS algorithms*

Datasets	HSGW			RSGW			ASGW			EGCACO		
	precisio	recall	f-score	precisio	recall	f-score	precisio	recall	f-score	precisi	recall	f-score
Breastcancer	<b>1</b>	<b>1</b>	<b>1</b>	0.9552	0.9678	0.9609	0.98	0.9891	0.9843	<b>1</b>	<b>1</b>	<b>1</b>
BreastEW	0.9539	0.9335	0.9422	0.9672	0.9573	0.9619	0.936	0.9315	0.9337	<b>0.9689</b>	<b>0.9736</b>	<b>0.9712</b>
CongressEW	0.9512	0.9512	0.9512	0.9662	0.9604	0.9632	<b>0.9714</b>	<b>0.9815</b>	<b>0.9759</b>	0.9384	0.9282	0.9332
Exactly	0.8143	0.8086	0.8113	0.7779	0.7779	0.7779	<b>0.9006</b>	<b>0.8734</b>	<b>0.8852</b>	0.8966	0.8593	0.8775
Exactly2	0.5505	0.5192	0.5007	0.6241	<b>0.5718</b>	0.5766	0.5609	0.534	0.5291	<b>0.6311</b>	0.5603	<b>0.6168</b>
HeartEW	0.5907	0.5917	0.5903	0.739	0.7417	0.7393	<b>0.85</b>	<b>0.85</b>	<b>0.85</b>	0.8037	0.8	0.8018
IonosphereE	0.9018	0.78	0.8045	<b>0.9327</b>	<b>0.86</b>	<b>0.8825</b>	0.9245	0.84	0.8639	0.8649	0.8304	0.8472
KrvskpEW	0.9646	0.9634	0.9639	0.9765	0.9764	0.9765	0.9704	<b>0.97</b>	0.9702	<b>0.98</b>	<b>0.985</b>	<b>0.9824</b>
Lymphograph	0.4234	0.4409	0.432	0.3946	0.4006	0.3944	0.4039	0.4069	0.401	<b>0.5216</b>	<b>0.5257</b>	<b>0.5236</b>
M-of-n	0.9379	0.9329	0.9353	0.8527	0.8443	0.8481	<b>0.9922</b>	<b>0.9865</b>	<b>0.9892</b>	0.899	0.8765	0.8876
PenglungEW	0.881	0.9048	0.8762	0.7929	0.7857	0.7732	0.7571	0.8095	0.7651	<b>0.8812</b>	<b>0.9241</b>	<b>0.9021</b>
SonarEW	<b>0.8413</b>	0.8249	0.8286	0.8612	<b>0.8513</b>	<b>0.8542</b>	0.8333	0.8295	0.8309	<b>0.864</b>	<b>0.8612</b>	<b>0.8625</b>
SpectEW	0.7778	0.7378	0.7545	0.7473	0.6924	0.7126	<b>0.7976</b>	<b>0.8171</b>	<b>0.8066</b>	0.7656	0.7516	0.7585
Tic-tac-toe	0.6864	0.645	0.6515	0.8995	0.8204	0.8433	0.9139	0.806	0.8325	<b>0.9789</b>	<b>0.8672</b>	<b>0.9196</b>
Vote	0.951	0.943	0.9467	0.9259	0.9459	0.9314	0.9647	<b>0.9647</b>	<b>0.9647</b>	<b>0.976</b>	0.941	0.9581
WaveformEW	0.811	0.8111	0.811	0.8202	0.8195	0.8187	<b>0.8367</b>	0.8361	<b>0.836</b>	0.8232	<b>0.844</b>	0.8334
WineEW	0.9697	<b>0.9762</b>	0.9718	0.9744	0.9762	<b>0.9743</b>	0.9697	0.9762	0.9718	<b>0.9751</b>	<b>0.9831</b>	<b>0.9790</b>
Zoo	<b>1</b>	<b>1</b>	<b>1</b>	0.9841	0.9286	0.944	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Average</b>	0.833	0.820	0.820	0.843	0.826	0.829	0.864	0.855	0.855	<b>0.876</b>	<b>0.864</b>	<b>0.869</b>

Table 5.33

*Rank performance on UCI datasets*

	<b>EGCACO</b>	<b>HSGW</b>	<b>RSGW</b>	<b>ASGW</b>	<b>BGWOPSO</b>	<b>WOA-CM</b>	<b>WOASAT</b>	<b>WOASAT-2</b>	<b>bALO-QR</b>
<b>Accuracy</b>	<b>2.612</b>	2.778	2.889	3.056	3.334	5.556	3.667	4.334	2.612
<b>Selected features</b>	<b>1.223</b>	5.889	6.723	6.167	3.167	3.5	4.612	3.166	3.056

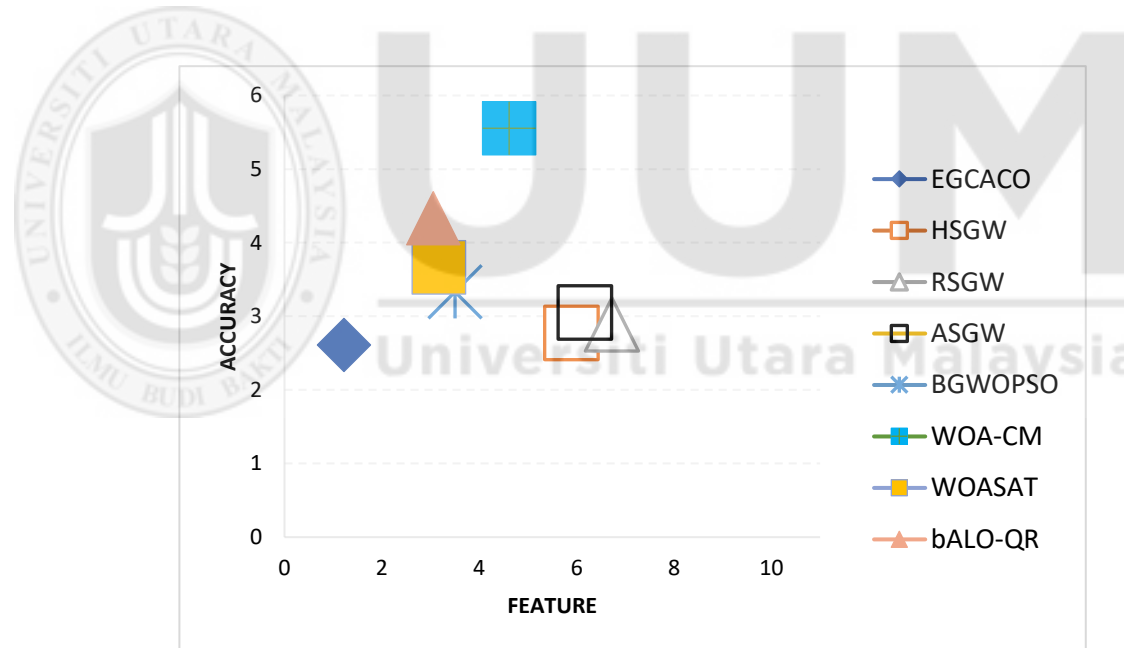


Figure 5.16: Performance rank plot using UCI datasets

### 5.6.2 Performance of EGCACO on the Microarray Datasets

Table 5.34 shows the experimental results of the average classification accuracy (Acc) and the standard deviation (Std) of the proposed EGCACO and eight (8) other FS algorithms. The experimental results are used to determine the best algorithm performance in terms of classification accuracy for microarray by using k-NN, classifier. The best result is highlighted for each dataset while the figures in parentheses indicate the rank of the algorithms.

Once again, the EGCACO algorithm outperforms its peers in terms of classification accuracy in microarray datasets. The proposed algorithm achieves the best results (first rank) in seven (7) datasets out of nine (9) (i.e., 77.78%) datasets. For the remaining two (2) datasets (i.e., CLL\_SUB\_111 and Lung), the proposed EGCACO algorithm comes second after the GA and DE algorithms. Moreover, the proposed EGCACO outperforms other algorithms (i.e., first rank) in terms of average classification accuracy, standard deviation, over the comparable benchmark algorithms. In addition, the results show the standard deviation values calculated for the accuracy results have very competitive small standard deviation values when compared to the other algorithms, indicating the stability of its performance.

Table 5.35 shows the experimental results of the average selected features using the proposed EGCACO and eight (8) benchmark FS algorithms. The best result for each dataset is highlighted.

Based on the results, the EGCACO algorithm demonstrates the lowest number of selected features in six (6) out of nine (9) datasets (i.e., 66.67%). In the case of the

remaining datasets, the ASO algorithm ranks first followed by the proposed EGCACO algorithm. The performances of the algorithms on all the datasets as a whole are also calculated as shown in the last row of the table. The proposed algorithm obtains the best (lowest) number of features. Thus, the proposed EGCACO performs well on these datasets which are considered as large size datasets.



Table 5.34

*Classification accuracy and standard deviation results on microarray datasets*

Dataset		EGCACO	TGA	GWO	ASO	PSO	HHO	GA	DE	BA
CLL_SUB_111	Avg	66.84 (2)	59.09 (3)	59.09 (3)	50 (5)	50 (5)	54.55 (4)	<b>68.18 (1)</b>	68.18 (2)	45.45 (6)
	std	0.0137	0.0296	0.0288	0.037	0.0282	0.0317	0.0262	0.0268	0.0264
Colon	Avg	<b>97.41 (1)</b>	75 (4)	83.33 (2)	81.67 (3)	75 (4)	75 (4)	83.33 (2)	75 (4)	75 (4)
	std	0.0027	0.0024	0.0052	0.0086	0.0025	0.0165	0.0036	0.0144	0.0026
Leukemia	Avg	<b>100 (1)</b>	92.86 (2)	85.71 (3)	71.43 (5)	85.71 (3)	85.71 (3)	92.86 (2)	78.57 (4)	85.71 (3)
	std	0.0067	0.0015	0.0022	0.0064	0.0012	0.0102	0.002	0.0023	0.0012
Lung	Avg	92.21 (2)	92 (3)	91.5 (3)	87.5 (5)	91.5 (3)	90 (4)	91.5 (3)	<b>92.5 (1)</b>	91.5 (3)
	std	0.0036	0.002	0.0034	0.0064	0.0019	0.0091	0.003	0.0036	0.0016
Lung_discrete	Avg	<b>89.86 (1)</b>	71.43 (4)	78.57 (3)	64.29 (5)	78.57 (3)	71.43 (4)	78.57 (3)	85.71 (2)	85.71 (2)
	std	0.0095	0.0064	0.0136	0.0096	0.0067	0.0124	0.0072	0.0137	0.0063
Lymphoma	Avg	<b>95.10 (1)</b>	89.47 (2)	78.95 (4)	78.95 (4)	89.47 (2)	89.47 (2)	89.47 (2)	84.21 (3)	84.21 (3)
	std	0.005	0.0016	0.0023	0.0104	0.0016	0.006	0.0036	0.0022	0.0012
nci9	Avg	<b>71.66 (1)</b>	50 (2)	33.33 (4)	25 (5)	41.67 (3)	50 (2)	33.33 (4)	50 (2)	50 (2)
	std	0.0137	0.0422	0.0401	0.0393	0.0394	0.0428	0.043	0.0375	0.0403
Prostate_GE	Avg	<b>95 (1)</b>	90 (3)	<b>95 (1)</b>	85 (4)	<b>95 (1)</b>	90 (3)	90 (3)	75 (5)	85 (4)
	std	0.0016	0.0016	0.0114	0.0089	0.0015	0.0151	0.0019	0.0135	0.0014
SMK_CAN_187	Avg	<b>91.37 (1)</b>	75.68 (4)	81.08 (3)	70.27 (6)	67.57 (7)	72.97 (5)	62.16 (8)	59.46 (9)	86.49 (2)

	std	0.0112	0.024	0.0226	0.0261	0.0197	0.0261	0.017	0.0225	0.0203
<b>Average</b>	<b>Avg</b>	<b>88.82</b>	77.28	76.28	68.23	74.94	75.45	76.6	74.29	76.56
	<b>std</b>	<b>0</b>	0.0123	0.0144	0.0169	0.0114	0.0188	0.0119	0.0151	0.0112

Table 5.35

*Results of number of selected features on microarray datasets*

<b>Dataset</b>	<b>EGCACO</b>	<b>TGA</b>	<b>GWO</b>	<b>ASO</b>	<b>PSO</b>	<b>HHO</b>	<b>GA</b>	<b>DE</b>	<b>BA</b>
CLL SUB	<b>357.5</b>	5662.63	6384.14	1103.75	5630.84	4399.46	5585.41	6901.91	5623.11
111	<b>(1)</b>	(7)	(8)	(2)	(6)	(3)	(4)	(9)	(5)
Colon	<b>42</b>	946.87	979.4	125.54	924.18	626.94	907.2	1032.07	944.51
	<b>(1)</b>	(6)	(7)	(2)	(4)	(8)	(3)	(9)	(5)
Leukemia	<b>43.793</b>	3400.45	3470.59	339.51	3356.92	1961.36	3240.73	3498.41	3404.03
	<b>(1)</b>	(6)	(8)	(2)	(5)	(3)	(4)	(9)	(7)
Lung	<b>179.26</b>	1566.4	1615.38	192.23	1533.21	915.67	1463.4	1629.41	1569.27
	<b>(1)</b>	(6)	(8)	(2)	(5)	(3)	(4)	(9)	(7)
lung_discrete	30.8	137.73	149.94	<b>25.31</b>	129.2	94.9	121.74	158.4	138.7
	(2)	(6)	(8)	<b>(1)</b>	(5)	(3)	(4)	(9)	(7)
Lymphoma	365	1890.76	1931.39	<b>155.81</b>	1844.89	1022.39	1574.2	1959.52	1903.45
	(2)	(6)	(8)	<b>(1)</b>	(5)	(3)	(4)	(9)	(7)
nci9	<b>1001</b>	4879.49	5232.5	1062.24	4799.88	3932.79	4720.81	6128.74	4811.22
	<b>(1)</b>	(7)	(8)	(2)	(5)	(3)	(4)	(9)	(6)

Prostate_GE	474 (2)	2883.27 (6)	3032.71 (8)	<b>424.93</b> <b>(1)</b>	2853.46 (5)	1841.66 (3)	2833.86 (4)	3086.76 (9)	2883.41 (7)
SMK CAN 187	<b>385.7 (1)</b>	9977.48 (7)	11069.31 (8)	1945.67 (2)	9945.69 (6)	7199.64 (3)	9922.05 (4)	11519.65 (9)	9945.36 (5)
<b>Average # of features</b>	<b>319.89</b>	3482.78	3762.81	597.22	3446.47	2443.86	3374.37	3990.54	3469.22



**UUM**  
Universiti Utara Malaysia

For the microarray datasets, Figures 5.17 and 5.18 provide a graphical comparison of the average classification accuracy and number of selected features by the EGCACO algorithm and eight (8) benchmark algorithms. It can be seen that the EGCACO algorithm obtains the best average classification accuracy and the fewest number of selected features.

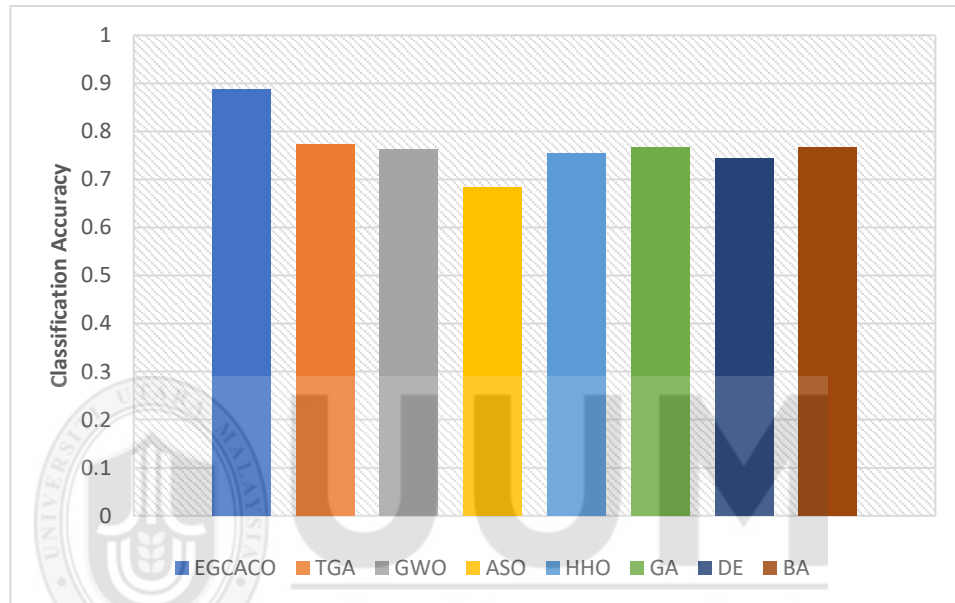


Figure 5.17: Graphical comparison for the average classification accuracy on microarray data

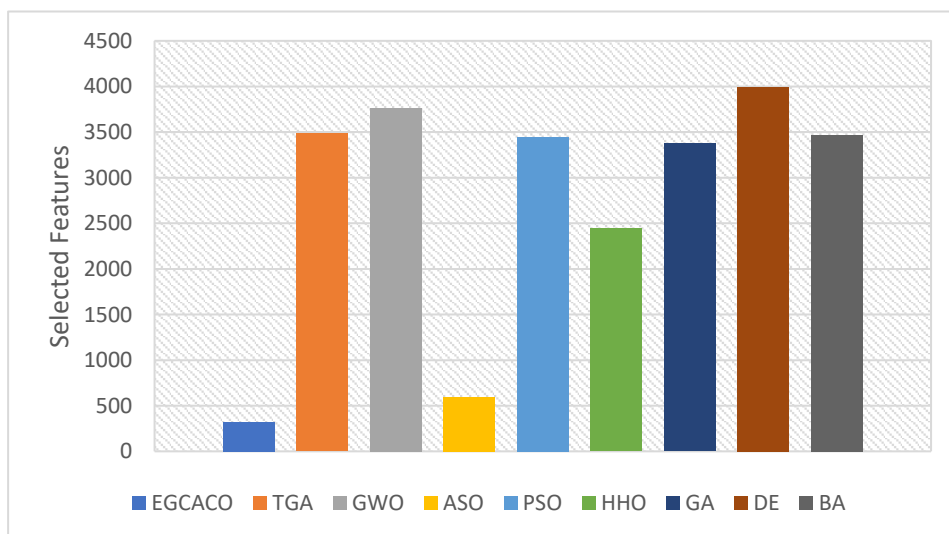


Figure 5.18: Graphical comparison for the average selected features on microarray data

The Friedman and Holm's post-hoc test rank is used to compare the performance of the proposed EGCACO algorithm. This test is to rank the algorithm based on the number of features and the highest accuracy. The highest rank is assigned to the algorithm with the least number of features and the highest accuracy. The ranking results are shown in Table 5.36. The highest ranks are highlighted. The EGCACO algorithm manages to obtain the highest rank for the classification accuracy and the number of selected features in this test.

Table 5.36

*Performance rank on microarray datasets*

	<b>EGCACO</b>	<b>TGA</b>	<b>GWO</b>	<b>ASO</b>	<b>PSO</b>	<b>HHO</b>	<b>GA</b>	<b>DE</b>	<b>BA</b>
<b>Accuracy</b>	<b>1.2223</b>	3	2.8889	4.6667	3.4445	3.4445	3.2223	3.5556	3.2223
<b>Selected features</b>	<b>1.334</b>	6.334	7.889	1.667	5.112	3.556	3.889	9	6.223

The results in Table 5.36 are displayed in Figure 5.19, where the accuracy is plotted against the number of selected features. The best algorithm that has managed to obtain a balance between classification accuracy and the selected features is indicated by the position closest to the origin. Thus, in this figure, the algorithm that has achieved the best balanced between the accuracy and the number of selected features is the proposed EGCACO.

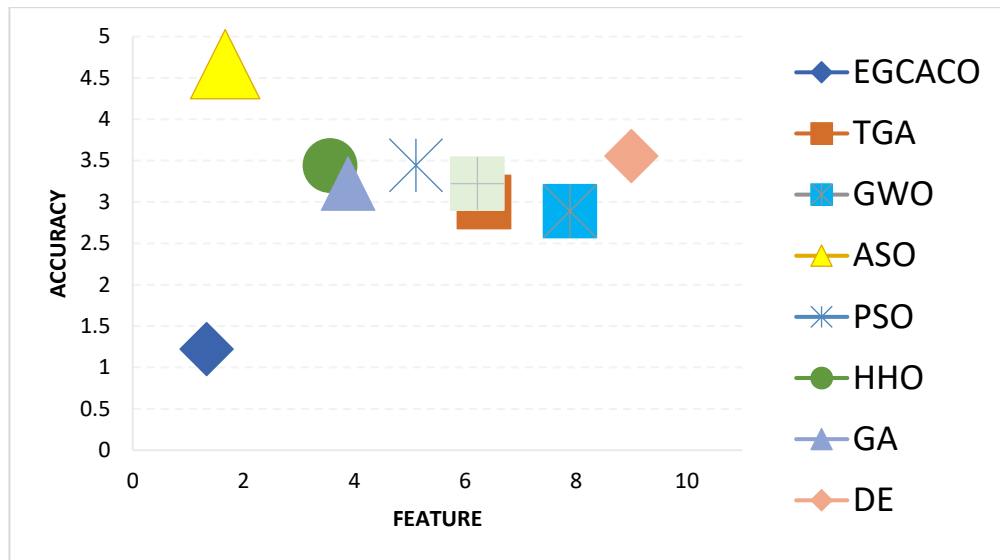


Figure 5.19: Rank plot for microarray datasets

In all the experiments conducted to evaluate the proposed EGCACO, the highest classification accuracy in medium, and large datasets (i.e., all the datasets in UCI) has been obtained. In addition, the achievement in reducing the dimension of features in such datasets is better than the several benchmark algorithms. From the above results, the proposed EGCACO has the ability to get the best result in very large datasets (i.e., Microarray). This demonstrates EGCACO is capable of search the feature space adaptively for optimal feature combinations. The EGCACO algorithm's performance can be justified by the fact that it makes use of the ACO-based feature clustering concept which benefits its intensification and diversification. Additionally, the ACO-based adaptive technique for selecting the appropriate parameter has the advantage of providing a more diverse capability in selecting the significant feature, which helps to avoid maximum redundancy and minimize the number of relevant features. The proposed EGCACO algorithm's good performance is also due to the hybridization of the genetic method in determining the final subset. This demonstrates its ability to generate sufficient diversity in finding the final subset.

## 5.7 Summary

The experimental findings demonstrate that the proposed EGCACO surpasses other FS algorithms in terms of classification performance and the number of selected features in both benchmark and microarray datasets. Integrating the ACO-based feature clustering method, ACO-based adaptive technique, and the genetic method to determine the final subset in the proposed EGCACO, a significant reduction in the number of selected features was obtained without losing important information, resulting in high classification accuracy for handling the problem of dimensionality datasets. All three modifications were tested by implementing EC-GCACCO, A-GCACCO, and G-GCACCO algorithms. Experimental results show the enhanced feature clustering method can effectively perform well in the space where the exploration to escape from local optima in grouping the highly correlated genes has been successful and was able to improve the classification accuracy. The adaptive ACO technique has the ability to select the appropriate value to be used in selecting significant features that can maximize dependency and minimize redundancy among the features and, therefore, enhance the classification accuracy of EGCACO. Furthermore, the genetic method (i.e., G-GCACCO) involves a subset determination to select an appropriate subset with the most informative features that improve the classification accuracy with the fewest number of selected features. In summary, the proposed EGCACO algorithm is able to minimize high dimensionality datasets while maintaining acceptable classification accuracy. The next chapter will summarize the significant findings and suggest future research directions.

# CHAPTER SIX CONCLUSION, LIMITATIONS AND FUTURE WORK

## 6.1 Introduction

The major purpose of this study is to develop an ACO-based algorithm for FS in DNA microarray. The proposed algorithm is made up of three components: ACO-based feature clustering, the ACO-based adaptive technique, and the genetic method. All components have contributed to the proposed EGCACO's ability to achieve high classification accuracy with the lowest number of selected features in DNA microarray datasets. To summarize the study, Section 6.2 highlights the main contributions, while Section 6.3 lists some of the limitations and recommendations for future research directions.

## 6.2 Research Contribution

This study has contributed to two main categories of contribution; knowledge and practical contributions.

### 6.2.1 Knowledge Contribution

The contributions from the research that can be categorized to the body of knowledge are:

The first contribution is the ACO-based feature clustering method which can be used to enhance the grouping of the highly correlated features into the same cluster. Through the use of the ACO-based feature clustering approach, the possibility of falling into local optima is reduced since many ants search for the best solution concurrently and stochastically. Each ant will gradually form its own cluster centres

for calculating heuristic values in an iteration level. Ants consider not only the pheromone levels but also the heuristic values of candidate nodes when selecting the next feature. This enables ants to find better solutions (i.e., groups of features). The experimental results revealed that the ACO-based feature clustering method outperforms its competitors that employ the greedy search strategy to cluster the features.

A new adaptive parameter selection technique based on ACO is introduced to control the selection of relevant features and avoid the redundancy of the features. This is based on three main aspects/traits in this technique. The first aspect is pheromone initialisation, where two pheromone matrices are initialised for the parameter value and the feature search space, respectively. This overcomes the searching complexity where the search space is reduced in size. The second aspect is the ant selection procedure, which utilises both probabilistic state transition rules and the greedy search method. This aspect clearly reinforces the appropriate dynamic adjustment of the parameter value in each selection. The last aspect is the pheromone updating rule for the quality of the selected parameter value. This enables the search of an optimal parameter value within the iterations. For most related ACO-based FS methods, the evaluation outcome demonstrated that the proposed algorithm is able to enhance classification accuracy with the proposed ACO-based adaptive selection of the parameter control, which depends on the feedback from the search behaviour.

A method for determining the final subset processes using a genetic algorithm has also been proposed. In the feature search space, the genetic method works by formulating a crossover with the subset evaluation and selection strategy for the best final subset that increases classification accuracy with fewer features. This enhancement removes

the static or predefined parameters. The experimental results have shown that the performance of the proposed genetic method is the best in obtaining the smallest number of features with the highest classification accuracy.

The EGCACO algorithm can, generally, be classified as another variant of the ACO optimisation algorithm and, specifically, a member of the GCACO algorithm which combines the advantages of the ACO-based feature clustering method, ACO-based adaptive selection technique, and the genetic method in determining the final subset. Thus, the proposed EGCACO algorithm has performed well for FS in DNA microarray data.

### **6.2.2 Practical Contribution**

The proposed EGCACO algorithm can be used in several application domains such as epidemiology, pathophysiology, and biological treatment. Due to the simplicity of the algorithm to identify the important features (i.e., genes), experts will be able to overcome the dimensional curse of the data. The EGCACO algorithm is able to eliminate unrelated and redundant genes and identifies the relevant ones. For instance, due to these redundant and irrelevant genes, the learning models (classifier) perform poorly, posing a challenge for data mining. Thus, identifying specific relevant features is significant and can help in the detection of DNA microarray. The EGCACO algorithm can also determine the most important features in other domains, with the purpose of extracting information from the data. Text mining, image processing, intrusion detection, business, economics classification, medical diagnosis, biological classification, and health care systems are examples of these domains.

### 6.3 Limitations and Future Work

This study has certain limitations, and this section recommends some future research directions to address the limitations.

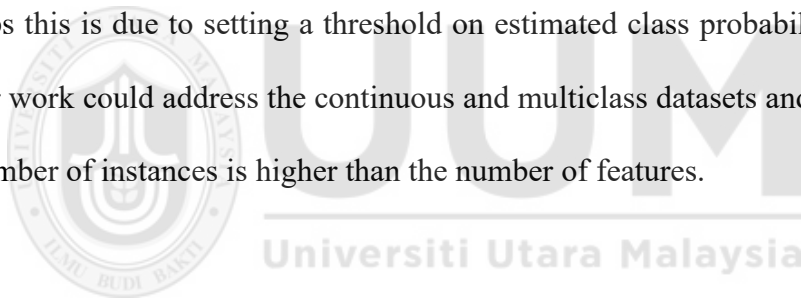
Overall, the ACO-based feature clustering performed well on datasets but did not obtain the best result in some datasets (i.e., Ionosphere, Madelon, and Hepatitis) especially when the support vector machines and k-NN classifiers were used. Thus, an improvement on specifying the appropriate number of clusters can be used to overcome this limitation.

The ACO-based adaptive technique has the ability to select the appropriate parameter value to be used in selecting significant features. Therefore, the ACO-based FS process considered the dependency of the chosen features and chose a feature subset with minimal redundancy between them. Thus, this has enabled best results to be obtained by the proposed algorithm for all classifiers. However, other predefined parameters are still required in the components of the EGCACO. Thus, a future research direction is to focus on the adaptive adjustment of several parameters that are related to the ACO-based FS algorithm.

The genetic method is able to produce the final subset from the whole dataset. This is done by arranging the most significant features and determining the number of features to be included in the final subset. However, it is still not flexible when filter subset evaluation is used in the selection strategy. With respect to the other algorithms, there are a different number of features that will be obtained in each classifier. Thus, the use of the classification performance as the principal characteristic of wrapper

methodologies to guide the selection strategy of the genetic method is suggested as future work.

The EGCACO algorithm for FS was able to deal with the different sizes of datasets. With respect to the UCI datasets, it displays good performance for small, medium, large, and very large size datasets when the number of features is higher than the number of instances. However, when the number of instances is higher than the number of features, the EGCACO did not get the best result, especially in small and medium sizes. On the other hand, for DNA microarray the best result was obtained in most of the datasets (i.e., very large size). Nevertheless, when the type of the datasets is continuous or multiclass, the classification accuracy of EGCACO is not superior. Perhaps this is due to setting a threshold on estimated class probability value. Thus, further work could address the continuous and multiclass datasets and the case where the number of instances is higher than the number of features.



## REFERENCES

- Aarabi, A., Wallois, F., & Grebe, R. (2006). Automated neonatal seizure detection: A multistage classification system through feature selection based on relevance and redundancy analysis. *Clinical Neurophysiology*, *117*(2), 328–340. <https://doi.org/10.1016/j.clinph.2005.10.006>.
- Abdelhady, S., Osama, A., Shaban, A., & Elbayoumi, M. (2020). A real-time optimisation of reactive power for an intelligent system using genetic algorithm. *IEEE Access*, *8*, 11991–12000.
- Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *IEEE Access*, *9*, 26766–26791. <https://doi.org/10.1109/ACCESS.2021.3056407>.
- Agrawal, R. K., Kaur, B., & Sharma, S. (2020). Quantum based Whale Optimisation Algorithm for wrapper feature selection. *Applied Soft Computing Journal*, *89*, 106092. <https://doi.org/10.1016/j.asoc.2020.106092>.
- Agrawal, S., & Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, *60*(1), 708–713. <https://doi.org/10.1016/j.procs.2015.08.220>.
- Ahmad, F. K., Norwawi, N. M., Deris, S., & Othman, N. H. (2008). A review of feature selection techniques via gene expression profiles. *Proceedings - International Symposium on Information Technology 2008, ITSIM*, *2*. <https://doi.org/10.1109/ITSIM.2008.4631678>.
- Ahmed, M., Mahmood, A. N., & Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, *55*(January 2019), 278–288. <https://doi.org/10.1016/j.future.2015.01.001>.
- Al-behadili, H. N. K., Ku-mahamud, K. R., & Sagban, R. (2020a). *Hybrid Ant Colony Optimisation and Genetic Algorithm for Rule Induction Hybrid Ant Colony Optimisation and Genetic Algorithm for Rule Induction*. July. <https://doi.org/10.3844/jcssp.2020.1019.1028>.
- Al-behadili, H. N. K., Ku-Mahamud, K. R., & Sagban, R. A. F. I. D. (2020). Hybrid ant colony optimisation and iterated local search for rules-based classification. *J. Theor. Appl. Inf. Technol*, *98*(04), 657-671.
- Al-Behadili, H. N. K., Ku-Mahamud, K. R., & Sagban, R. (2021). Genetic-based pruning technique for ant-miner classification algorithm. *International Journal on Advanced Science, Engineering and Information Technology*, *11*(1), 304.
- Al-Betar, M. A., Alomari, O. A., & Abu-Romman, S. M. (2020). A TRIZ-inspired bat algorithm for gene selection in cancer classification. *Genomics*, *112*(1), 114-126.
- Alelyani Salem, Jiliang Tang, & Huan Liu. (2013). Feature Selection for Clustering: A Review. *Data Clustering: Algorithms and Applications*, *29*, 110–121.

- Aleti, A., & Moser, I. (2011). Predictive parameter control. *Genetic and Evolutionary Computation Conference, GECCO'11, June 2014*, 561–568.  
<https://doi.org/10.1145/2001576.2001653>.
- Aleti, A., & Moser, I. (2013a). Entropy-based adaptive range parameter control for evolutionary algorithms. *GECCO 2013 - Proceedings of the 2013 Genetic and Evolutionary Computation Conference*, 1501–1508. <https://doi.org/10.1145/2463372.2463560>.
- Aleti, A., & Moser, I. (2013b). Studying feedback mechanisms for adaptive parameter control in evolutionary algorithms. *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, 3117–3124. <https://doi.org/10.1109/CEC.2013.6557950>.
- Alfi, A. (2012). Particle swarm optimisation algorithm with Dynamic Inertia Weight for online parameter identification applied to Lorenz chaotic system. *International Journal of Innovative Computing, Information and Control*, 8(2), 1191–1203.
- Algethami, H., & Landa-Silva, D. (2017, June). Diversity-based adaptive genetic algorithm for a Workforce Scheduling and Routing Problem. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1771-1778). IEEE, <https://doi.org/10.1109/CEC.2017.7969516>
- Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., ... & Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769), 503–511. <https://doi.org/10.1038/35000501>.
- Allahyari, M., Pouriye, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919.
- Almuallim, H., & Dietterich, T. G. (1994). Learning Boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1–2), 279–305. [https://doi.org/10.1016/0004-3702\(94\)90084-1](https://doi.org/10.1016/0004-3702(94)90084-1).
- Al-Tashi, Q., Kadir, S. J. A., Rais, H. M., Mirjalili, S., & Alhussian, H. (2019). Binary optimisation using hybrid grey wolf optimisation for feature selection. *Ieee Access*, 7, 39496-39508, <https://doi.org/10.1109/ACCESS.2019.2906757>.
- Alter, O., Brown, P. O., & Botstein, D. (2000). Singular value decomposition for genome-Wide expression data processing and modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 97(18), 10101–10106.  
<https://doi.org/10.1073/pnas.97.18.10101>
- Ambusaidi, M. A., He, X., & Nanda, P. (2015). Unsupervised feature selection method for intrusion detection system. *Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015, 1*, 295–301.  
<https://doi.org/10.1109/Trustcom.2015.387>.

- Amoozegar, M., & Minaei-Bidgoli, B. (2018). Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism. *Expert Systems with Applications*, *113*, 499–514. <https://doi.org/10.1016/j.eswa.2018.07.013>.
- Ang, J. C., Mirzal, A., Haron, H., & Hamed, H. N. A. (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. <https://doi.org/10.1109/TCBB.2015.2478454>.
- Anouncia, S. M., & Wiil, U. K. (Eds.). (2018). Knowledge Computing and Its Applications. Springer. <https://doi.org/10.1007/978-981-10-8258-0>
- Ashlock, D. (2006). Evolutionary computation for modeling and optimisation. Springer Science & Business Media, <https://doi.org/10.1007/0-387-31909-3>.
- Babatunde, O. H., Armstrong, L., Leng, J., & Diepeveen, D. (2014). A genetic algorithm-based feature selection.
- Back, T. (1992). The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. In Proc. 2nd Conference of Parallel Problem Solving from Nature, 1992. Elsevier Science Publishers.
- Bäck, T., & Schütz, M. (1996, June). Intelligent mutation rate control in canonical genetic algorithms. In International Symposium on Methodologies for Intelligent Systems (pp. 158-167). Springer, Berlin, Heidelberg.
- Bair, E., & Tibshirani, R. (2003). Machine learning methods applied to DNA microarray data can improve the diagnosis of cancer. *ACM SIGKDD Explorations Newsletter*, *5*(2), 48. <https://doi.org/10.1145/980972.980980>.
- Baldi, P., & Long, A. D. (2001). A Bayesian framework for the analysis of microarray expression data: Regularized t-test and statistical inferences of gene changes. *Bioinformatics*, *17*(6), 509–519. <https://doi.org/10.1093/bioinformatics/17.6.509>.
- Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). *Genetic programming: an introduction: on the automatic evolution of computer programs and its applications*. Morgan Kaufmann Publishers Inc.
- Bautu, E., Bautu, A., & Luchian, H. (2007, September). Adagep-an adaptive gene expression programming algorithm. In Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2007) (pp. 403-406). IEEE. <https://doi.org/10.1109/SYNASC.2007.51>.
- Ben-Bassat, M. (1982). Pattern recognition and reduction of dimensionality. *Handbook of Statistics*, *2*(1982), 773–910.
- Bharti, K. K., & Singh, P. kumar. (2014). A Survey on Filter Techniques for Feature Selection in Text Mining. *Advances in Intelligent Systems and Computing*, *236*(SocProS), 1117–1126. <https://doi.org/10.1007/978-81-322-1602-5>.

- Bihl, T. J., Bauer, K. W., & Temple, M. A. (2016). Feature Selection for RF Fingerprinting with Multiple Discriminant Analysis and Using ZigBee Device Emissions. *IEEE Transactions on Information Forensics and Security*, *11*(8), 1862–1874. <https://doi.org/10.1109/TIFS.2016.2561902>.
- Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., ... & Trent, J. (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, *406*(6795), 536-540. <https://doi.org/10.1038/35020115>
- Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10). <https://doi.org/10.1088/1742-5468/2008/10/P10008>.
- Boeringer, D. W., & Werner, D. H. (2002). Adaptive mutation parameter toggling genetic algorithm for phase-only array synthesis. *Electronics Letters*, *38*(25), 1618–1619.
- Bolón-Canedo, V., Sánchez-Marroño, N., Alonso-Betanzos, A., Benítez, J. M., & Herrera, F. (2014). A review of microarray datasets and applied feature selection methods. *Information Sciences*, *282*, 111–135. <https://doi.org/10.1016/j.ins.2014.05.042>
- Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimisation metaheuristics. *Information Sciences*, *237*, 82–117. <https://doi.org/10.1016/j.ins.2013.02.041>.
- Boutsidis, C., Mahoney, M. W., & Drineas, P. (2008). Unsupervised feature selection for principal components analysis. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 61–69. <https://doi.org/10.1145/1401890.1401903>.
- Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, *10*(6), 646–657.
- Brest, J., Žumer, V., & Maučec, M. S. (2006). Self-adaptive differential evolution algorithm in constrained real-parameter optimisation. *2006 IEEE Congress on Evolutionary Computation, CEC 2006, January 2006*, 215–222. <https://doi.org/10.1109/cec.2006.1688311>.
- Brunet, J. P., Tamayo, P., Golub, T. R., & Mesirov, J. P. (2004). Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the national academy of sciences*, *101*(12), 4164-4169.
- Buonvicino, D., Urru, M., Muzzi, M., Ranieri, G., Luceri, C., Oteri, C., Lapucci, A., & Chiarugi, A. (2018). Trigeminal ganglion transcriptome analysis in 2 rat models of medication-overuse headache reveals coherent and widespread induction of pronociceptive gene expression patterns. *Pain*, *159*(10), 1980–1988. <https://doi.org/10.1097/j.pain.0000000000001291>.
- Cai, D., Zhang, C., & He, X. (2010). Unsupervised feature selection for Multi-Cluster data. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 333–342. <https://doi.org/10.1145/1835804.1835848>.

- Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79. <https://doi.org/10.1016/j.neucom.2017.11.077>.
- Califano, A., Stolovitzky, G., & Tu, Y. (2000). Analysis of gene expression microarrays for phenotype classification. *Ismb*, *8*, 75–85.
- Caruana, R., & Freitag, D. (1994). Greedy Attribute Selection. *Machine Learning Proceedings 1994*, 28–36. <https://doi.org/10.1016/b978-1-55860-335-6.50012-x>.
- Castillo, O., Neyoy, H., Soria, J., García, M., & Valdez, F. (2013). Dynamic Fuzzy Logic Parameter Tuning for ACO and Its Application in the Fuzzy Logic Control of an Autonomous Mobile Robot. *International Journal of Advanced Robotic Systems*, *10*(1). <https://doi.org/10.5772/54883>.
- Chang, H. Y., Sneddon, J. B., Alizadeh, A. A., Sood, R., West, R. B., Montgomery, K., Chi, J. T., Van De Rijn, M., Botstein, D., & Brown, P. O. (2004). Gene expression signature of fibroblast serum response predicts human cancer progression: Similarities between tumors and wounds. *PLoS Biology*, *2*(2), 206–215. <https://doi.org/10.1371/journal.pbio.0020007>.
- Chang, R. S., Chang, J. S., & Lin, P. S. (2009). An ant algorithm for balanced job scheduling in grids. *Future Generation Computer Systems*, *25*(1), 20–27. <https://doi.org/10.1016/j.future.2008.06.004>.
- Chang, Y., Glass, K., Liu, Y. Y., Silverman, E. K., Crapo, J. D., Tal-Singer, R., Bowler, R., Dy, J., Cho, M., & Castaldi, P. (2016). COPD subtypes identified by network-based clustering of blood gene expression. *Genomics*, *107*(2–3), 51–58. <https://doi.org/10.1016/j.ygeno.2016.01.004>.
- Charlesworth, R. P. G., Agnew, L. L., Scott, D. R., & Andronicos, N. M. (2019). Celiac disease gene expression data can be used to classify biopsies along the Marsh score severity scale. *Journal of Gastroenterology and Hepatology (Australia)*, *34*(1), 169–177. <https://doi.org/10.1111/jgh.14369>.
- Chatterjee, A., & Siarry, P. (2006). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimisation. *Computers and Operations Research*, *33*(3), 859–871. <https://doi.org/10.1016/j.cor.2004.08.012>.
- Che, J., Yang, Y., Li, L., Bai, X., Zhang, S., & Deng, C. (2017). Maximum relevance minimum common redundancy feature selection for nonlinear data. *Information Sciences*, *409–410*, 68–86. <https://doi.org/10.1016/j.ins.2017.05.013>
- Chen, R., Yang, B., Li, S., & Wang, S. (2020). A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. *Computers & Industrial Engineering*, *149*, 106778.
- Cheng, Y., & Church, G. M. (2000, August). Biclustering of expression data. In *Ismb* (Vol. 8, No. 2000, pp. 93-103).

- Chinnaiyan, A. M., & Rubin, M. A. (2002). Gene-expression profiles in hereditary breast cancer. *Advances in Anatomic Pathology*, 9(1), 1–6. <https://doi.org/10.1097/00125480-200201000-00001>.
- Chinnaswamy, A., & Srinivasan, R. (2016). Hybrid feature selection using correlation coefficient and particle swarm optimisation on microarray gene expression data. *Advances in Intelligent Systems and Computing*, 424, 229–239. [https://doi.org/10.1007/978-3-319-28031-8\\_20](https://doi.org/10.1007/978-3-319-28031-8_20).
- Chuang, L. Y., Chang, H. W., Tu, C. J., & Yang, C. H. (2008). Improved binary PSO for feature selection using gene expression data. *Computational Biology and Chemistry*, 32(1), 29–38. <https://doi.org/10.1016/j.compbiolchem.2007.09.005>.
- Chuang, L. Y., Tsai, S. W., & Yang, C. H. (2011). Improved binary particle swarm optimisation using catfish effect for feature selection. *Expert Systems with Applications*, 38(10), 12699–12707. <https://doi.org/10.1016/j.eswa.2011.04.057>.
- Chuang, L. Y., Yang, C. H., & Li, J. C. (2011). Chaotic maps based on binary particle swarm optimisation for feature selection. *Applied Soft Computing Journal*, 11(1), 239–248. <https://doi.org/10.1016/j.asoc.2009.11.014>.
- Chuang, L. Y., Yang, C. H., & Yang, C. H. (2009). Tabu search and binary particle swarm optimisation for feature selection using microarray data. *Journal of Computational Biology*, 16(12), 1689–1703. <https://doi.org/10.1089/cmb.2007.0211>.
- Chusanapiputt, S., Nualhong, D., Jantarang, S., & Phoomvuthisarn, S. (2006). Selective self-adaptive approach to ant system for solving unit commitment problem. *GECCO 2006 - Genetic and Evolutionary Computation Conference*, 2, 1729–1736. <https://doi.org/10.1145/1143997.1144279>.
- Colorni, A., Dorigo, M., & Maniezzo, V. (1992). A genetic algorithm to solve the timetable problem. *Politecnico Di Milano, Milan, Italy TR*, 60–90.
- Cornuéjols, G., Fonlupt, J., & Naddef, D. (1985). The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming*, 33(1), 1–27. <https://doi.org/10.1007/BF01582008>.
- Cunliffe, H. E., Ringnér, M., Bilke, S., Walker, R. L., Cheung, J. M., Chen, Y., & Meltzer, P. S. (2003). The gene expression response of breast cancer to growth regulators: patterns and correlation with tumor expression profiles. *Cancer Research*, 63(21), 7158–7166.
- Dadaneh, B. Z., Markid, H. Y., & Zakerolhosseini, A. (2016). Unsupervised probabilistic feature selection using ant colony optimisation. *Expert Systems with Applications*, 53, 27–42. <https://doi.org/10.1016/j.eswa.2016.01.021>.
- Das, A. K., Goswami, S., Chakrabarti, A., & Chakraborty, B. (2017). A new hybrid feature selection approach using feature association map for supervised and unsupervised classification. *Expert Systems with Applications*, 88, 81–94. <https://doi.org/10.1016/j.eswa.2017.06.032>.

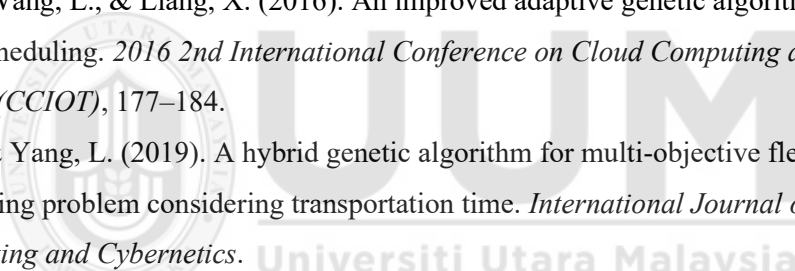
- Davis, L. D., De Jong, K., Vose, M. D., & Whitley, L. D. (2012). *Evolutionary algorithms* (Vol. 111). Springer Science & Business Media.
- Demestichas, P., Georgantas, N., Tzifa, E., Demesticha, V., Striki, M., Kilanioti, M., & Theologou, M. (2000). Computationally efficient algorithms for location area planning in future cellular systems. *Computer Communications*, 23(13), 1263–1280.
- Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 1–30.
- Devijver, P. A., & Kittler, J. (1982). *Pattern recognition: A statistical approach*. Prentice hall.
- Dhillon, I. S., Mallela, S., & Kumar, R. (2003). A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3, 1265–1287.
- Dhrif, H. (2019). Stability and Scalability of Feature Subset Selection using Particle Swarm Optimisation in Bioinformatics (Doctoral dissertation, University of Miami).
- Ding, C., & Peng, H. (2003). Minimum redundancy feature selection from microarray gene expression data. *Proceedings of the 2003 IEEE Bioinformatics Conference, CSB 2003*, 3(2), 523–528. <https://doi.org/10.1109/CSB.2003.1227396>.
- Ding, Y. M., & Wang, X. Y. (2008). Real-coded adaptive genetic algorithm applied to PID parameter optimisation on a 6R manipulators. *Proceedings - 4th International Conference on Natural Computation, ICNC 2008*, 1(3), 635–639. <https://doi.org/10.1109/ICNC.2008.82>.
- Dorigo, M., & Birattari, M. (2010). *Ant colony optimisation*. Springer.
- Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimisation. *IEEE computational intelligence magazine*, 1(4), 28–39. <https://doi.org/10.1109/MCI.2006.329691>.
- Dorigo, M., & Di Caro, G. (1999). Ant colony optimisation: a new meta-heuristic. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 2, 1470–1477.
- Dorigo, M., & Gambardella, L. M. (1997a). Ant colonies for the travelling salesman problem. *Biosystems*, 43(2), 73–81.
- Dorigo, M., & Gambardella, L. M. (1997b). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53–66.
- Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimisation by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(1), 29–41. <https://doi.org/10.1109/3477.484436>.
- Dorigo, M., & Stützle, T. (2019). Ant colony optimisation: Overview and recent advances. In *International Series in Operations Research and Management Science* (Vol. 272). [https://doi.org/10.1007/978-3-319-91086-4\\_10](https://doi.org/10.1007/978-3-319-91086-4_10).
- Dragoi, E. N., Curteanu, S., Galaction, A. I., & Cascaval, D. (2013). Optimisation methodology based on neural networks and self-adaptive differential evolution algorithm applied to an

- aerobic fermentation process. *Applied Soft Computing Journal*, 13(1), 222–238.  
<https://doi.org/10.1016/j.asoc.2012.08.004>.
- Dragoi, E. N., & Dafinescu, V. (2016). Parameter control and hybridization techniques in differential evolution: a survey. *Artificial Intelligence Review*, 45(4), 447–470.  
<https://doi.org/10.1007/s10462-015-9452-8>.
- Dudoit, S., van der Laan, M. J., Keleş, S., Molinaro, A. M., Sinisi, S. E., & Teng, S. L. (2003). Loss-Based Estimation with Cross-Validation : Applications to Microarray Data Analysis and Motif Finding Loss-Based Estimation with Cross-Validation : Applications to Microarray Data Analysis and Motif Finding. *Biostatistics*, 5(2), 56–68.
- Dudoit, S., Yang, Y. H., Callow, M. J., & Speed, T. P. (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, 12(1), 111–139.
- Dy, J. G., & Brodley, C. E. (2004). Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5, 845–889. [https://doi.org/10.1007/springerreference\\_302701](https://doi.org/10.1007/springerreference_302701).
- Eberhart, R. C., & Shi, Y. (2001). Tracking and optimizing dynamic systems with particle swarms. *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, 1*, 94–100. <https://doi.org/10.1109/cec.2001.934376>.
- Eiben, Á. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2), 124–141.
- Eiben, A. E., & Smit, S. K. (2011). Evolutionary algorithm parameters and methods to tune them. In *Autonomous search* (pp. 15-36). Springer, Berlin, Heidelberg.
- Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95(25), 14863–14868. <https://doi.org/10.1073/pnas.95.25.14863>.
- Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016a). Binary ant lion approaches for feature selection. *Neurocomputing*, 213, 54–65. <https://doi.org/10.1016/j.neucom.2016.03.101>.
- Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016b). Binary grey wolf optimisation approaches for feature selection. *Neurocomputing*, 172, 371–381.  
<https://doi.org/10.1016/j.neucom.2015.06.083>.
- Engelbrecht, A. P., & Pampará, G. (2007). Binary differential evolution strategies. *2007 IEEE Congress on Evolutionary Computation, CEC 2007*, 1942–1947.  
<https://doi.org/10.1109/CEC.2007.4424711>.
- Estévez, P. A., & Caballero, R. E. (1998). *A Niching Genetic Algorithm for Selecting Features for Neural Network Classifiers*. [https://doi.org/10.1007/978-1-4471-1599-1\\_45](https://doi.org/10.1007/978-1-4471-1599-1_45).
- Fahrudin, T. M., Syarif, I., & Barakbah, A. R. (2016). Ant colony algorithm for feature selection on microarray datasets. *Proceedings - 2016 International Electronics Symposium, IES 2016*, 351–356. <https://doi.org/10.1109/ELECSYM.2016.7861030>.

- Falaghi, H., & Haghifam, M. R. (2007, July). ACO based algorithm for distributed generation sources allocation and sizing in distribution systems. In 2007 IEEE Lausanne Power Tech (pp. 555-560). IEEE. <https://doi.org/10.1109/PCT.2007.4538377>.
- Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Al-Zoubi, A. M., Mirjalili, S., & Fujita, H. (2018). An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems, 154*, 43–67. <https://doi.org/10.1016/j.knosys.2018.05.009>.
- Feo, T. A., & Resende, M. G. C. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimisation, 6*(2), 109–133. <https://doi.org/10.1007/BF01096763>
- Feoktistov, V. (2006). *Differential evolution*. Springer.
- Fernandez-Prieto, J. A., Canada-Bago, J., Gadeo-Martos, M. A., & Velasco, J. R. (2011). Optimisation of control parameters for genetic algorithms to test computer networks under realistic traffic loads. *Applied Soft Computing Journal, 11*(4), 3744–3752. <https://doi.org/10.1016/j.asoc.2011.02.004>.
- Ferreira, A. J., & Figueiredo, M. A. T. (2012). An unsupervised approach to feature discretization and selection. *Pattern Recognition, 45*(9), 3048–3060. <https://doi.org/10.1016/j.patcog.2011.12.008>.
- Ferreira, C. (2001). Gene expression programming: a new adaptive algorithm for solving problems. *ArXiv Preprint Cs/0102027*.
- Fialho, Á., da Costa, L., Schoenauer, M., & Sebag, M. (2010). Analyzing bandit-based adaptive operator selection mechanisms. *Annals of Mathematics and Artificial Intelligence, 60*(1), 25–64. <https://doi.org/10.1007/s10472-010-9213-y>.
- Fister, I., Mernik, M., & Brest, J. (2011). Hybridization of Evolutionary Algorithms. In *Evolutionary Algorithms*. <https://doi.org/10.5772/15592>.
- Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research, 3*, 1289–1305.
- Fowlkes, E. B., Gnanadesikan, R., & Kettenring, J. R. (1988). Variable selection in clustering. *Journal of classification, 5*(2), 205-228.
- Franks, J. M., Cai, G., & Whitfield, M. L. (2018). Feature specific quantile normalization enables cross-platform classification of molecular subtypes using gene expression data. *Bioinformatics, 34*(11), 1868–1874. <https://doi.org/10.1093/bioinformatics/bty026>.
- Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance Author ( s ): Milton Friedman Source : Journal of the American Statistical Association , Vol . 32 , No . 200 , ( Dec . , 1937 ) , pp . 675- Published by : American St. *Journal of the American Statistical Association, 32*(200), 675–701.
- Fung, B. Y. M., & Ng, V. T. Y. (2003). Classification of heterogeneous gene expression data. *ACM SIGKDD Explorations Newsletter, 5*(2), 69. <https://doi.org/10.1145/980972.980982>

- Gambardella, L. M., & Dorigo, M. (2000). An ant colony system hybridized with a new local search for the sequential ordering problem. *INFORMS Journal on Computing*, 12(3), 237–255.
- Ganatra, A., Kosta, Y. P., Panchal, G., & Gajjar, C. (2011). *Initial Classification Through Back Propagation In a Neural Network Following Optimisation Through GA to Evaluate the Fitness of an Algorithm*. 3(1), 98–116.
- García-Torres, M., Gómez-Vela, F., Melián-Batista, B., & Moreno-Vega, J. M. (2016). High-dimensional feature selection via feature grouping: A Variable Neighborhood Search approach. *Information Sciences*, 326, 102–118.
- Ghaemi, M., & Feizi-Derakhshi, M. R. (2016). Feature selection using Forest Optimisation Algorithm. *Pattern Recognition*, 60, 121–129. <https://doi.org/10.1016/j.patcog.2016.05.012>.
- Ghamisi, P., & Benediktsson, J. A. (2015). Feature selection based on hybridization of genetic algorithm and particle swarm optimisation. *IEEE Geoscience and Remote Sensing Letters*, 12(2), 309–313. <https://doi.org/10.1109/LGRS.2014.2337320>.
- Gheyas, I. A., & Smith, L. S. (2010). Feature subset selection in large dimensionality domains. *Pattern Recognition*, 43(1), 5–13. <https://doi.org/10.1016/j.patcog.2009.06.009>.
- Ghimatgar, H., Kazemi, K., Helfroush, M. S., & Aarabi, A. (2018). An improved feature selection algorithm based on graph clustering and ant colony optimisation. *Knowledge-Based Systems*, 159(July), 270–285. <https://doi.org/10.1016/j.knosys.2018.06.025>.
- Ghosh, A., & Nath, B. (2004). Multi-objective rule mining using genetic algorithms. *Information Sciences*, 163(1–3), 123–133. <https://doi.org/10.1016/j.ins.2003.03.021>.
- Ghosh, M., & Sarkar, R. (2020). A wrapper-filter feature selection technique based on ant colony optimisation. *Neural Computing and Applications*, 32(12), 7839–7857. <https://doi.org/10.1007/s00521-019-04171-3>.
- Glenisson, P., Mathys, J., & De Moor, B. (2003). Meta-clustering of gene expression data and literature-based information. *ACM SIGKDD Explorations Newsletter*, 5(2), 101. <https://doi.org/10.1145/980972.980985>.
- Glover, F. (1986). Paths for Integer Programming. *Computers and Operations Research*, 13(5), 533–549. [https://doi.org/http://dx.doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/http://dx.doi.org/10.1016/0305-0548(86)90048-1).
- Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., & Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439), 531–527. <https://doi.org/10.1126/science.286.5439.531>.
- Good, B. H., De Montjoye, Y. A., & Clauset, A. (2010). Performance of modularity maximization in practical contexts. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 81(4), 1–20. <https://doi.org/10.1103/PhysRevE.81.046106>.

- Gu, Q., Li, Z., & Han, J. (2011). Generalized fisher score for feature selection. *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011*.
- Gudino-Penaloza, F., Gonzalez-Mendoza, M., Mora-Vargas, J., & Hernandez-Gress, N. (2013). Fuzzy hyperheuristic framework for GA parameters tuning. *Proceedings - 2013 12th Mexican International Conference on Artificial Intelligence, MICAI 2013*, 53–58. <https://doi.org/10.1109/MICAI.2013.48>.
- Guha, R., Ghosh, M., Chakrabarti, A., Sarkar, R., & Mirjalili, S. (2020). Introducing clustering based population in Binary Gravitational Search Algorithm for Feature Selection. *Applied Soft Computing Journal*, 93, 106341. <https://doi.org/10.1016/j.asoc.2020.106341>.
- Guha, R., Ghosh, M., Mutsuddi, S., Sarkar, R., & Mirjalili, S. (2020). Embedded chaotic whale survival algorithm for filter–wrapper feature selection. *Soft Computing*, 24(17), 12821–12843. <https://doi.org/10.1007/s00500-020-05183-1>.
- Hall, M. A. (1999). Correlation-based feature selection for machine learning.
- Hall, M. A. (2000). *Correlation-based feature selection of discrete and numeric class machine learning* (pp. 32–33).
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, 11(1), 10. <https://doi.org/10.1145/1656274.1656278>.
- Hanczar, B., Courtine, M., Benis, A., Hennegar, C., Clément, K., & Zucker, J. D. (2003). Improving classification of microarray data using prototype-based feature selection. *ACM SIGKDD Explorations Newsletter*, 5(2), 23–30.
- Harrison, K. R., Engelbrecht, A. P., & Ombuki-Berman, B. M. (2018). An adaptive particle swarm optimisation algorithm based on optimal parameter regions. *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings, 2018-Janua*, 1–8. <https://doi.org/10.1109/SSCI.2017.8285342>.
- Hashemi, S., Kiani, S., Noroozi, N., & Moghaddam, M. E. (2010). An image contrast enhancement method based on genetic algorithm. *Pattern Recognition Letters*, 31(13), 1816–1824.
- Hastie, T., Tibshirani, R., Botstein, D., & Brown, P. (2001). Supervised harvesting of expression trees. *Genome Biology*, 2(1), 1–12. <https://doi.org/10.1186/gb-2001-2-1-research0003>
- He, X., Cai, D., & Niyogi, P. (2005). Laplacian Score for feature selection. *Advances in Neural Information Processing Systems*, 507–514.
- Herman, G., Zhang, B., Wang, Y., Ye, G., & Chen, F. (2013). Mutual information-based method for selecting informative feature sets. *Pattern Recognition*, 46(12), 3315–3327. <https://doi.org/10.1016/j.patcog.2013.04.021>.
- Hinterding, R. (1995). Gaussian mutation and self-adaption for numeric genetic algorithms. *Proceedings of the IEEE Conference on Evolutionary Computation*, 1, 384–388. <https://doi.org/10.1109/icec.1995.489178>.

- Hinterding, R., Michalewicz, Z., & Eiben, A. E. (1997). Adaptation in evolutionary computation: A survey. *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 65–69. <https://doi.org/10.1109/icec.1997.592270>.
- Hu, Y., & Yang, S. X. (2004). A knowledge based genetic algorithm for path planning of a mobile robot. *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, 5, 4350–4355.
- Huang, C. L., & Huang, W. L. (2009). Handling sequential pattern decay: Developing a two-stage collaborative recommender system. *Electronic Commerce Research and Applications*, 8(3), 117–129. <https://doi.org/10.1016/j.elerap.2008.10.001>.
- Huang, C., Li, Y., & Yao, X. (2019). A Survey of Automatic Parameter Tuning Methods for Metaheuristics. *IEEE Transactions on Evolutionary Computation*, 1–1. <https://doi.org/10.1109/tevc.2019.2921598>.
- Huang, H., Yang, X., Hao, Z., & Cai, R. (2006, August). A novel ACO algorithm with adaptive parameter. In *International Conference on Intelligent Computing* (pp. 12-21). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/11816102\\_2](https://doi.org/10.1007/11816102_2).
- Huang, M., Wang, L., & Liang, X. (2016). An improved adaptive genetic algorithm in flexible job shop scheduling. *2016 2nd International Conference on Cloud Computing and Internet of Things (CCIOT)*, 177–184.
- Huang, X., & Yang, L. (2019). A hybrid genetic algorithm for multi-objective flexible job shop scheduling problem considering transportation time. *International Journal of Intelligent Computing and Cybernetics*.  <https://doi.org/10.1109/ijicc.2019.8711111>.
- Huang, Z. (2013). An improved differential evolution algorithm based on statistical log-linear model. *Sensors and Transducers*, 159(11), 277–281.
- Ideker, T., Thorsson, V., Siegel, A. F., & Hood, L. E. (2001). Testing for differentially-expressed genes by maximum-likelihood analysis of microarray data. *Journal of Computational Biology*, 7(6), 805–817. <https://doi.org/10.1089/10665270050514945>.
- Ienco, D., & Meo, R. (2008). Exploration and reduction of the feature space by hierarchical clustering. *Society for Industrial and Applied Mathematics - 8th SIAM International Conference on Data Mining 2008, Proceedings in Applied Mathematics 130*, 2, 577–587. <https://doi.org/10.1137/1.9781611972788.53>.
- Jaddi, N. S., & Abdullah, S. (2021). A Novel Auction-Based Optimisation Algorithm and Its Application in Rough Set Feature Selection. *IEEE Access*, 9, 106501–106514. <https://doi.org/10.1109/ACCESS.2021.3098808>.
- Jang, S. H., Roh, J. H., Kim, W., Sherpa, T., Kim, J. H., & Park, J. B. (2011). A novel binary ant colony optimisation: Application to the unit commitment problem of power systems. *Journal of Electrical Engineering and Technology*, 6(2), 174–181. <https://doi.org/10.5370/JEET.2011.6.2.174>.

- Jantzen, J. (2007). *Foundations of Fuzzy Control* John Wiley & Sons Ltd. England.
- Jenatton, R., Audibert, J. Y., & Bach, F. (2011). Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 12, 2777–2824.
- Jensen, T. R., & Toft, B. (2011). Graph colouring problems. John Wiley & Sons.  
<https://doi.org/10.1002/9781118032497>.
- Jiang, D., Pei, J., & Zhang, A. (2003, August). Interactive exploration of coherent patterns in time-series gene expression data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 565-570).  
<https://doi.org/10.1145/956804.956820>.
- Jiang, Y., & Ren, J. (2011). Eigenvalue sensitive feature selection. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, August*, 89–96.
- Jiao, B., Lian, Z., & Gu, X. (2008). A dynamic inertia weight particle swarm optimisation algorithm. *Chaos, Solitons and Fractals*, 37(3), 698–705.  
<https://doi.org/10.1016/j.chaos.2006.09.063>.
- Jörnsten, R., & Yu, B. (2003). Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics*, 19(9), 1100–1109.  
<https://doi.org/10.1093/bioinformatics/btg039>.
- Kabir, M. M., Shahjahan, M., & Murase, K. (2012). A new hybrid ant colony optimisation algorithm for feature selection. *Expert Systems with Applications*, 39(3), 3747–3763.  
<https://doi.org/10.1016/j.eswa.2011.09.073>
- Kamrath, N. R., Goldman, B. W., & Tauritz, D. R. (2013). Using supportive coevolution to evolve self-configuring crossover. *GECCO 2013 - Proceedings of the 2013 Genetic and Evolutionary Computation Conference Companion*, 1489–1496.  
<https://doi.org/10.1145/2464576.2482727>
- Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimisation (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimisation (PSO). *Proc. IEEE International Conference on Neural Networks, Perth, Australia*, 1942–1948.
- Khadhraoui, T., Ktata, S., Benzarti, F., & Amiri, H. (2016). Features Selection Based on Modified PSO Algorithm for 2D Face Recognition. *Proceedings - Computer Graphics, Imaging and Visualization: New Techniques and Trends, CGiV 2016, March*, 99–104.  
<https://doi.org/10.1109/CGiV.2016.28>.
- Khan, J. A., Van Aelst, S., & Zamar, R. H. (2007). Building a robust linear model with forward selection and stepwise procedures. *Computational Statistics and Data Analysis*, 52(1), 239–248. <https://doi.org/10.1016/j.csda.2007.01.007>.

- Khan, J., Wei, J. S., Ringnér, M., Saal, L. H., Ladanyi, M., Berthold, F., Schwab, M., Antonescu, C. R., & Meltzer, P. S. (2005). Expression Profiling and Artificial Neural Networks. *Cancer Research*, 7(6), 673–679. <https://doi.org/10.1038/89044>.
- Kim, S., & Xing, E. (2008). Feature selection via block-regularized regression. *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, UAI 2008*, 325–332.
- Kim, Y. B., & Gao, J. (2006). Unsupervised gene selection for high dimensional data. *Proceedings - Sixth IEEE Symposium on BioInformatics and BioEngineering, BIBE 2006*, 227–232. <https://doi.org/10.1109/BIBE.2006.253339>.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimisation by simulated annealing. *Science*. <https://doi.org/10.1016/j.ultsonch.2011.06.003>.
- Kora, P., & Yadlapalli, P. (2017). Crossover operators in genetic algorithms: A review. *International Journal of Computer Applications*, 162(10).
- Kotinis, M. (2014). Improving a multi-objective differential evolution optimiser using fuzzy adaptation and K-medoids clustering. *Soft Computing*, 18(4), 757–771. <https://doi.org/10.1007/s00500-013-1086-7>
- Kotsiantis, S. B. (2011). Erratum: Feature selection for machine learning classification problems: A recent overview (Artificial Intelligence Review (2011)). *Artificial Intelligence Review*, 42(1), 157. <https://doi.org/10.1007/s10462-011-9230-1>
- Kramer, O. (2010). Evolutionary self-adaptation: A survey of operators and strategy parameters. *Evolutionary Intelligence*, 3(2), 51–65. <https://doi.org/10.1007/s12065-010-0035-y>
- Kudo, M., & Sklansky, J. (2000). Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1), 25–41.
- Lai, C., Reinders, M. J. T., & Wessels, L. (2006). Random subspace method for multivariate feature selection. *Pattern Recognition Letters*, 27(10), 1067–1076. <https://doi.org/10.1016/j.patrec.2005.12.018>
- Lapointe, J., Li, C., Higgins, J. P., Van De Rijn, M., Bair, E., Montgomery, K., ... & Pollack, J. R. (2004). Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *Proceedings of the National Academy of Sciences*, 101(3), 811–816. <https://doi.org/10.1073/pnas.0304146101>
- Laporte, G. (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2), 231–247.
- Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., Schaetzen, V. De, Duque, R., Bersini, H., & Nowe, A. (2012). *survey of filter techniques for feature selection in MicroArrays.pdf*. 9(4), 1106–1119. <https://doi.org/10.1109/TCBB.2012.33>
- Lee, C. P., & Leu, Y. (2011). A novel hybrid feature selection method for microarray data analysis. *Applied Soft Computing Journal*, 11(1), 208–213. <https://doi.org/10.1016/j.asoc.2009.11.010>.

- Lee, W., Stolfo, S. J., & Mok, K. W. (2000). Adaptive intrusion detection: A data mining approach. *Artificial Intelligence Review*, *14*(6), 533–567.  
<https://doi.org/10.1023/A:1006624031083>.
- Leloup, L., Ehrlich, S. D., Zagorec, M., & Morel-Deville, F. (1997). Single-crossover integration in the *Lactobacillus sake* chromosome and insertional inactivation of the *ptsI* and *lacL* genes. *Applied and Environmental Microbiology*, *63*(6), 2117–2123.
- Leung, S. W., Zhang, X., & Yuen, S. Y. (2012). Multiobjective differential evolution algorithm with opposition-based parameter control. *2012 IEEE Congress on Evolutionary Computation, CEC 2012, 1*, 10–15. <https://doi.org/10.1109/CEC.2012.6256612>.
- Leung, Y., & Hung, Y. (2010). A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *7*(1), 108–117. <https://doi.org/10.1109/TCBB.2008.46>.
- Li, C., & Hung Wong, W. (2001). Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biology*, *2*(8), 1–11.  
<https://doi.org/10.1186/gb-2001-2-8-research0032>.
- Li, H., & Hong, F. (2001). Cluster-Rasch models for microarray gene expression data. *Genome Biology*, *2*(8), 1–13. <https://doi.org/10.1186/gb-2001-2-8-research0031>.
- Li, H., & Li, P. (2013). Self-adaptive ant colony optimisation for construction time-cost optimisation. *Kybernetes*, *42*(8), 1181–1194. <https://doi.org/10.1108/K-03-2013-0063>.
- Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys*, *50*(6).  
<https://doi.org/10.1145/3136625>.
- Li, R., & Chang, X. (2006). A modified genetic algorithm with multiple subpopulations and dynamic parameters applied in CVAR model. *2006 International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06)*, 151.
- Li, Y., Wang, G., Chen, H., Shi, L., & Qin, L. (2013). An Ant Colony Optimisation Based Dimension Reduction Method for High-Dimensional Datasets. *Journal of Bionic Engineering*, *10*(2), 231–241. [https://doi.org/10.1016/S1672-6529\(13\)60219-X](https://doi.org/10.1016/S1672-6529(13)60219-X).
- Liang, Z., Sun, J., Lin, Q., Du, Z., Chen, J., & Ming, Z. (2016). A novel multiple rule sets data classification algorithm based on ant colony algorithm. *Applied Soft Computing Journal*, *38*, 1000–1011. <https://doi.org/10.1016/j.asoc.2015.10.046>
- Liao, B., Jiang, Y., Liang, W., Zhu, W., Cai, L., & Cao, Z. (2014). Gene selection using locality sensitive Laplacian score. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *11*(6), 1146–1156. <https://doi.org/10.1109/TCBB.2014.2328334>

- Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining* (Vol. 454). Springer Science & Business Media.
- Liu, H., Wu, X., & Zhang, S. (2011). Feature selection using hierarchical feature clustering. *International Conference on Information and Knowledge Management, Proceedings*, 979–984. <https://doi.org/10.1145/2063576.2063716>
- Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4), 491–502. <https://doi.org/10.1109/TKDE.2005.66>.
- Liu, J., & Lampinen, J. (2005). A fuzzy adaptive differential evolution algorithm. *Soft Computing*, 9(6), 448–462. <https://doi.org/10.1007/s00500-004-0363-x>.
- Long, W., Jiao, J., Liang, X., Wu, T., Xu, M., & Cai, S. (2021). Pinhole-imaging-based learning butterfly optimisation algorithm for global optimisation and feature selection. *Applied Soft Computing*, 103(January), 107146. <https://doi.org/10.1016/j.asoc.2021.107146>.
- López-Ibáñez, M., & Blum, C. (2010). Beam-ACO for the travelling salesman problem with time windows. *Computers & Operations Research*, 37(9), 1570–1583.
- Löw, F., Michel, U., Dech, S., & Conrad, C. (2013). Impact of feature selection on the accuracy and spatial uncertainty of per-field crop classification using Support Vector Machines. *ISPRS Journal of Photogrammetry and Remote Sensing*, 85, 102–119. <https://doi.org/10.1016/j.isprsjprs.2013.08.007>.
- Mafarja, M., Aljarah, I., Faris, H., Hammouri, A. I., Al-Zoubi, A. M., & Mirjalili, S. (2019). Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Systems with Applications*, 117, 267–286. <https://doi.org/10.1016/j.eswa.2018.09.015>
- Mafarja, M. M., & Mirjalili, S. (2017). Hybrid Whale Optimisation Algorithm with simulated annealing for feature selection. *Neurocomputing*, 260, 302–312. <https://doi.org/10.1016/j.neucom.2017.04.053>.
- Mafarja, M. M., & Mirjalili, S. (2019). Hybrid binary ant lion optimiser with rough set and approximate entropy reducts for feature selection. *Soft Computing*, 23(15), 6249–6265. <https://doi.org/10.1007/s00500-018-3282-y>.
- Mafarja, M., & Mirjalili, S. (2018). Whale optimisation approaches for wrapper feature selection. *Applied Soft Computing*, 62, 441–453. <https://doi.org/10.1016/j.asoc.2017.11.006>.
- Mafarja, M., Qasem, A., Heidari, A. A., Aljarah, I., Faris, H., & Mirjalili, S. (2020). Efficient Hybrid Nature-Inspired Binary Optimisers for Feature Selection. *Cognitive Computation*, 12(1), 150–175. <https://doi.org/10.1007/s12559-019-09668-6>.
- Manbari, Z., Akhlaghian Tab, F., & Salavati, C. (2019a). Fast unsupervised feature selection based on the improved binary ant system and mutation strategy. *Neural Computing and Applications*, 31(9), 4963–4982. <https://doi.org/10.1007/s00521-018-03991-z>.

- Manbari, Z., Akhlaghian Tab, F., & Salavati, C. (2019b). Hybrid fast unsupervised feature selection for high-dimensional data. *Expert Systems with Applications*, *124*, 97–118. <https://doi.org/10.1016/j.eswa.2019.01.016>.
- Mandal, M., & Mukhopadhyay, A. (2013). An Improved Minimum Redundancy Maximum Relevance Approach for Feature Selection in Gene Expression Data. *Procedia Technology*, *10*, 20–27. <https://doi.org/10.1016/j.protcy.2013.12.332>.
- Manita, G., & Korbaa, O. (2020). Binary political optimiser for feature selection using gene expression data. *Computational Intelligence and Neuroscience*, *2020*. <https://doi.org/10.1155/2020/8896570>.
- Marchal, K., Engelen, K., De Brabanter, J., Aert, S., De Moor, B., Ayoubi, T., & Van Hummelen, P. (2002). Comparison of different methodologies to identify differentially expressed genes in two-sample cDNA microarrays. *Journal of Biological Systems*, *10*(4), 409–430. <https://doi.org/10.1142/S0218339002000731>.
- Marques, V., & Gomide, F. (2011). Parameter control of metaheuristics with genetic fuzzy systems. *Evolutionary Intelligence*, *4*(3), 183–202. <https://doi.org/10.1007/s12065-011-0059-y>.
- Martínez Sotoca, J., & Pla, F. (2010). Supervised feature selection by clustering using conditional mutual information-based distances. *Pattern Recognition*, *43*(6), 2068–2081. <https://doi.org/10.1016/j.patcog.2009.12.013>.
- Melin, P., Olivás, F., Castillo, O., Valdez, F., Soria, J., & Valdez, M. (2013). Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. *Expert Systems with Applications*, *40*(8), 3196–3206. <https://doi.org/10.1016/j.eswa.2012.12.033>.
- Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant colony optimisation for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation*, *6*(4), 333–346. <https://doi.org/10.1109/TEVC.2002.802450>.
- Mernik, M. (2007). *Self-Adapting Control Parameters in Differential Evolution : A Comparative Study on Numerical Benchmark Problems*. January. <https://doi.org/10.1109/TEVC.2006.872133>.
- Meyer, B. (2004). Convergence control in ACO. In *Genetic and Evolutionary Computation Conference (GECCO), Seattle, WA, Late-Breaking Paper Available on CD*.
- Mirzaei, M. R., Najafi, A., Arababadi, M. K., Asadi, M. H., & Mowla, S. J. (2014). Altered expression of apoptotic genes in response to OCT4B1 suppression in human tumor cell lines. *Tumor Biology*, *35*(10), 9999–10009. <https://doi.org/10.1007/s13277-014-2238-9>.
- Mo, D., & Huang, S. H. (2011). Feature selection based on inference correlation. *Intelligent Data Analysis*, *15*(3), 375–398. <https://doi.org/10.3233/IDA-2010-0473>.

- Mo, D., & Huang, S. H. (2012). Fractal-based intrinsic dimension estimation and its application in dimensionality reduction. *IEEE Transactions on Knowledge and Data Engineering*, 24(1), 59–71. <https://doi.org/10.1109/TKDE.2010.225>.
- Mohammadzadeh, H., & Gharehchopogh, F. S. (2021). A novel hybrid whale optimisation algorithm with flower pollination algorithm for feature selection: Case study Email spam detection. *Computational Intelligence*, 37(1), 176–209. <https://doi.org/10.1111/coin.12397>.
- Montalvo, I., Izquierdó, J., Peerez-Garcia, R., & Herrera, M. (2010). Improved performance of PSO with self-adaptive parameters for computing the optimal design of Water Supply Systems. *Engineering Applications of Artificial Intelligence*, 23(5), 727–735. <https://doi.org/10.1016/j.engappai.2010.01.015>.
- Mootha, V. K., Lindgren, C. M., Eriksson, K.F., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E., Ridderstråle, M., & Laurila, E. (2003). PGC-1 $\alpha$ -responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, 34(3), 267.
- Moradi, P., & Rostami, M. (2015a). A graph theoretic approach for unsupervised feature selection. *Engineering Applications of Artificial Intelligence*, 44, 33–45. <https://doi.org/10.1016/j.engappai.2015.05.005>.
- Moradi, P., & Rostami, M. (2015b). Integration of graph clustering with ant colony optimisation for feature selection. *Knowledge-Based Systems*, 84, 144–161. <https://doi.org/10.1016/j.knosys.2015.04.007>.
- Morin, S., Gagné, C., & Gravel, M. (2009). Ant colony optimisation with a specialized pheromone trail for the car-sequencing problem. *European Journal of Operational Research*, 197(3), 1185–1191.
- Morovvat, M., & Osareh, A. (2016). An Ensemble of Filters and Wrappers for Microarray Data Classification. *Machine Learning and Applications: An International Journal*, 3(2), 01–17. <https://doi.org/10.5121/mlaij.2016.3201>
- Mukherjee, S. N., Roberts, S. J., Sykacek, P., & Gurr, S. J. (2003). Gene ranking using bootstrapped P-values. *ACM SIGKDD Explorations Newsletter*, 5(2), 16. <https://doi.org/10.1145/980972.980976>
- Murali, T. M., & Kasif, S. (2002). Extracting conserved gene expression motifs from gene expression data. In *Biocomputing 2003* (pp. 77–88). World Scientific.
- Nadimpally, V., & Zaki, M. J. (2003). A novel approach to determine normal variation in gene expression data. *ACM SIGKDD Explorations Newsletter*, 5(2), 6. <https://doi.org/10.1145/980972.980975>.
- Najafi, A., Bidkhori, G., Bozorgmehr, J., Koch, I., & Masoudi-Nejad, A. (2014). Genome Scale Modeling in Systems Biology: Algorithms and Resources. *Current Genomics*, 15(2), 130–159. <https://doi.org/10.2174/1389202915666140319002221>.

- Naseri, A., & Hasheminejad, S. M. H. (2019). An unsupervised gene selection method based on multiobjective ant colony optimisation. *International Journal of Artificial Intelligence*, 17(2), 1–22.
- Nemati, S., Basiri, M. E., Ghasem-Aghaee, N., & Aghdam, M. H. (2009). A novel ACO-GA hybrid algorithm for feature selection in protein function prediction. *Expert Systems with Applications*, 36(10), 12086–12094. <https://doi.org/10.1016/j.eswa.2009.04.023>.
- Neshat, M. (2013). FAIPSO: Fuzzy adaptive informed particle swarm optimisation. *Neural Computing and Applications*, 23(SUPPL1), 95–116. <https://doi.org/10.1007/s00521-012-1256-z>.
- Niijima, S., & Okuno, Y. (2009). Laplacian linear discriminant analysis approach to unsupervised feature selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(4), 605–614. <https://doi.org/10.1109/TCBB.2007.70257>.
- Niknam, T., & Azad Farsani, E. (2010). A hybrid self-adaptive particle swarm optimisation and modified shuffled frog leaping algorithm for distribution feeder reconfiguration. *Engineering Applications of Artificial Intelligence*, 23(8), 1340–1349. <https://doi.org/10.1016/j.engappai.2010.02.005>.
- Novovicová, J., Pudil, P., & Kittler, J. (1996). Divergence based feature selection for multimodal class densities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2), 218–223. <https://doi.org/10.1109/34.481557>.
- Nutt, C. L., Mani, D. R., Betensky, R. A., Tamayo, P., Cairncross, J. G., Ladd, C., ... & Louis, D. N. (2003). Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer research*, 63(7), 1602–1607.
- O’Connell, M. (2003). Differential expression, class discovery and class prediction using S-PLUS and S+ArrayAnalyzer. *ACM SIGKDD Explorations Newsletter*, 5(2), 38. <https://doi.org/10.1145/980972.980979>.
- Oh, I. S., Lee, J. S., & Moon, B. R. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11), 1424–1437. <https://doi.org/10.1109/TPAMI.2004.105>.
- Okafor, A. (2005). Entropy based techniques with applications in data mining. University of Florida. <https://doi.org/10.1017/CBO9781107415324.004>.
- Olivas, F., Valdez, F., & Castillo, O. (2013). Particle swarm optimisation with dynamic parameter adaptation using interval type-2 fuzzy logic for benchmark mathematical functions. *2013 World Congress on Nature and Biologically Inspired Computing, NaBIC 2013*, 36–40. <https://doi.org/10.1109/NaBIC.2013.6617875>.
- Ortega, A., Escobar, J. J., Damas, M., Ortiz, A., & González, J. (2022). Ant colony optimisation for feature selection via a filter-randomized search heuristic. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 85–88.

- Osaba, E., Onieva, E., Carballedo, R., Diaz, F., Perallos, A., & Zhang, X. (2013). A multi-crossover and adaptive island based population algorithm for solving routing problems. *Journal of Zhejiang University: Science C*, 14(11), 815–821. <https://doi.org/10.1631/jzus.C1300184>.
- Otero, F. E. B., Freitas, A. A., & Johnson, C. G. (2013). A new sequential covering strategy for inducing classification rules with ant colony algorithms. *IEEE Transactions on Evolutionary Computation*, 17(1), 64–76. <https://doi.org/10.1109/TEVC.2012.2185846>.
- Padilha, V. A., & de Leon Ferreira, A. C. P. (2019). Experimental correlation analysis of bicluster coherence measures and gene ontology information. *Applied Soft Computing Journal*, 85, 105688. <https://doi.org/10.1016/j.asoc.2019.105688>.
- Padungweang, P., Lursinsap, C., & Sunat, K. (2012). A discrimination analysis for unsupervised feature selection via optic diffraction principle. *IEEE Transactions on Neural Networks and Learning Systems*, 23(10), 1587–1600. <https://doi.org/10.1109/TNNLS.2012.2208269>.
- Pahikkala, T., Airola, A., & Salakoski, T. (2010). Speeding up greedy forward selection for regularized least-squares. *Proceedings - 9th International Conference on Machine Learning and Applications, ICMLA 2010*, 325–330. <https://doi.org/10.1109/ICMLA.2010.55>.
- Pan, W., Lin, J., & Le, C. T. (2002). Model-based cluster analysis of microarray gene-expression data. *Genome Biology*, 3(2), 1–8. <https://doi.org/10.1186/gb-2002-3-2-research0009>.
- Pan, S. J. (2014). Transfer Learning. *Data Classification: Algorithms and Applications*, 21. <https://doi.org/10.1201/b17320>.
- Paniri, M., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2020). MLACO: A multi-label feature selection algorithm based on ant colony optimisation. *Knowledge-Based Systems*, 105285. <https://doi.org/10.1016/j.knosys.2019.105285>.
- Parpinelli, R. S., Plichoski, G. F., Da Silva, R. S., & Narloch, P. H. (2019). A review of techniques for online control of parameters in swarm intelligence and evolutionary computation algorithms. *International Journal of Bio-Inspired Computation*, 13(1), 1–20. <https://doi.org/10.1504/IJBIC.2019.097731>.
- Parsa, M. G., Zare, H., & Ghatee, M. (2020). Unsupervised feature selection based on adaptive similarity learning and subspace clustering. *Engineering Applications of Artificial Intelligence*, 95, 103855.
- Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 8, 1226–1238.
- Peng, H., Ying, C., Tan, S., Hu, B., & Sun, Z. (2018). An Improved Feature Selection Algorithm Based on Ant Colony Optimisation. In *IEEE Access* (Vol. 6, pp. 69203–69209). <https://doi.org/10.1109/ACCESS.2018.2879583>.

- Peng, Y., Luo, X., & Wei, W. (2014). A new fuzzy adaptive simulated annealing genetic algorithm and its convergence analysis and convergence rate estimation. *International Journal of Control, Automation and Systems*, 12(3), 670–679.  
<https://doi.org/10.1007/s12555-011-0022-9>.
- Pereira, A. G. C., Roveda, J. A. F., Amorim Carlos, L., Simioli Medeiros Campos, V., & Roveda, S. R. M. M. (2013). Convergence analysis of an elitist non-homogeneous genetic algorithm with mutation probability adjusted by a fuzzy controller. *Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting, IFSA/NAFIPS 2013, February 2016*, 19–23.  
<https://doi.org/10.1109/IFSA-NAFIPS.2013.6608368>.
- Piatetsky-Shapiro, G., & Tamayo, P. (2003). Microarray data mining. *ACM SIGKDD Explorations Newsletter*, 5(2), 1. <https://doi.org/10.1145/980972.980974>.
- Podgorelec, V., Brest, J., & Kokol, P. (2000). Power of heterogeneous computing as a vehicle for implementing E3 medical decision support systems. *International Journal of Medical Informatics*, 58, 179–190.
- Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimisation. *IEEE Transactions on Evolutionary Computation*, 13(2), 398–417. <https://doi.org/10.1109/TEVC.2008.927706>
- Rajappan, S., & Rangasamy, D. (2017). Adaptive Genetic Algorithm with Exploration-Exploitation Tradeoff for Preprocessing Microarray Datasets. *Current Bioinformatics*, 12(5), 441–451.
- Ramaswamy, S., Ross, K. N., Lander, E. S., & Golub, T. R. (2003). A molecular signature of metastasis in primary solid tumors. *Nature Genetics*, 33(1), 49–54.  
<https://doi.org/10.1038/ng1060>
- Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C. H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P., Poggio, T., Gerald, W., Loda, M., Lander, E. S., & Golub, T. R. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America*, 98(26), 15149–15154. <https://doi.org/10.1073/pnas.211566398>.
- Ramos, J., Castellanos-Garzón, J. A., González-Briones, A., de Paz, J. F., & Corchado, J. M. (2017). An Agent-Based Clustering Approach for Gene Selection in Gene Expression Microarray. *Interdisciplinary Sciences: Computational Life Sciences*, 9(1), 1–13.  
<https://doi.org/10.1007/s12539-017-0219-6>.
- Rasheeduddin, S., & Rao, K. R. (2019). Extreme Learning Machine for Thyroid Nodule Classification with Graph Cluster Ant Colony Optimisation based Feature Selection. *International Journal of Recent Technology and Engineering*, 8(2), 1478–1488.  
<https://doi.org/10.35940/ijrte.b2115.078219>.

- Raymer, M. L., Punch, W. F., Goodman, E. D., Kuhn, L. A., & Jain, A. K. (2000). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(2), 164–171. <https://doi.org/10.1109/4235.850656>.
- Rostami, M., Berahmand, K., & Forouzandeh, S. (2021). A novel community detection based genetic algorithm for feature selection. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-020-00398-3>.
- Rostami, M., Forouzandeh, S., Berahmand, K., & Soltani, M. (2020). Integration of multi-objective PSO based feature selection and node centrality for medical datasets. *Genomics*, 112(6), 4370–4384. <https://doi.org/10.1016/j.ygeno.2020.07.027>.
- Roy, R., Singh, R., Chattopadhyay, E., Ray, A., Sarkar, N. De, Aich, R., Paul, R. R., Pal, M., & Roy, B. (2016). MicroRNA and target gene expression based clustering of oral cancer, precancer and normal tissues. *Gene*, 593(1), 58–63. <https://doi.org/10.1016/j.gene.2016.08.011>.
- Saeyns, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19), 2507–2517. <https://doi.org/10.1093/bioinformatics/btm344>.
- Sagban, R. (2016). *Reactive Approach for Automating Exploration and Exploitation in Ant Colony Optimisation*. Unpublished doctoral dissertation. Universiti Utara Malaysia, Malaysia.
- Sagban, R., Ku-Mahamud, K. R., & Bakar, M. S. A. (2017). Unified strategy for intensification and diversification balance in ACO metaheuristic. *ICIT 2017 - 8th International Conference on Information Technology, Proceedings*, 139–143. <https://doi.org/10.1109/ICITECH.2017.8079991>.
- Sahoo, A., & Chandra, S. (2017). Multi-objective Grey Wolf Optimiser for improved cervix lesion classification. *Applied Soft Computing Journal*, 52, 64–80. <https://doi.org/10.1016/j.asoc.2016.12.022>.
- Sahu, B., Dehuri, S., & Jagadev, A. (2018). A Study on the Relevance of Feature Selection Methods in Microarray Data. *The Open Bioinformatics Journal*, 11(1), 117–139. <https://doi.org/10.2174/1875036201811010117>.
- Saidi, R., Bouaguel, W., & Essoussi, N. (2019). *on the Genetic Algorithm and Pearson Correlation Coefficient* (Issue January). Springer International Publishing. <https://doi.org/10.1007/978-3-030-02357-7>.
- Salama, K., & Freitas, A. (2012). ABC-Miner: An ant-based Bayesian classification algorithm. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7461 LNCS, 13–24. [https://doi.org/10.1007/978-3-642-32650-9\\_2](https://doi.org/10.1007/978-3-642-32650-9_2).

- Salem, H., Attiya, G., & El-Fishawy, N. (2017). Classification of human cancer diseases by gene expression profiles. *Applied Soft Computing Journal*, 50, 124–134. <https://doi.org/10.1016/j.asoc.2016.11.026>.
- Segal, E., Battle, A., & Koller, D. (2002). Decomposing gene expression into cellular processes. In *Biocomputing 2003* (pp. 89–100). World Scientific.
- Setyawan, N., Kadir, R. E. A., & Jazidie, A. (2017, August). Adaptive Gaussian parameter particle swarm optimisation and its implementation in mobile robot path planning. In 2017 *International Seminar on Intelligent Technology and Its Applications (ISITIA)* (pp. 238–243). IEEE. <https://doi.org/10.1109/ISITIA.2017.8124087>.
- Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1), 3–55.
- Sharma, A., Imoto, S., & Miyano, S. (2011). A top-r feature selection algorithm for microarray gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(3), 754–764.
- Sharapov, R. R. (2007). *Genetic algorithms: Basic ideas, variants and analysis*. INTECH Open Access Publisher.
- Sheikhpour, R., Sarram, M. A., Gharaghani, S., & Chahooki, M. A. Z. (2017). A Survey on semi-supervised feature selection methods. *Pattern Recognition*, 64, 141–158. <https://doi.org/10.1016/j.patcog.2016.11.003>.
- Shelokar, P. S., Jayaraman, V. K., & Kulkarni, B. D. (2004). An ant colony approach for clustering. *Analytica Chimica Acta*, 509(2), 187–195. <https://doi.org/10.1016/j.aca.2003.12.032>.
- Shen, Q., Mei, Z., & Ye, B. X. (2009). Simultaneous genes and training samples selection by modified particle swarm optimisation for gene expression data classification. *Computers in Biology and Medicine*, 39(7), 646–649. <https://doi.org/10.1016/j.combiomed.2009.04.008>.
- Shen, X., & Huang, H. C. (2010). Grouping pursuit through a regularization solution surface. *Journal of the American Statistical Association*, 105(490), 727–739. <https://doi.org/10.1198/jasa.2010.tm09380>.
- Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimisation. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 3, 1945–1950.
- Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Aguiar, R. C. T., Gaasenbeek, M., Angelo, M., Reich, M., Pinkus, G. S., Ray, T. S., Koval, M. A., Last, K. W., Norton, A., Lister, T. A., Mesirov, J., Neuberg, D. S., Lander, E. S., Aster, J. C., & Golub, T. R. (2002). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1), 68–74. <https://doi.org/10.1038/nm0102-68>.

- Shukla, A. K., Tripathi, D., Reddy, B. R., & Chandramohan, D. (2019). A study on metaheuristics approaches for gene selection in microarray data: algorithms, applications and open challenges. *Evolutionary Intelligence*, 0123456789. <https://doi.org/10.1007/s12065-019-00306-6>.
- Silwattananusarn, T., Kanarkard, W., & Tuamsuk, K. (2016). Enhanced Classification Accuracy for Cardiocogram Data with Ensemble Feature Selection and Classifier Ensemble. *Journal of Computer and Communications*, 04(04), 20–35. <https://doi.org/10.4236/jcc.2016.44003>.
- Simon, R. (2003). Supervised analysis when the number of candidate features ( $p$ ) greatly exceeds the number of cases ( $n$ ). *ACM SIGKDD Explorations Newsletter*, 5(2), 31. <https://doi.org/10.1145/980972.980978>.
- Simons, C. L., & Parmee, I. C. (2010). Dynamic parameter control of interactive local search in UML software design. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, November 2010*, 3397–3404. <https://doi.org/10.1109/ICSMC.2010.5642479>.
- Sivagaminathan, R. K., & Ramakrishnan, S. (2007). A hybrid approach for feature subset selection using neural networks and ant colony optimisation. *Expert Systems with Applications*, 33(1), 49–60. <https://doi.org/10.1016/j.eswa.2006.04.010>.
- Smętek, M., & Trawiński, B. (2011, September). Investigation of self-adapting genetic algorithms using some multimodal benchmark functions. In *International Conference on Computational Collective Intelligence* (pp. 213-223). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-23935-9\\_21](https://doi.org/10.1007/978-3-642-23935-9_21).
- Song, Q., Ni, J., & Wang, G. (2013). A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 25(1), 1–14. <https://doi.org/10.1109/TKDE.2011.181>
- Song, X. F., Zhang, Y., Gong, D. W., & Gao, X. Z. (2021). A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimisation for high-dimensional data. *IEEE Transactions on Cybernetics*.
- Spears, W. M. (1995). Adapting Crossover in Evolutionary Algorithms. *Proceedings of the 4th Conference on Evolutionary Programming, July 1998*, 367–384.
- Srinivas, M., & Patnaik, L. M. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4), 656–667.
- Storey, J. D., & Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America*, 100(16), 9440–9445. <https://doi.org/10.1073/pnas.1530509100>.
- Stützle, T., & Hoos, H. H. (2000). MAX–MIN ant system. *Future Generation Computer Systems*, 16(8), 889–914.

- Stützle, T., López-Ibáñez, M., Dorigo, M., Cochran, J. J., Cox, L. A., Keskinocak, P., Kharoufeh, J. P., & Smith, J. C. (2011). A concise overview of applications of ant colony optimisation. *Wiley Encyclopedia of Operations Research and Management Science*, 2, 896–911.
- Swarnkar, T., & Mitra, P. (2015). Graph-based unsupervised feature selection and multiview clustering for microarray data. *Journal of Biosciences*, 40(4), 755–767. <https://doi.org/10.1007/s12038-015-9559-8>.
- Swets, D. L., & Weng, J. J. (1995). Efficient content-based image retrieval using automatic feature selection. *Proceedings of the IEEE International Conference on Computer Vision*, 85–90. <https://doi.org/10.1109/iscv.1995.476982>.
- Tabakhi, S., & Moradi, P. (2015). Relevance-redundancy feature selection based on ant colony optimisation. *Pattern Recognition*, 48(9), 2798–2811. <https://doi.org/10.1016/j.patcog.2015.03.020>.
- Tabakhi, S., Moradi, P., & Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimisation. *Engineering Applications of Artificial Intelligence*, 32, 112–123. <https://doi.org/10.1016/j.engappai.2014.03.007>.
- Tabakhi, S., Najafi, A., Ranjbar, R., & Moradi, P. (2015). Gene selection for microarray data classification using a novel ant colony optimisation. *Neurocomputing*, 168, 1024–1036. <https://doi.org/10.1016/j.neucom.2015.05.022>.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.
- Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., & Golub, T. R. (1999). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences of the United States of America*, 96(6), 2907–2912. <https://doi.org/10.1073/pnas.96.6.2907>.
- Tan, M., Tsang, I. W., & Wang, L. (2013). Minimax sparse logistic regression for very high-dimensional feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 24(10), 1609–1622. <https://doi.org/10.1109/TNNLS.2013.2263427>.
- Tang, J., Alelyani, S., & Liu, H. (2014b). Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, 37.
- Tang, K., Li, Z., Luo, L., & Liu, B. (2015). Multi-strategy adaptive particle swarm optimisation for numerical optimisation. *Engineering Applications of Artificial Intelligence*, 37, 9–19. <https://doi.org/10.1016/j.engappai.2014.08.002>.
- Tang, W., Shi, Z., & Wu, Y. (2013). Regularized simultaneous forward-backward greedy algorithm for sparse unmixing of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 52(9), 5271–5288. <https://doi.org/10.1109/TGRS.2013.2287795>

- Taradeh, M., Mafarja, M., Heidari, A. A., Faris, H., Aljarah, I., Mirjalili, S., & Fujita, H. (2019). An evolutionary gravitational search-based feature selection. *Information Sciences*, 497(May), 219–239. <https://doi.org/10.1016/j.ins.2019.05.038>
- Tarokh, M., & Zhang, X. (2014). Real-time motion tracking of robot manipulators using adaptive genetic algorithms. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 74(3–4), 697–708. <https://doi.org/10.1007/s10846-013-9860-4>
- Teo, J. (2006). Exploring dynamic self-adaptive populations in differential evolution. *Soft Computing*, 10(8), 673–686. <https://doi.org/10.1007/s00500-005-0537-1>
- Thaher, T., Heidari, A. A., Mafarja, M., Dong, J. S., & Mirjalili, S. (2020). Binary Harris Hawks optimiser for high-dimensional, low sample size feature selection. In *Evolutionary machine learning techniques* (pp. 251-272). Springer, Singapore. [https://doi.org/10.1007/978-981-32-9990-0\\_12](https://doi.org/10.1007/978-981-32-9990-0_12)
- Theodoridis, S., & Koutroumbas, K. (2009). Pattern recognition. 2003. *Elsevier Inc.*
- Tibshirani, R., Hastie, T., Narasimhan, B., & Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, 99(10), 6567–6572. <https://doi.org/10.1073/pnas.082099299>
- Too, J., Abdullah, A. R., Saad, N. M., & Ali, N. M. (2018). Feature selection based on binary tree growth algorithm for the classification of myoelectric signals. *Machines*, 6(4). <https://doi.org/10.3390/machines6040065>.
- Too, J., & Rahim Abdullah, A. (2020). Binary atom search optimisation approaches for feature selection. *Connection Science*, 32(4), 406–430. <https://doi.org/10.1080/09540091.2020.1741515>.
- Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, 98(9), 5116–5121. <https://doi.org/10.1073/pnas.091062498>.
- Unler, A., Murat, A., & Chinnam, R. B. (2011). Mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences*, 181(20), 4625–4641. <https://doi.org/10.1016/j.ins.2010.05.037>.
- Vafaei, F., & Nelson, P. C. (2009). A genetic algorithm that incorporates an adaptive mutation based on an evolutionary model. *8th International Conference on Machine Learning and Applications, ICMLA 2009*, 101–107. <https://doi.org/10.1109/ICMLA.2009.101>.
- Vajda, P., Eiben, A. E., & Hordijk, W. (2008). Parameter Control Methods for Selection. *Methods*, 1, 1–10.
- van Dam, S., Vösa, U., van der Graaf, A., Franke, L., & de Magalhães, J. P. (2018). Gene co-expression analysis for functional classification and gene-disease predictions. *Briefings in Bioinformatics*, 19(4), 575–592. <https://doi.org/10.1093/bib/bbw139>.

- Vengatesan, K., Singh, R. P., Mahajan, S. B., & Sanjeevikumar, P. (2017). Statistical analysis of gene expression data using biclustering coherent column. *International Journal of Pure and Applied Mathematics*, 114(9 Special Issue), 447–454.
- Vinay Kumar, S. B., & Rao, P. V. (2017). Interactive Self Improvement Based Adaptive Particle Swarm Optimisation. *New Review of Information Networking*, 22(1), 13–33.  
<https://doi.org/10.1080/13614576.2017.1297732>.
- Wan, Y., Wang, M., Ye, Z., & Lai, X. (2016). A feature selection method based on modified binary coded ant colony optimisation algorithm. *Applied Soft Computing Journal*, 49(August 2016), 248–258. <https://doi.org/10.1016/j.asoc.2016.08.011>.
- Wang, S., Pedrycz, W., Zhu, Q., & Zhu, W. (2015). Subspace learning for unsupervised feature selection via matrix factorization. *Pattern Recognition*, 48(1), 10–19.  
<https://doi.org/10.1016/j.patcog.2014.08.004>.
- Wang, Z., Li, M., & Li, J. (2015). A multi-objective evolutionary algorithm for feature selection based on mutual information with a new redundancy measure. *Information Sciences*, 307, 73–88. <https://doi.org/10.1016/j.ins.2015.02.031>.
- Wei, J., Zhang, R., Yu, Z., Hu, R., Tang, J., Gui, C., & Yuan, Y. (2017). A BPSO-SVM algorithm based on memory renewal and enhanced mutation mechanisms for feature selection. *Applied Soft Computing Journal*, 58, 176–192. <https://doi.org/10.1016/j.asoc.2017.04.061>.
- Wu, X., Ye, Y., & Zhang, L. (2003). Graphical modeling based gene interaction analysis for microarray data. *ACM SIGKDD Explorations Newsletter*, 5(2), 91.  
<https://doi.org/10.1145/980972.980984>.
- Xiang, S., Nie, F., Meng, G., Pan, C., & Zhang, C. (2012). Discriminative least squares regression for multiclass classification and feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 23(11), 1738–1754. <https://doi.org/10.1109/TNNLS.2012.2212721>
- Xu, F., Pan, L., Zong, R. U. I., Gao, H. A. O., & Lu, H. (2021). *A Hybrid Feature Selection Algorithm Based on a Discrete Artificial Bee Colony for Parkinson 's Diagnosis*. 21(3).
- Xu, R., Damelin, S., Nadler, B., & Wunsch, D. C. (2008). Clustering of high-dimensional gene expression data with feature filtering methods and diffusion maps. *BioMedical Engineering and Informatics: New Development and the Future - Proceedings of the 1st International Conference on BioMedical Engineering and Informatics, BMEI 2008*, 1(May), 245–249.  
<https://doi.org/10.1109/BMEI.2008.256>.
- Xu, Z., King, I., Lyu, M. R. T., & Jin, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. *IEEE Transactions on Neural Networks*, 21(7), 1033–1047.  
<https://doi.org/10.1109/TNN.2010.2047114>
- Xue, Y., Zhang, M., Liao, Z., Li, M., Luo, J., & Hu, X. (2018). A contiguous column coherent evolution biclustering algorithm for time-series gene expression data. *International Journal*

- of Machine Learning and Cybernetics*, 9(3), 441–453. <https://doi.org/10.1007/s13042-015-0487-6>
- Yan, X., Nazmi, S., Erol, B. A., Homaifar, A., Gebru, B., & Tunstel, E. (2020). An efficient unsupervised feature selection procedure through feature clustering. *Pattern Recognition Letters*, 131, 277–284.
- Yancang, L. I. (2009). A Novel ACO with Average Entropy. *Journal of Software Engineering and Applications*, 02(05), 370–374. <https://doi.org/10.4236/jsea.2009.25049>.
- Yang, J. B., & Ong, C. J. (2011). Feature selection using probabilistic prediction of support vector regression. *IEEE Transactions on Neural Networks*, 22(6), 954–962. <https://doi.org/10.1109/TNN.2011.2128342>.
- Yang, K., Zheng, J., Yang, M., Zhou, R., & Liu, G. (2013). Adaptive Genetic Algorithm for Daily Optimal Operation of Cascade Reservoirs and its Improvement Strategy. *Water Resources Management*, 27(12), 4209–4235. <https://doi.org/10.1007/s11269-013-0403-3>.
- Yang, L., Li, K., Zhang, W., & Ke, Z. (2016a). Ant colony classification mining algorithm based on pheromone attraction and exclusion. *Soft Computing*, 1–13. <https://doi.org/10.1007/s00500-016-2151-9>
- Yang, L., Li, K., Zhang, W., & Ke, Z. (2016b). Ant colony classification mining algorithm based on pheromone attraction and exclusion. *Soft Computing*, 1–13. <https://doi.org/10.1007/s00500-016-2151-9>
- Yang, L., LI, K., Zhang, W., Wang, Y., & Ke, Z. (2015). Application of a New mAnt-MinerPR Algorithm in Classification Rule Mining. *Journal of Digital Information Management*, 13(5).
- Yang, L., Widyantoro, D. H., Ioerger, T., & Yen, J. (2001). An entropy-based adaptive genetic algorithm for learning classification rules. *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*, 2, 790–796.
- Yang, X. S. (2010). Nature-Inspired Metaheuristic Algorithms Second Edition. In *Luniver Press* (Issue May). <https://doi.org/10.1016/j.chemosphere.2009.08.026>.
- Yang, X., Yuan, J., Yuan, J., & Mao, H. (2007). A modified particle swarm optimiser with dynamic adaptation. *Applied Mathematics and Computation*, 189(2), 1205–1213. <https://doi.org/10.1016/j.amc.2006.12.045>.
- Yang, Y., Shen, H. T., Ma, Z., Huang, Z., & Zhou, X. (2011).  $\ell_{2,1}$ -Norm regularized discriminative feature selection for unsupervised learning. *IJCAI International Joint Conference on Artificial Intelligence*, 1589–1594. <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-267>.
- Yasmin, M., Mohsin, S., & Sharif, M. (2014). Intelligent image retrieval techniques: A survey. *Journal of Applied Research and Technology*, 12(1), 87–103. [https://doi.org/10.1016/S1665-6423\(14\)71609-8](https://doi.org/10.1016/S1665-6423(14)71609-8).

- Yasuda, K., Yazawa, K., & Motoki, M. (2010). Particle swarm optimisation with parameter self-adjusting mechanism. *IEEJ Transactions on Electrical and Electronic Engineering*, 5(2), 256–257. <https://doi.org/10.1002/tee.20525>.
- Yi, Y., Zhou, W., Cao, Y., Liu, Q., & Wang, J. (2016). Unsupervised feature selection with graph regularized nonnegative self-representation. *Chinese Conference on Biometric Recognition*, 591–599.
- YiMing, C. A. O., Man, H. U., & YongLi, X. U. (2020). Correlation analysis between gene locus combinations and intraocular pressure data based on ant colony algorithms. *Journal of Beijing University of Chemical Technology*, 47(4), 94.
- Yu, L., & Liu, H. (2003). Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. *Proceedings, Twentieth International Conference on Machine Learning*, 2, 856–863.
- Yuan, L., Yuan, C.-A., & Huang, D.-S. (2017). FAACOSE: a fast adaptive ant colony optimisation algorithm for detecting SNP epistasis. *Complexity*, 2017.
- Yuan, M., Yang, Z., & Ji, G. (2019). Partial maximum correlation information: A new feature selection method for microarray data classification. *Neurocomputing*, 323, 231–243. <https://doi.org/10.1016/j.neucom.2018.09.084>.
- Zafarani, R., Abbasi, M. A., & Liu, H. (2014). Social media mining: An introduction. *Social Media Mining: An Introduction*, 9781107018, 1–320. <https://doi.org/10.1017/CBO9781139088510>.
- Zeng, Z., Wang, X., Zhang, J., & Wu, Q. (2016). Semi-supervised feature selection based on local discriminative information. *Neurocomputing*, 173, 102–109. <https://doi.org/10.1016/j.neucom.2015.05.119>.
- Zhan, Z. H., & Zhang, J. (2010). Self-adaptive differential evolution based on PSO Learning strategy. *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference, GECCO '10*, 39–46. <https://doi.org/10.1145/1830483.1830490>.
- Zhang, H., Zhou, J., Fang, N., Zhang, R., & Zhang, Y. (2013). An efficient multi-objective adaptive differential evolution with chaotic neuron network and its application on long-term hydropower operation with considering ecological environment problem. *International Journal of Electrical Power and Energy Systems*, 45(1), 60–70. <https://doi.org/10.1016/j.ijepes.2012.08.069>.
- Zhang, J., & Ding, X. (2011). A multi-swarm self-adaptive and cooperative particle swarm optimisation. *Engineering Applications of Artificial Intelligence*, 24(6), 958–967. <https://doi.org/10.1016/j.engappai.2011.05.010>.
- Zhang, J., & Sanderson, A. C. (2009). JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5), 945–958. <https://doi.org/10.1109/TEVC.2009.2014613>.

- Zhang, J., Chen, W. N., Zhan, Z. H., Yu, W. J., Li, Y. L., Chen, N., & Zhou, Q. (2012). A survey on algorithm adaptation in evolutionary computation. *Frontiers of Electrical and Electronic Engineering in China*, 7(1), 16–31. <https://doi.org/10.1007/s11460-012-0192-0>.
- Zhang, J., Ma, Z., Sun, Q., & Yan, J. (2018). Research Review on Algorithms of Community Detection in Complex Networks. *Journal of Physics: Conference Series*, 1069(1). <https://doi.org/10.1088/1742-6596/1069/1/012124>.
- Zhang, X., Zhang, R., Wang, J., & Wang, L. (2018). An adaptive particle swarm optimisation algorithm based on aggregation degree. *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, 11(4), 443–448.
- Zhang, Y., Li, S., Wang, T., & Zhang, Z. (2013). Divergence-based feature selection for separate classes. *Neurocomputing*, 101, 32–42. <https://doi.org/10.1016/j.neucom.2012.06.036>.
- Zhang, Y., Ma, Y., & Yang, X. (2005). Multi-label feature selection based on mutual information. *ICNC-FSKD 2018 - 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, 27(8), 1379–1386. <https://doi.org/10.1109/FSKD.2018.8687220>.
- Zhao, J., Lu, K., & He, X. (2008). Locality sensitive semi-supervised feature selection. *Neurocomputing*, 71(10–12), 1842–1849. <https://doi.org/10.1016/j.neucom.2007.06.014>.
- Zhao, X., Deng, W., & Shi, Y. (2013). Feature selection with attributes clustering by maximal information coefficient. *Procedia Computer Science*, 17, 70–79. <https://doi.org/10.1016/j.procs.2013.05.011>.
- Zhao, Z. (2010). *Spectral feature selection for mining ultrahigh dimensional data*. Citeseer.
- Zhao, Z., & Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. *ACM International Conference Proceeding Series*, 227, 1151–1157. <https://doi.org/10.1145/1273496.1273641>.
- Zhaoquan, C., & Han, H. (2008). Ant colony optimisation algorithm based on adaptive weight and volatility parameters. *Proceedings - 2008 2nd International Symposium on Intelligent Information Technology Application, IITA 2008*, 2, 75–79. <https://doi.org/10.1109/IITA.2008.371>.
- Zhe, C., Sun, A., & Xiao, X. (2019). Community detection on large complex attribute network. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2041–2049. <https://doi.org/10.1145/3292500.3330721>.
- Zielinski, K., & Laur, R. (2007). Adaptive parameter setting for a multi-objective particle swarm optimisation algorithm. *2007 IEEE Congress on Evolutionary Computation*, 3019–3026.
- Zinck, A. (2009). Library of Congress Cataloging-in-Publication Data.