# An Improved Bees Algorithm Local Search Mechanism for Numerical dataset


**Prepared By:**

**Aras Ghazi Mohammed Al-dawoodi**

**Supervisor:**

**Dr. Massudi bin Mahmuddin**




**DISSERTATION SUBMITTED TO THE AWANG HAD SALLEH GRADUATE**

**SCHOOL OF COMPUTING**

**UUM COLLEGE OF ARTS AND SCIENCES**

**UNIVERSITI UTARA MALAYSIA**

**IN PARITIAL FULFILLMENT OF THE REQUIREMENT FOR THE**

**DEGREE OF MASTER IN (INFORMATION TECHNOLOGY)**

**2014 - 2015**

## Permission to Use

In presenting this dissertation report in partial fulfilment of the requirements for a postgraduate degree from University 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 report in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) 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 report or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to University Utara Malaysia for any scholarly use which may be made of any material from my report.

Requests for permission to copy or to make other use of materials in this project report, 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
University Utara Malaysia
06010 UUM Sintok

# Abstrak

Bees Algorithm (BA), satu prosedur pengoptimuman heuristik, merupakan salah satu teknik carian asas yang berdasarkan kepada aktiviti pencarian makanan lebah. Algoritma ini menjalankan sejenis eksploitasi di tetangga digabungkan dengan gelintaran penerokaan rawak. Walau bagaimanapun, isu utama BA ialah ia memerlukan masa pengiraan yang lama serta pelbagai proses pengiraan untuk mendapatkan penyelesaian yang baik, terutamanya dalam isu-isu yang lebih rumit. Pendekatan ini tidak menjamin apa-apa penyelesaian optimum bagi masalah terutamanya masalah kekurangan ketepatan. Untuk menyelesaikan isu ini, gelintaran setempat dalam BA itu disiasat menggunakan Simple swap, 2-Opt dan 3-Opt telah dicadangkan sebagai kaedah asal untuk Bees Algorithm Feature Selection (BAFS). Dalam kajian ini, cadangan lanjutan kaedah asal adalah 4-Opt sebagai gelintaran yang dibentangkan. Cadangan ini telah dilaksanakan dan membandingkan secara komprehensif dan menganalisis prestasi mereka berkaitan dengan kejituan dan masa. Tambahan pula, dalam kajian ini algoritma pemilihan ciri dilaksanakan dan diuji menggunakan set data paling popular dari (UCI) Machine Learning Repository. Keputusan yang diperolehi daripada kerja-kerja eksperimen mengesahkan bahawa cadangan lanjutan carian komuniti termasuk pendekatan 4 Opt telah menyediakan ramalan ketepatan yang lebih baik dengan masa yang sesuai daripada BAFS asal.



**Kata Kunci :** Bees Algorithm (BA), Feature selection, Local search, Simple swap, 2-Opt and 3-Opt, 4-Opt.

# Abstract

Bees Algorithm (BA), a heuristic optimization procedure, represents one of the fundamental search techniques is based on the food foraging activities of bees. This algorithm performs a kind of exploitative neighbourhoods search combined with random explorative search. However, the main issue of BA is that it requires long computational time as well as numerous computational processes to obtain a good solution, especially in more complicated issues. This approach does not guarantee any optimum solutions for the problem mainly because of lack of accuracy. To solve this issue, the local search in the BA is investigated by Simple swap, 2-Opt and 3-Opt were proposed as Massudi methods for Bees Algorithm Feature Selection (BAFS). In this study, the proposed extension methods is 4-Opt as search neighbourhood is presented. This proposal was implemented and comprehensively compares and analyse their performances with respect to accuracy and time. Furthermore, in this study the feature selection algorithm is implemented and tested using most popular dataset from Machine Learning Repository (UCI). The obtained results from experimental work confirmed that the proposed extension of the search neighbourhood including 4-Opt approach has provided better accuracy with suitable time than the Massudi methods.



**Keywords:** Bees Algorithm (BA), Feature selection, Local search, Simple swap, 2-Opt and 3-Opt, 4-Opt approaches.

## Acknowledgement

بسم الله الرحمن الرحيم

*"In the name of Allah, the Most Gracious, the Most Merciful"*

All praises and thanks to the Almighty, Allah (SWT), who helps me to finish this study, Allah gives me the opportunity, strength and the ability to complete my study for a Master degree after a long time of continuous work.

There are several people who made this thesis possible and to whom I wish to express my gratitude. Firstly, I would like to thank my supervisor, Dr. Massudi bin Mahmuddin, for his excellent supervision, support, and encouragement during my study here at University Utara Malaysia (UUM).

Finally, it would not be possible for me to complete the study and this dissertation without the help of Allah and then supporting and encourage from my family and friends. First and foremost, my great thanks go to my parent (Ghazi and Brewin) to motivate and support me, may Allah bless them. To my brother (Alan) and my sisters (Alaa) for supporting me and had a great influence to finish my master study. I would like to thanks my great friends Mr.Wisam Dawood, Mr.Saifuldeen, Mr. Zakaria Noor, Mr.Ahmed Abdul Rahman, Mr.Ahmad Mahdi, Mr.Mahmoud Mehdi and Mr.Muhammad Sami, thanks for standing beside me and giving support in all periods of study

My gratitude is extended to my Colleagues in the Ministry of Higher Education and Scientific and especially Tikrit University and University Utara Malaysia for innumerable discussions, suggestions, critics, and helps during my study that create a stimulating and enjoyable atmosphere to work in.

Last but not least, I would like to express the warmest love of my wife Evin and kids Rowand and Sanarya, for their love, courage, and always backing me when I need it most. Finally, to my relatives and friends in Kurdistan and Iraq.

Thanks for all persons who helped or contributed to finish my Master program.

**Aras Ghazi Mohammed Al-dawood - 2015**

# Table of Contents

# List of Tables

# List of Figure

# Abbreviations

| | |
|---|---|
| BA | Bees Algorithm |
| FS | Feature Selection |
| BAFS | Bees Algorithm Feature Selection |
| GA | Genetic Algorithm |
| ML | Machine Learning |
| MLP | Multilayer Perceptron |
| PSO | Particle Swarm Optimisation |
| TSP | Travel Salesman Problem |
| *N* | Total number of features in a data set |
| *Ns* | Total number of evaluated features |
| *Nt* | Total number of selected features |
| *m* | Number of sites selected for neighbourhood search |
| *e* | Number of best "elite" sites out of m selected sites |
| *nep* | Number of bees recruited for the best e sites |
| *nsp* | Number of bees recruited for the other (m-e) selected sites |
| R | Maximum iterations of Bees Algorithm |
| KDD | Knowledge Discovery in Databases |

## List of Appendices

# CHAPTER ONE
# INTRODUCTION

## 1.1 Optimisation Algorithms

Nature-inspired optimisation algorithms have gained considerable attention in recent years [1]. Its role is crucial and manifold in a wide number of research areas as varied as computer science, operational research, mathematics, and artificial intelligence where it is used as an optimum solution for complex problems [1][2]. A number of optimisation algorithms have been proposed to solve varied problems including real-time issues like Traveling Salesman Problem (TSP), Cutting Stock Problem, Packing Problems, Minimum Spanning Tree (MST) and timetabling problems, which are difficult to resolve in traditional way [3][4].

One of the common ways of resolving optimisation problems is the use of Swarm-based optimisation algorithms, such as Bees Algorithm (BA) [5], Ant Colony Optimisation [6], Bat Algorithm [7], Particle Swarm Optimisation [8], Firefly Algorithm [9], Cuckoo search [10] and so on. However, there is no algorithm that can single-handedly resolve all sorts of optimisation problems [1][11][12][13][14], mainly due to the massive amount of data and their applications with each introducing different types of problem that requires different algorithm to bring out solutions. This has further led to the development of various optimisation methods to resolve different optimisation problems. In order to choose the best method for a given problem, one must first identify and understand the type of the problem [15]. The challenge here is that for each problem, there are different algorithms offering the optimum result [1].

The situation is further complicated as the development of massive data has made data analysis and inference all the more difficult. Moreover, the process of data analysis and inference is not consistent with its rapid growth due to the frequent alterations in optimisation in the form of relevant, irrelevant, and redundant features [16]; (a) relevant: this class of features has strong impact on the output; (b) irrelevant: opposite to relevant features, irrelevant features do not have any bias on the output; (c) redundant: a redundancy occurs when a feature captures the functionality of other. All of these complexities significantly affect the performance of the results in terms of accuracy and time-efficiency [2].

Confirmed by [17], the quality of the data is one of the most important factors influencing the performance of the results. While data collection and problem representation have been increasing, feature selection is also being used in many machine learning tasks [18]. Machine learning is the most commonly used technique to address larger and more complex tasks by analysing the most relevant information already present in databases [4].



Over the last few years, researchers have been actively using machine learning techniques for data analytics. Machine learning generates models for learned data-based prediction out of which a new pattern can be predicted. However, employing a machine learning model alone does not guarantee the best possible solutions [2]. For example, a neural network learning algorithm needs an optimisation technique to obtain the best result. Referring to [19][2], optimisation is the most significant component required to produce the best machine learning outcome. In other words, machine learning methods are used for big data mining to optimize a performance criterion using sample data or past experience [2].

One of the serious challenges in machine learning is the selection of relevant features and elimination of irrelevant ones [20]. Massive data normally are characteristically contain a large amount of irrelevant features. Such data can be time-consuming and labour intensive during the Knowledge Discovery in Databases (KDD) processes [2][19][21]. One possible solution to eliminate the unwanted features from the raw data is to use the Feature Selection (FS) technique [22]. At the pre-processing phase of machine learning, FS is used for selecting a subset from the original set of features in order to form patterns in the training dataset. FS has been considered as a significant field of research and development since 1970s. In recent years, FS has been successfully applied in various types of problem scenarios, such as data mining applications, information retrieval processing, pattern classification, text categorization, and genomic analysis [2][23][24]. The benefits of FS are manifold: it speeds up algorithm for data mining, improves predictive accuracy, develops inductive learning, reduces the complexity of the induced model, and increases the comprehensibility [25][2][25][17].

In general, FS can be viewed in three approaches: 'Wrapper', 'Filter' and 'Embedding' approaches [22]. A comparative analysis of these approaches is important to understand their accuracy and efficiency [2]. A learning algorithm with the optimisation that uses the Wrapper approach incorporates an optimisation tool and evaluates a model, whereas the filters approach are similar to wrappers in the search approach, but instead of evaluating against a model, a simpler filter is evaluated. In the other words, inductive algorithms are used by wrapper methods as the evaluation function whereas filter methods are independent of the inductive algorithm [16]. In the context of FS, the Filter approach is faster but less accurate and computationally intensive than the Wrapper approach [18][2][26]. The Wrapper approach is one of the most widely used approaches due to its adequate results and efficiency in handling larger and more complex dataset as compared to the Filter approach [27]. However, it is an expensive technique as it involves a complex process of building a classifier with hundreds of items to evaluate one feature subset and dispensing huge numbers of features [28][29]. In the Embedding approach, the learning algorithm builds its own optimisation tools inside its framework. This approach is found in various genetics-based machine learning approaches such as [30][23][31]. Examples of the use of the Wrapper approach are [18][5] and [32][33] in case of the Filter approach. According to [34], the Wrapper approach achieves relatively superior results than Filter and Embedded approaches.

While working through these optimisation techniques and approaches, optimisation problems are commonly resolved by Swarm-based optimisation algorithms. In order to resolve complex optimisation problems, bio-inspired optimisation techniques are used due to their robustness, simplicity, and efficiency [35][36]. One of the most widely used bio-inspired optimisation techniques is the Bees Algorithm (BA) [22]. It represents one of the fundamental search techniques and is based on the food foraging activities of bees. In order for the BA to compete successfully with other optimisation algorithms, it is important to analyse the issues and challenges faced by the same. One of the main issues of the BA is the long computational time required during the process. The BA features a neighbourhood search where each bee is assigned to a number of random sites. For selecting the bees that can produce the best solution, BA calculates the fitness of each site. The selected bees are then assigned to the group of the most efficient bees known as “elite” bees. This further consumes more computational time.

In order to resolve these issues, it is important to identify and eliminate all unnecessary repetition impeding the process. However, this doesn't guarantee a better processing time. In some worse cases, the process fails to find an optimal solution. Another solution is to present some of the total iterations. But that obstructs the algorithm from receiving the optimal solution [4].

An improvement of the BA is needed to avoid repetitive neighbourhood search as well as to reduce the total computational time and lack of accuracy. In order to overcome these limitations, the current study uses the Wrapper approach for optimisation. It also develops extension of swapping mechanism operators for a faster and better solution [2][3]. The purpose of proposed an extension approach is not only to accelerate processing but also to maintain the abilities to overcome the drawbacks of the local optimal solution. The study aims to evaluate the proposed results in terms of accuracy and time, and compare the results with Massudi methods that was done by [2].



## 1.2 Problem Statement

With data being available massively across the globe, some of it may be of inaccurate formatting or missing attribute [2]. Recent years have witnessed the accessibility to data on the Internet on a larger scale [13][17]. These data contain large number of features which can be relevant, irrelevant, and redundant [17][19]. In addition, the availability of massive data represents a challenge for analysing them [25][24].

These data consume relatively more time and lack accuracy during the KDD process [17][27][28]. During various KDD phases, the pre-processing phase may require estimating a significant amount of parameters with each feature used in the process adding an independent set of information. This is the ideal situation but in reality, features are often highly correlated, and thus may increase redundancy in the available information. This further impedes data accuracy as well as consumes longer time to get the optimal results [34][31].

In order to resolve these issues, the BA optimisation approach can be used. But it requires appropriate mathematical proof to establish its effectiveness [18][2][29]. As revealed by past literature, due to repetitive iteration, the BA approach takes longer execution time to get the optimal result, especially in local search neighbourhood procedures [2], [23][5]. Bees spend a lot of time identifying the global optimal solution or choosing a good location for producing the best fitness [4][5].



Moreover, BA involves a huge number of computational processes to obtain a good solution, especially in more complicated issues. The approach does not guarantee any optimum solutions for the problem mainly because of lack of accuracy [2][4]. Therefore, a proper action plan is needed to overcome these challenges and propose new operators to: a) reduce computational processes, b) increase accuracy, and c) improve speed [2][18][5].

### 1.3 Research Questions

In order to analyse and describe the research problem, the following research questions are addressed in this study:

1. How can improve the Bees algorithm feature selection (BAFS) in terms of speed and accuracy?

### 1.4 Research Objectives

The main objective of the current study is to enhance BA using Wrapper feature selection approach. To achieve this, the study focuses on the following factors need to accomplish:

- To identify the appropriate wrapper techniques for improving speed and accuracy of BAFS.
- To develop extensions of swapping mechanism of BAFS.
- To evaluate the proposed algorithm in item of speed and accuracy of BAFS by comparing the results with Massudi methods [2].



### 1.5 Significance of Research

Bees Algorithm optimisation has an important role in the data analysis process in machine learning and data mining. The goal of this research has described and explained how the Bees Algorithm, an optimisation tool, was used to support optimisation in terms of speed and accuracy in the Data mining process phase, the pre-processing phase from KDD requires preparing and finding the optimal or near-optimal solution for dataset. However, irrelevant or unnecessary features may reduce the performance of the system. The Bees Algorithm was studied to discover its best characteristics for the feature selection optimisation problem. The Bees Algorithm framework was used to solve the feature selection problem for numerical dataset. Therefore, the current study focuses on improving the speed and increasing the accuracy by identifying relevant and remove irrelevant and redundant data through an improved local search mechanism for BAFS approach. The main disadvantage of the Bees Algorithm, its long computational time in finding the optimum, has also been addressed by proposed an extension swapping operator.

## 1.6 Research Outcomes

The study aims at achieving the following outcomes:

1- Obtain data with better accuracy through BA with Wrapper approach.

2- Speed up the output result.

## 1.7 Scope of the Study

In the context of massive data mining, KDD has varied phases and this study focuses only on its pre-processing phase. The research concentrates on reducing the irrelevant and redundant data in order to increase the speed and improve the accuracy of feature selection methods supported by BA. This study also proposes an extension local search technique mechanism based on the BA architecture to reduce the limitations and drawbacks of optimisation techniques. There are two types of optimisation associated with the BA methods: continuous and combinatorial optimisation problem. The current study focuses on the combinatorial problem of optimisation.

# CHAPTER TWO
# LITERATURE REVIEW

## 2.1 Knowledge Discovery in Databases (KDD)

With the significant and speedy growth of massive data around the world, it is difficult to discover knowledge in a traditional way. This situation motivates researchers to look for alternative automated methods to identify and learn new knowledge through knowledge discovery in databases (KDD). As shown in Figure 2.1, KDD involves five phases: preparing, cleaning, reducing, mining, and evaluating data. The current study focuses on data reduction at the pre-processed data phase [2].



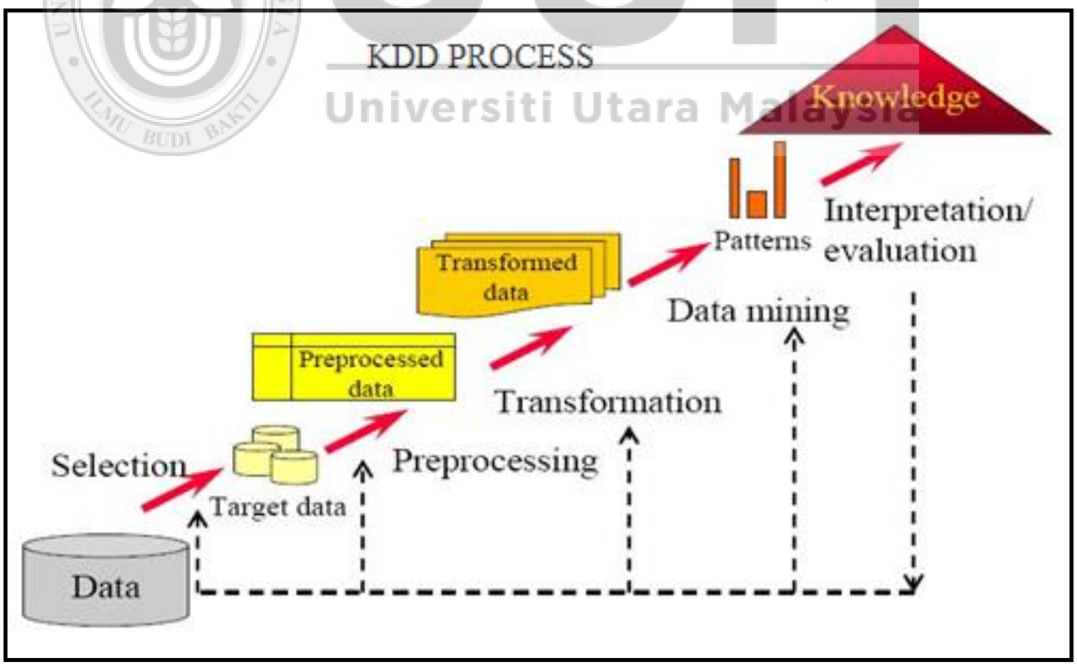


*Figure 2.1.* The knowledge discovery process (Reproduced from [2] as cited in [33])

The representation of data in the pre-processing phase of KDD often uses too many features out of which only a few may be related to the target concept. The features also include irrelevant (data with no useful information [36]) and redundant (data with no significant contribution to the characteristics of the entire dataset [36][35]) data [2][32][23]. Data pre-processing removes such unnecessary data and prepares the dataset for further processing [34]. In other words, the pre-processing data phase prepares the subset data and supports it to move to the data mining process phase, as shown in Figure 2.1. Data pre-processing is significant as it influences the quality of information mined out of the dataset in the process of successful data mining [36][35].

In addition, the pre-processing phase involves enhancement of data reliability. In case of massive data mining, some of the patterns may be corrupted as there are missing or extra items in the patterns. This phase executes data correction by adding or removing the required items, resulting in further reduction in the size of existing data by removing irrelevant and redundant data through an optimisation algorithm. Data that are 'believed' not to contribute to knowledge discovery are removed in this phase [32]. In order to understand the mechanisms of knowledge discovery, it is relevant to know the learning methods on data.

### 2.1.1 Learning Methods on Data

Over the last few years, various approaches have been proposed for the learning of object categories based on training data. A learning paradigm has three influential learning approaches: Supervised, Unsupervised, and Reinforcement learning [2]. Supervised or teacher-based learning is a traditional way of providing more explanatory information based on training data for better performance. Suitable for numerical dataset, this learning paradigm has a class label for a set of input patterns with a large training set which is used as a direction to build a generalisation model of the domain problem. Examples of its use include neural network, statistical model, decision tree, and genetics-based models. The current study adopts a supervised learning approach for FS from numerical dataset.

Based on reward (or punishment) [37] during the learning process, reinforcement learning is a trial-and-error learning in a dynamic environment. In this technique, feedback is given based on observation or the impact of the environment. For example, in case of playing chess, the goal is to win the game by defeating the opponent. In order to win the game, a player should have plans and possible solutions so that they know how to map their actions based on the current move. Details of this approach can be seen in [38].

In contrast with the above learning methods, no reward or penalty is awarded in unsupervised learning. It does not require any 'teacher' to understand patterns in the data. Class labels are usually limited (as unlabelled data) or not provided at all. Learning is based on the auto-association of each pattern in the unlabelled data. The unlabelled data is then evaluated, based on predefined criteria such as distance or error, by minimizing error generalization or 'loss regression estimation'. Clustering and self-organising map algorithms use this technique. For example, the learning process in the K-means algorithm is based on the error of Euclidean distance among objects in the training set to the nearest centroid (cluster centre). Current study includes unlabelled data with no classes. For optimisation problem analysis, the study focuses on combinatorial optimisation problems.

### 2.1.2 Combinatorial Problems

Combinatorial problems are an integral part in the field of computer science [39]. Operations that use these optimisation problems are: searching, sorting, generating permutations, generating subsets, generating graphs, and job scheduling.

Combinatorial problems are frequently seen in our everyday life. For example, a regular working parent has a string of daily responsibilities: to drop off his child at school, pick up from the dry cleaners, visit to the health centre and reach office on time to attend his morning meeting. To do all of this, the parent may synchronise the activities in the following way: drop off his child at school, then pick up from the dry cleaners, then visit the health centre, and finally attend his meeting. Or he may pick up from the dry cleaners first, then drop off his child, and so on. In other words, there are numerous possible ways for the parent to optimally carry out his responsibilities. The optimal solution to it depends totally on what the parent chooses to do first in order to save the most time. This is how combinatorial optimisation works.



## 2.2 Optimisation

Optimisation is a process that studies countless possible solutions of a real function in order to bring out the best solution by maximising or minimising the objective function. The objective function can be single or multiple (known as multi-objective optimisation). The values affecting the objective function are its variables. One of the constraints in the objective function is that the nature of the problem is limited and optional in optimisation. So an optimisation problem can be defined as a process to find the optimum solution by minimizing or maximizing the objective function(s) in order to overcome the constraints [2][23] such as combinatorial (discrete) and continuous [40], [41].

## 2.3 Discrete Optimisation Problems

Discrete optimisation is also known as combinatorial optimisation. It looks for an optimal solution in a finite or countable infinite set of potential solutions. Search for the optimal solution is based on a number of criterion functions that are minimized or maximized. Examples of the criterion function are given below:

- Minimization: Distance, cost, weight, traversal span, processing time, energy consumption, material, number of objects.
- Maximization: Value, profit, output, revenue, return, utility, capacity, efficiency, number of objects.
- Solution: Combinatorial structures such as arrangements, combinations, sequences, object choices, chains, subsets, sub-graphs, network routes, assignments, job schedules, packing schemes, and so on.



### 2.3.1 Optimisation and Machine Learning

The significance of optimisation in machine learning is shown by Bennett et al. in 2006 [42]: "Optimisation lies at the heart of machine learning. Most machine learning problems reduce to optimisation problems." Machine learning uses dataset to build models during the learning process. The new generalized model is used to predict future cases. The performance of the model is evaluated on the basis of some generalization error of the target and the original labelled classes. In case of any errors generated through the learning process, penalties are incurred.

Machine learning is closely associated with optimisation, especially when it comes to determining the optimal or near-optimal solution. Optimisation is used by various machine learning algorithms to determine optimal criteria [42]. The Support Vector Machine (SVM) learning process uses optimisation to choose appropriate learning parameters [43]. Optimisation is used for the same purpose by neural networks [44] and Radial Basis Function (RBF) [45]. The current study includes a multilayer perceptron (MLP) network as the learning algorithm to guide FS.

### 2.3.2 Optimisation Techniques

Optimisation can be broadly grouped into three main approaches: heuristic, mathematical, and natural computation. Heuristic-based algorithms are observed in hill climbing, Tabu search, GA, simulated annealing, ant colony optimisation, and particle swarm optimisation. Natural optimisation approaches are observed in swarm-based optimisation (ACO, BA and PSO) and genetic programming.



Optimisation depends greatly on the nature of the problem. For instance, numerical optimisation is only suitable for numerical data with either real or integer values. Numerical optimisation can be applied to a combinatorial problem since these are concerned with finding the best possible combination of a set of space solutions. In case of multi-objective optimisation, the objective function is minimised or maximised simultaneously. Multi-objective optimisation is applied in areas as varied as process control, risk management, equipment design, industrial product design, and robot design [2].

### 2.3.3 Optimisation Algorithms

Development of different optimisation algorithms has a long history to study. To start with, heuristic-based algorithms are widely used by researchers as natural (for simplicity and efficiency) phenomenon [46] and modern (as a scientific method for optimisation) phenomenon [47]. Between 1940s and 1960s, heuristic methods have been used in various applications. Evolutionary algorithms were developed by Rechenberg & Schwefel in 1963 [48][49] Evolutionary programming was developed in 1966 [50][51]. Genetic algorithms were developed by Holland in 1960s and 1970s [52].

Based on Darwin's principle of survival of the fitness, Holland's seminal book on genetic algorithms was published in 1975. Meta heuristic algorithms were developed between 1980s and 1990s. Pioneered by Richard Feynman's proposal of quantum computing system [53] inspired by quantum mechanics, physics-inspired optimisation algorithms were developed in 1982. In 1983, simulated annealing (SA) was developed, pioneered by Kirkpatrick et al. and inspired by the annealing process of metals [49]. Farmer et al. developed artificial immune systems in 1986. Pioneered by Glover in Tabu search in the 1980s, memory-based meta-heuristics were developed. His book on Tabu search was published later [54]. Tabu search records Tabu moves in a Tabu list without revisiting previous solutions [49].

During 1992s witnessed the development of Ant Colony optimisation (ACO) in Dorigo's PhD thesis on optimisation and natural algorithms [49], [55]. ACO is inspired by the swarm intelligence of social ants that use pheromone as a chemical messenger. In 1992, a book on genetic programming published by Koza of Stanford University introduced a whole new area of machine learning that revolutionised computer programming. Kennedy and Russell developed particle swarm optimisation [56] in 1995. It was based on the behaviour of fish and bird swarms. Differential evolution (DE) was developed by Storn and Price in 1996 and 1997. It is a vector-based evolutionary algorithm that is more efficient than genetic algorithms in many applications [48], [57].

According to [9] developed Harmony search (HS) algorithm in 2001. It is a music-inspired algorithm based on the way a musician adjusts his instruments for great harmony. In 2002, bacteria foraging algorithm was developed by [49]. In 2004, honey bee algorithm was proposed by Nakrani and Tovey for Internet hosting centre optimisation. This was followed by the development of BA by [56]. Artificial bee colony (ABC) optimisation was developed by Karaboga in 2005. In 2008, the author of this study developed firefly algorithm (FA). Cuckoo search (CS) algorithm was developed by [56] in 2009. Kaveh and Talatahari proposed CSS [5] in 2010. It is based on electrostatic theorems such as Coulomb's law, Gauss's law, and superposition principle from electrostatics and Newton's laws of motion. Spiral Galaxy-Based Search Algorithm (GbSA) was developed by [33] in 2011. Salem developed Base Optimisation algorithm in 2012. It is based on mathematics-based algorithms. 2013 witnessed Chemical Reaction Algorithm developed by Melin and Gases Motion Optimisation developed by [58].



### 2.3.4 Search for Optimality

Referring to [49] the formulation of an optimisation problem is followed by the search for optimal solutions using the appropriate mathematical techniques. Searching for the optimal solution is, in a figurative sense, like treasure hunting. Imagine a treasure hunt in a hilly landscape within a time limit. In case of no guidance to find the hidden treasure, and even extreme, if the treasure is placed at the highest peak of a known region, the climb would be directly up to the steepest cliff to reach the highest peak. This scenario refers to the classic hill-climbing technique. In most cases when we do not know where to look for the treasure, we would do a random walk in the hilly region to search the treasure instead of searching every single square inch of it. Such random searches are relevant in modern search algorithms. In case of searching the treasure hunting alone or individually, the search can be termed as a trajectory-based search (for example, simulated annealing).

Alternatively, treasure hunt search can be done collectively or in a group of people where information (if any treasure found) is shared. This search uses swarm intelligence and can be termed as particle swarm optimisation and/or firefly algorithm. Certain other assumptions are also relevant in the search process. Firstly, if the treasure is really important and the search area is extremely large, the search process may take a very long time. Secondly, if there is no time limit and any part of the region is accessible, it is theoretically possible to find the ultimate treasure. This can be termed as the global optimal solution. Moreover, the search strategy can be refined further. Some hunters are better than others. If we keep the better hunters and recruit new ones, this search corresponds to the genetic algorithms or evolutionary algorithms where search agents improve. All of these search options correspond to modern meta-heuristic algorithms. In such algorithms, the best solutions or agents are used by randomising or replacing the not-so-good ones which is done by evaluating each individual's competence (fitness) and system history (use of memory). This strategy is accurate to design better and efficient optimisation algorithms [49].

## 2.4 Feature Selection Techniques

Features are important in describing a piece of data without which data is meaningless. The aim of this study is to deal with numerical features and explain the relationship of FS and performance. Numerical features can be arranged into three main types: discrete, continuous, and complex. These data were being further used in a classification learning process [2].

FS is an approach of dimension reduction techniques that reduces irrelevant dimensions of features. In order to achieve this, optimisation is relevant in FS. The current study explains optimisation and its role in determining the best feature subset. There are two different learning approaches to generate feature subset: supervised and unsupervised. The current study deals with supervised learning. A supervised learning technique is used for error estimation during the learning process of generating a feature subset. The main purpose of using supervised techniques is to maximize classification accuracy [23][59].



The easy and extensive availability of high-dimensional data on the Internet makes it difficult for machine learning methods to deal with larger number of input features, which is a serious challenge for researchers. In order to efficiently use machine learning methods, pre-processing of the data is essentially required. FS is one of the most frequent and important techniques in data pre-processing, which is also an indispensable component of the machine learning process [1]. FS is also termed as "attribute selection", "variable subset selection", and "variable selection" in statistics and machine learning [60]. FS removes irrelevant and redundant data while detecting relevant features. Irrelevant features are those that provide no useful information whereas redundant features are those that provide no more information than the currently selected features [40][55]. It accelerates data mining algorithms, improves predictive accuracy, and increases comprehensibility.

Ideally, the FS algorithm selects those features from the set of the original features which are necessary and relevant for classification. This selection process further improves the performance of classification process in terms of speed, accuracy, and simplicity [61].

### 2.4.1 General Feature Selection Process

The process of FS removes irrelevant and redundant features from data as they are not important to the class concept, such as microarray data analysis. When the number of samples is countably less than the features, the search space gets meagrely populated which makes machine learning difficult. This further fails the model to differentiate accurately between noise and relevant data [62][60][20]. FS involves two main approaches: individual evaluation and subset evaluation. Individual evaluation ranks the features [13] by assigning the weight of an individual feature according to its degree of relevance. In case of subset evaluation, search strategy is used for constructing candidate feature subsets. The general procedure for FS includes four key steps as shown in Figure 2.2: a) subset generation, b) subset evaluation, c) stopping criteria, and d) result validation. Heuristic search is used for subset generation.  Each state specifies a candidate subset which is evaluated in the search space.

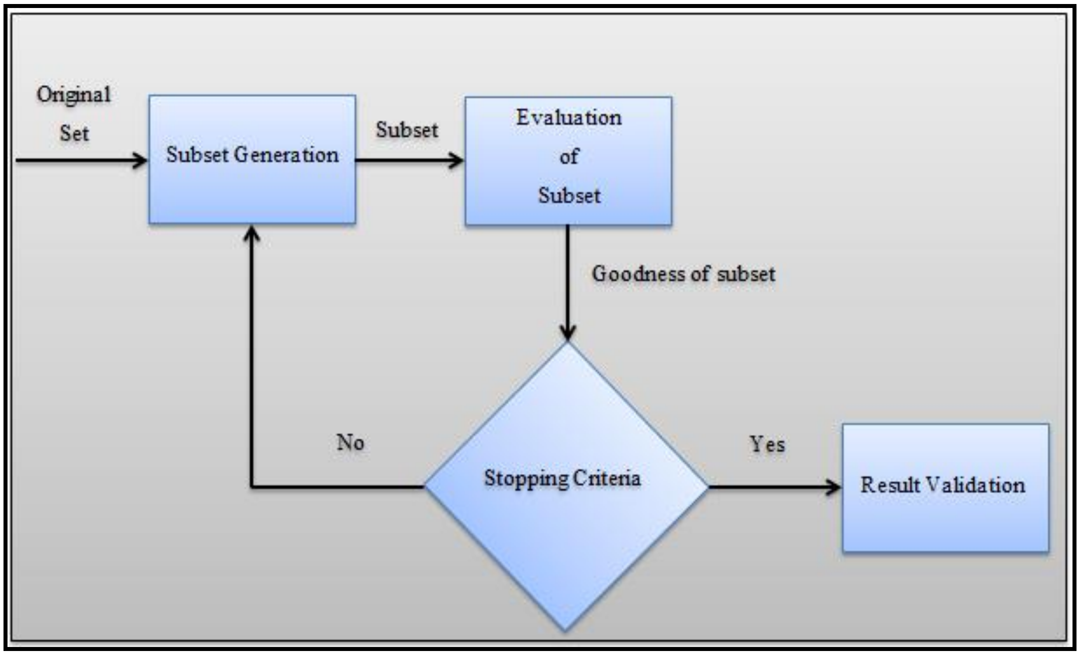


*Figure 2.2.* Four key steps for the FS process (Reproduced from [36])

The nature of the subset generation process is determined by two basic issues:

a) Successor generation decides the search starting point that impacts the search direction. The search starting points at each state are decided through a number of forward, backward, compound, weighting, and random methods [21].

b) Search organization is responsible for the FS process with a specific strategy, which can be sequential, exponential, or random search [27].

As evaluation criteria are relevant to evaluate the newly generated subset, many evaluation criteria have been proposed so far. Evaluation criteria can be categorized into two groups on the basis of their dependency on mining algorithms: independent and dependent criteria [19]. Independent criteria do not involve mining algorithms to evaluate feature set of the training data whereas dependent criteria involve predetermined mining algorithm and its performance for FS. Further, stop criteria must be determined to stop the selection process. The final step of the FS process is result validation. Although validation is not part of the FS process, it is significant for FS method and done through various tests and comparative studies (previously established results or results of competing methods using artificial datasets and/or real world datasets).

FS involves three general approaches: a) the Filter approach evaluates training data without mining algorithm [17], b) the Wrapper approach brings out an optimal feature subset from the specific mining algorithm [29], and c) the Embedded approach selects feature subset from the training data using a specific learning algorithm. Additionally, The applications of FS can be seen in various domains of pattern recognition, including image processing systems, text mining, bioinformatics, and manufacturing [2][63].

### 2.4.2 Feature Selection and Performance

Machine learning uses FS as one of its primary research approaches. During the training process, it uses "semi supervised" learning techniques which is a combination of supervised and unsupervised learning. Semi-supervised learning is used in the context of incomplete data which occurs due to missing or insufficient data for an input-output pattern set, such as in [2][60].

The ML model includes an FS module which is used for evaluating the features of a dataset. As a component of machine learning, FS identifies feature accuracy for a predictive model during the learning process and also to enhance future prediction. In order to find subset of m collection from N number of features ($m < N$, $m \neq 0$), ($2^N$-2) possible combinations are required [2]. This makes FS a highly challenging task for a high dimension dataset (e.g. dataset with at least 100 attributes). This challenge can be minimised by using the BA as an optimisation tool to identify the optimal subset features.

However, there are still many issues for learning techniques with regard to obtaining high performance and more specifically accuracy. For instance, imbalanced raw or pre-processed data is a serious issue. It may happen due to the availability of too much or too little data, noise, and feature excessiveness. In the first case, excessive data may cause the learning algorithm (classifier) to learn nothing. In case of too little data, generalisation during the learning process may be incorrect due to insufficient training. Noisy data diverts the learning algorithm resulting in incorrect generalisation. As shown by Jouve and Nicoloyannis, noise and redundant features can be removed when FS is used in clustering [64].

Data mining process spends around 80% of its resources in cleaning and pre-processing [65]. Because pre-processing does not include any standard procedure. Pre-processing reduces data dimension to a manageable size which is achieved by the reduction of the total number of evaluated features. During the learning process, large and complex features consume more resources. The learning algorithm or the classifier for supervised learning requires more efficiency and a longer time to evaluate the entire dataset.

Larger feature data tend to reduce the accuracy of performance due to which mining such data is not advisable. Moreover, larger feature data with sample dimension cause an over-fitting problem. Ideally, a small sample dataset suffices to describe the concept of the target data [60], [66]. The current study focuses on FS by reducing irrelevant features which can negatively affect the accuracy of the original data.

FS has been studied in various fields related to high dimensional data. The definition of a small and a large number of feature dimensions is still under debate in the FS fraternity. For example, [67] define small (0-19 features), medium (20-49 features), and large scale (over 50 features) features. However, [68] describe the small dataset is less than 40 features. [69] define a "not too large" feature with a dataset of less than 10 features. [70] describe a large feature dataset with more than hundred features. The current study defines a dataset with small (less than 10 features), medium (between 11 and 100 features), and large (more than 100 features) features.

Data mining and knowledge discovery require appropriate techniques for knowledge inference from data. Data mining involves an iterative process to extract the informative structure from a dataset. With an aim to ensure the relevance of the final results to the users' needs, [71] identify three stages of knowledge discovery development where:

a) FS is employed prior to a pre-processing phase to ensure that good quality data is fed during the mining process.
b) FS is applied during the evaluating stage to ensure an effective and efficient performance.
c) FS is used to ensure reliable and comprehensive results.



FS produces reduced features while retaining better accuracy of the newly generated features. FS can be categorised into two main search strategies: complete and sub-optimal search strategies. A complete strategy uncovers all possible features derived from the total number of features. However, the strategy consumes a long time for computation and a longer time in case of larger feature data.

FS is an NP-hard problem as argued by [72],[73][74]. FS is also termed as a combinatorial optimisation problem [75]. As mentioned earlier, FS includes four basic stages of generation, evaluation, stopping, and validation as shown in Figure 2.2. Feature generation produces a subset of features for evaluation which may use (i) forward selection, (ii) backward elimination, or (iii) random generation. At the evaluation stage, the process computes the goodness of the features and replaces the old one with the newest best feature subset. Then the stopping criterion limits the total number of the run. Next, a validation procedure conducts a validity test to compare the results of the selection methods using artificial or real dataset.

The primary objective of FS is to improve performance including learning and accuracy. Moreover, it generates a lesser m number of features of the original data N where $m < N$, $m \neq 0$. As feature dimensionality was reduced, fewer features were being evaluated causing less computational time and thus faster learning. Besides, it produces simple rules (general model) which are more meaningful to the user due to the preference of an easy description instead of a complicated one.

Previous studies on FS included simple but expensive and a brute force approach to identify the best subset features. But this approach is particularly unsuitable for a large dimensional dataset. If there is N number of features, it requires $2^N - 2$ feature combinations. If the dataset includes 32 features, it would generate 232-2 = 4,294,967,294 feature combinations, which is impossible to evaluate. Then there is a statistical technique-based approach that requires some statistical formulation to discover the best feature subset. Packianather and Drake [76] used statistical techniques to evaluate the best features for defect image detection of wood veneer. Recent research on statistical approaches can be seen in [77][78].

Normally, FS can use either a statistical or a classification approach. In the statistical technique, selected features are not dependent on its learning algorithm. The technique uses Castleman's criteria [79] to determine the correlation of inter-class feature variation.

### 2.4.3 Feature Selection Objectives

FS algorithms include the following objectives used by researchers:

- Find the minimal feature subset that suffices the target concept
- Select a subset of N features from a set of M features where N < M and the value of a criterion function is optimized over all subsets of size N
- Choose a feature subset to improve prediction accuracy and decrease the structure size without compromising on the prediction accuracy of the selected feature-based classifier
- To reduce the size of the problem by reducing the computation time and space required to run the algorithms
- To improve classifiers by removing noisy or irrelevant features, and reducing the likelihood of overeating to noisy data
- To identify the features relevant to a specific problem, for example, to demonstrate which gene expressions are relevant in a certain disease.



### 2.4.4 Relevant Features

An optimal feature subset includes all relevant features which leads to the requirement of a proper definition of their relevance. Previous studies classified features by their relevance with three qualifiers: irrelevant, weakly relevant, and strongly relevant, as shown in the graphical representation in Figure 2.3 [50]. To answer the question “relevant to what?” [18], the definition of feature relevance along with the degree of relevance were discussed in the current study. A general algorithm for FS is shown on Figure 2.4.

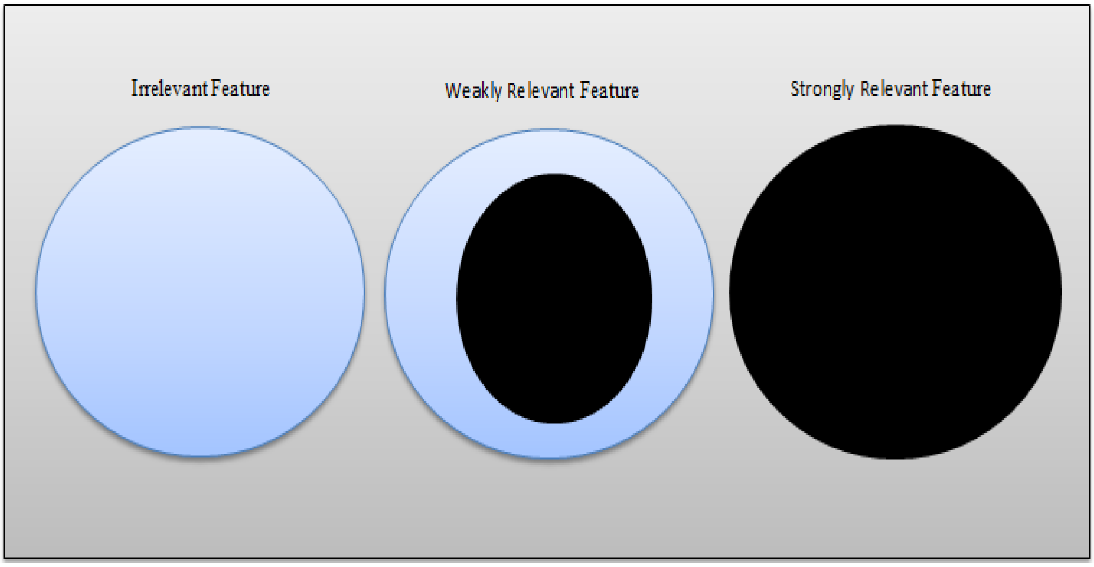


*Figure 2.3.* A view of relevance feature (Reproduced from [36])

***INPUT:***

$X$ : *Set of features of a data set having n features*
$SG$ : *Successor Generator Operator*
$E$ : *Evaluation measure (dependent or independent*
$\theta$ : *Stopping Criteria*

***OUTPUT:***

$X_{opt}$ : *Optimal feature set or weighted features*



***Initialize:***

$X' := Start\_point(X);$
$X_{opt} := \{Best\ of\ X'\ using\ E\ \};$

***Repeat:***

$X' := Search\_Strategy\ (X', SG(E), X);$
$X_{opt} := \{Best\ of\ X'\ according\ to\ E\ \};$
$If\ E(X') \geq E(X_{opt})\ or\ (E(X') == E(X_{opt}) \& |X'| < |X_{opt}|)$
$Then\ X_{opt} = X';$

***Until*** *Stop criteria is not found;*

*Figure 2.4.* General algorithm for FS (Adopted from [36])

### 2.4.5 General Approaches for Feature Selection

Generally, feature selection is the process of selecting relevant features, or a candidate subset of features. FS has three general approaches discussed in the sections below.

#### 2.4.5.1 Wrapper Feature Selection

In 1997, Kohavi and John introduced the Wrapper FS approach [74]. Wrapper FS differs from Filter FS in terms of usage of the learning algorithm. Wrapper FS relies solely on maximizing prediction accuracy of the learning algorithm as shown in Figure 2.6. The general algorithm for Wrapper algorithm is shown in Figure 2.5. This approach uses a learning algorithm known as the "black-box" [80]. During its evaluation process, feedback is used to find the subset features in feature space. In case of Filter FS approaches, the learning algorithm is separated from the main selection process.

FS is also an optimisation problem [81]. Currently, there are many optimisation techniques available including the genetic algorithm (GA), simulated annealing (SA), ant colony optimisation (ACO), and particle swarm optimisation (PSO) [2]. In the current study, the BA is applied to FS due to the promising results it demonstrated in existing research.



Feature space search influences the performance of the Wrapper technique, especially its promptness to find the best subset features in order to avoid an exhaustive search. Wrapper feature selection approaches include the following strategies [16]:

- Forward selection: Its evaluation starts after all features have been considered. In other words, it starts from an empty set of features and adds one feature at a time that meets the selection criterion [2][59].

- Backward elimination: It starts with all features. It is slower but more Table than forward selection in selecting optimal features. It starts from all the features and deletes one feature at a time that adversely affects the selection criterion [2][59].
- Stochastic search: It is based on the specific searching strategy of the particular algorithm. For instance, in the genetic search of GA approaches, a feature mask defines each state in order to perform a genetic operation (such as crossover, and mutation) [2].

The combination of Filter and Wrapper FS techniques can be seen in a number of works. For instance, [82] combined a filter and a GA-wrapper algorithm and tested the proposed hybrid algorithm in Chinese character pattern recognition. Huang et al. [83] used mutual information and GA in their work which resulted in better target accuracy. More works can be seen in [81] and an hybrid statistical FS in [69], [77], [78]. Furthermore, the advantages and disadvantages of the Wrapper, Filter and Embedded approaches have shown in Table 2.1. was adopted from [84] and [85] as cited in [86].



```
INPUT:
D = {X, L}          // a training data set with n number of features where
                    //  X = {f1, f2, f3, ..., fn} and L labels
X'                  // predefined initial feature subset (X' ⊂ X or X' = {φ})
θ                   // a stopping criterion
OUTPUT:   X'opt     // an optimal subset
Begin:
Initialize:
     Xopt = X';
     φopt = E(X', A);      // evaluate X'  by using mining algorithm A
do begin
    Xg = generate(X); // Subset generation for evaluation
    φ = E(Xg, A) ;     //  Xg  current subset evaluation by A
   If (φ > φopt)
       φopt = φ;
       X'opt =  Xg;
repeat (until θ is not reached );
end
 return X'opt;
end;
```

*Figure 2.5.* A general algorithm for wrapper approach (adopted from [36])

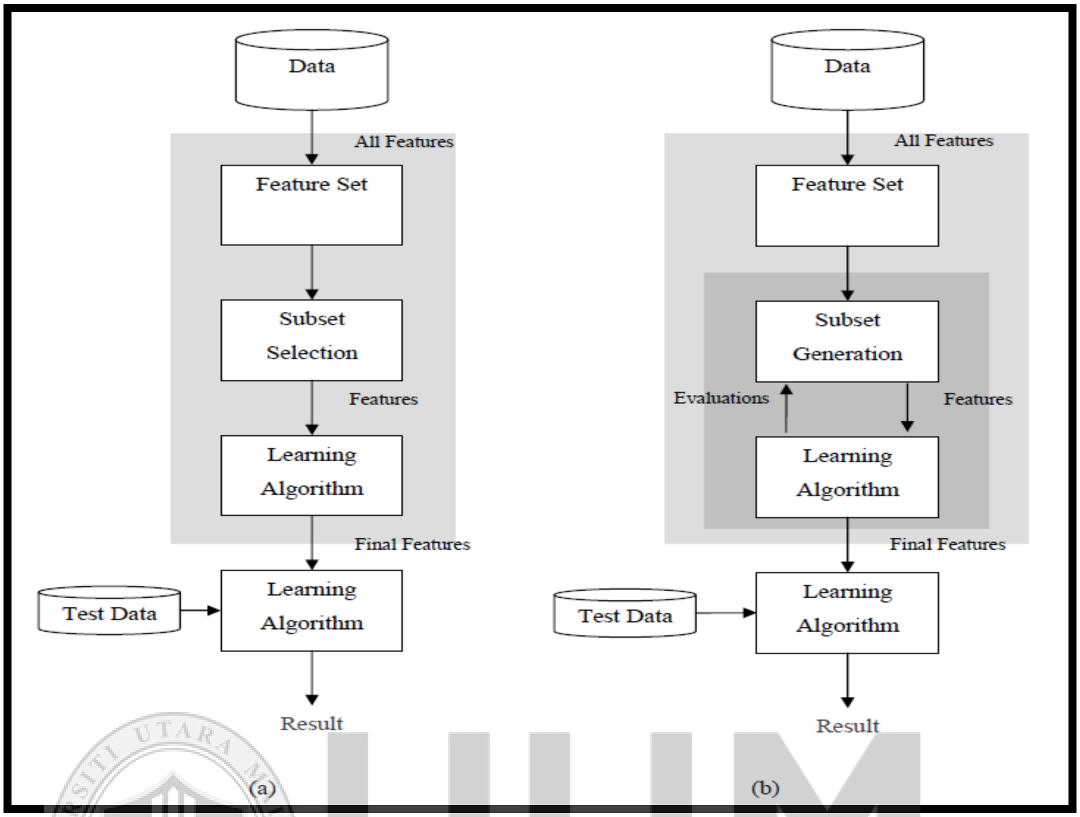


*Figure 2.6.* Feature selection methods (a) filter and (b) wrapper FS (Reproduced from [2])



#### 2.4.5.2 Filter Feature Selection

The Filter approach incorporates an independent measure to evaluate feature subsets without any learning algorithm. This approach is computationally efficient but it can miss features that are useful as combined features [87]. Figure 2.4 shows the graphical representation of the filter model. The general algorithm for Filter approach is shown in Figure 2.7.

The filter approach selects feature subsets free of induction algorithms. In some cases, algorithm-specific measures can be devised and computed efficiently.

Figure 2.4 shows the Filter approach where a pre-processing step is used for FS. However, Filter approach completely ignores how of the selected feature subset influences the performance of the induction algorithm [74]. Furthermore, the advantages and disadvantages of the Wrapper, Filter and Embedded approaches have shown in Table 2.1. was adopted from [84] and [85] as cited in [86].

***INPUT:***
$D = \{X, L\}$ *// a training data set with n number of features where*
*//* $X = \{f_1, f_2, f_3, \ldots, f_n\}$ *and L labels*
$X'$ *// predefined initial feature subset* ($X' \subset X$ *or* $X' = \{\phi\}$)
$\theta$ *// a stopping criterion*
***OUTPUT:*** $X'_{opt}$ *// an optimal subset*

***Begin:***
***Initialize:***
$X_{opt} = X'$;
$\varphi_{opt} = E(X', I_m)$; *// evaluate* $X'$ *by using an independent measure* $I_m$
***do begin***
$X_g = generate(X)$; *// Subset generation for evaluation*
$\varphi = E(X_g, I_m)$ ; *//* $X_g$ *current subset evaluation by* $I_m$
***If*** $(\varphi > \varphi_{opt})$
$\varphi_{opt} = \varphi$;
$X'_{opt} = X_g$;
*repeat (until* $\theta$ *is not reached );*
*end*
*return* $X'_{opt}$;
*end;*

*Figure 2.7.* A general algorithm for filter approach (Adopted from [36])



#### 2.4.5.3 Embedded Feature Selection

Embedded approach includes learning algorithm at a lower computational cost as compared to the Wrapper approach. While considering feature dependencies, it takes into account the relations between input and output features, and the features with better local discrimination. Independent criteria are used in this approach to decide the optimal subsets for a known cardinality. It also uses the learning algorithm to select the final optimal subset across different cardinality [88][89]. Figure 2.4 shows a general embedded algorithm. Furthermore, the advantages and disadvantages of the Wrapper, Filter and Embedded approaches have shown in Table 2.1. was adopted from [84] and [85] as cited in [86].

Table 2.1

*The advantages and disadvantages of filter, wrapper and embedded proach (adopted from [84] [85] as cited in [86])*

| Model | Advantage | Disadvantage | Examples |
|---|---|---|---|
| Filter | Fast, Scalable, independent of classifier, better computational complexity and cheaper than other technique. | Ignores interaction with classifier, ignores feature dependent, less scalable than other techniques. | Correlation-based feature selection (CFS), Euclidean distance, Information gain and Fast correlation-based feature selection (FCBF) |
| Wrapper | Simple, interest with classifier, model feature dependent, better accuracy classification and minimize computational cost. | Intensive computational, risk is over fitting and more expensive. | Sequential forward selection (SFS), Sequential backward elimination (SBE). |
| Embedded | Better computational complexity than others and model feature dependent. | Classifier dependent selection | Decision trees, Weighted naive Bayes, Feature selection using the weight vector of SVM |

**INPUT:**
$D = \{X, L\}$ // a training data set with $n$ number of features where
// $X = \{f_1, f_2, f_3, \ldots, f_n\}$ and $L$ labels
$X'$ // predefined initial feature subset ($X' \subset X$ or $X' = \{\phi\}$)
$\theta$ // a stopping criterion
**OUTPUT:** $X'_{opt}$ // an optimal subset

**Begin:**
**Initialize:**
$X_{opt} = X'$;
$\varphi_{opt} = E(X', I_m)$; // evaluate $X'$ by using independent evaluation measure
$\delta_{opt} = E(X', A)$; // evaluate $X'$ by using mining algorithm $A$
$C_0 = C(X')$; // cardinality calculation of $X'$
*do begin*
*for* $k = C_0 + 1$ *to* $n$
*for* $i = 0$ *to* $n - k$
$X_g = X_{opt} \cup \{f_i\}$; // Subset generation for evaluation with cardinality $k$
$\varphi = E(X_g, I_m)$ ; // evaluation the current subset $X_g$ by $I_m$
*If* $(\varphi > \varphi_{opt})$
$\varphi_{opt} = \varphi$;
$X'_{opt} = X_g$;
*end*
$\delta = E(X'_{opt}, A)$; // evaluating subset $X'_{opt}$ by $A$ learning algorithm
*If* $(\delta > \delta_{opt})$
$X'_{opt} = X_{opt}$;
$\delta_{opt} = \delta$;
*else*
*break and return* $X'_{opt}$
*end*
*return* $X'_{opt}$;
*end*



*Figure 2.8.* A general algorithm for embedded approach (adopted from [36]).

## 2.5 Nature and Non-Nature Inspired Optimisation Algorithms

### 2.5.1 Sources of Inspiration

Nature has inspired researchers in varied ways, thus making both nature-inspired and non-nature-inspired algorithms major sources of inspiration. Figure 2.9 shows examples of non-nature-inspired algorithms: Scatter Search (SS), Iterated Local Search (ILS), Guided Local Search (GLS), and Path Relinking (PR). New algorithms are usually nature-inspired [90] and capable of clarifying and resolving difficult relationships from initial conditions and rules based on limited or no knowledge of the search space. Nature is the ideal model for representing optimisation. Nature-inspired optimisation algorithms examine each and every phenomenon in nature to detect the optimal strategy while treating complex interaction among organisms containing microorganism, human beings, sustaining the ecosystem, keeping on diversity, adaptation, and physical phenomenon [39].

Optimisation algorithms can include different levels of classifications, depending on the relevance of the sources of inspiration, and details and number of sub-sources to be used in the process. For instance, in the context of simplicity, the process used the highest level sources including biology, physics, chemistry or other algorithms. By far, successful characteristics of biological system are used in the majority of nature-inspired algorithms due to which they can also be termed as bio-inspired algorithms [90]. Bio-inspired computation represents a contemporary phase with vast applications covering topics as varied as computer-networking, security issues, robotics, and bio-medical engineering.



Other sources of inspiration for algorithms are physical and chemical systems, as shown in the Figure 2.9 some of which may be based on music as well [91].

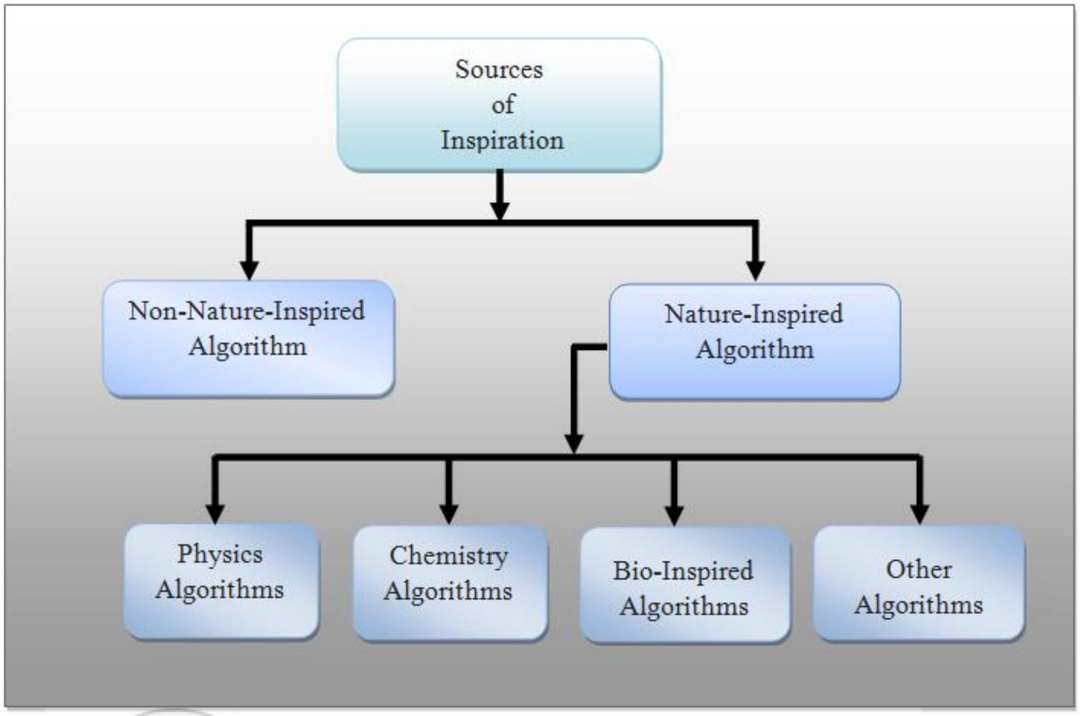


*Figure 2.9.* Classification of sources of inspiration algorithms

### 2.5.2 Classification of Algorithms

All existing algorithms can be broadly divided into many major categories: a) bio-inspired, b) physics, and c) chemistry algorithms along with other algorithms, as shown in Figure 2.5. However, this is not a unique classification as it should be based largely on the focus and perspectives of the algorithms. For example, keeping the trajectory of the search path as the focus and perspective, algorithms can be classified as: trajectory-based (e.g., simulated annealing) and population-based (e.g., particle swarm optimisation and firefly algorithms). Based on the interaction of multiple agents, algorithms can be classified as: attraction-based (e.g., firefly algorithm that uses the attraction of light and fireflies) and non-attraction-based (e.g., genetic algorithms).

Based on the updating equations, algorithms can be classified as: equation-based (e.g., particle swarm optimisation and cuckoo search algorithms) and rule-based (e.g., genetic algorithms as they do not have explicit equations for crossover and mutation). However, this is not a unique classification. For example, firefly algorithm uses three explicit rules which can be converted explicitly into a nonlinear and single updating equation. These classifications show that classification of algorithms depends on the actual perspective and motivations. The primary emphasis of the current study is on the sources of inspiration.

### 2.5.3 Bio-Inspired Algorithm

Bio-inspired algorithms are the meta-heuristics that imitate the nature's method to resolve optimisation problems. These algorithms use varied nature-inspired processes such as the evolution of species, emergent behaviour of biological societies, and functioning of the vertebrate immune system [55]. The optimisation techniques used in bio-inspired algorithms are: Bees Algorithm (BA) [95] Simulated Annealing (SA) [96], Genetic Algorithm (GA) [88], Particle swarm optimisation (PSO) [47], [49], [56], Big Bang Big Crunch (BBBC) [90], Artificial Bee Colony (ABC) optimisation [97]–[99], and Ant Colony optimisation (ACO) [63]. Bio-inspired algorithms are used to overcome many restrictions of traditional algorithms and also adopted by the fields of science, engineering, and business. Recent years have witnessed researchers use bio-inspired algorithms and applications for solving problems of data mining [27][100], data clustering [100][61], classification [41][18], economic emissions load dispatch [101], and travelling salesman issues [48].

Bio-inspired approaches are widely adopted (more recently in computer science) in literature to solve an impressive array of problems. Figure 2.10 shows successful implications of bio-inspired approaches including Bio-inspired Algorithm, Evolutionary Algorithms, Swarm-based Algorithms, Bacterial Foraging Algorithm, Ecology, Artificial immune system, and other algorithms that are inspired by the collective behaviour and natural evolution in animals.

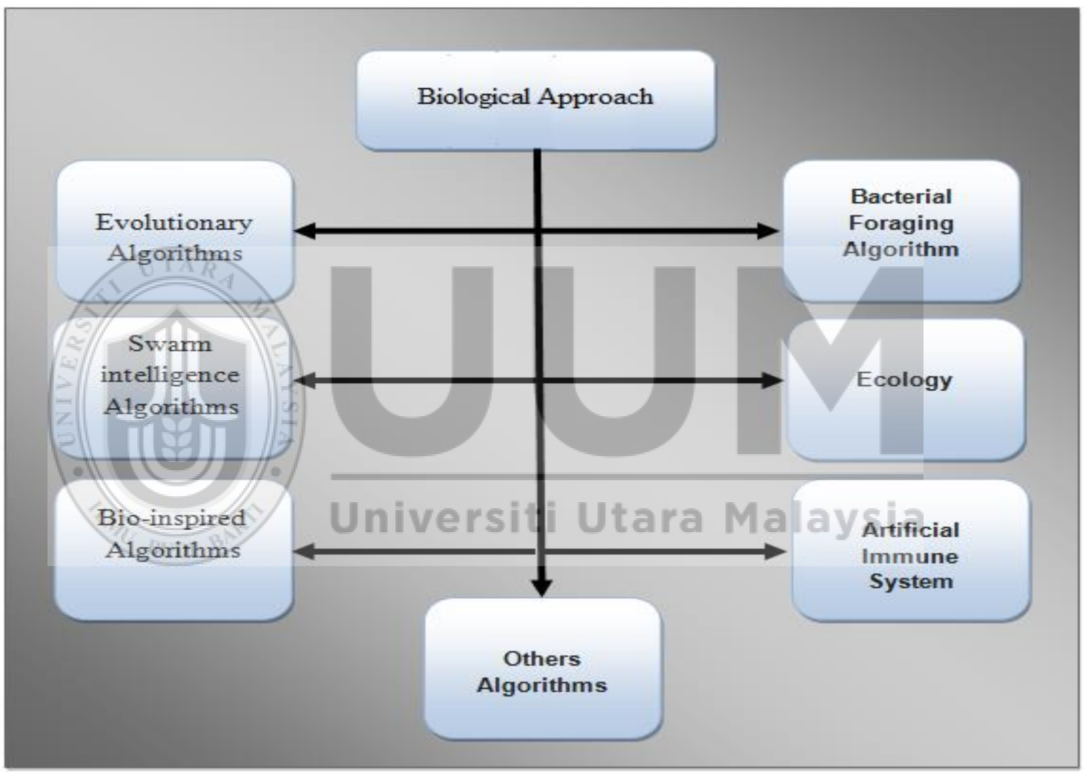


*Figure 2.10.* Classification of biological approach (Reproduced from [45])

### 2.5.4 Bees Algorithm (BA)

Swarm insects are an example of a successful self-organized social group with their structure and activities, which has attracted many researchers. The people of ancient Egypt used honey more than 4000 years ago. There are more than 16,000 different species of bees found in nature including bumblebees, sting-less bees, honeybees, and African bees. These bees, especially the honeybees, collect nectar and pollen from neighbouring flowers and bring them to their hive. These foods from nectar and pollen are largely stored and kept in beeswax in the hive [2].

In the hierarchical structure of bees, a group or a colony is comprised of a "queen" bee, workers (female bees), and drones (male bees). There is only one "queen" bee at one time in a group. She is much bigger than the female bees and lays thousands of eggs per day in order to maintain the colony's capacity and requirement. Adult worker bees provide food for her offspring and construct their own nest [102]. The "queen" bee and the adult female bees have a special 'mother-daughter' relationship, known as eusocial, whose primary concern is to nurture the young bees. Solitariness is the most important aspect of this eusocial relationship, which can also be seen in other swarming inserts including ants, wasps, and some species of termites.

The scout bees begin foraging activities in a colony by sending worker bees to search for promising flower patches which contain large amounts of food (nectar or pollen). Patches with more food tend to visited by more bees with less effort, whereas patches with less food receive fewer bees. During this search, a percentage of the population as scout bees is kept back. When they return to the hive, a certain amount of their food is deposited. Later the scout bees perform the "waggle dance" on the "dance floor", as shown in Figure 2.11.

The waggle dance is essential in the bees' communication as it contains the following information given by the dancer scout bee regarding a flower patch (food source): a) the direction in which it had been found, b) its distance from the hive, and c) its quality rating (or fitness). This information is crucial for other bees to know about the food source and the outside environment by which the colony can evaluate the food sources based on the food quality and the amount of energy invested to collect it. After the waggle dance, the scout bee goes back to the flower patch followed by the recruiter bees. In case of more promising patches, more recruiter bees are sent, thus resulting in quicker and efficient collection of food for the colony.

This process of harvesting observes the food level of the flower patch. This observation is important for the next waggle dance to be performed by the scout bees upon their return to the hive. The scout bees advertised the food source when the flower patch remained at a comfortably high level as a food source.

The foraging activities of bees can be greatly influenced by various weather conditions such as fog, heavy rain, humidity, and temperature. In case of unfavourable weather conditions, foraging activities were less. Moreover, foraging distance varies according to the species and size of the bees. Larger bees can collect more nectar and pollen than the smaller ones. They also take more time to search the food source. [103] observed in their study that 10%, 50%, 25%, and 10 of the bees could forage within 0.5 km, 6km, 7.5 km, and over 9.5 km from the hive respectively.

Bees communicate with each other if they meet one or more bees at the same flower. This communication is again based on dance, as shown in Figure 2.11. Bees perform three main dance rituals based on the distance of the food source from the hive: a) the round dance refers to a relatively short distance (less than 50m), b) the sickle dance is used for a medium distance (50-150m), and c) the waggle dance is used for longer distances (more than 150m) [49].

The bees also communicate at the hive. For example, honeybees communicate with other bees to dispense information on the food source and its location after they return from successful foraging trips. Navigation of the bees is determined by the sun positions throughout the day, as shown in Figure 2.12. Based on the information of the successful bees, more bees were deployed to the location for more food collection. This helps the bees find a food source in shorter time, especially in case of random foraging.

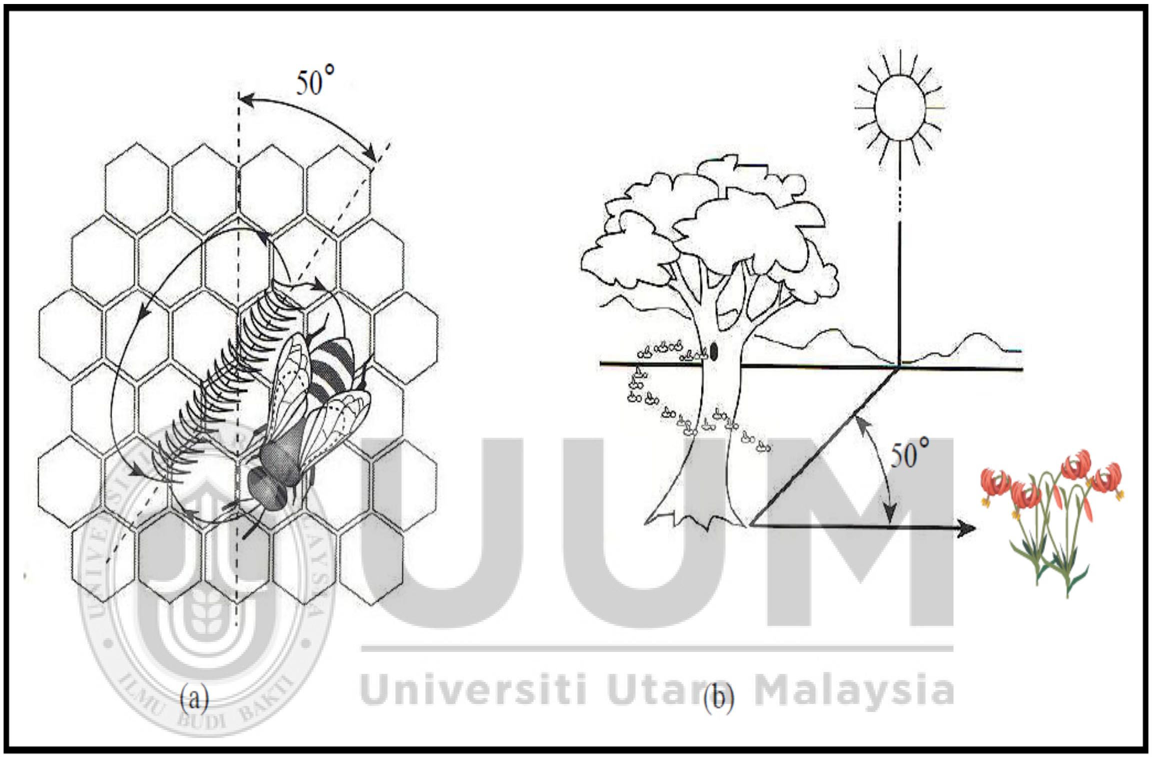




*Figure 2.11.* Illustration shows (a) waggle dance give (b) direction to the source of food by the sun as estimate location to the source of food to other bees (adapted from [2])

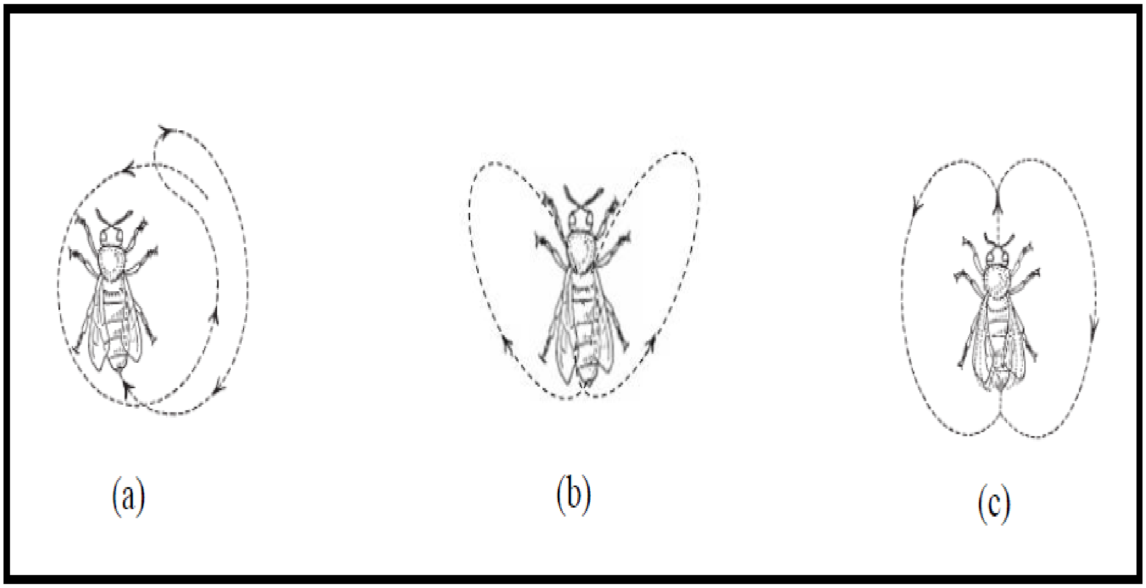


*Figure 2.12.* A bees shows her (a) round (b) sickle dance and (c) waggle dance (adapted from [2])



#### 2.5.4.1 Bio-Bees Inspired Searching Algorithm

According to Mitra Mahdiani et al. in 2014 [102], the foraging activities of bees prompted researchers to attempt an imitation of their activities to find the best possible solution. This requires a few adjustments in a computer and programming environment: firstly, a predefined number of bees were dispersed to the food source. Unlike the factors of natural bees (weather conditions, humidity and the total number of bees in the hive), the total number of bees in computer programming depends on the nature of the problem. This initialization process of the bio-bees searching algorithm also occurs in many optimisation algorithms including GA and PSO [2].

Secondly, in terms of computer programming, bees are known as a swarm agent and the food is known as a problem function. The swarm agent goes to every potential site (solution in programming) and returns with the solution that best meets the criteria of the problem function. The initialization solution is randomly generated to make it more robust and cover most of the possible solutions [2].

The swarm-based algorithm is relevant in problem solving in the sense that swarm involves a multiple swarm agent in which each of the agents collectively forms a distributed system. Here, many agents are used for providing solutions of the same problem out of which only the best solution had been chosen. These agents uniquely 'communicate' with each other in order to identify the best solution of the problem. An intelligent swarm-based algorithm is 'anti-classical' intelligence that works differently from any normal artificial intelligence algorithm. Swarm-based intelligence works collectively towards global information so that each agent can make their own judgment based on this information. Swarm-based intelligence is more flexible, decentralized, and self-organized as compared to older methods [2][102].

#### 2.5.4.2 Bees Algorithm in Action



A group of researchers first developed the Bees Algorithm at the Manufacturing Engineering Centre, Cardiff University [104][5][40][105][106]. The algorithm mimics the food foraging behaviour of swarms of honey bees. In its basic version, the algorithm performs a kind of neighbourhood search combined with random search and can be used for both combinatorial optimisation and functional optimisation" [95][107][108].

The BA has been applied in many numerical optimisation problems. Numerical optimisation deals with real data that requires numerical answer, such as parameter optimisation for learning algorithms [5], homogenous data formation for clustering problems [104], and preliminary design problems on the generation of multiple feasible solutions [109].

The BA can be broadly divided into four main phases:

- Parameter initialisation
- Fitness evaluation
- Organisation
- Neighbourhood deployment

These phases are shown in details in Figure 2.14 and Figure 2.13 below with illustrations of the pseudo code and flowchart. Table 2.2 shows parameter initialization. In this phase, the following parameters are set [2][110][111]:

a) Total number of scout bees (*n*)

b) Total number of sites selected for neighbourhood search (*m*)

c) Total number of the best "elite" sites out of m selected sites (*e*)

d) Total number of bees recruited for the best e sites (*nep*)

e) Total number of bees recruited for the other (*m*-*e*) selected sites (*nsp*)

f) A stopping criterion



During the phase of fitness evaluation of the bees, fitness of each bee is calculated. Based on their fitness, the organisation phase organises each bee by sorting in an ascending order.

Next, the neighbourhood deployment phase chooses the bees with the highest fitness and labels them as "selected bees" and selected sites. For neighbourhood search, each selected bee is later assigned a total number of bees.

Table 2.2

*Parameter description of BA (adopted from* [2]*).*

| No | Parameters | Descriptions |
|---|---|---|
| 1. | n | Number of scout bees |
| 2. | m | Number of sites selected for neighbourhood search |
| 3. | e | Number of best "*elite*" sites out of m selected sites |
| 4. | nep | Number of bees recruited for best e sites |
| 5. | nsp | Number of bees recruited for the other (*m*-*e*) selected sites |
| 6. | R | Maximum iterations |
| 7. | n1 | Number of bees around selected locations |
| 8. | n2 | Number of bees around each "elite" locations |
| 9. | dim | Dimension of search space |
| 10. | start_x | Minimum range of search space |
| 11. | end_x | Maximum range of search space |
| 12. | ngh | Neighbourhood search domain |
| 13. | bpos | Bees position |
| 14. | bNghPos | Bees position in neighbourhood search domain |



At step 1, the algorithm takes an initial population of n number of scout bees where each bee represents a set of a function problem. Step 2, evaluates the fitness of the sites visited by the scout bees. Step 3 labels bees with the highest fitness as "selected bees" and selects their sites for neighbourhood search. Steps 4 and 5 conduct a neighbourhood search of the selected sites where more bees are assigned to search the best e sites. The selection of bees happens directly according to the fitness of the sites visited by the bees. Alternatively, these fitness values determine the selection of the bees. In case of the best e sites with the most promising solutions, more bees are recruited in the neighbourhood search. Step 6 selects the bee with the highest fitness for each site in order to form the next bee population.

```
Initialise parameters.
Step 1. Initialise population.
Step 2. Evaluate fitness of the population.
Do
      Step 3. Select elite bees and neighbourhood search.
      Step 4. Select other sites for neighbourhood search.
      Step 5. Recruit bees around selected sites and evaluate
              fitness.
      Step 6. Select fittest bee's site from each site.
      Step 7. Assign remaining bees to search randomly and evaluate
              their fitness.
While stopping criterion not met.
```

*Figure 2.13.* A typical simplified pseudo-code of the BA (Reproduced from [2])

Step 7 randomly assigns the remaining bees in the population to search for new potential solutions so that at the end of each iteration, the colony has: a) representatives from each selected site and b) scout bees assigned to conduct random searches.



Steps 3 to 7 are repeated until the best fitness value has been stabilized or the iteration has reached its maximum number. The BA stops as soon as the stopping criterion is met, which means, it exceeds either the total number of iterations or the target total fitness set earlier. Most applications use the total number of iterations as their stopping criteria as it is easier than the target total fitness to determine. Moreover, determining the target total fitness requires the user a clear understanding of the relationship between fitness and the problem.

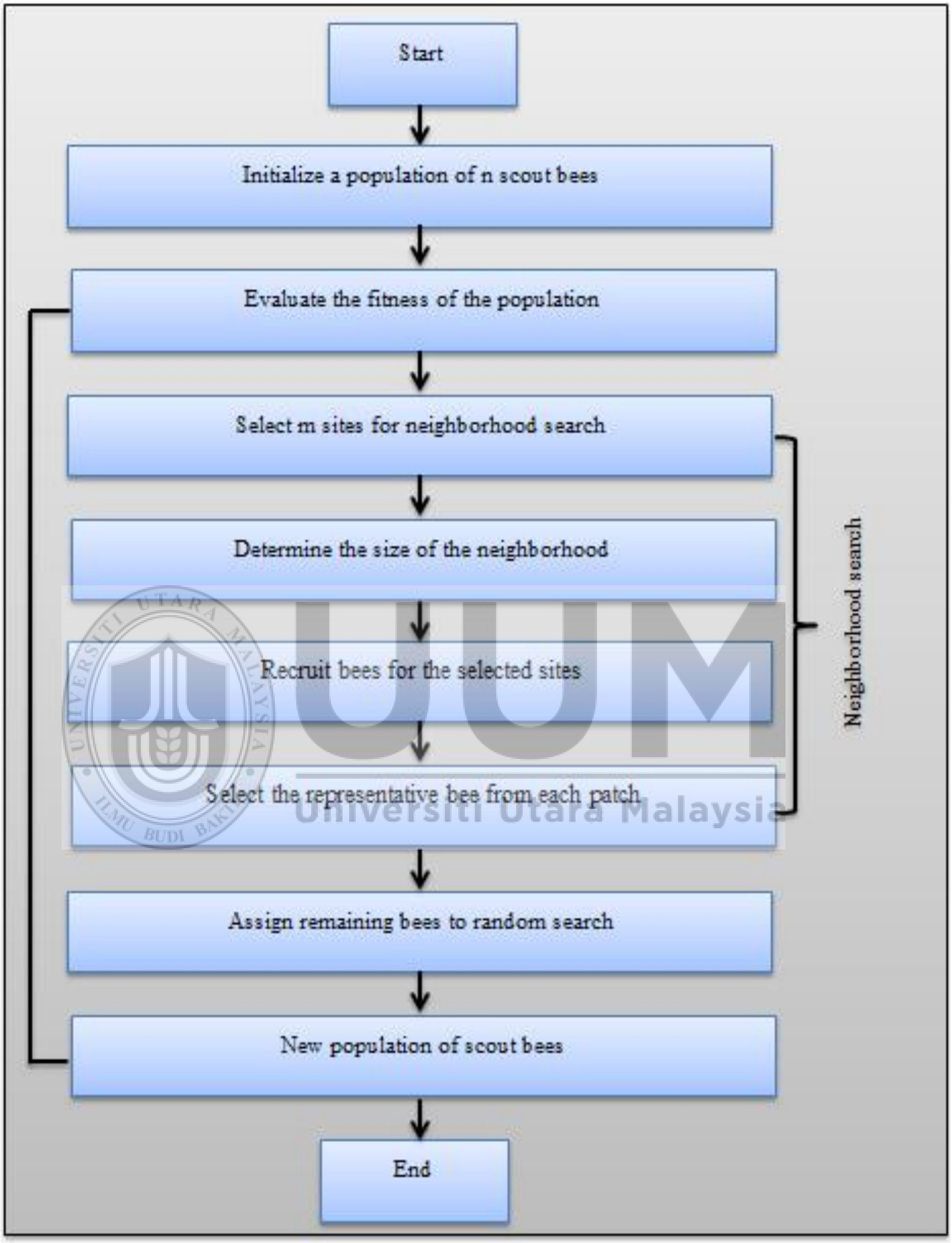


*Figure 2.14.* The flowchart of BA (Reproduced from [108])

### 2.5.4.3 Bees-based Approaches

The foraging activities of bees have attracted a great deal of attention of researchers [2][18][23][28]. These algorithms imitate various styles, but mostly the foraging activities of real bees are imitated [49]. The bee-based algorithm applications include many variations such as Fuzzy Bee System (FBS), Bee System, Bee Colony Optimisation (BCO), Beehive, Bees Swarm Optimisation (BSO), Virtual Bee Algorithm (VBA), Artificial Bee Colony Algorithm (ABC), Bumblebees, and Bees Algorithm (BA).

Wedde [112] proposed Beehive which is a system that imitates the filtering and sharing of communication among bees in a computer system with no centralised information database. Teodorović and Dell'orco offered BCO [95] that imitates natural bee behaviour for the ride-matching problem in transportation engineering. Wong implemented Bee System [88] to solve travelling salesman problem (TSP). Yang developed VBA [49] to determine the intensity of communication among the bees for engineering application problems. Drias proposed BSO [2], [49] based on the behaviour of real bees especially in nectar harvesting from the easiest accessible source while favouring the richest. It deals with operation research and economics of network analysis.

The bio bee-based algorithm has been applied to various types of optimisation problem, including routing, resource allocation, and computer communication, and the basic function optimisation as mentioned above and listed in Table 2.3.

Table 2.3
*The Description for many of bio-inspired based BA adapted (adopted from [2])*

| NO | Algorithm/ Author/ Year | Description | Advantages | Disadvantages | Type of Problem |
|---|---|---|---|---|---|
| **1.** | Fuzzy Bee System (FBS), Sato and Hagiwara (1997) | In this respect Fuzzy Bee System (FBS) was introduced where the agents use approximate reasoning and rules of fuzzy logic in their communication and acting. In the case of FBS the agents (artificial Bees) use approximate reasoning and rules of fuzzy logic in their communication and acting. | It is possible to calculate solution component attractiveness even if some of the input data were only approximately known. | Algorithm have a few weakness like initial partition selection and local optima convergence. | Ride-Matching problem. |
| **2.** | Honeybees Mating Optimization (HBMO) Abbass, (2001) | To simulate the social behaviour found among honeybees. In the original HBMO, a drone mates with a queen probabilistically through the use of an annealing function. In general, a honeybee community consists of three types of members: the queen, ale honeybees (or drones), and neuter/undeveloped female honeybees (or workers). | Feasibility of finding global optimum for several problems, Availability to combine the hybrid algorithms with EA and others, Implementation with several optimization problems, Availability for real and binary problems. | Stability and convergence of algorithm is based on recombination and mutation rates, Algorithm has a weakness on local search, It has a difficult encoding scheme. | Vehicle routing problem. financial classification problems |

| | | | | | |
|---|---|---|---|---|---|
| **3.** | Bee system Lucic and Teodorovic (2001) | It is aimed to explore the possible applications of collective bee intelligence in solving complex traffic and transportation engineering problems.<br>Bee System is an improved version of the Genetic Algorithm (GA). The main purpose of the algorithm is to improve local search while keeping the global search ability of GA. | These techniques are efficient and flexible.<br>The intakes are located.<br>Lowest cost for active lake quality improvements alternatives by use of solar energy. | Requires 20 mixers to achieve needed aeration.<br>Cause significant interference to boat traffic. | Travel salesman problem (TSP). |
| **4.** | Bee colony optimization (BCO) Teodorovi´c and Dell'Orco (2002) | The BCO represents the new Meta heuristic capable to solve difficult combinatorial optimization problems. | BCO technique avoids locally optimal solution. It searches for the best solution obtains by the entire bee colony. It is adaptive to changes in the environment. | In BCO, It is difficult to examine it hypothetically and its probability changes from iteration to iteration. | Max-Routing.<br>Wavelength assignment. |
| **5.** | BeeHive Wedde et al (2004) | Presented a novel routing algorithm, called BeeHive, which was the inspiration gained from communicative and evaluative methods and procedures of honeyBees. | The most important advantage of BeeHive is the distribution of the traffic to different routes proportional to their quality and capacity. This is done by a very simple mechanism, without wasting much computing time and energy. | A higher memory use for storing every forager. Although they are really small it is more than storing every route only once. | Telecommunication network routing.<br>Job-Shop scheduling. |

| | | | | | |
|---|---|---|---|---|---|
| **6.** | Bees Algorithm (BA) Pham et al (2005) | Bees Algorithm is one of the optimization algorithms inspired from the natural foraging ways of the honey bees of finding the best solution. It is a series of activities based on the searching algorithm in order to access the best solutions. It is an iteration algorithm. | Easy to use and to implementation. It is very efficient in finding optimal solutions. Good ability to deal with several optimization problems. | It is suffering from slow convergence. Lack of accuracy. Time-consuming. The other downside of the Bee Algorithm is that it has needless computation. | Controller design optimization. Data clustering. File search optimization. Filter design optimization. Manufacturing system optimization. Mechanical design optimization. Scheduling optimization. |
| **7.** | Bees swarm optimization (BSO) Drias et al (2005) | It is inspired from the behaviour of real Bees especially harvesting the nectar of the easiest sources of access while always privileging the richest. | Respond time where it shows its superiority on the exact algorithm. Very flexible and not complex. | Never guarantee the optimal solutions. Slow computation response. | Operational research. Economic network analysis. |
| **8.** | Virtual bee algorithm (VBA) Yang (2005) | The Virtual Bees Algorithm (BA) is an optimization algorithm inspired by the natural foraging behaviour of honey Bees to find the optimal solution. Natural | A virtual (VBA) which is effective on function optimization problem. It is simple than other nature | It has computational cost and some of the output and input need to be specified. It has lack of physical meaning. | Engineering problems. |

| | | | | | |
|---|---|---|---|---|---|
| | | inspired algorithms have some similarity with genetic algorithms, but it has multi agents that work independently and thus it is much more efficient than the genetic algorithms due to the parallelism of the multiple independent Bees. | inspired Algorithms due to less parameter setting.<br>It is more power and effectiveness in many application. | | |
| **9.** | Artificial Bee Colony Algorithm (ABC)<br>Karaboga (2005) and Basturk (2006) | The basic idea of designing ABC is to mimic the foraging behaviour (such as exploration, exploitation, recruitment and abandonment) of honeybees.<br>ABC algorithm has widely used to solve other problem such as multi-dimensional numeric problems, real-parameter optimization, constrained numerical optimization problems. | Easy to implement.<br>High flexibility, which allows adjustments and the introduction of specific knowledge of the problem by observing nature.<br>Robust against initialization, regardless of feasibility and distribution of the initial solutions population.<br>The structure of the algorithm is favourable for parallel processing, thus saving time.<br>Ability to explore local solutions.<br>Global optimizer, with effective search process even under<br>High complexity, and with low risk of premature convergence. | Lack of use of secondary information about the problem (gradients).<br>High number of objective function evaluations.<br>Slow down when used in sequential processing.<br>Slow to obtain accurate solutions.<br>The population of solutions increases the computational cost due to slowdown, many iterations and memory capacity required.<br>Deterministic methods have higher accuracy in finding solutions when it does not get stuck in a local minimum. | Continuous Optimization. |

| | | | | | |
|---|---|---|---|---|---|
| **10.** | BumbleBees, Comellas and Martinez (2009) | Is a member of the bee genus Bombus, in the family Apidae. Our algorithm associates possible solutions to the problem considered with individual bumbleBees living in an artificial world which evolves following a set of simplified rules based on the behavior of a real bumblebee colony. Like in some other optimization algorithms, a fitness or cost function measures the quality of the solution, but in our algorithm this fitness is also tied to the lifespan of the evolving individual. | Is characterized by the low cost. It is easy to adapt it to new tasks. It is possible to establish different gradients of different environmental parameters. | Its disadvantage is that there is no such technique that guarantees that we can get out of the environment of a local minima. If small surroundings were chosen, they would be never and a better solution in the next iteration because there still a stick in the area of a local minima. | Vehicle routing problem. |
| **11.** | Honeybee social foraging (HBSF) Quijano and Passino (2010) | Modeling social foraging for nectar involves representing the environment, activities during bee expeditions (exploration and foraging), unloading nectar, dance strength decisions, explorer allocation, recruitment on the dance floor, and accounting for interactions with other hive functions. | It is able to achieve an ideal free distribution situation which could maximize the uniform temperature allocation. | It is suffering from slow convergence. | Resource Allocation. |
| **12.** | OptBees Algorithm | It is based on the processes of collective decision-making by bee colonies to solve | Algorithm is capable of generating and maintaining the diversity and | It is suffering from slow convergence. | Continuous Optimization |

| | | | | | |
|---|---|---|---|---|---|
| | Maia et al. (2012) | multimodal continuous optimization problem. | consequently obtaining multiple local optima solutions without losing the ability of global optimization. OptBees is also a competitive tool in the context where the goal is just to obtain the best possible solution, without being necessary to determine locally optimal solutions and/or multiple global optimal solutions. | It has a difficult encoding scheme. | Problems |
| **13.** | Bees life algorithm (BLA) Bitam and Mellouk (2013) | Two bees' behaviours are employed in the proposed algorithm, i.e., reproduction and foraging behaviours. In addition, the former incorporated the crossover and mutation operators, while the latter used a neighbourhood search approach. | Algorithm outperforms the others in terms of the solution quality and complexity when it Compared with other CI algorithms (such as GA, BeA, and HBMO) | Still need more Improvement to overcome it is limitation on optimal way of scheduling for users and using a reproduction and food foraging behavior in bees life algorithm. | Job Scheduling in Cloud Computing. Vehicular ad hoc network (VANET) problem. In particular, the quality of service multicast routing problem (QoS-MRP). |

## 2.6 Local Search Algorithm

Local search is a metaheuristic method for solving computationally hard optimization problems. Local search can be used on problems that can be formulated as finding a solution maximizing a criterion among a number of candidate solutions [2]. According to [113] "local search starts with some feasible solution of the problem and tries to progressively improve it. Each step of the procedure carries out a movement from one solution to another one with a better value. The method terminates when, for a solution, there is no other accessible solution that improves it". Local search algorithms move from solution to solution in the space of candidate solutions by applying local changes as simple search, 2-opt, 3-opt Lin–Kernighan heuristic, WalkSAT and so forth, this process is repeated until a solution deemed optimal is found or a time bound is elapsed [2].

Local search algorithms are widely applied to numerous hard computational problems, including problems from computer science (particularly artificial intelligence), mathematics, operations research, engineering, and bioinformatics. According to [114][115], [116] most problems can be formulated in terms of search space and target in several different manners. Some problems where local search has been applied are:

- The travelling salesman problem, a solution can be a cycle and the criterion to maximize is a combination of the number of nodes and the length of the cycle. But a solution can also be a path, and being a cycle is part of the target.
- The vertex cover problem, in which a solution is a vertex cover of a graph, and the target is to find a solution with a minimal number of nodes.

- The nurse scheduling problem where a solution is an assignment of nurses to shifts which satisfies all established constraints.

- The Boolean satisfiability problem, in which a candidate solution is a truth assignment, and the target is to maximize the number of clauses satisfied by the assignment; in this case, the final solution is of use only if it satisfies all clauses.

- The k-medoid clustering problem and other related facility location problems for which local search offers the best known approximation ratios from a worst-case perspective.

This study addresses the local search for bees algorithm feature selection for numerical dataset, the original Bees Algorithm was initially designed for numerical data. Recently, the Bees Algorithm Density developed for works in the searching process of the combinatorial problem but still need more enhancement for better result. The main focus is to design a new or extend local search technique mechanism based on the Bees Algorithm architecture. This is an important process for any optimisation algorithm. Determining the best local search method enables an overall solution of a problem. For instance the GA uses some local search operators including crossover, mutation, or standard selection. FS is known as an instance of the combinatorial problem. The local search in the Bees Algorithm is therefore adapted from the simple swap, 2-opt and 3-opt local search approach used for the complex problems [2].

### 2.6.1 Simple Swap Search Approach

This process is implemented by employing a simple swap of features to be evaluated with swap values at two random points, $L_1$ and $L_2$ where $L_2 > L_1$. The value of the feature at the index of $L_1$ $F_{L1}$ was swapped with the value of the feature at the index of $L_2$ $F_{L2}$. This idea was proposed by [2]. Figure 2.15 shows an example of the simple swap operation. Figure 2.16 shows the pseudo code of neighbourhood procedure of FS in the modified BA by simple swap and mutation.

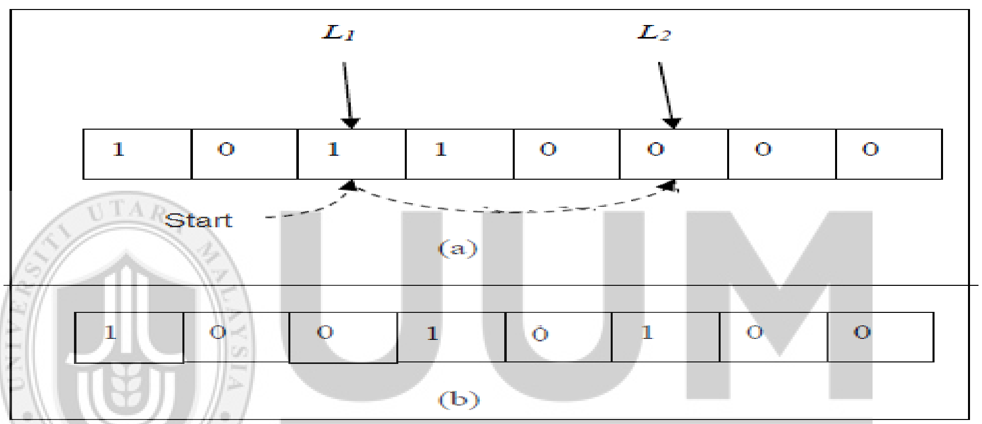




*Figure 2.15.* A 2-Points-Swap move: (a) original binary feature and (b) feature after resulting move

```
For (itr=1 to maxRun){
    If (totalFeat > 2^totalFeat -2 ) stop process
    For(i=1 to m) {
       If (totalFeat > 2^totalFeat -2) stop process
       For (j=1 to recuited bee) {
          If (totalFeat > 2^totalFeat -2) stop process
          Do simple swap(F i )
          Do SimpleMutation(F i )
              If feat new is not exit, save in tabu list
              Evaluate and calculate feature, F new
If fitness F new <= Ft i save feature, new F
    }
  }
}
```

*Figure 2.16.* Pseudo code of neighbourhood procedure of FS in modified BA by Simple swap and Mutation

### 2.6.2 2-Opt-Based Search Approach

It is a simple local search algorithm first proposed by Crows in 1958 for solving the traveling salesman problem. The 2-Opt operator is the simplest and easiest of all operators in the k-opt family to implement for solving a problem [117]. The main objective of its implementation is to make small changes on the tour and check if the solution quality improves [118]. Although the basic move had already been suggested by [119], this move deletes two edges, thus breaking the tour into two paths, and then reconnects those paths in the other possible way [120].

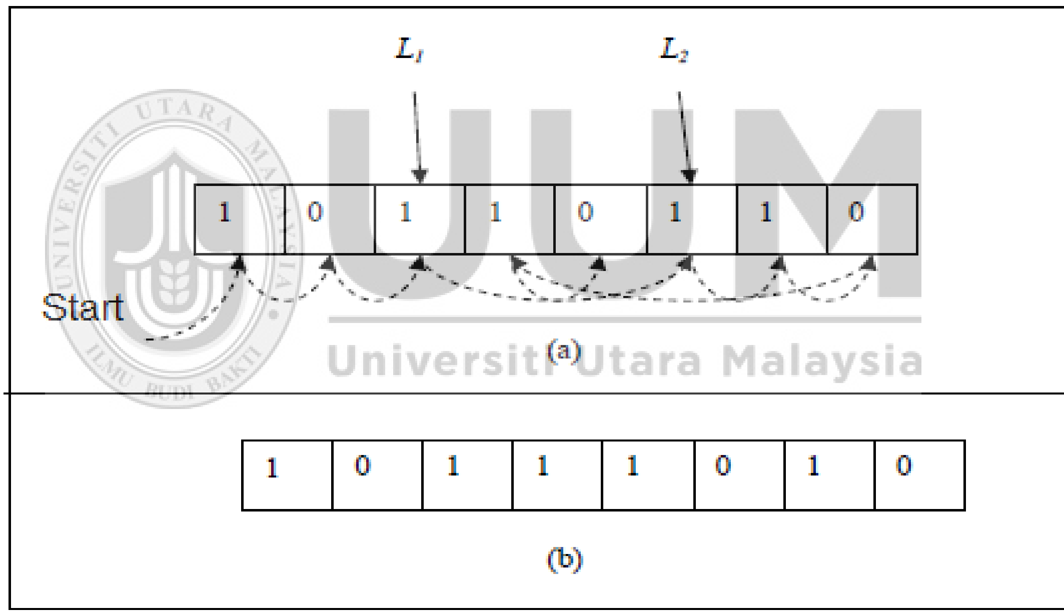


*Figure 2.17. A 2-Points-Swap move: (a) original binary feature and (b) feature after resulting move*

The 2-Opt-based local search approach is also used in the old proposed algorithm [2]. In this study, the search starts by initialising two points of swapping, $L_1$ and $L_2$ where $L_2 > L_1$. The process is implemented by swapping the feature values at indices $L_1$ ($F_{L1}$) and L$_2$ ($F_{L2}$). The swapping process continues with ($F_{L1+1=} F_{L2-1}$) until $L_1$ and $L_2$ have the same value. Figure 2.17, represents a simple example of this 2-Opt-based operation. Figure 2.18, illustrates the pseudo code of neighbourhood procedure of FS in the modified BA by 2-Opt-based approach and mutation.

```
For (itr=1 to maxRun){
    If (totalFeat > 2^totalFeat -2 ) stop process
    For(i=1 to m) {
        If (totalFeat > 2^totalFeat -2) stop process
        For (j=1 to recuited bee) {
            If (totalFeat > 2^totalFeat -2) stop process
            Do 2-points-swap(F i )
            Do SimpleMutation(F i )
                If feat new is not exit, save in tabu list
                Evaluate and calculate feature, F new
If fitness F new <= Fti save feature, new F
        }
    }
}
```

*Figure 2.18. Pseudo code of neighborhood procedure of FS in modified BA by Two Points Swap Approach and Mutation ([2])*

### 2.6.3 3-Opt-Based Search Approach

3-Opt-based search approach was proposed by [121] and [122]. It is an algorithm that creates three tour segments by removing three edges from the tour. This allows the addition of a new element to the method, thus reconnecting the tour segments in different ways [117] to locate the best possible way. This makes the 3-Opt-based search relatively slower than the 2-Opt-based search, but it creates tours with higher quality than 2-Opt [123]. Therefore, in this proposed algorithm, 3-Opt-based search approach was also applied as an old idea used by [2]. This search approach involves random generation of reference points at $L_1$, $L_2$ and $L_3$ where $L_1 > L_2 > L_3$. Feature values at index $L_1$ and $L_2$ ($F_{L1}$ and $F_{L2}$ respectively) use a 2-Opt-based operation. Features between $fea_{L2+1}$ and $fea_{L3}$ were moved into features between $F_{L1}$ and $F_{L1+(L3-L2)}$. Figure 2.19, gives a simple example of the 3-Opt-based operation.

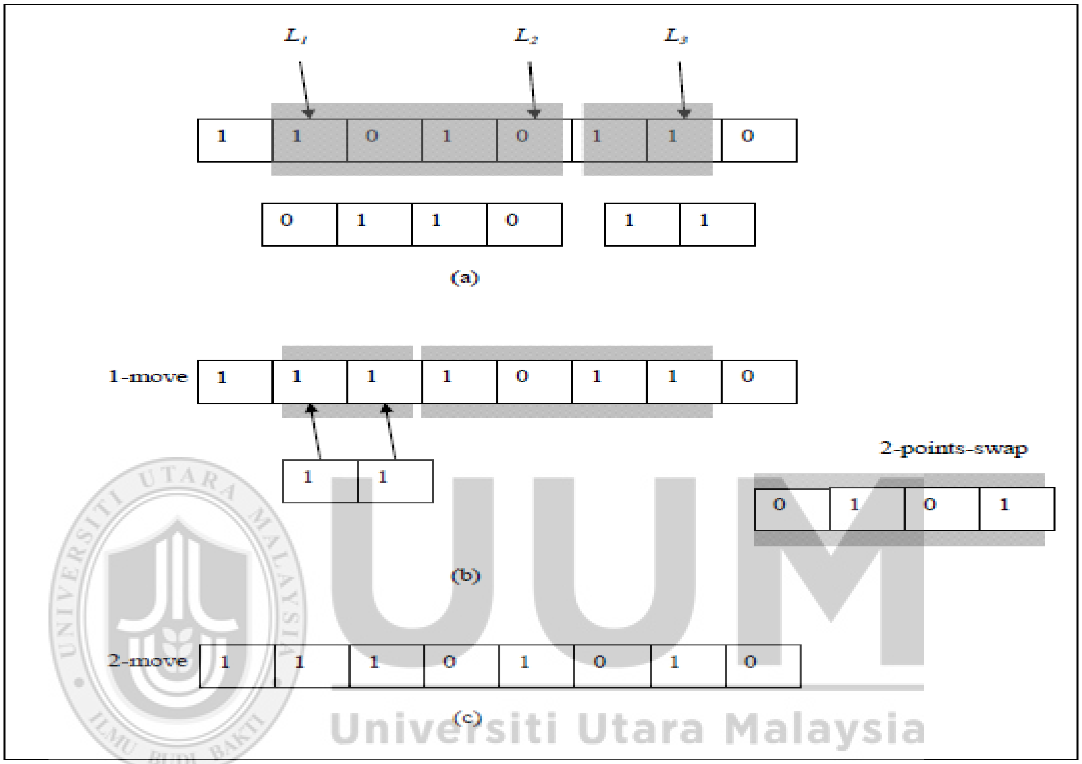


*Figure 2.19.* A 3-Points-Swap move: (a) original binary feature, (b) 2-points-swap and (b) final feature after resulting move

The proposed algorithm also uses a modified 2-Opt-liked moving operation. 2-Opt involves two index points that are generated randomly. The movement starts with the first feature, $\boldsymbol{fea_1}$ and continues with the next feature until the first index point, $\boldsymbol{L_1}$ is found at feature $\boldsymbol{F_{L1}}$. The next move starts at second index, $\boldsymbol{L_2}$ at feature $\boldsymbol{F_{L2}}$ where the next move feature is read. The feature movement process of 2-Opt continues until the last feature, $\boldsymbol{F_{total}}$. The next feature to be read is next to index $\boldsymbol{L_1}$, at $\boldsymbol{fea_{L1+1}}$. These moves stop at feature, $\boldsymbol{F_{L2-1}}$ by which time all features have been covered. Figure 2.20

illustrates the pseudo code of neighbourhood procedure of FS in the modified BA by 3-Opt-based operation and mutation.

```
For (itr=1 to maxRun){
  If (totalFeat > 2totalFeat  -2 ) stop process
  For(i=1 to m) {
     If (totalFeat > 2totalFeat  -2) stop process
     For (j=1 to recuited bee) {
       If (totalFeat > 2totalFeat  -2) stop process
       Do 3-points-swap(F i )
         Do SimpleMutation(F i )
         If feat new is not exit, save in tabu list
         Evaluate and calculate feature, F new
If fitness F new <= Fti save feature, new F
    }
  }
}
```

*Figure 2.20.* Pseudo code of neighborhood procedure of FS in modified BA by Simple swap and Mutation ([2])

## 2.7 Multilayer Perceptron Model

Kordos [124] defines the artificial neural network as a general mathematical computing paradigm that models the operations of biological neural systems. Research on artificial neural networks was originated in 1943 by McCulloch and Pitts, who proposed the first mathematical model of a neuron. In 1958, Rosenblatt introduced the first neural network known as perceptron. All neural network models that have been proposed over the years share a common building block known as a neuron and a networked interconnection structure. Neural networks are preferred over conventional programming for their ability to solve problems which have no algorithmic solution or a complicated existing solution. MLP is the most widely used neural network that is based on several sequentially connected layers of perceptron.

Previous studies infer that a single layer perceptron can classify only linearly separable data. For instance, Minsky and Papert in their famous book "Perceptrons" (1969) observed that the single layer perceptron cannot solve the Xor problem. The book contributed to the stagnation in research on neural networks for certain time. MLP was perceived as a technique that would solve linear non-separable problems, but efficient algorithms for the training of MLPs were not known at that time. The first successful algorithm, known as back propagation, was developed several years later signalling the rapid development of the field of neural networks [124][125].

MLP is a network that consists of usually two or three layers of neurons and an additional input layer. Some researchers consider the input layer as a separate network layer. The current study includes a three-layer network, as shown in Figure 2.21. It includes a network of two layers of neurons that is based on McCulloch and Pitts' model, and one additional input layer of neurons which distributes the input signals.

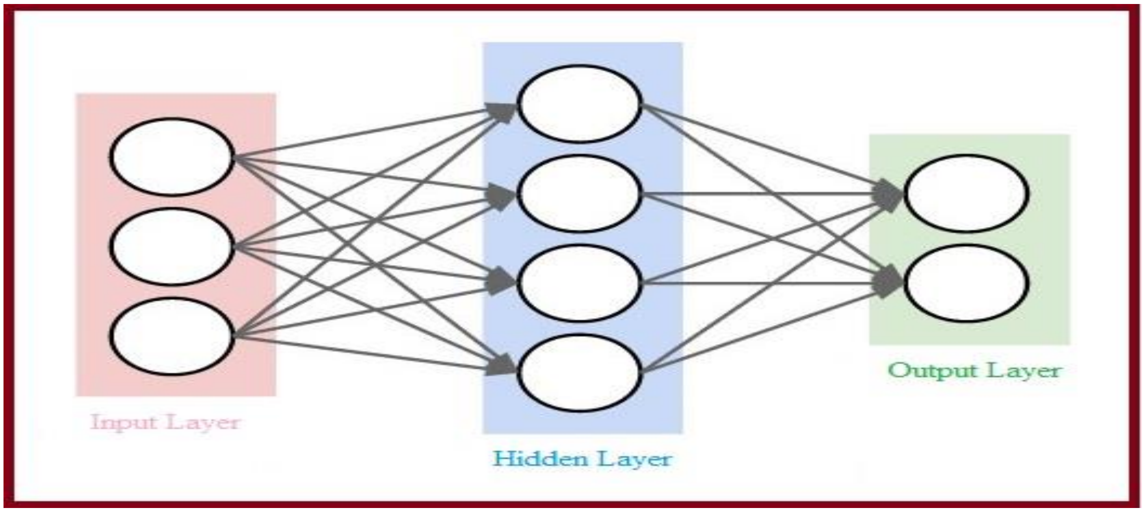


*Figure 2.21.* Three-layer, fully connected MLP network. Vertical arrows symbolize biases (Reproduced from [124])



In practical implementations, there is one input layer and one output layer. The number of hidden layers can be zero, one or two. During the training process, the weights of the output layer along with all hidden layers are optimised. Connecting two successive layers are optional here. In addition, irrelevant weights can be removed during or after the network learning process. When every node in a given layer is connected to every node in the following layer, it means that the MLP network is fully connected.

MLP networks are the most widely used neural networks that are applied to varied fields including medical diagnosis and image recognition, time series prediction, data compression, defect detection in materials, bankruptcy prediction, music classification, solar collectors sensitivity analysis, handwriting recognition, viruses and Internet worm detection, and so on [124], [125]. Known as one of the most popular learning techniques, MLP is used mainly for supervised learning [50].

### 2.7.1 Applications of MLP in Feature Selection (FS)

Features in data are usually stored in a vector matrix of attributes. The correctness of the features subset in this matrix is assessed by the Wrapper FS method using a learning algorithm (classifier) for numerical dataset. The MLP networks in FS can be observed in the following approaches: Embedded FS [126]–[128][129]–[131], Wrapper FS [2][18][132]–[134], and Filter FS [135], [136]. Applications of MLP have been widely used by researchers. Liu applied Embedded FS in a three-layer feedforward neural network with penalty [137]. [129] used a modified MLP network for eliminating irrelevant features. [138] used the Wrapper neural network with the weight analysis-based ANNIGMA (artificial neural net input gain measurement approximation) in their work. Sudhwani et al. used the Wrapper approach, and compared MLP and SVM utilities as a classifier with maximum output information [139].

[140] and [141] compared three approaches – Filter, Wrapper, and Embedded – to find out that the Wrapper approach performed better in estimating accuracy. Wrapper FS classifier-based techniques are also effective in improving the performance of the FS system with minimal effort [142]. The current study uses a neural network based on Wrapper feature selection approach.

Finding the best of N features in an exhaustive method computationally requires at least **O ($2^N$-2)** time. Although there are quicker versions of the exhaustive approach such as branch and bound [143], it is still an impractical solution with larger features sets [144]. This is why researchers prefer the Wrapper approach which integrates a learning algorithm with an optimisation tool. Optimisation tools can be classified in two different searching solution techniques: single and population-based. Single optimisation uses a possible solution throughout the process whereas population-based optimisation uses multiple possible solutions approach that chooses only one best solution as the final answer. The criterion for the best solution selection is based on an evaluation function $\boldsymbol{f}$. Applications of the population-based optimisation can be found in genetic-evolutionary algorithm and swarm-based approaches including GA, ACO, PSO, and BA. GA is one of the most popular tools used in various Wrapper FS approaches. [145] used GA with AQ15 (a rule induction classifier) to select a significant number of features. Feng [146] used GA and SVM to reduce features for the recognition problem in Chinese handwriting. [86] studied the possibilities of FS by providing a basic taxonomy of FS techniques and discussing their use, varieties, and potential in a number of both common as well as upcoming bioinformatics applications. A number of bioinformatics problems, such as skin tumour recognition [147], also used population-based optimisation with a chosen classifier to get better solutions. The DNA microarray problem [148] used an integrated approach using a GA and a neural network to produce better accuracy.



Wrapper FS methods using a genetic-based technique are combined with a classifier in order to randomise the features searching techniques. This approach has been used in many GA methods [149][150][74], [151], [152] and genetic programming [153], [154][155], [156]. PSO has also been used in FS problems by researchers [157]–[160]. Wang [161] combined PSO and rough sets as a selection strategy for reducing data dimensions. Chuang [71] used binary PSO and k-NN to evaluate gene datasets in the gene classification problem. Researchers prefer to use PSO because it requires simple mathematical operators [161][162] in contrast to the complex mathematical operators required in GA methods.

In terms of finding the best relevant features, ACO has attracted the attention of the machine learning fraternity. [163] used ACO as a weight selector and neural network as classifier. [164] used ACO by splitting the search process into two stages: initialisation and generation of the feature subset. Another study combined ACO and probabilistic neural network [165] to identify relevant features in bearing fault diagnosis problem.

### 2.7.2 MLP Applications in Supervised Classification

Applications of artificial neural networks are extensive and varied in different learning algorithms and pattern recognition. Development of neural networks was inspired by the biological human brain [166]. Artificial neural networks generated much enthusiasm in the 1980s and 1990s. Based on network flow, neural networks can be broadly categorised into two types: feedforward neural networks and recurrent neural networks. Feedforward neural networks have unidirectional flows from the input until the output, such as MLP, learning vector quantisation, and adaptive linear neuron [2].



MLP is a collection of multiple single-layer networks connected together, and it has been used successfully for many applications. Due to its limited capabilities, a single-layer network can only be applied for a linearly separable problem. Figure 2.21 shows a typical architecture of an MLP network. MLP is known as a universal approximator. Applications of MLP networks are implemented in supervised learning processes in order to update the weights of each connection in each network layer. This updating process is repeated until the stopping criterion is met, which is normally a total number epoch. The connection weight is adjusted by minimising the difference between the input and the desired output. During each epoch, the network requires mapping between the input and output of data to update weights.

The MLP network has an outstanding capability to extract patterns and interpret the meaning of data which is too complicated to recognise. The network relies on iterative learning from the initial experience of a given data pattern. The neural network is also capable of predicting and detecting the trends of a complex dataset which can be seen in many pattern recognition applications including the gesture recognition system and material science [2].

## Summary

This chapter in details was described swarm-based and population-based optimisation algorithms and in particular BAFS. It also highlighted the BA local search. This study provided information and background to the contents of subsequent chapters. Nevertheless, none of the survey especially those are related to the bees-inspired algorithm studied and discussed the search neighbourhood. For this reason, the contents of the subsequent chapters primarily focused on this issue.

# CHAPTER THREE
# METHODOLOGY

## 3.1 Research Methodology

The previous chapters present an overview of the works which have a direct relationship with the objectives of this research. Choosing an appropriate research methodology is an essential part in defining the steps to be taken to answer the research questions. The literatures give an extensive understanding of the background while demonstrating how to select a specific method to perform a specific activity [95]. The main objective for this research is to obtain speed up of FS and increase accuracy of larger dataset. Therefore, this study is considered as one of exploratory research. The study used the following research framework to achieve the objectives discussed in section 1.4.

## 3.2 Research Framework

The research framework of this study describes the methods that were adopted here: implementation tools and validation steps. Figure 3.1 shows the research framework used in this study. It includes seven main stages that were executed in this study. In addition, the seven phases in this methodology is adopted from research design [2] which are all linked with comprehension activities in accomplishing the objectives. **Phase 1** dealt with identifying the problems and determined their factors. **Phase 2** includes data collection through FS algorithm which was implemented and tested using most popular datasets from different fields. The same data was used in the Massudi work of BAFS [2]. **Phase 3** identified appropriate wrapper techniques to improve the speed and increase the accuracy of the BA. **Phase 4** developed and proposed an extension of swapping mechanism of BAFS. **Phase 5** evaluated the proposed swapping mechanism of BAFS in terms of speed and accuracy, by tested and compared the results with Massudi methods [2]. **Phase 6** analysed the result of proposed swapping mechanism of BAFS. **Phase 7** added the scope of this study for future works. The next section discussed the activities involved in each phase.

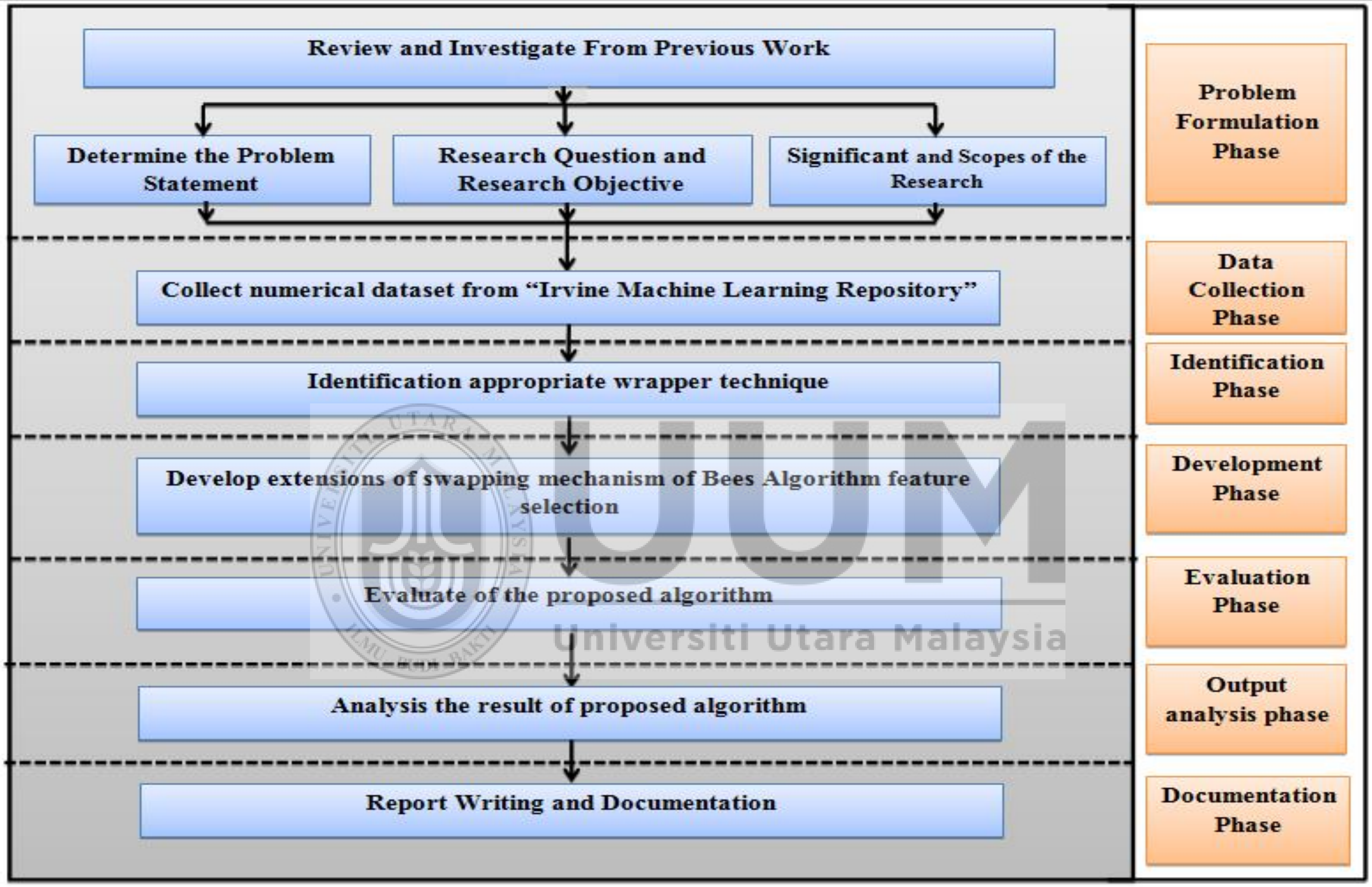


*Figure 3.1.* Proposed Research Framework (adopted from [2])

### 3.2.1 Problem Formulation Phase

This phase focuses on identifying the main problem while reviewing and investigating previous work in order to determine the problem statement, research questions, and research objectives of the study, and the significance and scope of the research.

The current study aims to identify and study the obstacles of BAFS which have already been explained in previous chapters in this study. It takes longer execution time during the process of FS due to an iteration of the BA module, especially in local search neighbourhood procedures [2], [23][5]. Bees spend most of the time in identifying the global optimal solution or choosing a good location to produce the best fitness [4][5].

Moreover, the BA involves a large number of computational processes to obtain a good solution, especially in complicated issues that consume a lot of time to access suitable results. Besides, there is a lack of accuracy [2], [4], [23] which does not guarantee to reach the optimum solutions or get better results for the problem.

These issues was be processed in order to get a new technique for enhancing the last results of BAFS that was done by [2] and obtaining better solution in terms of accuracy and speed [2], [23][167].

All these problems were addressed in this study with the aim to detect important features while removing irrelevant and redundant features. Local search method of BA was applied to reduce high-dimensionality of feature space from dataset, which led to increase accuracy and improve speed while attempting to produce optimal or near-optimal solution on the dataset.

### 3.2.2 Data Collection

The data to be used in this research comes from many sources one of which is the University of California at Irvine (UCI) repository [168]. The UCI repository contains more than 100 datasets used by many researchers [169]. The current study used feature selection algorithm to implement and test datasets from fields as varied as Iris, Pima Indians Diabetes, Motorola, Soybean, Spect Heart, Haberman, Hepatitis, Wavefrom, and Yeast Datasets which were used in the Massudi work of BAFS [2].

These datasets are based on several criteria and objectives. For instance, the total number of classes, instances, and features (variables) is diverse, as shown in Table 3.1.

Table 3.1
*Datasets from different fields of universal ([170])*

| No | Dataset name | Number of Feature | Number of Instance | Data type | Year | Cite |
|---|---|---|---|---|---|---|
| 1. | Pima Indians Diabetes Dataset | 8 | 768 | Integer, Real | 1990 | [171] |
| 2. | SPECT Heart Dataset | 22 | 267 | Continuous | 2001 | [172] |
| 3. | Hepatitis Dataset | 19 | 155 | Categorical, integer, real | 1988 | [173] |
| 4. | Soybean Dataset | 35 | 47 | Real | 1987 | [174] |
| 5. | Lung-cancer Dataset | 56 | 32 | Integer | 1992 | [175] |

1. **Pima Indians Diabetes Dataset**: This dataset contains several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. ADAP is an adaptive learning routine that generates and executes digital analogs of perceptron-like devices. It has 768 instances, and 8 features and it is one of the integer, real data type [2] [171].

2. **Soybean Dataset**: This is a small version of the soybean dataset contains 47 instances, 35 features and it is one of the real data type [2][170][176].

3. **SPECT Heart Dataset:** Another dataset from the University of California Irvine repository of Single Proton Emission Computed Tomography (SPECT) images contains 267 instances and 22 features to identify patient normal or abnormal heart. It is one of the continuous data type [2][170][177].

4. **Hepatitis Dataset**: About the hepatitis database and BILIRUBIN problem is continuous attribute (= the number of it is "values" in the ASDOHEPA.DAT file is negative); "values" are quoted because when speaking about the continuous attribute there is no such thing as all possible values. However, they represent so called "boundary" values; according to these "boundary" values the attribute can be discretized. At the same time, because of the continuous attribute, one can perform some other test since the continuous information is preserved. This is data that consists of a 19 continuous feature with distinctive two classes. A total number of example or instances are 155. It is one of the Categorical, integer and real data type [2] [173].

5. **Lung-cancer Dataset**: This data was used by Hong and Young [178] to illustrate the power of the optimal discriminant plane even in ill-posed settings. It is one of the Integer data type [2][170][178].

### 3.2.3 Identification Appropriateness of Wrapper Technique

Wrapper FS has been popularized by [73]. It differs from filter FS in terms of usage of the learning algorithm. Wrapper FS relies solely on maximising prediction accuracy as produced by the learning algorithm.

A learning algorithm with the optimisation that uses the Wrapper approach incorporates an optimisation tool and evaluates a model, whereas the filters approach are similar to wrappers in the search approach, but instead of evaluating against a model, a simpler filter is evaluated. In the other words, inductive algorithms are used by wrapper methods as the evaluation function whereas filter methods are independent of the inductive algorithm [16]. In the context of FS, the Filter approach is faster but less accurate and computationally intensive than the Wrapper approach [18][2][26]. The Wrapper approach is one of the most widely used approaches due to its adequate results and efficiency in handling larger and more complex dataset as compared to the Filter approach [27]. However, it is an expensive technique as it involves a complex process of building a classifier with hundreds of items to evaluate one feature subset and dispensing huge numbers of features [28][29].

Searching in feature space influences the performance of wrapper technique, especially its quickness to find the best subset features to avoid an exhaustive search. The wrapper FS approaches include three popular strategies: a) forward selection, b) backward elimination, and c) stochastic search. Forward selection evaluates from no features until all features have been considered. Backward elimination starts with all features. Stochastic approaches totally depend on the specific searching strategy of the particular algorithm. For instance, in genetic search that utilises GA approaches, each state is defined by a feature mask so that a genetic operation can be performed (such as crossover, and mutation). The current study chose "forward selection" to work with a BA in order to improve speed and increase accuracy.

### 3.2.4 Development of Bees Algorithm Local Search

Defined as one of the latest bio-inspired optimisation algorithms, the BA has been used to explain varied problems (such as, time consummation and accuracy). One possible solution to deal with these problems is to identify and eliminate all unnecessary repetition. In this phase, the extension of the swapping mechanism was developed for BAFS.



The main focus is to redesign local search technique mechanism based on the BA architecture. This is an important process for any optimisation algorithm. The best local search method is determined in order to provide an overall solution of a problem. GA uses some local search operators including crossover, mutation, or standard selection. FS is Bees Features Selected Fitness value which is known as an instance of the combinatorial problem.

The local search in the BA is therefore adapted from the simple swap, 2-Opt and 3-Opt local search approaches used for the TSP problem [179][180], selection features for manufacturing data [18], vehicle routing problem [108] and so on. To optimise the fitness possible solution, a suggestive 4-Opt as a proposed extension is used in the proposed algorithm. In this study, local search in the proposed algorithm is a combination of mutation and a swapping operation.

### 3.2.5 Parameters Setting for BA and MLP

MLP has been chosen in this work due to a number of reasons. MLP has been one of the most popular and widely used learning algorithms for the last decade in many machine learning applications as well as general application domains. MLP as a neural network is unique in its adaptive ability to the environment by adjusting the connection weight. Moreover, MLP is also known as a learning algorithm that has ability to estimate a desired precision. The parameter setting used for these experiments is given below. Table 3.2 shows the standard parameter values set for the experiments.

*Table 3.2*
*Parameter Setting for Multilayer Perceptron (MLP)*

| NO | Parameters | Values |
|---|---|---|
| **1.** | Number of Hidden Layer | 1 |
| **2.** | Desired Error | 0.001 |
| **3.** | Learning Momentum | 0.1 |
| **4.** | Learning Rate | 0.3 |
| **5.** | Number of Epoch | 500 |
| **6.** | Cross-Validation | 10 |

The BA requires a number of parameters and properties to be initialised, as shown in Table 2.2. Details of these parameters can be seen in section 2.5.4.2. These parameters and properties require customisation in order to ensure that the proposed algorithm can run properly and produce reliable results. The current study does not include all of the BA standard properties in the combinatorial FS problem. The unused standard properties of his proposed algorithm are shown in Table 3.3 below:

Table 3.3
*Unused standard properties of BA ([2])*

| NO | Parameters | Description |
|---|---|---|
| **1.** | *Dim* | Total dimension of space is not needed since the total space of limited to $2^N$-2 possible solution. |
| **2.** | *start_x* | Minimum ranges of feature start when none is selected. This situation cannot be represented in continuous ways. |
| **3.** | *end_x* | Maximum range of feature end when all full features are selected. |
| **4.** | *Ngh* | Neighborhood search domain is not applicable in the combinatorial problem. |
| **5.** | *bPos* | There is no need for bees to know the location of the position. |
| **6.** | *bNghPos* | Since bPos is not be used, this property is also not essential. |



Notice that all the properties mentioned above are only suitable for a continuous optimisation problem. The nature of the combinatorial problem is the main reason why such properties are not required [2]. In contrast, in this study the BA require a number of parameters to be set manually. Table 3.4 shows all the parameters used in the experiments.

*Table 3.4*
*Parameters setting for the BA*

| NO | Parameters | Values |
|---|---|---|
| **1.** | *n* | 25 |
| **2.** | *m* | 5 |
| **3.** | *e* | 2 |
| **4.** | *nep* | 15 |
| **5.** | *nsp* | 20 |



Stopping criteria for both used and unused standard properties are set differently in order to ensure that the overall proposed approach works well and delivers the correct solutions. To achieve the primary concerns of minimising the computational time and increasing the accuracy, the number of iterations in the training process of the neural network is set at only 500 epoch. Although this was resulted in an 'undertrained' condition, the main purpose of using a neural network in this study is to acquire the total mean square of the selected features. A large number of epochs results in a longer time for the proposed algorithm to produce its best possible subset features. The stopping criteria of the BA are set to the total number of bees and the total number of iterations. Table 3.4 shows the parameters used in the experiment.

### 3.2.6 Evaluation of the Proposed Algorithm

This phase is an important stage to testify the robustness and reliability of this work. An important issue in the developing phase is validation. The validation techniques "include testing the model under known input conditions and comparing model output with system output. In other words, validation means that the model compares the performance under known conditions with the performance of the real system" [107]. In this study, the validation of the result depends on two metrics: speed and accuracy. This was determined by comparing the obtained result between original methods from Massudi methods and the extension proposed of swapping mechanism of BAFS results.

#### 3.2.6.1 Experimental Setting

Experiments with the proposed approach was undertaken to evaluate its effectiveness and robustness. Evaluation of the proposed approach on the numerical dataset was performed by comparing the results with those given when using the same approach on the numerical dataset.



The experiments used CORE i7-2450M CPU 2.50 GHZ with 6.00 GB RAM, Windows 7 Home Premium. While C++ was adopted as the implementation programming language. The MLP classifier is used in this approach for training and learning data. A BA and MLP parameters setting were explained with more details in section 4.5.7.

The fitness function for numerical dataset was calculated based on many factors. FS for labelled data was based on mean square error (MSE). The MSE calculation is based on the error from an MLP. This study used Weka application to obtain the accuracy for proposed algorithm. The classification accuracy obtained for a particular dataset and the number of features in the dataset gives the fitness of the corresponding bee as follows:

$$Fitness = \frac{1}{k_1 \times MSE + k_2 \times \frac{N_S}{N_t}} \tag{3.1}$$

where $k_1$ and $k_2$ are weighting factors and MSE is the mean squared error of the MLP in classifying the test data. The term $N_t/N_s$ is included in Eq (4.1) to reflect the fact that it is more desirable to have small feature subsets. $k_1$ and $k_2$ enable the scaling of the contribution of MSE and $Nt/Ns$ to suit particular problems [2][18].

### 3.2.7 Output Analysis Phase

This phase attempts to collect the results from the previous phase and compare them with the latest developments using Bees technique.

### 3.2.8 Documentation Phase

This is the final phase of the research process in this study. The documentation phase aims to document the produced proposed extension to determine the scope for future work.

## Summary

The methodology of this work has been explained in this chapter. The main problems of BA have been explained as the first step in the methodology. The second step has been discussed the types of dataset that were collected from UCI. The next step is about identifying appropriate way from three types of wrapper FS. Step number 4 is mentioned about the way of developing the local search approach by proposing extension that was adopted from Massudi methods. The fifth phase is very important that dealt with evaluating the proposed extension. Next two phases were dealt with analysing and documenting the proposed extension to determine the scope for future work.

# CHAPTER FOUR
# PROPOSED BEES ALGORITHM FEATURE SELECTION (BAFS)

## 4.1 Introduction

Previous chapters on the literature review of the BA issues and their techniques reveal that long computational time interrupts the performance of the proposed approach. Processing all features in the real data might contain irrelevant features that affect the performance of the algorithm. One possible solution is to involve a pre-processing step, known as feature selection (FS), to reduce unrelated features. The Wrapper technique is one of the approaches used by FS for reducing unwanted features. One of the advantages of Wrapper FS is that it can be implemented on different types of dataset. In this chapter, the BA is studied in the context of Wrapper FS problem that focuses on the arrangement of the BA operators. Before going any further, it is important to understand the neural systems to deal with optimisation problems.



## 4.2 Proposed Bees Algorithm Wrapper Feature Selection

The general structure of the proposed Wrapper FS method was discussed in this section. Wrapper FS using the BA on dataset was described in the rest of this section. As mentioned in the previous section, the proposed method uses MLP networks as learning algorithms to guide FS in the form of feedback to the selection process as to how well a given set of features characterises patterns from various classes [2]. For the FS process, the method requires a dataset comprising of patterns, each with $\boldsymbol{N_t}$ features. All the patterns used in the training set are of known classes. Based on the original dataset, new dataset is constructed in which patterns only have a subset of the original features. In other words, a pattern in a new dataset had $N_s$ features **($1 \leq N_s \leq N_t$)** selected from the original set of $N_t$ features [2][18].

A bee represents a subset of ***Ns*** features that can be uniquely identified by a binary string (e.g. 010110111) where the total number of bits is $N_t$ and the total number of non-zero bits is $N_s$. The position i ($1 \leq i \leq Ns$) of a bit along the string indicates a particular feature. If a feature is selected to form a new dataset, the corresponding bit is 1. Otherwise, it is zero [52].

FS starts with the random generation of a population of binary strings (or bees). For each string, a new dataset is constructed using the selected features specified in the string. The training data of the dataset are used in training an MLP whereas the remaining data or the test data are employed to evaluate the classification accuracy of the trained MLP. Figure 4.1 shows the FS method of the Bees Algorithm.

The proposed algorithm started with the parameter initialisation phase that includes the following parameters: the total number of Bees (n), the total number of "elite" Bees (e), the number of sites selected for neighbourhood search (m), the number of Bees around selected location (*nsp*), and the number of Bees around each "elite" locations (*nep*). Setting of these parameters is already discussed in the previous section [2].

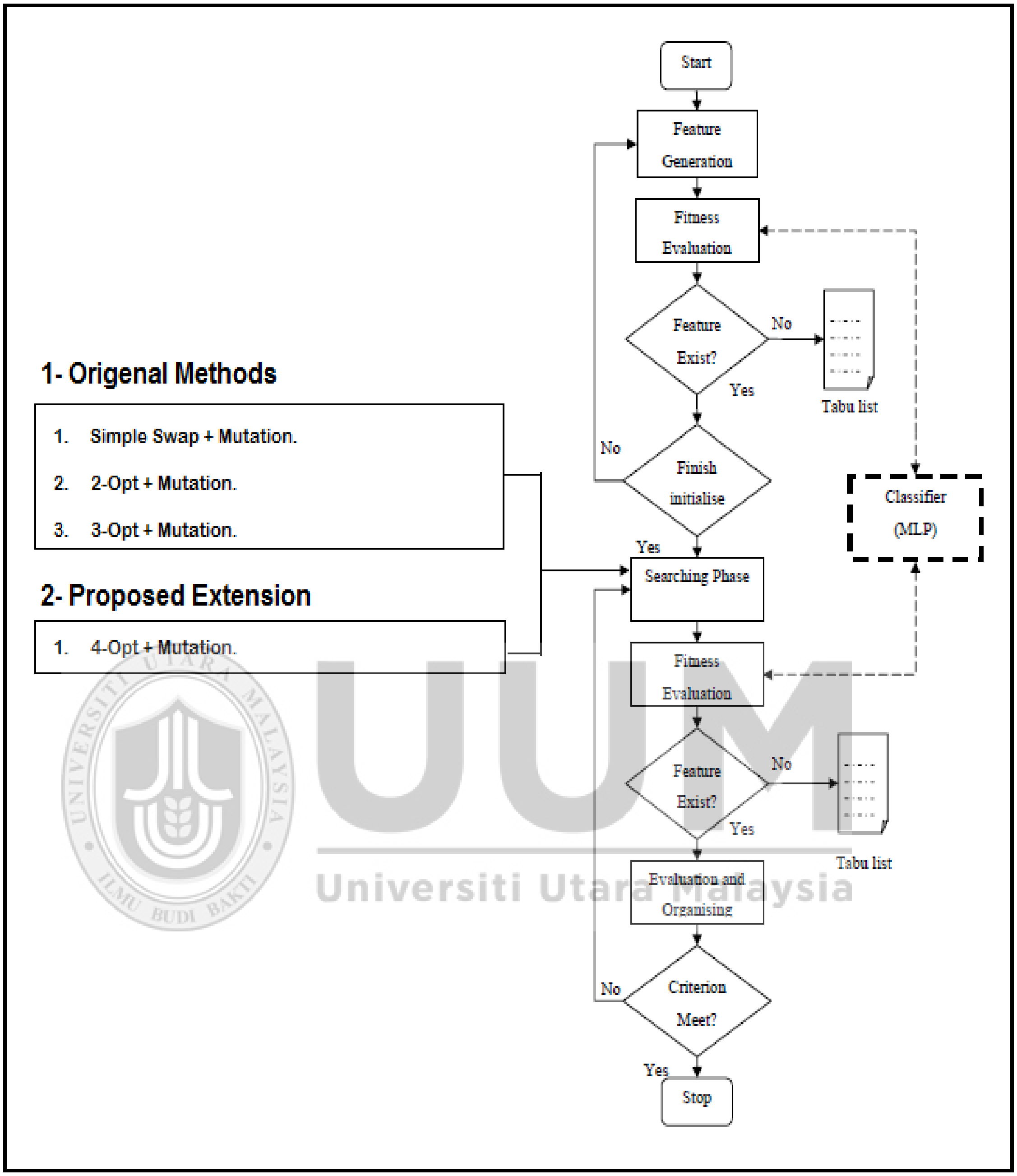


*Figure 4.1.* Flow chart of proposed BAFS method (Adopted by [2])

The initialisation process generated subset features for each bee – $\boldsymbol{feat_1, feat_2, \ldots, feat_n}$. This phase generates random subset features and calculates the total MSE and fitness, based on Eq (4.1).

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N+1} x_i \quad (4.1)$$

The next step in this algorithm is organisation of the bees based on the fitness value acquired from the initialisation process. Some of the new bees were assigned to an elected number of features (also known as 'sites' in the original BA terminology). The top quality features were identified by evaluating the fitness value of each bee. This best feature subset is known as "elite" (*e*) which becomes the target feature for other bees as they attempt to find possible better features from the existing ones. A total number of bees (*nep*) was assigned to these potential features. Each of these new bees was assigned new possible better subset features. The pseudo code of the initialisation process can be seen in Figure 4.2.



```
For (i=1 to n)
Do
      Generate random feature, feat_i
      Calculate total feature selected, totalFeat of feature, feat_i
While ((totalFeat == 0) OR (totalFeat == totalAllFeat)) AND (feat_i  is
        not exist)
Add feature, feat_i if not exist
Calculate error and fitness of feature, feat_i
Save error and fitness of feature, feat_i in bee_i
```

*Figure 4.2.* General Pseudo code of initialization process of each bee of wrapper FS

The assignment of new features is based on the existing fitness features. The neighbourhood search process started with the generation and identification of the best features. A newly generated feature $\boldsymbol{feat_{new}}$ was evaluated using the same principle equation as in the initialisation phase. A local search method is applied to generate this new feature, $\boldsymbol{feat_{new}}$ from the existing feature $\boldsymbol{feat_i}$. The local search method implemented a combination method for new feature generation process including the following:

- **Massudi Method:** (Simple Swap + Mutation, 2-Opt + Mutation, 3-Opt + Mutation) as an Massudi idea developed in [2]

- **Extension proposed:** (4-Opt + Mutation, Simple Swap) as an extension idea that was described in detail under Local Search Methods in Section 4.5.

Differences between the Massudi idea and the extension idea are described in Figure 4.3 where: (a) describes the Massudi idea done by [2], (b) describes the 4-Opt extension idea for this work.

The remaining n-m features of other bees are filled by some other new candidates. These newly recruited bees have entirely new features which are randomly generated. The fitness of the newly generated features of each bee is calculated here.

In the next phase, features of each bee (old and newly generated) are organized once more to find the best features in the list of new bees. The process continues until it surpasses the total number of iterations.

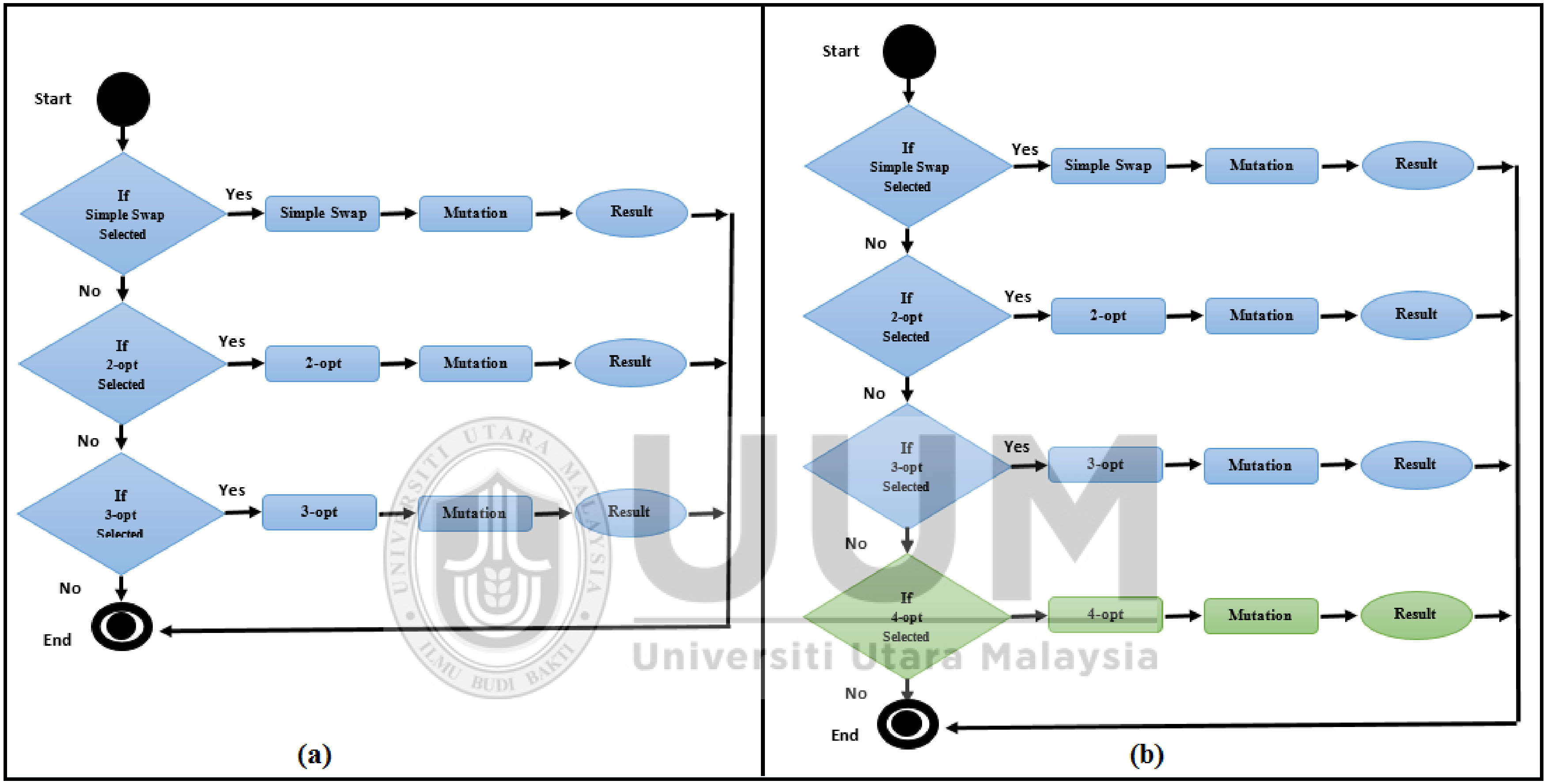


*Figure 4.3.* Difference between origin idea and extension idea of proposed BA FS method. (a) is an origin idea , (b) is 4-Opt extension idea

## 4.3 Feature Representation in Bees Algorithm

Representation of a feature subset in the BA requires individual feature representation. As a result, each feature is represented in binary form. ‘1’ indicates the selection of a particular feature of the whole feature set, or else it was ‘0’. For example, if the total number of the original features is 8, and only 7 of them are selected and a few representations of the new subset features are generated, it used: a) ‘01111111’ which means that the first feature from the total of the original features will not be selected and b) ‘1111110’ which means that only the first seven features are selected. Figure 4.4 illustrates how the features are presented in the BA.

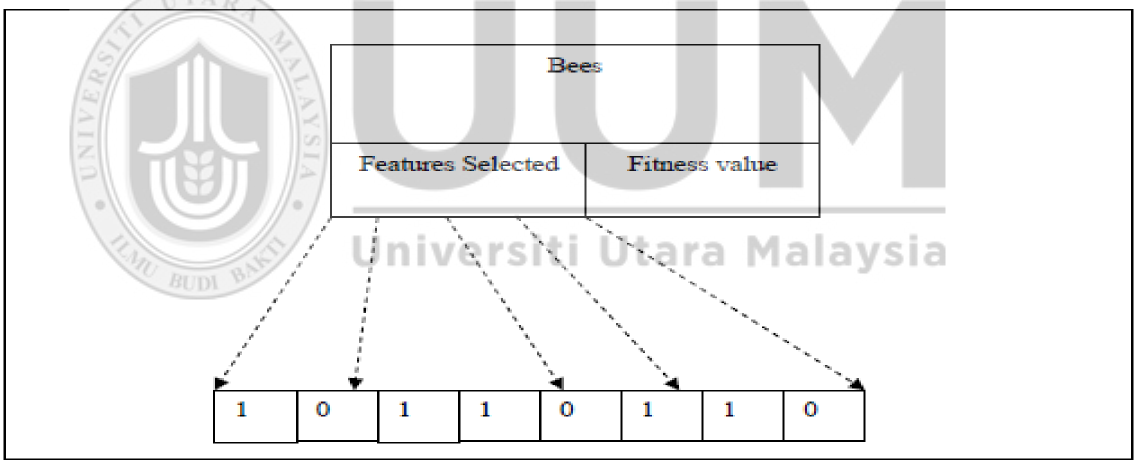


*Figure 4.4.* Structure of feature representation of each bee in the proposed algorithm. ‘1’ indicate the particular feature is selected, otherwise ‘0’ (Adopted from [2])

## 4.4 Local Search Methods

This section describes the implementation of the local search methods on the numerical dataset. In case of combinatorial domains, there is no mathematical distance definition for the neighbourhood search unlike continuous domains. Since the BA was developed for continuous domains, it is necessary to modify the neighbourhood part by simply replacing the patch with a local search operator [108][181]. Determining the best local search method is important as it enables an overall solution of a problem [2]. The current study aims to design an extension of the local search technique mechanism based on the BA architecture.

Coming back to the combinatorial domain, FS is known as a good example of the combinatorial problem. Therefore, local search in the BA is adapted from a number of methods including, the simple swap, 2-Opt, 3-Opt, and 4-Opt local search approach used for the TSP problem. The use of a combination method is a relatively new idea as it was not included in the previous works conducted on the field. To optimise the fitness possible solution, the current study used all of the local search approaches mentioned above and suggest a combination among them, including 4-Opt as the extension idea of the local search technique. It used the proposed algorithm to improve the speed and increase the accuracy of BAFS. To do this, the study used simple swap, 2-Opt, 3-Opt operators and add 4-Opt as an extension as the best candidates to be implemented in BAFS. Generally, the 2-Opt and 3-Opt local searches pass through all possible combinations (pairs, triplets and quadruplets) while calculating the savings for the new range of probability for each combination (pairs, triplets and quadruplets), find the global best improvement, and perform the move (apply the best found improvement to the current solution) of the BA. Therefore, local search in the proposed algorithm is a combination of a swapping operation and mutation [18][2]. In order to understand the combination idea, it is relevant to individually know the local search approaches.

### 4.4.1 4-Opt-Based (Double-Bridge Move) Search Approach

There are numerous techniques and mechanisms that can improve local search while avoiding common problems. One of them is the double-bridge move or 4-Opt move that was first mentioned by Lin and Kernighan in 1973 [182] as an example of simple move which cannot be normally generated by 3-Opt or Lin-Kernighan algorithm. This move is used by different modern algorithms in order for its ability to escape from the local optima [183].

The 4-Opt-based search approach (Double-Bridge Move) is an extension to simple swap, 2-Opt, and 3-Opt operators. In the cases of the simple swap and 2-Opt operator, there is only one way to reconnect the tour fragments after deleting the two selected edges [2][117]. The 3-Opt operator chooses the best triple edges that are not yet connected to the current tour [123]. In contrast, the ‘double-bridge move’ (4-Opt) is used as the perturbation technique, and a stochastic 2-Opt is used as the embedded local search heuristic. The double-bridge move involves partitioning a permutation into 4 pieces (a, b, c, d) and putting it back together in a specific and jumbled ordering (a, d, c, b) in the TSP problem [96].



Therefore, this method can be termed as an extension from the previous methods proposed by Mahmuddin 2009 [2]. For this, reference points are generated randomly at $L_1$, $L_2$, $L_3$ and $L_4$ where $L_1 > L_2 > L_3 > L_4$. Feature values at index $L_1$, $L_2$, $L_3$ and $L_4$ ($F_{L1}$, $F_{L2}$, $F_{L3}$ and $F_{L4}$, respectively) use a 2-Opt-based operation as two sequential parts. The process is implemented by swapping the feature values at indices $L_1$ ($F_{L1}$) and $L_2$ ($F_{L2}$) as the first part and $L_3$ ($F_{L3}$) and $L_4$ ($F_{L4}$) as the second part. The swapping process continues with ($F_{L1+1} = F_{L2-1}$) and ($F_{L3+1} = F_{L4-1}$) until ($L_1$ and $L_2$) and ($L_3$ and $L_4$) have the same value. Figure 4.5 represents a simple example of this 4-Opt-based operation. Figure 4.6 shows the pseudo code of neighbourhood procedure of FS in the modified BA by 4-Opt-based operation and mutation.

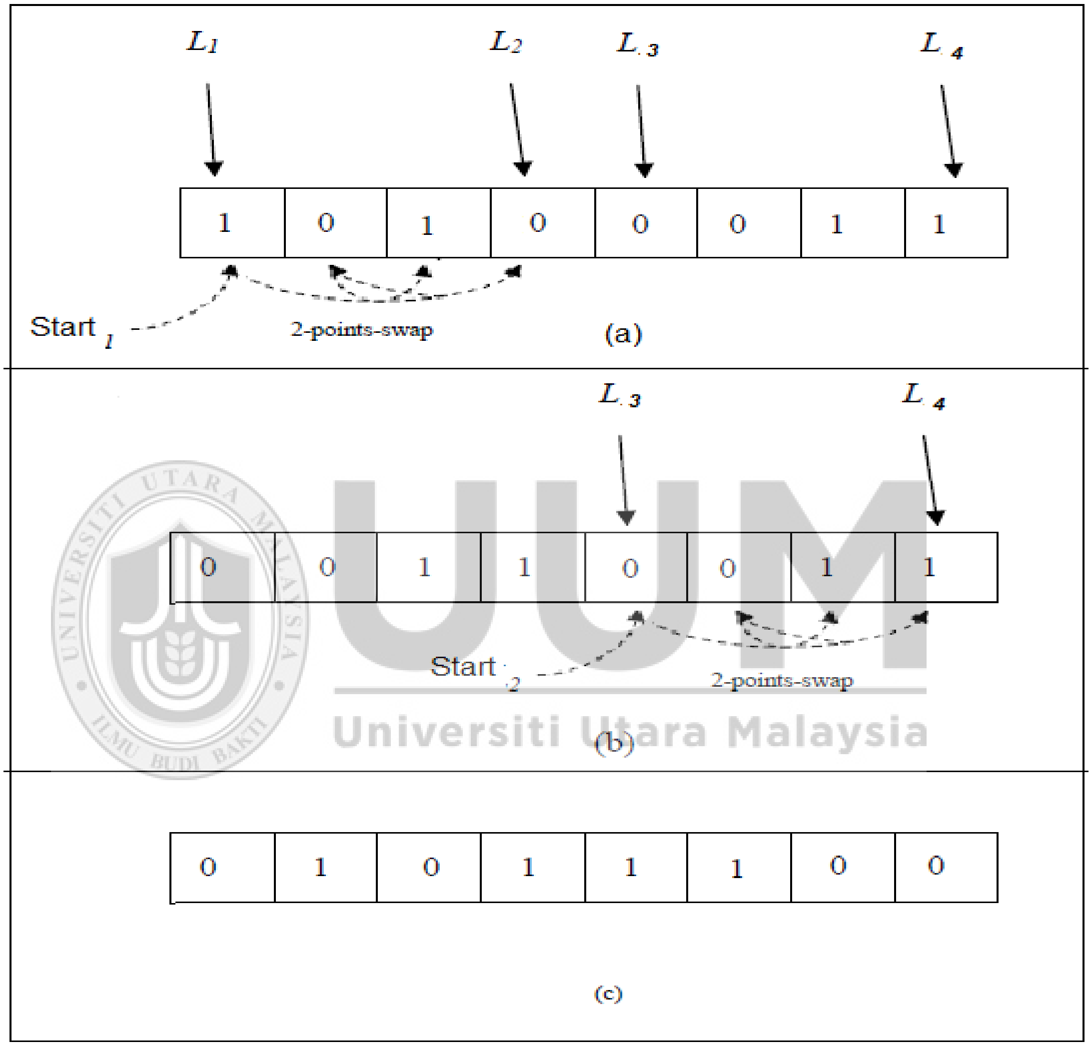


*Figure 4.5.* A 4-Points-Swap move: (a) original binary feature, (b) The first 2-Points-swap and (c) the second 2-Points-swap feature after resulting movie

```
For (itr=1 to maxRun)
{
  If (totalFeat > 2totalFeat-2)
     stop process
  For (i=1 to m) {
     If (totalFeat > 2totalFeat-2)
        stop process
     For (j=1 to recuited bee) {
        If (totalFeat > 2totalFeat-2)
           stop process
        Do 4-Points-swap(F i )
        Do SimpleMutation(F i )
        If feat new is not exit,
           save in tabu list
        Evaluate and calculate feature, F new
        If fitness F new <= Fti save feature, new F
    }
  }
}
```

*Figure 4.6.* Pseudo code of neighbourhood procedure of FS in modified BA by 4-Points-Swap Approach (Double Bridge Move) and Mutation



### 4.4.2 Multi-Bits Mutation Approach

In this approach, the mutation operator is adopted from the one used in GA. A simple mutation operates by changing the value of the target index of features. If the feature is selected (indicated with a '1'), the mutation was transformed to an unselected feature. In case of an unselected feature (indicated with a '0'), it was altered to '1'. Other GA operators such as the crossover are inappropriate in the BA framework feature since crossover requires two parents. Evaluations in the BA are based on single rather than multiple features. In addition, the proposed algorithm is built in this manner.

```
Do {
      total_bit = Generate random
} while (total_bit == number_of_features)
For (i=1 to total_bit) {
    Generate randomly index bit location, indx
    If indx is not exist in Loc_i then
    Loc_i = indx
}
For (i=1 to total_bit) {
    If F_Loc_i =='0' then F_Loc_i ='1'
    Else F_Loc_i ='0'
}
```

*Figure 4.7.* Pseudo code of mutation operations in the proposed algorithm



The mutation approach starts with the random generation of the total number of bits that were mutated. This total number is later used to mutate bits in the selected features based on the mutation process that has been described above. The pseudo code of the proposed mutation can be seen in Figure 4.7.

## Summary

This chapter has proposed BA Wrapper feature selection regarding to proposed extension method is 4-Opt and combination approaches based on Simple swap, 2-Opt, 3-Opt and 4opt with Mutation which derived from the Massudi methods method. FS in this chapter is based on the classifier error criteria. In the proposed algorithm, MLP is used for error calculation to train datasets. To verify the proposed extension by compared the result with respect to time and accuracy with Massudi methods that was done by [2] including Simple Swap, 2-Opt and 3-Opt.

# CHAPTER FIVE
# FINDINGS AND DISCUSSION

## 5.1 Introduction

This study evaluated the proposed algorithm in terms of speed and accuracy of BAFS by comparing the results with Massudi methods [2]. In this study, there are four types of local search techniques approaches that were used in the proposed algorithm. The Simple Swap, 2-Opt and 3-Opt as an origin idea and 4-Opt approach as an extension idea that was described in detail in the previous section. To evaluate the effectiveness of the proposed local search techniques, a sequence of experiments was undertaken for each of these local search techniques. Evaluation of the robustness of the proposed approaches is made using numerical dataset. This experiment used five different types of datasets like Pima Indians Diabetes, SPECT-Heart, Hepatitis, Soybean and Lung-cancer. These dataset were chosen such that they had different data characteristics, including total number of features, data type and total number of instances are start from (8, 19, 22, 35 and 56 feature) and (768, 80, 155,47 and 32) respectively. A detailed description of the dataset can be seen in section 3.1.1.2.

These datasets are a well-known dataset for classification and they have been chosen here because it contains the only medium scale number of features that was explained previously in Section 2.5.2. This medium scale number of features were selected so that obtaining all possible solution using standard parameters for manageable task. Each technique had three runs in order to select the most suitable results.

## 5.2 Results of Proposed Local Search

As mentioned in the previous chapters, the proposed Bees Algorithm feature selection had been implemented using C++ language on a personal computer by Visual Studio 2013 professional, whose CPU is Intel(R) core (TM) i5 2.53 GHz and memory size is 4 GB and the operating system is Windows 7 Ultimate. According to this characteristics the propose Algorithm was applied on different dataset. The dataset assumed as medium range of data because it is ranged between (8, 19, 22, 35 and 56) for (Pima Indians Diabetes, SPECT Heart, Hepatitis, Soybean and Lung-cancer Dataset) respectively.

### 5.2.1 Experimental Results for Pima Indians Diabetes Dataset

Experimental results clarified different local search method as an operator in BA on Pima Indians Diabetes Dataset. All the results between Massudi and proposed methods in terms of accuracy, time, error rate, and fitness, have been clearly shown as provided in Figure 5.1. More description of these results can be seen in Table 5.1.



#### 5.2.1.1 Accuracy and Speed

Figure 5.1 illustrate the highest accuracy that obtained from applying all local search mechanism is proposed extension represented a manner that double-bridge search move (4-opt and Mutation) was 75.5%. This result has achieved in second running and solely 3 features has been selected from 8 features. The proposed extension methods has implemented in 44.4s with error rate 0.0000. Further, the fitness rate was 0.8888.

Whilst, highest accuracy of Massudi methods (number 3 (3-Opt and Mutation)) was 67%. This result was obtained during third running with 2 features selected from 8 features. The Massudi methods have implemented during 50.3s with an error rate 0.0000 and the fitness rate was 0.7999 as shown in Table 5.1.

Therefore, it can be concluded that the method double-bridge search move from proposed extension is increased slightly 8.5% of accuracy with 5.9s less time out of the from Massudi methods.

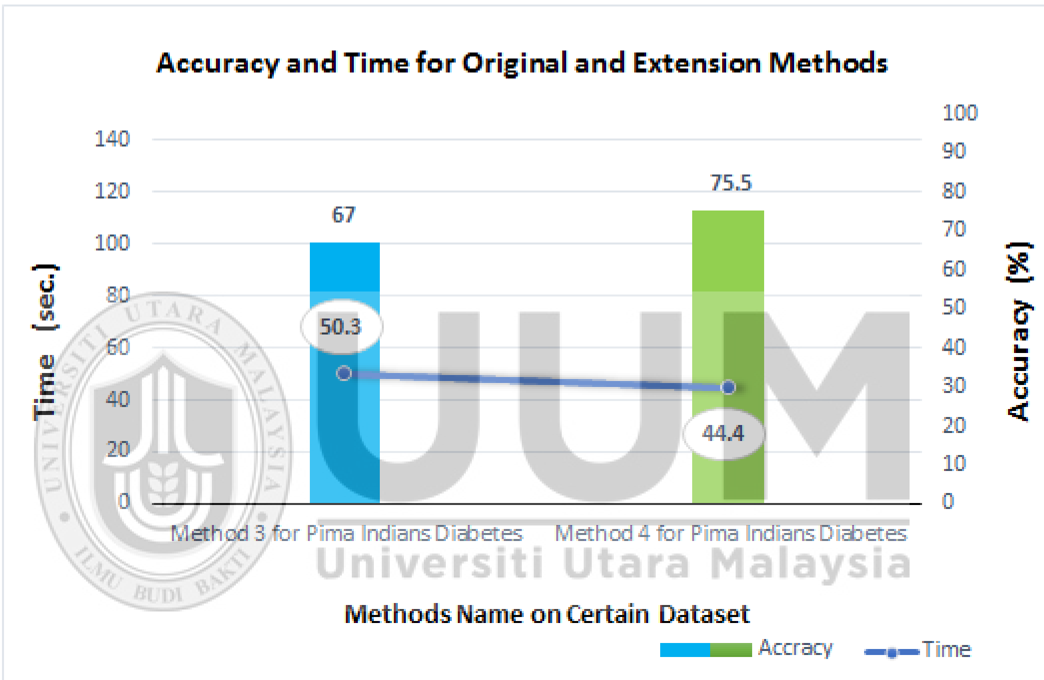


*Figure 5.1.* Maximum Accuracy and Minimum Time for Massudi and Extension Methods on Pima Indians Diabetes Dataset

Table 5.1

*Experimental Result of different local search methods on Pima Indians Diabetes Dataset.*

| **Dataset Name** | **Methods Name** | **Type of Methods** | **No Of Runs** | **No Of Selection** | **Error (MSE)** | **Fitness** | **Time (Second)** | **Accuracy (%)** |
|---|---|---|---|---|---|---|---|---|
| Pima Indians Diabetes Dataset | 3-Opt And Mutation | Massudi Method | 3 | 2/8 | 0.0000 | 0.7999 | 50.3 | 67.0 |
| Pima Indians Diabetes Dataset | 4-Opt And Mutation | Proposed Extension | 2 | 3/8 | 0.0000 | 0.8888 | **44.4** | **75.5** |

*Note: MSE = Root Mean Square Error.*

### 5.2.2 Experimental Results for SPECT-Heart Dataset

The previous sections have shown the experimental result for Pima Indians Diabetes Dataset, while this section provided summarization of the results of a comparative study between Massudi and proposed methods in terms of accuracy and time for Spect-Heart dataset as shown in Figure 5.2. A more description of these results can be seen in Table 5.2.

#### 5.2.2.1 Accuracy and Speed

The results obtained are presented in Figure 5.2 clarifies the superior accuracy obtained from the extension proposed by method number 4-Opt and Mutation was 83.8% took 27.2s of the time, as well as the errors and the fitness are 0.1882, 0.6638 respectively through the second running with 7 features selected from 22 features. Meanwhile, the maximum amount of accuracy obtained from the original method by number 3-Opt and Mutation was 79.4% took 29.1s of the time, instead of the error and the fitness are 0.1941, 0.6612 respectively in the second running with 7 features selected from 22 features.

Therefore, the both methods produce almost the same selected feature. However, the conclusion from these results is that the proposed extension overcome the original one in the highest result of accuracy at a rate of 4.4%. as well as the results also showed that the method which based on extension proposed had proved effective in terms of time comparing with the original method even it was small amount 1.9s.

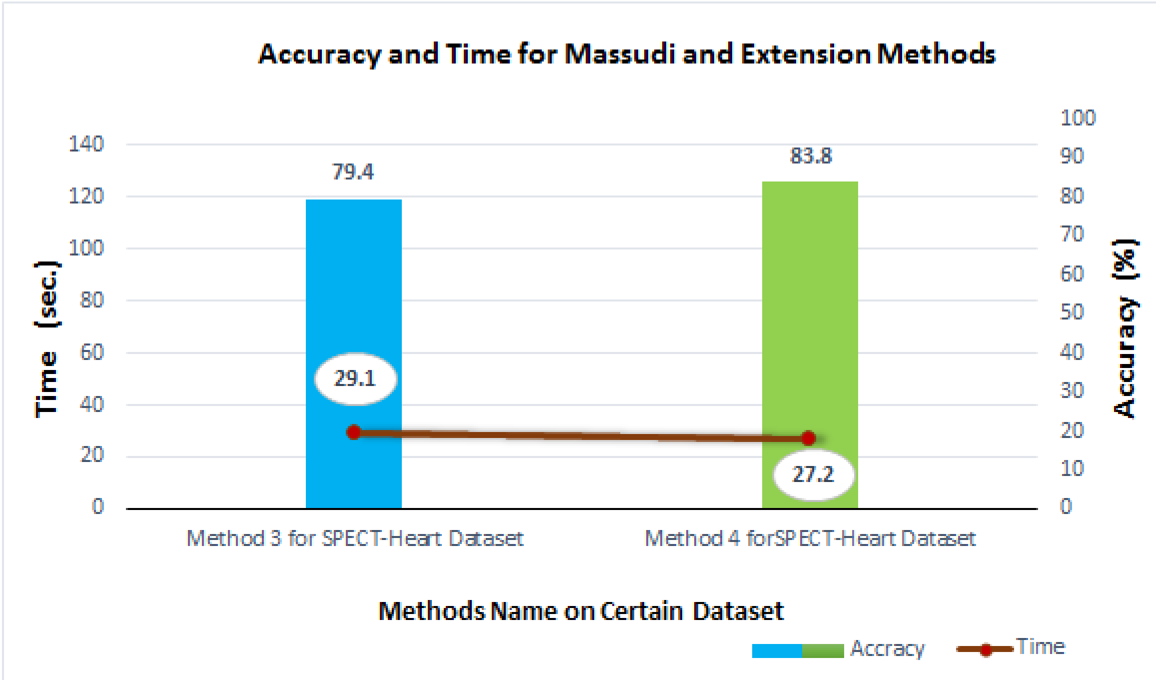


*Figure 5.2.* Maximum Accuracy and Minimum Time for Massudi and Extension Methods on SPECT-Heart Dataset

Table 5.2
*Result of different local search methods on SPECT-Heart dataset*

| No | Dataset Name | Methods Name | Type of Methods | No Of Runs | No Of Selection | Error (MSE) | Fitness | Time (Second) | Accuracy (%) |
|---|---|---|---|---|---|---|---|---|---|
| 1. | SPECT Heart Dataset | 3-Opt And Mutation | Massudi Method | 2 | 7/22 | 0.1941 | 0.6612 | 29.1 | 79.4 |
| 2. | SPECT Heart Dataset | 4-Opt And Mutation | Proposed Extension | 1 | 7/22 | 0.1882 | 0.6638 | **27.2** | **83.8** |

*Note: MSE = Root Mean Square Error.*

### 5.2.3 Experimental Results for Soybean Dataset

Experimental results explained diverse local search methods as an operator in BA to test Hepatitis Dataset. This section shows empirical comparisons of the relative accuracy and the value of time that was spent in each method to achieve the optimal results. The overall results that presented between Massudi and extension proposed methods had been clearly provided in Figure 5.3. Further description of these results can be seen in Table 5.3.

#### 5.2.3.1 Accuracy and Speed

From the previous two experimental results for Pima Indians Diabetes and SPECT Heart datasets illustrated overwhelming success by the proposed extension has come up with better results compared with the original proposed in terms of speed and accuracy. However, Soybean Dataset shows the surprising results. Both methods obtained the same values in terms of the accuracy which were 100%, But those methods produce these results by difference time However, However, the proposed extension represented by method 4-Opt and Mutation had improved slightly in term of the time than Massudi methods at 2.9s. The all information for both methods illustrated in table 5.3 and Figure 5.3.

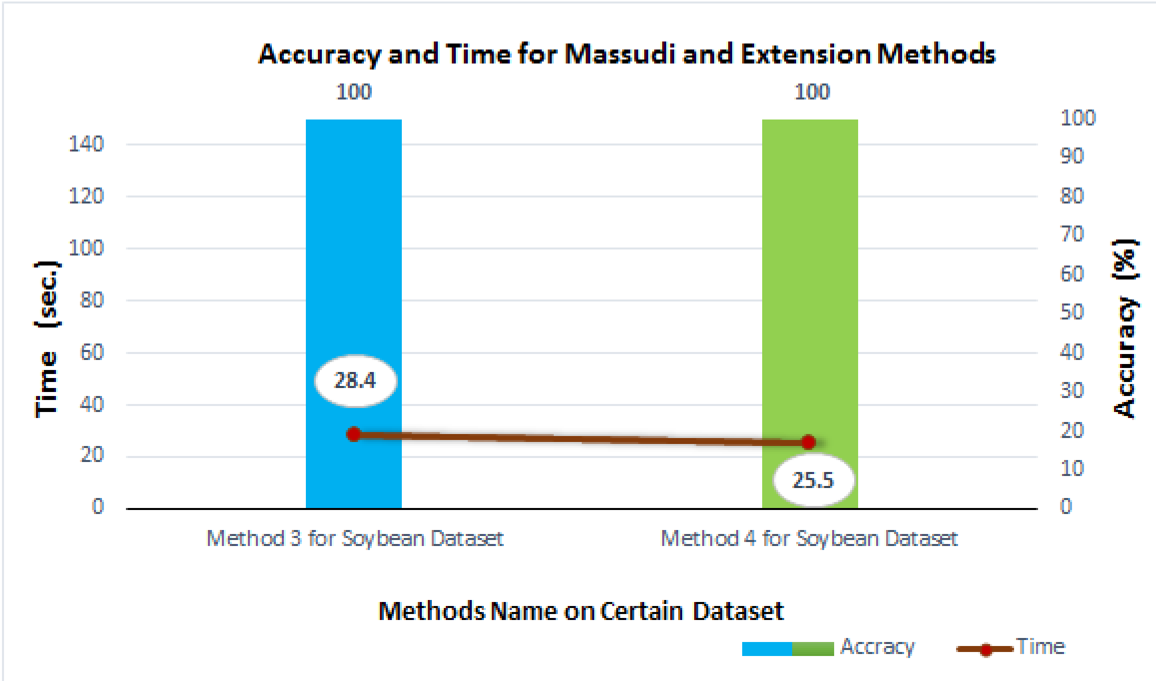


*Figure 5.3.* Maximum Accuracy and Minimum Time for Massudi and Extension Methods on Soybean Dataset

Table 5.3
*Result of different local search method on Soybean Dataset*

| No | Dataset Name | Methods Name | Type of Methods | No Of Runs | No Of Selection | Error (MSE) | Fitness | Time (Second) | Accuracy (%) |
|---|---|---|---|---|---|---|---|---|---|
| 1. | Soybean Dataset | 3-Opt And Mutation | Origin Method | 1 | 11/35 | 0.0034 | 0.7590 | 28.4 | **100** |
| 2. | Soybean Dataset | 4-Opt And Mutation | Proposed Extension | 2 | 10/35 | 0.0044 | 0.7745 | **25.5** | **100** |

*Note: MSE = Root Mean Square Error.*

### 5.2.4 Experimental Results for Hepatitis Dataset

Experimental results clarified different local search method as an operator in BA on Hepatitis dataset. All the results shows between Massudi and proposed methods in terms of accuracy and time (second) have been clearly shown as provided in Figure 5.4. An additional description of these results can be seen in Table 5.4.

#### 5.2.4.1 Accuracy and Speed

The proposed extension had overcome its counterpart clearly in terms of time and accuracy with the following data; Pima Indians Diabetes, SPECT Heart datasets. However, both proposed methods obtained the same results in term of accuracy with Soybean Dataset. Regarding to Figure 5.4 illustrated the overall approaches for Hepatitis dataset that gained optimal results of accuracy were close to 85%. While, the optimal result for Massudi methods were close to 82%. It can be noticed that there were a diversity of results in terms of time, error rate and fitness for both methods.



In the other hand, regarding Table 5.4, it can be noticed that the proposed extension obtained better result with respect to the time than original one. The method gain 32.4s which less than the time gained by Massudi method which was 34.9s. The extension method had obtained its time through third running that selected 6 features out of 19. The method's error rate and fitness were 0.4619 and 0.979 respectively. Meanwhile, the method 3 had achieved its time through first running that selected 7 features out of 19. The method's error rate and fitness were 0.4303 and 0.5404 respectively.

From the above experimental results, it has shown that the BAFS works well with the proposed extension to save time and increase the accuracy. However, the proposed extension method outperform his competitor methods. From Table 5.4 shows the results that the proposed algorithm is able to estimate the total number of features correctly with satisfactory results. Improved its results comparing with Massudi methods in terms of accuracy and time; which were 2.5%, 2.5s, respectively.

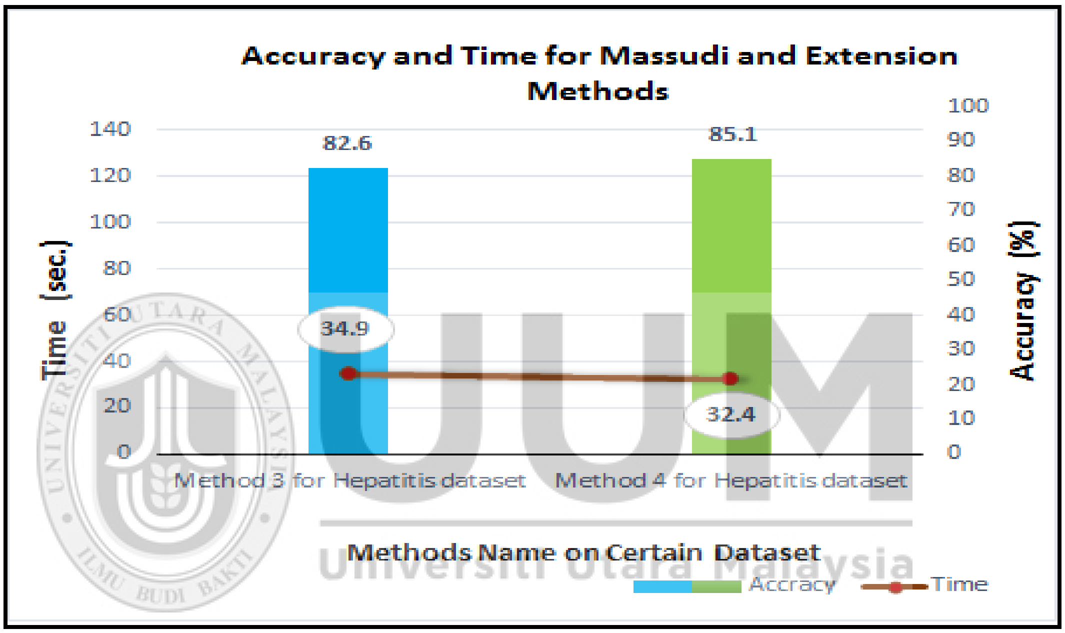


*Figure 5.4.* Maximum Accuracy and Minimum Time for Massui andxtension Methods on Hepatitis Dataset

Table 5.4
*Result of different local search method on Hepatitis dataset*

| **No** | **Dataset Name** | **Methods Name** | **Type of Methods** | **No Of Runs** | **No Of Selection** | **Error (MSE)** | **Fitness** | **Time (Second)** | **Accuracy (%)** |
|---|---|---|---|---|---|---|---|---|---|
| 1. | Hepatitis Dataset | 3-Opt And Mutation | Massudi Method | 1 | 7/19 | 0.4303 | 0.5404 | 34.9 | 82.6 |
| 2. | Hepatitis Dataset | 4-Opt And Mutation | Proposed Extension | 3 | 7/19 | 0.4619 | 0.979 | **32.4** | **85.1** |

*Note: MSE = Root Mean Square Error.*

### 5.2.5 Experimental Results for Lung-cancer Dataset

Experimental results clarified different local search method as an operator in BA for the Lung-cancer dataset. In this section, the relative accuracy and the value of time that was spent in each method to achieve the optimal results were compared empirically. All the results that showed between Massudi and proposed methods had been clearly provided in Figure 5.5. An additional description of these results can be seen in Table 5.5.

#### 5.2.5.1 Accuracy and Speed

The topmost result of the accuracy of the Lung-cancer dataset are illustrated in Figure 5.5. This figure Clarifies the highest accuracy value obtained from extension method which was 50% in the second running with 24 features selected from 56 features. This superior accuracy has achieved with 0.0003 error rate, 0.7347 of fitness and 17.0s of time. In contrast, the superior results from Massudi methods is 46.9% for accuracy which obtained by method 2-Opt and Mutation with 24 features selected from 56 features. The results achieved by this method which based on the amount of error rate and fitness are 0.0600 and 0.6882 during 17.0s time.

From Table 5.5, It can be determined from the above that the extension method proposed and Massudi method obtained same value in terms of number of features selection and the time consuming to get the results which are 24/56 features and 17.0 second. The enhancement just happened in term of accuracy which is 3.1% better than the higher results of the Massudi methods.

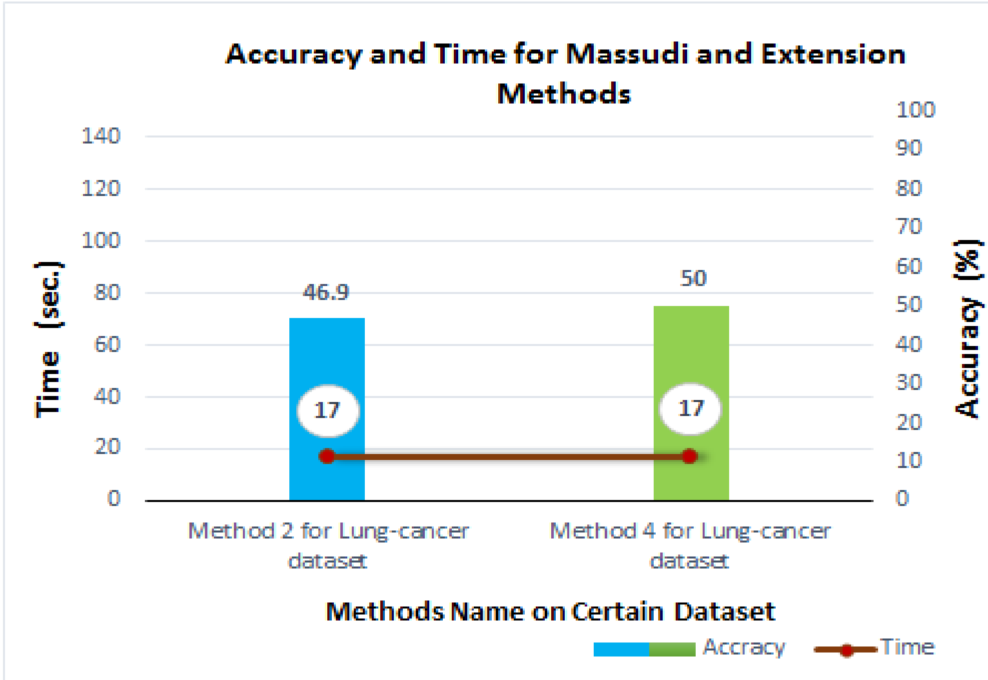


*Figure 5.5.* Maximum Accuracy and Minimum Time for Massudi and Extension Methods on Lung-cancer Dataset

Table 5.5
*Result of different local search method on Lung-cancer dataset*

| No | Dataset Name | Methods Name | Type of Methods | No Of Runs | No Of Selection | Error (MSE) | Fitness | Time (Second) | Accuracy (%) |
|---|---|---|---|---|---|---|---|---|---|
| 1. | Lung-cancer Dataset | 2-Opt And Mutation | Massudi Method | 1 | 24/56 | 0.0574 | 0.6729 | **17.0** | 46.9 |
| 2. | Lung-cancer Dataset | 4-Opt And Mutation | Proposed Extension | 2 | 24/56 | 0.0003 | 0.7347 | **17.0** | **50.0** |

*Note: MSE = Root Mean Square Error.*

## 5.3 Discussion Result

The performance of this study in terms of speed and accuracy achieved on the basis of the features' ability to discriminate between the good, normal, and defective. As earlier reported, one way of improving the performance, which is to select the most appropriate features from a given feature set for the purpose of training and testing so that better results can be achieved overall. In this research, an extension of the selection approach of wrapper-based feature using Bees Algorithm local search for numerical datasets is presented. Based on the earlier discussion, Bees Algorithm is a swarm-based optimisation technique mimicking the foraging behaviour of honey bees found in nature. In order to demonstrate the wrapper-based feature selection procedure, a 4-opt (Double-bridge) is used as a local search in this study.

The first experiment revealed that this study cannot be conducted on the data that have small feature selection as in Iris, Haberman, Badges, and Balance Scale and so on as shown in Table 1 in the appendix. As mentioned in section 2.4.2, there are many ideas suggested by researchers to divide the data in terms of number of feature selection, including: small, medium, and large [2] [67]. However, other academics have agreed on dividing them into large and small only [68] [69] [70]. Therefore, this study considered the medium range because the tested data ranged from 8 until 56 features. Based on the literature, it can be concluded that the small value of feature in this study means smaller than or equal to 4 features for data, while the large one means bigger than or equal to 100 features.

Consequently, the main reasons that justify why this study did not deal with small and large features are as follows:

Firstly, in order to find subset of m collection from N number of features ($m < N$, $m \neq 0$), ($2^N$-2), possible combinations are required. So, the maximum number of small feature is 4 that is meant "($2^4$-2) = (2*2*2*2-2) = (16-2) = 14". Thus, this amount of number of feature does not need working with local search mechanism to obtain the optimal and near optimal features. The original Bees algorithm with wrapper-based feature selection is quite enough for this amount of features.

Secondly, the large numbers of features also are not within the range of this study. As for the second experiment, this study was applied on huge data (large features), the study recurred more than one time to implement one test. This delay was occurred because the Parameter Setting for MLP Bees Algorithm (BA) were not prepared and arranged to deal with huge data. For instance, the number of iteration n, m, e, nep, nsp and number of Epoch in Multilayer Perceptron are very important to be arranged to deal with huge data. However, the increment number of inputs needs the most powerful ways to get the best result.



On the one hand, on the basis of the obtained results, it can be noted that classification accuracies comparable with those for the full-feature cases were achieved despite large reductions in the number of features. This confirms the ability of the proposed method to choose informative features. This study has empirically shown the effect of applying the proposed extension in Figure 5.6.

The accuracy and time of the algorithm were computed with maximum absolute differences of the best results. This table also shows that majority of the tested dataset produce results closer to 90%.

On the other hand, Table 5.6 shows that the proposed extension had overcome its counterparts clearly in terms of time than the original approach with the following datasets: Pima Indians Diabetes, Soybean, Spect-Heart, Hepatitis and Diabetes. However, a same result is obtained from Lung-cancer dataset. Figure 5.6 indicates that BAFS works well with the proposed extension and in the same time the accuracy.

Table 5.6

*Summarize Results of various datasets that is used in the experiment both proposed methods*

| No | Dataset Name | Methods Name | Type of Methods | No Of Runs | No Of Selection | Error (MSE) | Fitness | Time (Second) | Accuracy (%) |
|---|---|---|---|---|---|---|---|---|---|
| **1.** | Pima Indians Diabetes Dataset | 3-Opt and Mutation | Massudi Method | 3 | 2/8 | 0.0000 | 0.7999 | 50.3 | 67.0 |
| | | 4-Opt and Mutation | Proposed Extension | 2 | 3/8 | 0.0000 | 0.8888 | **44.4** | **75.5** |
| **2.** | SPECT Heart Dataset | 3-Opt and Mutation | Massudi Method | 2 | 7/22 | 0.1941 | 0.6612 | 29.1 | 79.4 |
| | | 4-Opt and Mutation | Proposed Extension | 1 | 7/22 | 0.1882 | 0.6638 | **27.2** | **83.8** |
| **3.** | Soybean Dataset | 3-Opt and Mutation | Origin Method | 1 | 11/35 | 0.0034 | 0.7590 | 28.4 | **100** |
| | | 4-Opt and Mutation | Proposed Extension | 2 | 10/35 | 0.0044 | 0.7745 | **25.5** | **100** |
| **4.** | Hepatitis Dataset | 3-Opt and Mutation | Massudi Method | 1 | 7/19 | 0.4303 | 0.5404 | 34.9 | 82.6 |
| | | 4-Opt and Mutation | Proposed Extension | 3 | 7/19 | 0.4619 | 0.979 | **32.4** | **85.1** |
| **5.** | Lung-cancer Dataset | 2-Opt and Mutation | Massudi Method | 1 | 24/56 | 0.0574 | 0.6729 | **17.0** | 46.9 |
| | | 4-Opt and Mutation | Proposed Extension | 2 | 24/56 | 0.0003 | 0.7347 | **17.0** | **50.0** |

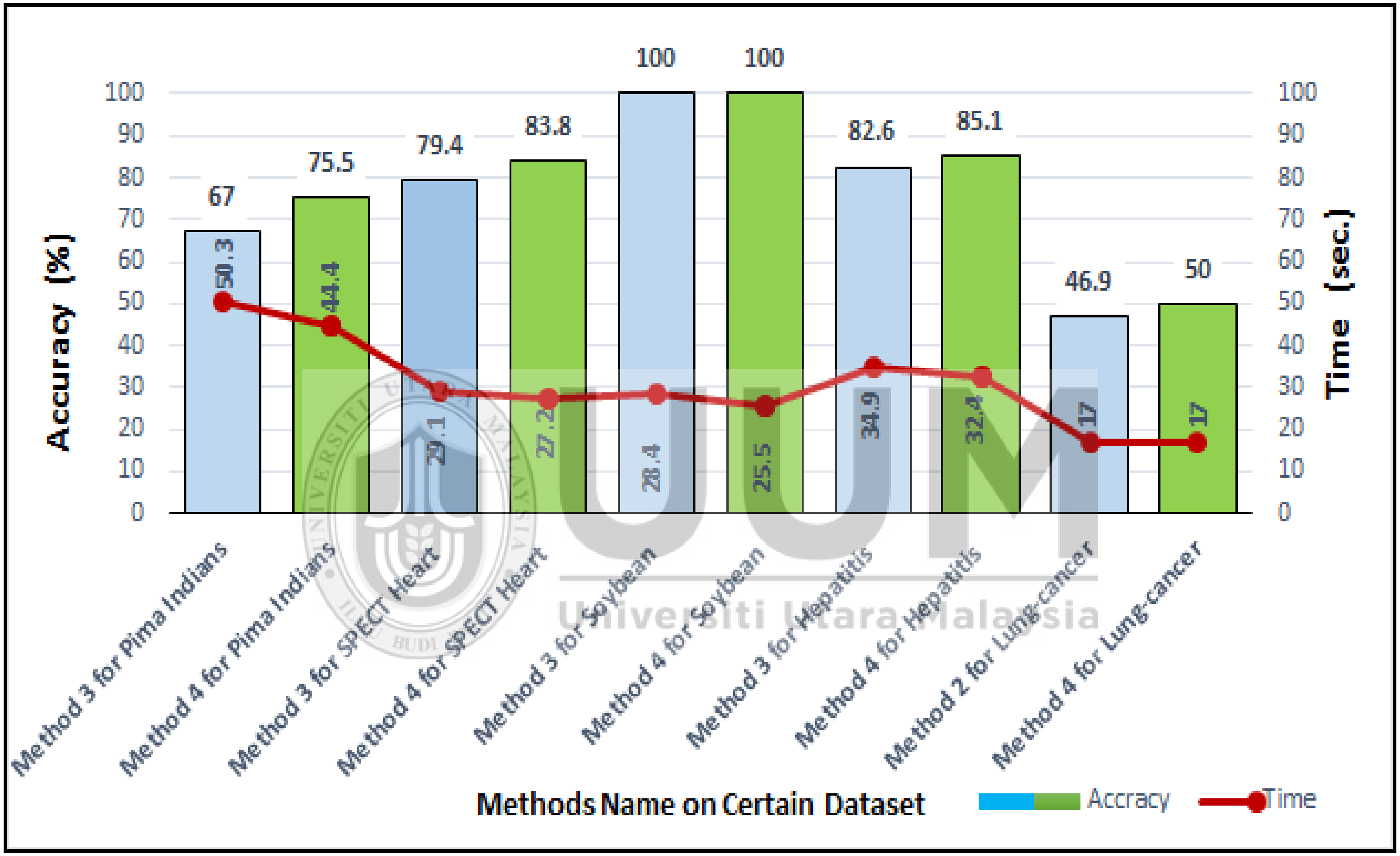


*Figure 5.6.* Maximum Accuracy and Minimum Time for Massudi and Extension Methods on Different Numerical Dataset

## Summary

This chapter has presented a method of reducing the number of features by selecting only those most relevant in a dataset. This chapter has presented a comparative study between all methods. The revealed results are based on two metrics, namely time and accuracy. All methods have been applied to various numerical datasets that were adopted from the UCI repository. The overall measurement results are summarized in Table 5.6. From the obtained results, it can be noted that classification accuracies comparable with those for the full-feature cases were achieved despite large reductions in the number of features. This confirms the ability of the proposed method to choose informative features. As shown in Table 5.6, this study has empirically revealed that the effect of applying the proposed extension and its combination on numerical dataset significantly improves the time and increase the accuracy of BAFS.

# CHAPTER SIX
# CONCLUSION AND FUTURE WORK

## 6.1 Introduction

Optimization has an important role in the data analysis process in machine learning and data mining. The study has described and explained how the BA, an optimization tool, was used for numerical datasets fields. In summary, the BA has been studied to find the subset of the best features in a FS problem. The main disadvantage of the BA; its long computational time in finding the optimum solution also has been addressed. This chapter summarizes the conclusions of this study. It also provides suggestions for further research.

## 6.2 Conclusions

A swarm-based intelligent optimisation procedure called the Bees Algorithm (BA) is presented in this study. The algorithm mimics the food foraging behaviour of swarms of honey bees. In its basic version, the algorithm performs a kind of neighbourhood search combined with random search and is able to find promising solutions to complex optimization problems. The purpose of neighbourhood search is to intensify the search effort around promising solutions.  Also given are further details of the local and global search methods used in the algorithm. Details of the improvements made to local search methods are presented in this study, by proposing extension methods and combining among all operators. The performance of the algorithm is evaluated on benchmark results, comparing them to those results achieved by other well-known algorithms in the literature [2].

- BA has been employed for FS to obtain the most relevant subset of features from a dataset. FS is needed to reduce processing and computational time compared to the case with all features included in the data pre-processing phase. Fewer features to be evaluated means the use of fewer resources. In this study, a wrapper technique for FS was implemented using the BA. The wrapper FS approaches include three popular strategies: a) forward selection, b) backward elimination, and c) stochastic search. The current study was choosing "forward selection" to work with BAFS (**Objective 1**). The main job for Forward selection is evaluated from no features until all features have been considered. Furthermore, A number of numerical datasets were tested to evaluate the effectiveness of the proposed algorithm.

- An enhancements to neighbourhood search and parameter numbers are represented in the BAFS. An extension operator 4-Opt and combination methods are introduced to reduce time consuming and increase the accuracy (**Objective 2**). From the overall measurement results obtained, it can be noted that classification accuracies and time execution comparable with those for the full-feature cases were achieved despite large reductions in the number of features. This confirms the ability of the proposed method to choose informative features. This study has empirically shown the effect of applying proposed extension and their combination of numerical dataset.

- In this study, the validation of the result depends on two metrics: speed and accuracy. This determined by comparing the result was obtain between original methods from Massudi methods and the extension proposed of swapping mechanism of BAFS. Finally, the experimental results obtained confirmed that the proposed extension of the search neighbourhood that include 4-Opt and their combination approaches have provided better prediction accuracy with suitable time than the Massudi methods (**Objective 1**). Finally, the results achieved in this study can make these approaches be successfully used in the local search mechanism.

## 6.3 Future Work

There is a potential future research arising from this study. Therefore, a number of issues can be investigated in order to improve the BAFS, including: firstly, BA is a simple optimization approach which still needs a mathematical proof to show that it is an effective optimization algorithm. Additional work is also required to differentiate the algorithm clearly from other nature-inspired optimization techniques such as Particle Swarm Optimization; besides the main advantages of hybrid algorithms to benefit from each other's strength. Secondly, further study is needed to reduce the number of parameters of the BA. At present, the user has to set all the parameters manually to obtain the best result, which is a problem for novice users. Reducing the number of parameters without reducing the performance of the BA is a research challenge for the future. Thirdly, the current algorithm has only been tested on numerical data including real, continuous and integer datasets. Further research is needed to validate the algorithm on other data types, such as mixed datasets and categorical or multi-class data types. Fourthly, a new local search algorithm must be developed for combinatorial domains to increase the efficiency of the BA. Finally, most studies in the BA were carried out to improve the neighbourhood search stage (local search). The future research studies on the BA can focus on the global search process stage using both the original and improved versions for several different industrial applications to extend the scope of the algorithm.

# REFERENCES


[1] B. Yuce, D. T. Pham, M. S. Packianather, and E. Mastrocinque, “An enhancement to the Bees Algorithm with slope angle computation and Hill Climbing Algorithm and its applications on scheduling and continuous-type optimisation problem,” *Prod. Manuf. Res.*, vol. 3, no. 1, pp. 3–19, 2015.

[2] M. Mahmuddin, “Optimisation Using Bees Algorithm on Unlabelled Data Problem,” Cardiff University, United Kingdom, 2009.

[3] K. I. Rufai, R. C. Muniyandi, and Z. A. Othman, “Improving Bee Algorithm Based Feature Selection in Intrusion Detection System Using Membrane Computing,” *J. Networks*, vol. 9, no. 3, pp. 523–529, 2014.

[4] N. M. Hammash, “Performance Comparison of Parallel Bees Algorithm on Rosenbrock Function,” Universiti Utara Malaysia, 2012.

[5] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, “The bees algorithm-a novel tool for complex optimisation problems,” in *Proceedings of the 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006)*, 2006, pp. 454–459.

[6] A. Colorni, M. Dorigo, and V. Maniezzo, “Distributed optimization by ant colonies,” in *Proceedings of the first European conference on artificial life*, 1991, vol. 142, pp. 134–142.

[7] X.-S. Yang, “A new metaheuristic bat-inspired algorithm,” in *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, 2010, pp. 65–74.

[8] J. Kennedy, “Particle swarm optimization,” in *Encyclopedia of Machine Learning*, Springer, 2010, pp. 760–766.

[9] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[10] X.-S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 210–214.

[11] I. R. M. Association, *Bioinformatics: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2013.

[12] A. Alazzam and H. W. Lewis III, "A New Optimization Algorithm For Combinatorial Problems," *Int. J. Adv. Res. Artif. Intell.*, vol. 2, no. 5, 2013.

[13] J. F. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. Mcarthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz, "The Expanding Digital Universe: A Forecast of Worldwide Information Growth through 2010, IDC Whitepaper," *MA IDC*, 2007.

[14] J. W. Herrmann, "A genetic algorithm for minimax optimization problems," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, 1999, vol. 2, pp. 21–30.

[15] redcedartech, "How to Select the Right Optimization Method for Your Problem." [Online]. Available: http://www.redcedartech.com/pdfs/Select_Optimization_Method.pdf. [Accessed: 10-Oct-2015].

[16] M. R. Hossain, A. M. T. Oo, and A. B. M. Ali, "The Effectiveness of Feature Selection Method in Solar Power Prediction," *J. Renew. Energy*, p. 9, 2013.

[17] J. F. Gantz, "The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things," *EMC Corporation*, 2015. [Online]. Available: Retrieved From.

[18] D. T. Pham, M. Mahmuddin, S. Otri, and H. Al-Jabbouli, "Application of the bees algorithm to the selection features for manufacturing data," *3rd Int. Virtual Conf. Intell. Prod. Mach. Syst.*, 2007.

[19] I. Cheng, "Hybrid Methods for Feature Selection," Western Kentucky University, 2013.

[20] K. I. Rufai, R. C. Muniyandi, and Z. a. Othman, "Improving Bee Algorithm Based Feature Selection in Intrusion Detection System Using Membrane Computing," *J. Networks*, vol. 9, no. 3, pp. 523–529, Mar. 2014.

[21] M. S. IP.Manimaran, "Feature Subset Selection Algorithm for Large Volumes of Data Based on Clustering," *Int. J. Adv. Res. Comput. Sci. Technol.*, vol. 1, no. 2, 2014.

[22] M. S. Packianather and B. Kapoor, "A wrapper-based feature selection approach using Bees Algorithm for a wood defect classification system," in *System of Systems Engineering Conference (SoSE), 2015 10th*, 2015, pp. 498–503.

[23] M. Mahmuddin and Y. Yusof, "A Hybrid Simplex Search and Bio-Inspired Algorithm for Faster Convergence," in *International Conference on Machine Learning and*

*Computing, IACSIT Press, Singapore*, 2009.

[24] I. Mandli and M. Panchal, “Selection of Most Relevant Features from High Dimensional Data using IG-GA Hybrid Approach,” vol. 3, no. 2, pp. 827–830, 2014.

[25] S. Alelyani, “On Feature Selection Stability: A Data Perspective,” Citeseer, Arizona State University, USA, 2013.

[26] P. Gupta, S. Jain, and A. Jain, “A Review Of Fast Clustering-Based Feature Subset Selection Algorithm,” *Int. J. Sci. Technol. Res.*, vol. 3, no. 11, pp. 20–26, 2014.

[27] M. N. S. Sherin Mary Varghese1, “Efficient Feature Subset Selection Techniques for High Dimensional Data,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 3, 2014.

[28] Q. Song, J. Ni, and G. Wang, “A fast clustering-based feature subset selection algorithm for high-dimensional data,” *Knowl. Data Eng. IEEE Trans.*, vol. 25, no. 1, pp. 1–14, 2013.

[29] S. Dreyer, “Evolutionary Feature Selection,” Norwegian University of Science and Technology, Norwegian, 2013.

[30] L. Özbakir, A. Baykasoğlu, and P. Tapkan, “Bees algorithm for generalized assignment problem,” *Appl. Math. Comput.*, vol. 215, no. 11, pp. 3782–3795, 2010.

[31] S. Kashef and H. Nezamabadi-pour, “An advanced ACO algorithm for feature subset selection,” *Neurocomputing*, 2014.

[32] Y. A. Alsultanny, “Database Preprocessing and Comparison between Data Mining Methods,” *Int. J. New Comput. Archit. their Appl.*, vol. 1, no. 1, pp. 61–73, 2011.

[33] W. Y. Tang, “Feature selection algorithms for very high dimensional data and mixed data,” School of Electrical and Electronic Engineering, singapore, 2008.

[34] S. Abraham and I. Simi, “An Efficient CMST-Fuzzy Based Feature Selection Algorithm with Feature Interaction for High-Dimensional Data,” *Int. J. Sci. Res. Educ.*, vol. 2, no. 07, 2014.

[35] R. Tiwari and M. P. Singh, “Correlation-based attribute selection using genetic algorithm,” *Int. J. Comput. Appl.*, vol. 4, no. 8, 2010.

[36] V. Kumar and S. Minz, “Feature Selection,” *SmartCR*, vol. 4, no. 3, pp. 211–229, 2014.

[37] K. Sigmund, C. Hauert, and M. A. Nowak, “Reward and punishment,” *Proc. Natl. Acad.*

*Sci.*, vol. 98, no. 19, pp. 10757–10762, 2001.

[38] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, pp. 237–285, 1996.

[39] A. George, "Analysis and Implementation of Lambda Opt Heuristic Framework for Combinatorial Optimization," MSc in Artificial Intelligence, 2011.

[40] D. T. Pham, E. Koc, J. Y. Lee, and J. Phrueksanant, "Using the bees algorithm to schedule jobs for a machine," in *Proceedings Eighth International Conference on Laser Metrology, CMM and Machine Tool Performance, LAMDAMAP, Euspen, UK, Cardiff*, 2007, pp. 430–439.

[41] D. T. Pham, E. Koç, A. Ghanbarzadeh, and S. Otri, "Optimisation of the weights of multi-layered perceptrons using the bees algorithm," in *Proceedings of 5th international symposium on intelligent manufacturing systems*, 2006, pp. 38–46.

[42] K. P. Bennett and E. Parrado-Hernández, "The interplay of optimization and machine learning research," *J. Mach. Learn. Res.*, vol. 7, pp. 1265–1281, 2006.

[43] T. Eitrich and B. Lang, "Efficient optimization of support vector machine learning parameters for unbalanced datasets," *J. Comput. Appl. Math.*, vol. 196, no. 2, pp. 425–436, 2006.



[44] D. T. Pham, S. Otri, A. Ghanbarzadeh, and E. Koc, "Application of the bees algorithm to the training of learning vector quantisation networks for control chart pattern recognition," in *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, 2006, vol. 1, pp. 1624–1629.

[45] D. T. Pham, A. Ghanbarzadeh, E. Koc, and S. Otri, "Application of the bees algorithm to the training of radial basis function networks for control chart pattern recognition," in *Proceedings of 5th CIRP international seminar on intelligent computation in manufacturing engineering (CIRP ICME'06), Ischia, Italy*, 2006, pp. 711–716.

[46] X.-S. Yang, "Metaheuristic Optimization," *Scholarpedia*, 08-Aug-2011. [Online]. Available: http://www.scholarpedia.org/article/Metaheuristic_Optimization. [Accessed: 07-Apr-2015].

[47] A. Kaveh and A. Nasrollahi, "Charged system search and particle swarm optimization hybridized for optimal design of engineering structures," *Sci. Iran.*, vol. 21, no. 2, pp.

295–305, 2014.

[48] T. Bartz, Beielstein, J. Branke, J. Mehnen, and O. Mersmann, “Evolutionary Algorithms,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 4, no. 3, pp. 178–195, 2014.

[49] X.-S. Yang, *Nature-Inspired Optimization Algorithms*, 2014, Firs. Elsevier, 2014.

[50] O. Olofintoye, J. Adeyemo, and F. Otieno, “Evolutionary Algorithms and Water Resources Optimization,” in *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, Springer, 2013, pp. 491–506.

[51] G. Fogel, D. Fogel, and L. Fogel, “Evolutionary programming,” *Scholarpedia*, 10-Apr-2011. [Online]. Available: http://www.scholarpedia.org/article/Evolutionary_programming. [Accessed: 07-Apr-2015].

[52] G. K. Brar and A. K. Virk, “Deployment of nodes for Maximum Coverage in Heterogeneous Wireless Sensor Network Using Genetic Algorithm,” *Int. J.*, vol. 2, no. 6, 2014.

[53] S. Lloyd, “The universe as quantum computer,” *arXiv Prepr. arXiv1312.4455*, 2013.

[54] F. Glover, M. Laguna, and R. Martí, “Fundamentals of scatter search and path relinking,” *Control Cybern.*, vol. 39, no. 3, pp. 653–684, 2000.

[55] T. Chengyi, “Gravitational Search Algorithm Based on Simulated Annealing,” *Int. Assoc. Information, Cult. Hum. Ind. Technol.*, vol. 9, no. 2, 2014.

[56] S. C. Varma, K. S. L. Murthy, and K. SriChandan, “Gaussian particle swarm optimization for combined economic emission dispatch,” in *Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference on*, 2013, pp. 136–140.

[57] P. Wang, T. Weise, and R. Chiong, “Novel evolutionary algorithms for supervised classification problems: an experimental study,” *Evol. Intell.*, vol. 4, no. 1, pp. 3–16, 2011.

[58] B. Xing and W.-J. Gao, *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*. Springer, 2014.

[59] K.-L. Du and M. N. S. Swamy, *Neural networks in a softcomputing framework*. Springer,

2006.

[60] V. Kumar, “Feature Selection: A literature Review,” *Smart Comput. Rev.*, vol. 4, p. 3, 2014.

[61] M. Shokouhifar and S. Sabet, “A hybrid approach for effective feature selection using neural networks and artificial bee colony optimization,” in *3rd international conference on machine vision (ICMV 2010)*, 2010, pp. 502–506.

[62] E. M. Karabulut, S. A. Özel, and T. İbrikçi, “A comparative study on the effect of feature selection on classification accuracy,” *Procedia Technol.*, vol. 1, pp. 323–327, 2012.

[63] M. Kabir, M. Shahjahan, and K. Murase, *Ant Colony Optimization Toward Feature Selection*, 1st ed. InTech, 2013.

[64] P.-E. Jouve and N. Nicoloyannis, “A filter feature selection method for clustering,” in *Foundations of Intelligent Systems*, Springer, 2005, pp. 583–593.

[65] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Data Preprocessing for Supervised Leaning,” vol. 1, no. 2, pp. 111–117, 2006.

[66] L. Yu and H. Liu, “Feature Selection for High-Dimensional Data : A Fast Correlation-Based Filter Solution,” 2003.

[67] M. Kudo and J. Sklansky, “Comparison of algorithms that select features for pattern classifiers,” *Pattern Recognit.*, vol. 33, no. 1, pp. 25–41, 2000.

[68] M. Bressan and J. Vitria, “On the selection and classification of independent features,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 25, no. 10, pp. 1312–1317, 2003.

[69] S. J. Raudys and A. K. Jain, “Small sample size effects in statistical pattern recognition: Recommendations for practitioners,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 252–264, 1991.

[70] M. A. Hall and G. Holmes, “Benchmarking attribute selection techniques for discrete class data mining,” *Knowl. Data Eng. IEEE Trans.*, vol. 15, no. 6, pp. 1437–1447, 2003.

[71] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, “Improved binary PSO for feature selection using gene expression data,” *Comput. Biol. Chem.*, vol. 32, no. 1, pp. 29–38, 2008.

[72] A. L. Blum and R. L. Rivest, “Training a 3-node neural network is NP-complete,” in

*Machine learning: From theory to applications*, Springer, 1993, pp. 9–28.

[73] H. Zhang and G. Sun, "Feature selection using tabu search method," *Pattern Recognit.*, vol. 35, no. 3, pp. 701–711, 2002.

[74] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1, pp. 273–324, 1997.

[75] C. Sima and E. R. Dougherty, "What should be expected from feature selection in small-sample settings," *Bioinformatics*, vol. 22, no. 19, pp. 2430–2436, 2006.

[76] M. S. Packianather and P. R. Drake, "Neural networks for classifying images of wood veneer. Part 2," *Int. J. Adv. Manuf. Technol.*, vol. 16, no. 6, pp. 424–433, 2000.

[77] J. Fan and R. Li, "Statistical challenges with high dimensionality: Feature selection in knowledge discovery," *arXiv Prepr. math/0602133*, 2006.

[78] M. Boulle, "Khiops: A statistical discretization method of continuous attributes," *Mach. Learn.*, vol. 55, no. 1, pp. 53–69, 2004.

[79] W. O. Saxton, Tj. Pitt, and M. Horner, "Digital image processing: the Semper system," *Ultramicroscopy*, vol. 4, no. 3, pp. 343–353, 1979.

[80] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[81] S. R. Hasani, Z. A. Othman, S. Mostafa, and M. Kahaki, "HYBRID FEATURE SELECTION ALGORITHM FOR INTRUSION DETECTION SYSTEM," vol. 10, no. 6, pp. 1015–1025, 2014.

[82] L.-X. Zhang, J.-X. Wang, Y.-N. Zhao, and Z.-H. Yang, "A novel hybrid feature selection algorithm: using ReliefF estimation for GA-Wrapper search," in *Machine Learning and Cybernetics, 2003 International Conference on*, 2003, vol. 1, pp. 380–384.

[83] J. Huang, Y. Cai, and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," *Pattern Recognit. Lett.*, vol. 28, no. 13, pp. 1825–1844, 2007.

[84] O. L. Garnes, "Feature Selection for Text Categorisation," Institutt for datateknikk og informasjonsvitenskap, 2009.

[85] S. NAQVI, "A Hybrid Filter-Wrapper Approach for FeatureSelection," Örebro

University, 2011.

[86] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[87] P. Shunmugapriya, S. Kanmani, R. Supraja, and K. Saranya, "Feature selection optimization through enhanced Artificial Bee Colony algorithm," in *Recent Trends in Information Technology (ICRTIT), 2013 International Conference on*, 2013, pp. 56–61.

[88] R. Forsati, A. Moayedikia, A. Keikha, and M. Shamsfard, "A novel approach for feature selection based on the bee colony optimization," *Int. J. Comput. Appl.*, vol. 43, no. 8, pp. 13–16, 2012.

[89] M. S. Srivastava, M. N. Joshi, and M. Gaur, "A Review Paper on Feature Selection Methodologies and Their Applications," *IJCSNS*, vol. 14, no. 5, p. 78, 2014.

[90] S. Sakthivel, S. A. Pandiyan, S. Marikani, and S. K. Selvi, "Application of Big Bang Big Crunch Algorithm for Optimal Power Flow Problems," *Int J Eng Sci*, vol. 2, no. 4, pp. 41–47, 2013.

[91] H. M. Genc and A. K. Hocaoglu, "Bearing-only target tracking based on big bang–big crunch algorithm," in *Computing in the Global Information Technology, 2008. ICCGI'08. The Third International Multi-Conference on*, 2008, pp. 229–233.



[92] Y.-J. Zheng, S.-Y. Chen, Y. Lin, and W.-L. Wang, "Bio-inspired optimization of sustainable energy systems: a review," *Math. Probl. Eng.*, vol. 2013, 2013.

[93] D. Rai and K. Tyagi, "Bio-inspired optimization techniques: a critical comparative study," *ACM SIGSOFT Softw. Eng. Notes*, vol. 38, no. 4, pp. 1–7, 2013.

[94] S. Binitha and S. S. Sathya, "A survey of bio inspired optimization algorithms," *Int. J. Soft Comput. Eng.*, vol. 2, no. 2, pp. 137–151, 2012.

[95] A. Imanguliyev, "Enhancements for the Bees Algorithm." Cardiff University, 2013.

[96] J. Brownlee, *Clever algorithms: nature-inspired programming recipes*, 1st ed. Australia: Jason Brownlee, 2012.

[97] E. Gerhardt and H. M. Gomes, "Artificial bee colony (ABC) algorithm for engineering optimization problems," *EngOpt*, 2012.

[98] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm,"

*Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, 2009.

[99] A. Baykasoglu, L. Ozbakir, and P. Tapkan, “Artificial bee colony algorithm and its application to generalized assignment problem,” *Swarm Intell. Focus Ant Part. swarm Optim.*, pp. 113–144, 2007.

[100] M. V. S. Prasad, C. H. S. Kumar, and T. Maneesha, “Feature Selection Using An Effective Dimensionality Reduction Technique,” vol. 3, no. 5, pp. 480 – 485, 2014.

[101] Y.-C. Wu, W.-P. Lee, and C.-W. Chien, “Modified the performance of differential evolution algorithm with dual evolution strategy,” in *2009 International Conference on Machine Learning and Computing, IPCSIT*, 2011, vol. 3, pp. 57–63.

[102] M. Mahdiani, H. M. Judi, and N. S. Ashaari, “Selection of Shape Features in Control Chart Pattern Classification Using Bees Algorithm.,” *Int. J. Inf. Process. Manag.*, vol. 5, no. 3, 2014.

[103] M. Beekman and F. L. W. Ratnieks, “Long range foraging by the honey bee, Apis mellifera L.,” *Funct. Ecol.*, vol. 14, no. 4, pp. 490–496, 2000.

[104] D. T. Pham and A. Ghanbarzadeh, “Multi-objective optimisation using the bees algorithm,” in *3rd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2007): Whittles, Dunbeath, Scotland*, 2007, vol. 242, pp. 111–116.



[105] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, “The bees algorithm. Technical note,” *Manuf. Eng. Centre, Cardiff Univ. UK*, pp. 1–57, 2005.

[106] D. T. Pham, S. Otri, A. Afify, M. Mahmuddin, and H. Al-Jabbouli, “Data clustering using the bees algorithm,” Liverpool, UK., 2007.

[107] D. T. Pham and M. Castellani, “The bees algorithm: modelling foraging behaviour to solve continuous optimization problems,” *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 223, no. 12, pp. 2919–2938, 2009.

[108] J. Phrueksanant, “Machine scheduling using the Bees Algorithm.” Cardiff University, 2013.

[109] H. A. Le Thi, H. M. Le, and T. P. Dinh, “Feature selection in machine learning: an exact penalty approach using a Difference of Convex function Algorithm,” *Mach. Learn.*, pp. 1–24, 2014.

[110] M. S. K. K. Lenin, B. Ravindranath Reddy, "Dwindling of real power loss by using Improved Bees Algorithm," *Int. J. Recent Res. Electr. Electron. Eng.*, vol. 1, no. 1, pp. (34–42), 2014.

[111] E. Mastrocinque, B. Yuce, A. Lambiase, and M. S. Packianather, "A multi-objective optimisation for supply chain network using the Bees Algorithm," *Int. J. Eng. Bus. Manag.*, vol. 5, pp. 1–11, 2013.

[112] H. F. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth, R. Jeruschkat, M. Duhm, L. Bensmann, and G. Kathagen, "T. B" ning. BeeHive–An Energy-Aware Scheduling and Routing Framework," Technical report-pg439, LSIII, School of Computer Science, University of Dortmund, 2004.

[113] R. Martí and G. Reinelt, "Heuristic Methods," in *The Linear Ordering Problem*, Springer, 2011, pp. 17–40.

[114] M. Gruber, "Exact and heuristic approaches for solving the bounded diameter minimum spanning tree problem," Vienna, Austria, 2009.

[115] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4135–4151, 2011.

[116] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003.

[117] J. Fosin, D. Davidović, and T. Carić, "A gpu implementation of local search operators for symmetric travelling salesman problem," *PROMET-Traffic&Transportation*, vol. 25, no. 3, pp. 225–234, 2013.

[118] G. A. Croes, "A method for solving traveling-salesman problems," *Oper. Res.*, vol. 6, no. 6, pp. 791–812, 1958.

[119] M. M. Flood, "The traveling-salesman problem," *Oper. Res.*, vol. 4, no. 1, pp. 61–75, 1956.

[120] D. S. Johnson and L. A. McGeoch, "The traveling salesman problem: A case study in local optimization," *Local search Comb. Optim.*, vol. 1, pp. 215–310, 1997.

[121] F. Bock, "An algorithm for solving'traveling salesman'and related network optimization problems, presented at the 14th National Meeting of the Operations Research Society of America, St," *Louis, Missouri*, 1958.

[122] S. Lin, “Computer solutions of the traveling salesman problem,” *Bell Syst. Tech. Journal,* vol. 44, no. 10, pp. 2245–2269, 1965.

[123] C. Walshaw, “A Multilevel Approach to the Travelling Salesman Problem,” INFORMS, 2002.

[124] M. Kordos, “Search-based algorithms for multilayer perceptrons.” The Silesian University of Technology, 2005.

[125] E. Romero and J. M. Sopena, “Performing feature selection with multilayer perceptrons,” *Neural Networks, IEEE Trans.*, vol. 19, no. 3, pp. 431–441, 2008.

[126] G. Kumar and K. Kumar, “A Multi-objective Genetic Algorithm Based Approach for Effective Intrusion Detection Using Neural Networks,” in *Intelligent Methods for Cyber Warfare*, Springer, 2015, pp. 173–200.

[127] S. Ravuri and A. Stolcke, “Neural Network Models for Lexical Addressee Detection,” in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[128] Z. N. S. Vanini, K. Khorasani, and N. Meskin, “Fault detection and isolation of a dual spool gas turbine engine using dynamic neural networks and multiple model approach,” *Inf. Sci. (Ny).*, vol. 259, pp. 234–251, 2014.

[129] E. Gasca, J. S. Sánchez, and R. Alonso, “Eliminating redundancy and irrelevance using a new MLP-based feature selection method,” *Pattern Recognit.*, vol. 39, no. 2, pp. 313–315, 2006.

[130] A. Verikas and M. Bacauskiene, “Feature selection with neural networks,” *Pattern Recognit. Lett.*, vol. 23, no. 11, pp. 1323–1335, 2002.

[131] G. Castellano and A. M. Fanelli, “Variable selection using neural-network models,” *Neurocomputing,* vol. 31, no. 1, pp. 1–13, 2000.

[132] M. M. Kabir, M. M. Islam, and K. Murase, “A new wrapper feature selection approach using neural network,” *Neurocomputing*, vol. 73, no. 16, pp. 3273–3283, 2010.

[133] R. Chakraborty and N. R. Pal, “Feature selection using a neural framework with controlled redundancy,” 2014.

[134] C. Chang, P. A. Verhaegen, and J. R. Duflou, “A Comparison of Classifiers for Intelligent Machine Usage Prediction,” in *Intelligent Environments (IE), 2014*

*International Conference on*, 2014, pp. 198–201.

[135] Z. Yan, Z. Wang, and H. Xie, "The application of mutual information-based feature selection and fuzzy LS-SVM-based classifier in motion classification," *Comput. Methods Programs Biomed.*, vol. 90, no. 3, pp. 275–284, 2008.

[136] G. Kim, Y. Kim, H. Lim, and H. Kim, "An MLP-based feature subset selection for HIV-1 protease cleavage site analysis," *Artif. Intell. Med.*, vol. 48, no. 2, pp. 83–89, 2010.

[137] R. Setiono and H. Liu, "Neural-network feature selector," *Neural Networks, IEEE Trans.*, vol. 8, no. 3, pp. 654–662, 1997.

[138] C.-N. Hsu, H.-J. Huang, and S. Dietrich, "The ANNIGMA-wrapper approach to fast feature selection for neural nets," *Syst. Man, Cybern. Part B Cybern. IEEE Trans.*, vol. 32, no. 2, pp. 207–212, 2002.

[139] V. Sindhwani, S. Rakshit, D. Deodhare, D. Erdogmus, J. C. Principe, and P. Niyogi, "Feature selection in MLPs and SVMs based on maximum output information," *Neural Networks, IEEE Trans.*, vol. 15, no. 4, pp. 937–948, 2004.

[140] D. M. Santoro, E. R. Hruschska Jr, and M. do Carmo Nicoletti, "Selecting feature subsets for inducing classifiers using a committee of heterogeneous methods," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, 2005, vol. 1, pp. 375–380.

[141] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell.*, vol. 97, no. 1, pp. 245–271, 1997.

[142] S. Piramuthu, "Evaluating feature selection methods for learning in data mining applications," *Eur. J. Oper. Res.*, vol. 156, no. 2, pp. 483–494, 2004.

[143] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *Comput. IEEE Trans.*, vol. 100, no. 9, pp. 917–922, 1977.

[144] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 19, no. 2, pp. 153–158, 1997.

[145] H. Vafaie and K. De Jong, "Genetic algorithms as a tool for feature selection in machine learning," in *Tools with Artificial Intelligence, 1992. TAI'92, Proceedings., Fourth International Conference on*, 1992, pp. 200–203.

[146] J. Feng, Y. Yang, H. Wang, and X.-M. Wang, "Feature selection based on genetic

algorithms and support vector machines for handwritten similar Chinese characters recognition," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, 2004, vol. 6, pp. 3600–3605.

[147] H. Handels, T. Roß, J. Kreusch, H. H. Wolff, and S. J. Poeppl, "Feature selection for optimized skin tumor recognition using genetic algorithms," *Artif. Intell. Med.*, vol. 16, no. 3, pp. 283–297, 1999.

[148] D. J. Skarzynski, M. M. Bah, K. M. Deptula, I. Woclawek-Potocka, A. Korzekwa, M. Shibaya, W. Pilawski, and K. Okuda, "Roles of tumor necrosis factor-α of the estrous cycle in cattle: an in vivo study," *Biol. Reprod.*, vol. 69, no. 6, pp. 1907–1913, 2003.

[149] P. L. Lanzi, "Fast feature selection with genetic algorithms: a filter approach," in *Evolutionary Computation, 1997., IEEE International Conference on*, 1997, pp. 537–540.

[150] S. F. Da Silva, M. X. Ribeiro, J. D. E. S. B. Neto, C. Traina-Jr, and A. J. M. Traina, "Improving the ranking quality of medical image retrieval using a genetic feature selection method," *Decis. Support Syst.*, vol. 51, no. 4, pp. 810–820, 2011.

[151] W. F. Punch III, E. D. Goodman, M. Pei, L. Chia-Shun, P. D. Hovland, and R. J. Enbody, "Further Research on Feature Selection and Classification Using Genetic Algorithms.," in *ICGA*, 1993, pp. 557–564.

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

[153] P. Zhang, B. Verma, and K. Kumar, "Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection," *Pattern Recognit. Lett.*, vol. 26, no. 7, pp. 909–919, 2005.

[154] D. P. Muni, N. R. Pal, and J. Das, "Genetic programming for simultaneous feature selection and classifier design," *Syst. Man, Cybern. Part B Cybern. IEEE Trans.*, vol. 36, no. 1, pp. 106–117, 2006.

[155] J. Yu, J. Yu, A. A. Almal, S. M. Dhanasekaran, D. Ghosh, W. P. Worzel, and A. M. Chinnaiyan, "Feature selection and molecular classification of cancer using genetic programming," *Neoplasia*, vol. 9, no. 4, pp. 292–IN3, 2007.

[156] R. J. Nandi, A. K. Nandi, R. M. Rangayyan, and D. Scutt, "Classification of breast

masses in mammograms using genetic programming and feature selection," *Med. Biol. Eng. Comput.*, vol. 44, no. 8, pp. 683–694, 2006.

[157] A. Seal, S. Ganguly, D. Bhattacharjee, M. Nasipuri, and C. Gonzalo-Martin, "Feature Selection using Particle Swarm Optimization for Thermal Face Recognition," in *Applied Computation and Security Systems*, Springer, 2015, pp. 25–35.

[158] H. B. Nguyen, B. Xue, I. Liu, and M. Zhang, "Filter based backward elimination in wrapper based PSO for feature selection in classification," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, 2014, pp. 3111–3118.

[159] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Syst.*, vol. 64, pp. 22–31, 2014.

[160] C.-J. Tu, L.-Y. Chuang, J.-Y. Chang, and C.-H. Yang, "Feature selection using PSO-SVM," *IAENG Int. J. Comput. Sci.*, vol. 33, no. 1, pp. 111–116, 2007.

[161] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 459–471, 2007.

[162] H. A. Firpi and E. D. Goodman, "Swarmed Feature Selection.," in *AIPR*, 2004, pp. 112–118.

[163] M. Karnan, K. Thangavel, R. Sivakuar, and K. Geetha, "Ant colony optimization for feature selection and classification of microcalcifications in digital mammograms," in *Advanced Computing and Communications, 2006. ADCOM 2006. International Conference on*, 2006, pp. 298–303.

[164] R. Bello, A. Puris, A. Nowe, Y. Martínez, and M. M. Garcia, "Two step ant colony system to solve the feature selection problem," in *Progress in Pattern Recognition, Image Analysis and Applications*, Springer, 2006, pp. 588–596.

[165] Y. H. Gai and G. Yu, "A Hybrid Feature Selection Based on Ant Colony Optimization and Probabilistic Neural Networks for Bearing Fault Diagnostics," *Appl. Mech. Mater.*, vol. 10, pp. 573–577, 2008.

[166] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biol.*, vol. 52, no. 1–2, pp. 99–115, 1990.

[167] M. S. Uzer, N. Yilmaz, and O. Inan, "Feature selection method based on artificial bee colony algorithm and support vector machines for medical datasets classification," *Sci. World J.*, vol. 2013, 2013.

[168] "UCI Machine Learning Repository: Data Sets," *UCI*, 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets.html. [Accessed: 10-Oct-2014].

[169] Y.-S. Hwang, "Wrapper-based Feature Selection Using Support Vector Machine," *Life Sci. J.*, vol. 11, no. 7, 2014.

[170] "UCI Machine Learning Repository: Heart Disease Data Set," *UCI*, 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Heart+Disease. [Accessed: 10-Oct-2014].

[171] "UCI Machine Learning Repository: Pima Indians Diabetes Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes. [Accessed: 24-Nov-2015].

[172] "UCI Machine Learning Repository: SPECT Heart Data Set," *University of California, Irvine*, 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SPECT+Heart. [Accessed: 19-Apr-2015].

[173] "UCI Machine Learning Repository: Hepatitis Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Hepatitis. [Accessed: 24-Nov-2015].

[174] "UCI Machine Learning Repository: Soybean (Small) Data Set," *University of California, Irvine,* 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Soybean+%28Small%29. [Accessed: 19-Apr-2015].

[175] "UCI Machine Learning Repository: Lung Cancer Data Set," *University of California, Irvine*, 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Lung+Cancer. [Accessed: 19-Apr-2015].

[176] R. S. Michalski and R. L. Chilausky, "Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis," *Int. J. Policy Anal. Inf. Syst.*, vol. 4, no. 2, pp. 125–161, 1980.

[177] L. A. Kurgan, K. J. Cios, R. Tadeusiewicz, M. Ogiela, and L. S. Goodenday,

"Knowledge discovery approach to automated cardiac SPECT diagnosis," *Artif. Intell. Med.*, vol. 23, no. 2, pp. 149–169, 2001.

[178] Z.-Q. Hong and J.-Y. Yang, "Optimal discriminant plane for a small number of samples and design method of classifier on the plane," *pattern Recognit.*, vol. 24, no. 4, pp. 317–324, 1991.

[179] A. Blazinskas and A. Misevicius, "COMBINING 2-OPT, 3-OPT AND 4-OPT WITH K-SWAP-KICK PERTURBATIONS FOR THE TRAVELING SALESMAN PROBLEM," *Kaunas Univ. Technol. Dep. Multimed. Eng. Studentu St*, pp. 50–401, 2009.

[180] D. Bonachea, E. Ingerman, J. Levy, and S. McPeak, "An Improved Adaptive Multi-Start Approach to Finding Near-Optimal Solutions to the Euclidean TSP.," in *GECCO*, 2000, pp. 143–150.

[181] E. Koc, "Bees Algorithm: theory, improvements and applications." Cardiff University, 2010.

[182] S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem," *Oper. Res.*, vol. 21, no. 2, pp. 498–516, Apr. 1973.

[183] F. Glover, "Finding a best traveling salesman 4-opt move in the same time as a best 2-opt move," *J. Heuristics*, vol. 2, no. 2, pp. 169–179, 1996.