A HYBRID ARTIFICIAL BEE COLONY ALGORITHM FOR THE MINIMUM DOMINATING SET PROBLEM

LINA TALIB ABDULAMEER

UNIVERSITI KEBANGSAAN MALAYSIA

A HYBRID ARTIFICIAL BEE COLONY ALGORITHM FOR THE MINIMUM DOMINATING SET PROBLEM

LINA TALIB ABDULAMEER

PROJECT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE (ARTIFICIAL INTELLIGENCE)

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI
2022

ALGORITMA KOLONI LEBAH TIRUAN HIBRID UNTUK MASALAH SET PENGUASAAN MINIMUM

LINA TALIB ABDULAMEER

PROJEK DIKEMUKAKAN UNTUK MEMENUHI SEBAHAGIAN DARIPADA SYARAT MEMPEROLEHI IJAZAH SARJANA SAINS KOMPUTER (KECERDASAN BUATAN)

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI
2022

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

12 February 2022 LINA TALIB ABDULAMEER
P103913

ACKNOWLEDGEMENT

First and foremost, praise be to Almighty Allah for all his blessings for giving me patience and good health throughout this master’s journey. Secondly, I am grateful to Malaysia, the great country, for giving me the opportunity to study under its umbrella. Also, I am grateful to my great country, Iraq, my office (Karbala University) for giving me the financial, logistic support. Moreover, I would like to express my high appreciation to my supervisor, Associate prof. Dr. Ravie Chandren Muniyandi for his supervision, who, without his guidance, I would not be able to produce the thesis in this way.

A word of thank is given to all my professors at FTSM, who I took courses under them. Thank you for helping me to gain knowledge.

My family in Iraq my brothers (Wissam and Husham), I thank you for your continuous encouragement until I reached this level of education. To all those for whom I hold feelings of love, respect and appreciation in my heart, thank you.

ABSTRAK

Keupayaan mengawal elemen rangkaian kompleks dengan hanya menggunakan sebilangan kecil nod telah muncul sebagai bidang penyelidikan dan menjadi cabaran yang penting dalam sistem sains rangkaian. Set penguasaan minimum (MDS) adalah topik rangkaian moden yang penting dalam konteks ini. Terdapat beberapa kaedah untuk menyelesaikan masalah MDS telah dibangunkan sejak sedekad yang lalu. Masalah set penguasaan minimum (MDS) ialah masalah pengoptimuman gabungan NP yang ketara dengan pelbagai aplikasi. Kajian ini bertujuan untuk menangani masalah MDS secara efisien dengan mencadangkan penggunaan algoritma Hybrid Binary Artificial Bee Colony (HBABC). Algoritma ABC biasa mengendalikan pemboleh ubah secara berterusan. Oleh itu, kajian ini dijalankan untuk membangunkan ABC binari (BABC) untuk disesuaikan dengan cadangan penyelesaian masalah MDS. BABC yang diwujudkan adalah berdasarkan populasi penyelesaian. Ini bermakna, BABC berupaya untuk meneroka ruang masalah MDS dengan lebih berkesan. Walau bagaimanapun, untuk mengelakkan ketidakseimbangan antara prosedur penerokaan dan eksploitasi, pengkaji mencadangkan BABC hibrid (HBABC) dengan menggabungkan algoritma yang dikenali sebagai Late Acceptance Hill Climbing (LAHC). Satu siri eksperimen telah dijalankan untuk membuktikan keberkesanan prosedur yang telah disepadukan dengan algoritma ABC. Selain itu, pengkaji turut membandingkan HBABC yang dicadangkan dengan algoritma lain dalam kajian tinjauan lepas berdasarkan penanda aras set data.Dua kumpulan set data telah digunakan, yang dikategorikan mengikut pemboleh ubah yang dikenali sebagai julat. Hasil dapatan kajian mendapati bahawa algoritma HBABC mampu mengatasi algoritma lain apabila julat nodnya adalah berbeza. Selain itu, HBABC adalah setanding atau lebih baik berbanding daripada kaedah lain apabila pemboleh ubah julatnya adalah sama untuk rangkaian yang diuji. Oleh itu, kaedah yang dicadangkan mempunyai kesan yang signifikan dalam menentukan MDS.

ABSTRACT

Controlling the elements of complex networks with a small number of nodes has recently emerged as a significant area of research and a significant challenge in network science. The minimum dominating set (MDS) is an essential modern network topic in this context. Several methods for solving the MDS problem have been developed over the last decade. The minimum dominating set (MDS) problem is a significant NP-hard combinatorial optimization problem with a wide range of applications. This study aims to address the MDS efficiently by proposing a hybrid binary artificial bee colony (HBABC) algorithm. Typical ABC algorithm handles continuous variables; thus, this study develops a binary ABC (BABC) to suit the solution representation of the MDS problem. The BABC is based on population of solutions, and thus, it explores the problem space effectively. However, to avoid imbalance between exploration and exploitation procedures, we propose a hybrid BABC (HBABC) by incorporating a local search algorithm called late acceptance hill climbing (LAHC). A series of experiments was conducted to prove the impact of the procedures integrated to the ABC algorithm. Also, we compare the proposed HBABC to other algorithms from the literature based on benchmark datasets. Two groups of datasets were used, which are categorized based on a variable called range. Results show that the HBABC algorithm has outperformed other algorithms when nodes' ranges are different. That is, HBABC gained improved by 78.5% compared to the rested data. Thus, the proposed method has a significant impact on determining theMDS.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABC	Artificial bee colony
ACO	Ant colony optimization algorithm
ACOLS-LS	Ant colony optimization Local Search- Local Search
ACOLS-LS-S	Ant colony optimization Local Search -Local Search-Specific
AMO	Animal Migration Optimization
BA	Bat algorithm
HBA	Hybrid Bat Algorithm
BCO	Bee colony optimization
BSO	Bee swarm optimization
CS	Cuckoo search
DS	Dominating Set
EB	Employed Bees
EHABC	Enhance Hybrid Artificial Bee Colony
FA	Firefly algorithm
HABC	Hybrid Artificial Bee Colony
HBABC	Hybrid Binary Artificial Bee Colony
HBCA	Honey bee colony algorithm
HGA	Hybrid Genetic Algorithm
IQABC	Improved Quick Artificial Bee Colony
KFABC	knowledge fusion Artificial bee colony
LAHC	Late Acceptance Hill-Climbing
MBA	Mine Blast Algorithm
MCDS	Minimum Connected Dominating Set
MSBA	Mutable smart bee algorithm
MWDS	Minimum Weight Dominating Set
OB	Onlooker Bees
PSO	Particle Swarm Optimization

SAMDS	Simulated Annealing MDS
SAMSABC	Surrogate Assisted Multi-Swarm Artificial Bee Colony
SB	Scout Bees
SD	Standard Deviation
UDG	Unit Disc Graphs
VBA	Virtual bee algorithm
VRP	Vehicle Routing Problem
WSN	Wireless Sensor Network
MDS	Minimum Dominating Set

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

In graph theory, the concept of dominance has been extensively studied (Hao et al. 2020). Meanwhile, the number of queens '$n \times n$' chess board required to identify an earlier historical origin of the dominating set was determined. Zhao et al. (2020) defined the dominating set (DS) as a subset of a given graph with connections to all other vertices. The minimum DS (MDS) is the domination number when studied by Padma and Karthica, and it is denoted by $\gamma(G)$ (Padma & Karthica 2020). G is defined as $G =$ a set of vertices (V) and a set of edges (E) (V, E). Adjacencies of a vertex are only dominated by it and vice versa. In other words, If v and u are both in V , and the pair (*v, u)* belongs to E then either one of them is a dominating vertex. That is, $DS \subset V$ and $V/DS = NDS$, which is adjacent to DS. This work addresses the problem of minimising the DS, where ND is the set of non-dominant nodes. A given graph's (G) DS is a subset of vertices connected to all other vertices. The DS with the fewest vertices is chosen as the minimum dominating set (MDS). MDS is a problem of optimization that has applications in network clustering (Oprişa et al. 2018), wireless networking (Priyadarshini & Sivakumar 2018), big data (Subramanyam & Somayajulu 2017), and social networks (Bouamama & Blum 2021). DS Members, for example, serve as cluster heads in a wireless sensor network (WSN), by collecting data and transmitting it to a gateway. By allowing only a small number of nodes with sufficient energy (i.e., the MDS) to access the gateway, this process helps to extend network life.

Exact methods have been used to implement small instances of graphs effectively (Raviv et al. 2020). Because these methods are limited to small instances, approximate methods are used for large instances (Sartori & Buriol 2020). Greedy methods can provide an approximation in a short amount of time. Nevertheless, this approximation degrades for large graph instances especially of irregular structure (S. Wang et al. 2019). Thus, a stochastic search method has been proposed to provide an accurate solution for the MDS in a reasonable amount of time to address this issue.

The current study proposes the artificial bee colony (ABC) to characterize the MDS problem, which is a population-based algorithm (Karaboga et al. 2007). ABC algorithms are capable of effectively exploring the space state of a given problem. However, to balance the search process, it is necessary to intensifies a solution's neighbourhood and the problem space (Abualigah & Diabat 2021). As a result, this research combines ABC with a local search to broaden the search for newly developed solutions. The main focus of this study is to enforce the ABC exploitation process using hybridization with a local search aiming at MDS. This process enables the algorithm to manipulate adjacent DSs which, in principle, have the potential to be the MDS of the graph. Extensive experiments on various graph instances are used to assess the performance of the proposed method. The purpose of these experiments is to demonstrate the performance of local search and how it compares to greedy methods.

1.2 BACKGROUND

The minimum dominating set problem has recently received attention, particularly in industrial applications. Users in social media were modelled as nodes, with edges representing user relationships. Companies typically monitor their customers' activities, such as conversation and interaction. The company's ability to monitor all customers is hampered by a large number of users (Dijkmans et al. 2020). A possible solution is to create a subset, such as a typical dominating set, to carry out this task. However, in the case of social network scale, building a dominating set is still prohibitively expensive due to the possibility of a sizeable dominating set. As a result, we must consider the k-dominating set D_k such as either each vertex belongs to D_k or to be at least connected to one member of $D_k$via a path that is connected to no more than k edges. When $k = 1$, the classical minimum dominating set corresponds to a particular case. For values $k > 1$, the cardinality of a k-dominating set is less than that of a $D1$-dominating set: $|D_k||D1|$, lowering the network's monitoring cost (Zverovich et al. 2021).

In recent years, researchers have proposed a number of optimization algorithms, which can be classified into two types: traditional optimization algorithms and modem intelligent algorithms (N. Xu et al. 2002). Standard optimization algorithms are typically used to solve convex optimization problems with a known global optimum. On the other hand, modern intelligent algorithms are well suited to solving nonconvex optimization problems, particularly multi-extremum problems. Most fields, such as economics, industrial production, and network optimization, have many nonconvex optimization problems (Ju et al. 2016). The application of modern intelligent algorithms, such as artificial neural networks, artificial immune algorithms, particle swarm algorithms (Bonyadi & Michalewicz 2017)(Bonyadi & Michalewicz 2017)(Bonyadi & Michalewicz 2017)(Bonyadi & Michalewicz 2017)(Bonyadi & Michalewicz 2017), genetic algorithms, ant colony algorithms, artificial fish swarm algorithms, cultural algorithms, tabu search algorithms, and simulated annealing algorithms, has naturally become the focus of attention in various fields (Yu et al. 2018). Each of these algorithms has distinct advantages for resolving these numerous challenges. However, no single intelligent algorithm can take advantage of all of these benefits, and all have drawbacks such as dimensional difficulties, high memory requirements, inability to handle nonlinear characteristics, premature phenomena, collapsing into local optima, and excessive computation time(Yuan et al. 2014).

1.3 PROBLEM STATEMENT

The dominating set (DS) is a subset of the vertices of a particular graph, where all other nodes are connected to this set. DS has been used to mine networks that encompasses many applications, including wireless sensor networks (WSN) and social networks. The cluster head may be located using DS to reduce energy usage by reducing the DS size. Alternatively, choosing a smaller DS size to reduce the expense of the diffusion could increase the influence of diffusing in a social network (Stoica et al. 2020). The reduction of the DS size to save energy or money is a typical problem in both of the scenarios outlined above. The DS reduction size is the gap that academics have been attempting to bridge in order to reach the minimum dominating set (MDS), which is classified as an Np-hard problem (Chalupa 2018). There is still an opportunity for more research into meeting MDS criteria using small graph instances and certain topologies. Because of the exponential expansion with rising input size, the huge network poses a new issue for the algorithms. Researchers who have proposed approximation methodologies to provide a solution for the MDS problem have questioned this problem. Using greedy algorithms, for example, could result in a solution in polynomial time; however, with this class of algorithms overestimates the size of the MDS. The worst-case situation may emerge due to MDS worsening with the provided graph's input size. As a result, alternative approximations based on standard methods have been offered; nonetheless, obtaining the exact solution remains a long way off.

The ant colony algorithm (ACO) is one of the earliest meta-heuristics that used for the DS problems (Ho et al. 2006) . The preference of applying the ACO to MDS is due to its compatibility with the DS problems as they are graph-based and there are available heuristics the problem. These facts motivated more studies to investigate the ACO for such problems. However, the ACO was converging fast to a low-quality local optimum solution, and thus the DS size still required to be minimized. The converging problem can affect other population-based meta-heuristic algorithms, especially if they do not consider the balance between exploitation and exploration (Eftimov & Korošec 2019). Also, partitioning the graph into subsets is studied to improve the convergence of the bat algorithm (BA) when it is applied to the MDS (Abed & Rais 2019). The partitioning scheme in Abed & Rais (2019) showed improved performance of the BA in term of running time. However, partitioning a graph can lead to loss of data since each partition was tackled independently.

In the context of swarm intelligence, artificial bee colony (ABC) is a population-based algorithm that mimics the swarm behaviour of the bee colony, where all members of the swarm move toward a rich source of food (Karaboga & Basturk 2007). The recent success of applying the ABC for community detection in the study of Che et al. (2021).The study of Che et al. (2021) has motivated us to investigate its performance for the MDS problem. Since the ABC is based on the swarm behaviour, it explores the problem space more than exploiting it. However, ABC is prone to falling into local minima when dealing with complex problems. Bees' search pattern is good for exploration but poor for exploitation (Rahnema & Gharehchopogh 2020). To achieve balance between the exploration and exploitation, we hybridize the ABC with a local search, where the latter is able to intensify the search around an eligible solution.

1.4 RESEARCH QUESTIONS

This research aims to provide effective searching algorithm for the MDS. Thus, the following research questions are raised:

1. How to balance between exploration and exploitation procedures of the artificial bee colony to find the smallest DS for large graph instances?

2. How hybridization of artificial bee colony performs effectively when the MDS problem is considered?

1.5 RESEARCH OBJECTIVES

The main aim of this study is to provide a stochastic algorithm that can scale for large graph instances. To achieve this goal, the following sub objectives are determined:

1. To improve the exploitation procedure of the artificial bee colony to obtain a balance between the exploration and exploitation.

2. To evaluate the proposed method (Hybrid artificial bee colony algorithm) and compare it against the algorithms.

1.6 RESEARCH SCOPE

This study focuses on MDS problem for un-weighted connected graphs. In these graphs, each node is reachable from any node throughout a path in the graph. Thus, graphs of multiple components are excluded in this research. In this study, MDS is addressed by a hybrid ABC algorithm in which we incorporate a local search algorithm called late acceptance hill-climbing with the ABC algorithm by 10%. The output of the hybrid algorithm is then used to visualise the graph with the fewest number of dominating sets. Using this method, we get the fewest dominating sets.

1.7 SIGNIFICANCE OF STUDY

Graph theory is an essential branch of mathematics. As a result of numerous research activities, it has been grown rapidly. Over the last few decades, graph theory and combinations have accounted for one-third of all research papers published in mathematics. Dominance in graphs is a well-studied branch of graph theory with applications in computer and communications, social networks, molecular physics and chemistry, biological sciences, engineering, and many other areas of graph theory. This approach is primarily due to the large number of new parameters derived from the fundamental definition of dominance.

MDS is an optimization problem found in network clusters, wireless networks, big data, and social networks. For example, in the Wireless Sensor Network (WSN), DS members act as cluster heads, collecting and transmitting data to a gateway. By allowing only a small number of nodes with sufficient power (i.e., MDS) to access the gateway, this process helps to extend the network's life.

As a result, it is an excellent opportunity to conduct these studies on various graph examples to determine whether the proposed method is an effective solution to the MDS problem. Consequently, the importance of this project is that enables new studies to achieve efficient solution to the MDS applications.

1.8 ORGANIZATION OF THE THESIS

The thesis is composed of five chapters including the current one. The first chapter contained an introduction, background, problem statement, research questions, objectives, research scope, and significance of study. The literature review is described in Chapter II which includes introduction, artificial bee colony algorithm, discussion of minimum dominating set definition, and the summary. The methodology is described in Chapter III. This chapter includes an introduction, research methodology, problem identification, problem representation, the proposed method, ABC algorithm, conversion to binary, hybridized ABC with local search algorithm, model evaluation, and summary. Chapter IV presents overview, experimental design, evaluation datasets and metrics (datasets and metrics), results analysis, summary. Chapter V is composed of the overall summarization of the thesis, recommendations, and future work.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

The purpose of Chapter 2 is to describe the core elements of the title (a hybrid artificial bee colony method for the least dominant set problem) and to provide support for the research aims. The following subjects are covered: ABC algorithm and summarising past research on balancing ABC algorithm exploitation and exploration.

Chapter 2 defines the term minimal dominant set, provides an overview of different techniques used to address the MDS problem, and provides a brief assessment of some related publications.

This chapter is divided into the following sections: Section 2.2 defines the Artificial Bee Colony (ABC) algorithm. Sections 2.3 provide an overview of recent efforts to resolve the MDS problem, a discussion of the related works to the MDS problem, and a summary of this chapter.

2.2 ARTIFICIAL BEE COLONY ALGORITHM

Karaboga (2005) created the Artificial Bee Colony (ABC) algorithm, a swarm-based meta-heuristic technique for improving numerical problems. Honey bees' creative foraging activity inspired it. Three critical components comprise the ABC model: employed and unemployed foraging bees, as well as food sources. The first two components, employed and unemployed foraging bees, are responsible for the third component, the hunt for abundant food sources. Additionally, the model outlines two dominant types of behaviour required for self-organization and collective intelligence:

Positive feedback is generated when foragers are recruited to rich food sources, while negative feedback is generated when foragers abandon low food sources.

2.2.1 ABC Components

The workforce of ABC algorithm is divided into three categories: employed bees (EB), onlooker bees (OB), and scout bees (SB). These categories could be termed as ABC algorithm components. In theory, the EB component is responsible for narrowing search to neighbourhoods that are already stored in memory. The OB component is responsible for exploiting possible neighbourhoods, whereas the SB component is responsible for randomly generating solutions from far-fetched unseen neighbourhoods to stimulate exploration in ABC. Positive reinforcement and multiple interactions are two critical components of collective intelligence that are implemented through the EB and OB components. The SB component of the ABC algorithm is responsible for implementing the last two aspects of the method: negative feedback and fluctuation. The EB and OB components showed fluctuation or randomness into the swarm. The scale of randomization is frequently managed by a coefficient that can be increased or decreased to provide greater or lesser variation. Compared to the EB and OB components, the SB component has a substantially higher degree of variability because solutions are created entirely randomly (S. F. Hussain et al. 2020).

2.2.2 ABC Mechanism

In ABC, a colony of artificial forager bees (agents) searches for abundant artificial food sources (reasonable solutions for a given problem). To use ABC, first, transform the optimization problem at hand into the problem of determining the best parameter vector that minimises an objective function. The process starts with randomly selection of artificial bees of a population with an initial solution vectors. Then, the process continues to improve ABC using iteration leading to employing the strategies. Finally, the process moves towards better solutions via a neighbour search mechanism while abandoning poor solutions.

The ABC deployment mechanism begins with exploiting the nectar of food sources, followed by continuous exploitation, which eventually causes them to become exhausted. The employed bee exploiting the exhausted food source then becomes a scout bee searching for new food sources. In other words, when the employed bee's food source is depleted, they become a scout bee. In ABC, a food source's position represents a potential solution to a problem, and the nectar amount of a food source corresponds to the quality (fitness) of the associated answer. Because each employed bee is associated with one and only one food source, the number of employed bees equals the number of food sources (solutions) (Karaboga et al. 2007).

2.2.3 Optimization of ABC

Optimization problems have appeared in various fields, including engineering, economics, and management. Effective and efficient optimization algorithms are always required to tackle increasingly complex real-world optimization problems. In recent years, some swarm intelligence algorithms inspired by the social behaviours of birds, fish, or insects, such as particle swarm optimization (PSO), ant colony optimization (ACO), artificial bee colony (ABC), and firefly algorithm (FA), have been proposed to solve optimization problems (Qawqzeh et al. 2021). According to a recent study, ABC is found to be superior or comparable to other swarm intelligence algorithms (Jahwar & Ahmed 2021).

ABC has been used to solve various problems (Gu et al. 2020). ABC, like other stochastic algorithms, faces some complex issues. During the search process, ABC, for example, exhibits slow convergence speed. A new candidate solution is generated by updating a random dimension vector of its parent solution due to the unique search pattern of bees. As a result, the offspring (new candidate solution) is similar to its parent, and the convergence rate slows (Gu et al. 2020).

Furthermore, ABC is prone to falling into local minima when dealing with complex multimodal problems. Bees' search pattern is good for exploration but poor for exploitation. On the other hand, a good optimization algorithm should balance exploration and exploitation during the search process (Rahnema & Gharehchopogh 2020).

Based on Ab Wahab et al. (2015), swarm-based algorithms have grown to metaheuristic research showing outperforming evolutionary in population-based metaheuristic algorithms. Until 2019, there were more than fifty swarm-based metaheuristic algorithms were successfully applied in a variety of areas, including engineering, design, energy utilization, transportation, electrical and electronics engineering, business and economics, and arts (K. Hussain et al. 2019).

In comparison to more established and more recent methods, ABC has demonstrated promising outcomes. K. Hussain et al. (2017) demonstrated the effectiveness of ABC as a fuzzy neural network optimizer. Moreover, Garg (2014) used ABC for various structural engineering design challenges and discovered that it was more effective in reducing design costs than other metaheuristic algorithms . Garg (2014) developed a two-stage efficient strategy for resolving trustworthy redundancy allocation challenges. ABC was utilised to determine the optimal solution for reliability-redundancy allocation in this technique, improved in the second phase. The ABC method was used to optimise the parameters of an industrial system based on fuzzy logic. ABC outperformed both conventional methods and other evolutionary algorithms. The findings established ABC's advantage over PSO.

Numerous modifications and hybrids of the ABC algorithm have been proposed in the literature to generate sufficient population diversity to achieve the desired trade-off between exploration and exploitation. However, most ABC research focuses on high-level experimental analysis (Bansal et al. 2018).

2.2.4 Artificial bee colony (ABC) algorithm and variants

The objective of ABC is to locate the patch of flowers that contains the most nectar (optimal solution). To accomplish this, ABC divides the bee swarm into three groups: employed bees, onlooker bees, and scout bees. Each bee represents a D-dimensional solution; other bees will most likely follow the bee that discovers the most valuable food source to the most advantageous location. As with PSO, ABC uses the concept of memory to save user-defined best locations.

A bee visits a new location and then compares it to its previous best location. If the new location is preferable, the old one is forgotten, and the new one is remembered; if the new location is not preferable, the memory remains unchanged. ABC begins by randomly assigning bees to locations. When employed bees return to the beehive, they communicate with onlooker bees, who select the employed bee to follow based on the probability of selection (2.1):

$$P_i = \frac{f_i t_i}{\sum_{n=1}^{SN} f_i t_n} \qquad \text{…(2.1)}$$

Where, i and SN indicate the total number of food sources, where $f_i t_i$ denotes the objective function value (nectar amount). The more nectar a worker bee shares, the more likely it is that onlooker bees will choose it. With the assistance of Equation (2.2), the onlooker bee moves to a new location v_i, guided by the chosen employed bee:

$$v_i = x_i + Rand_i(x_i - x_j) \qquad \text{…(2.2)}$$

Where x_i is the current location in memory, x_j is the randomly chosen employed bee, and randomness is added to find nectar around location x_j. After several iterations, any bee that cannot find a better food source for an extended period is replaced with scout bee v_{new}. The failed bee's scout bee roams around any random or unexplored region to explore the environment as in Equation (2.3):

$$v_{new} = lb_i + Rand_i(ub_i - lb_j) \qquad \text{…(2.3)}$$

Where lb_i, ub_i, and $Rand_i$ are the lower, upper, and randomness bounds [0,1]. The next cycle begins with employed bees visiting the neighbourhoods of locations remembered by Algorithm 1 (chapter 3, page 34) summarises the three-step ABC procedure.

Y. Xu et al. (2013) improve search pattern by proposing a series of reasonable solutions in which, they employed bees and onlooker bees processes. Hence, new criteria were generated around the best solutions to improve the speed of the algorithm's convergence. The new ABC (NABC) has shown improved search efficiency when it was compared to ABC.

In another trial, ABC algorithm was altered to solve problems with constraints and then, by comparing the results to state-of-the-art meta-heuristics such as genetic algorithms, a differential POS was evolved (Karaboga & Akay 2011). The new variant resulted in two changes to ABC. The first change was enabling employed updating bee's parameter. The second change was creating Deb's tournament selection rules. The statistical analysis, which included ANOVA and ANOM, recommended the best parameters for the variant. In another article, Karaboga & Akay (2011), the same authors showed the improved ABC with the changes in the perturbation process. Hence, this process solved the issue of ABC's poor convergence on non-separable and composite constrained optimization problems. When the results of the experiments were compared to the standard and variants of ABC and other popular population-based algorithms, the proposed modification significantly improved ABC performance.

Sharma et al. (2016) have proposed another necessary modification to ABC, in which lévy flight was used as a balanced randomization strategy to avoid extra exploration and poor convergence. The variant included the lévy flight local search strategy as the final (4th) step after the employed bees, onlooker bees, and scout bees steps.

2.2.5 Discussion of ABC Algorithm

Table 2.1 summarizes ten research papers on the ABC algorithm from a different perspective. The papers enclosed are distributed according to the year of publication as follows: 1 (2015), 4 (2019), 3 (2020), and 2 (2021). The paper distribution shows that 90% of the selected papers were published within the last three years, while only one paper (10%) was selected because it contains essential basic information about the ABC algorithm. Hancer et al. (2015) has detailed one of the earliest modifications to ABC algorithm called a binary modification.

This modification was developed by integrating evolutionary-based similarity search mechanisms into an existing binary ABC variant. In 2015, the gap discussed was eliminating the irrelevant (redundant) features. Since 2019, there have been several modifications. A modified hybrid artificial bee colony (HABC) algorithm was used for numerical optimization by enhancing the accuracy (Pan et al. 2019). The other modified ABC algorithm was conducted by Huang et al. (2019) to enhance the ABC algorithm (EHABC) because of the optimization problems. EHABC was specifically used for solving continuous numerical optimization. The control of the ABC algorithm was discussed by Kumar & Nagalla (2020). To improve the qualities of the final solutions and convergence characteristics compared to the standard implementations of the ABC algorithms.

This control was powered by the decision-making process of the employed bees managing transitions to the dance area is modelled. The control procedure creates a more flexible transition mechanism. In the same year of 2019, Aslan et al. (2019) worked on the ABC algorithm to reach improved quick ABC (iqABC) to exploit a better mechanism based on the ant's food management. However, there is still no full solution to the complex behaviours of foraging for the food of the honey bees. The modified algorithm iqABC balances the local and the global search ability to attain optimization. The other modification was conducted by Sun et al. (2020) by surrogate-assisted multi-swarm artificial bee colony for complex numerical optimization problems.

The methodology used in this modification includes employing a diversity of enhanced local exploitation capability with an orthogonal method for better competition. The proposed algorithm can well keep the balance between exploration and exploitation with an outstanding performance which, compared to Aslan et al. (2019), represents a higher level of coordinating the algorithm performance.

Away from modifying ABC, S. F. Hussain et al. (2020) have introduced clustering optimization to build similarity between rows and columns and fill the gap associated with lacking optimization problems for data clustering. The performance of the clustering approach has attained very noticeable performance.

Another approach for better treating the ABC algorithm was adaptive neighbourhood search and Gaussian perturbation (Xiao et al. 2021). The process has some limitations, such as weak exploitation ability and slow convergence. The contribution of involving adaptive neighbourhood search is proposing Gaussian perturbation ABC algorithm.

Further, H. Wang et al. (2021) proposed an artificial bee colony algorithm based on knowledge fusion using knowledge to sense the search status, a learning mechanism. In this approach, weak exploitation capabilities of the traditional ABC algorithm have been evolved; however, the fusion approach results in a novel ABC algorithm based on knowledge fusion (called KFABC). The last article performed by Xiao et al. (2021) discusses binary optimization, a similar issue discussed earlier by Hancer et al. (2015). Xiang et al. (2021), an artificial bee colony algorithm was investigated with a pure crossover operation for binary optimization using information sharing and removing random perturbation in Xiang's article. The experimental results demonstrate that the proposed ABC is superior to other state-of-the-art approaches in terms of solution accuracy, convergence speed, and robustness and, consequently, creates ABC to successfully solve the binary structure of the (UFLP).

As conclusion for ABC Algorithm Table 2.1 summarizes ten research papers published between 2015 (Hancer et al. 2015) and 2021 (Xiang et al. 2021). The papers discuss the improvement of ABC and the modified BABC. The methodology used a comparison, optimization, and proposing various hybridization techniques. The results were denoted by successful attempts to enhance previous findings. The paper has mainly contributed to show the enhancement of ABC algorithm and developing different models. It is important to embark on the gaps mentioned in these articles. The eliminating of irrelevant features, low accuracy, difficulty of solving complex problems, convergence problem, and minimizing the limitation of ABC.

Table 2.1 Literature review of ABC Algorithm

#	Author/Year Title	Objective	Methodology	Gaps	Results	Contribution
1	Hancer et al. (2015). A modified binary ABC algorithm based on advanced similarity scheme for feature selection	To enhance the binary ABC algorithm's performance for feature selection problems by incorporating evolutionary similarity search mechanisms into an existing binary ABC variant	Comparing with some well-known variants of (PSO) and ABC algorithms	The pre-processing task of eliminating irrelevant or redundant features	Obtaining higher classification performance by eliminating irrelevant and redundant features	The modified ABC motivates and addresses feature selection problems
2	Pan et al. (2019). A hybrid artificial bee colony algorithm with modified search model for numerical optimization	To propose hybrid (ABC) to improved strategies and better utilization	HABC tests a new search model (random mutation scheme) and to generate multiple dimensions.	ABC standard suffers from low accuracy and slow convergence rate.	HABC is superior to other three ABC algorithms.	The limited ability of the traditional ABC results in creating HABC
3	Huang et al. (2019). An enhanced hybridized artificial bee colony algorithm for optimization problems	To present (EHABC) algorithm for optimization problems.	EHABC enhanced convergence speed and the information exchange between bees using mutation operator.	Some problems of optimization exist.	EHABC outperfoms ABC GABC, HABC, and EABC in solving continuous numerical optimization problems.	By utilising the crossover function of GA, we can enhance optimization.
4	Aslan (2019). A Transition Control Mechanism for Artificial	To improve the qualities of the final solutions and	The ABC algorithm is strengthened by the addition of a new control mechanism that simulates	Difficulties in managing the control transition behaviour.	Employing foragers to integrate standard serial and parallel	Developing a new model for the employed bees' decision-making process,

… to be continued

… continuation

	Bee Colony (ABC) Algorithm	convergence characteristics compared to the standard ABC algorithms	The decision-making. process of employed bees managing transitions to the dance area		implementations of the ABC algorithm	dubbed the transition control mechanism.
5	Aslan et al. (2019). Improved quick artificial bee colony (iqABC) algorithm for global optimization	To enhance the improved quick ABC performance.	Employing ABC algorithm by exploiting better mechanism based on food management	ABC algorithm does not fully solve the complex behaviours of honey bees in foraging	The proposed method outperforms the traditional implementation of the ABC algorithm and its other variants significantly.	Proposing an enhanced fast ABC (iqABC) method to strike a balance between local and global search capabilities.
6	Sun et al. (2020). A modified surrogate-assisted multi-swarm artificial bee colony for complex numerical optimization problems	To proposes a surrogate-assisted multi-swarm artificial bee colony (SAMSABC)	By utilising diversity to strengthen the local exploitation potential through an orthogonal approach, greater competitiveness can be achieved.	ABC suffers from slow convergence rate which limits its real-world applications	The proposed algorithm is capable of striking an excellent balance between exploration and exploitation.	Developing a modified ABC for solving complex optimization problems in order to maintain population diversity
7	S. F. Hussain et al. (2020). Co-clustering optimization using Artificial Bee Colony (ABC) algorithm	To propose the use of higher order correlations to generate similarity between rows and columns that are dependent on one another.	Measuring the embedded similarities to optimizing the co-clusters by exploring the vicinity of the solutions produced by the ABC algorithm	ABC algorithm lacking optimization problems for data clustering.	Examining the performance of the ABC algorithm for the clustering of high-dimension datasets.	The optimization of clustering has been solved with improvements in the objective function
8	Xiao et al. (2021).	To investigate the	On the basis of the	ABC has some limitations	ABCNG is more	The ABCNG algorithm

… to be continued

… continuation

	Artificial bee colony algorithm based on adaptive neighbourhood search and Gaussian perturbation	effectiveness of the novel ABC with adaptive neighbourhood search and Gaussian perturbation (ABCNG)	Neighbourhood structure, a modified global best solution guided search strategy is constructed that incorporates a new Gaussian perturbation.	Such as a limited capacity for exploitation and a slow convergence rate.	Competitive than six other ABCs.	Is proposed as a new ABC with adaptive neighbourhood search and Gaussian perturbation.
9	H. Wang et al. (2021). Artificial bee colony algorithm based on knowledge fusion	To propose a novel ABC based on knowledge fusion (KFABC)	Using KFABC to sensing the search status and to propose adaptively select appropriate knowledge.	Exploitation capabilities of the tradition ABC algorithm is weak.	KFABC outperfoms nine ABC and three differential evolution algorithms	A novel ABC algorithm based on knowledge fusion (called KFABC)
10	Xiang et al. (2021). Artificial bee colony algorithm with a pure crossover operation for binary optimization	To propose a binary ABC called (BABC) as a means of resolving the problem of uncapacitated facility location (UFLP).	Utilizing a pure crossover operation to exchange information and eliminate random perturbation.	ABC algorithm approaches to continuous optimization omit binary continuous space from consideration.	The proposed BABC outperforms other state-of-the-art approaches in terms of solution accuracy, convergence speed, and robustness.	This paper proposes the creation of BABC in order to successfully solve the binary structure of the UFLPs.

2.3 MINIMUM DOMINATING SET

2.3.1 Definition of MDS

The dominating set of the graph $G\ (V, E)$ is a subset of vertices D that has the property that each vertex in G either belongs to D or is next to a vertex from D. The minimum dominating set is the one that has the smallest set size. One of the standard NP problems is one of the most influential groups on the graph. It is imperative to figure out the minimum dominant set of the graph to get the best graph. This approach is what all the algorithms used to solve this want to do. This definition has two main parts: DS and MDS. The DS is a group of vertices that connect to all other vertices. Problem: The minimum dominating set problem is to figure out the DS with the least number of points (MDS). Also, the MDS is defined as the minimum number of nodes that cover all other nodes in a given graph (Abed & Rais 2019).

In social networks, MDS is investigated for influence maximization which aims to diffuse information through networks (Alipour et al. 2020). People in the network may not talk directly to each other because of time or other constraints. In this case, we might need to reach all of the network's nodes in an emergency, but only a small number of people in the network can be directly connected. Emergency messages can be sent quickly if all network nodes can be reached by at least one person (or one of these people). Some nodes might have to be chosen in a social network to ensure that news does not spread by checking the spread of news in the network. In these situations, the goal is to pick the least number of these nodes. Thus, the main issue" is how to find the minimum members of the networks that influence others?

2.3.2 MDS Approaches

Dominating sets are an essential concept in graph theory and have applications in a variety of fields, particularly social networks (Chalupa 2018). In a social network, one common problem is to find the smallest group of influential individuals or a set of initial seeds so that all participants can be reached with only one hop from the seeds. This problem is equivalent to determining the network's minimum dominating set (minDS).

Under the assumption $P = NP$, the minDS problem is a classic non-deterministic polynomial-time hardness $(NP) - a$ hard problem that cannot be approximated with a constant ratio (Raz & Safra 1997). Furthermore, even when limited to power-law graphs, negative results for minDS approximation have been demonstrated (Gast et al. 2015). Several works on exact algorithms for minDS have been completed, with the primary goal of improving the upper bound of running time. Exact algorithms for minDS that are state-of-the-art are based on the branch and reduced paradigm and can achieve a run time of $O(1{:}4969n)$ (Van Rooij & Bodlaender 2011). Better complexity results have been obtained using fixed parameterized algorithms (Karthik & Inbal 2021). Theoretical aspects are the primary focus of such algorithms.

2.3.3 Heuristics for the MDS

The dominating set is classified into three types in graph theory: the minimum dominating set (MDS), the minimum connected dominating set (MCDS), and the minimum weight dominating set ($MWDS$). These variants have been successfully investigated in the recent past using meta-heuristic approaches (Lin et al. 2016). Although the MDS problem's primary objective is to find the smallest dominating set, several studies have taken into account the processing time required to find the MDS. Lin et al. (2016) developed a hybrid method that combines a greedy construction procedure with some stochastic operators to quickly find the highest-quality solution to the $MWDS$. On the other hand, Giap & Ha (2014) attempted to accelerate the solution of the MDS problem by parallelizing the genetic algorithm. As a result, they investigated a parallel scheme for generating the desired solutions for the MDS problem using genetic algorithm operators.

Heuristic approaches are frequently used in practise to obtain good solutions in a reasonable amount of time. While greedy algorithms are fast, yet the resulting dominating sets are far from satisfactory relatively to the input size. Sanchis (2002) compares several greedy MinDS heuristics. Typically, heuristic search algorithms return very good, if not optimal, results in a reasonable amount of time. To solve MinDS, well-known heuristic search methods such as genetic algorithms Hedar &

Ismail (2010) and ant colony optimization (Potluri & Singh 2013) were developed. Hyper meta-heuristic algorithms optimise performance by combining various heuristic search algorithms and preprocessing techniques (Abu-Khzam et al. 2017). These algorithms were evaluated on standard benchmarks containing thousands of vertices. Cai et al. (2021) recently applied the configuration checking (CC) strategy to MinDS, resulting in the development of two local search algorithms. Y. Wang et al. (2017) proposed the CC2FS algorithm for unweighted and weighted MinDS and demonstrated that it outperformed ACO-PP-LS on standard benchmarks (Potluri & Singh 2013). Following that, FastMWDS was proposed as a CC-based local search that significantly improved CC2FS on massive weighted graphs (Y. Wang et al. 2018). Chalupa(2018) proposed RLS and order-based randomised local search algorithms that outperformed ACO-LS and ACO-PP-LS (Potluri & Singh 2013) on unit disc graph benchmarks as well as some massive graphs. Fan et al. (2019) developed ScBppw, a local search algorithm based on two concepts: score checking and probabilistic random walk.

Small instances of a given problem can be solved intuitively more accurately and quickly than large instances of the same problem. As a result, some studies have recommended that problem instances be divided into smaller ones to facilitate problem-solving. For example, specific instances of the vehicle routing problem (VRP) had been partitioned into sectors and solved concurrently (Rabbouch et al. 2020).

Table 2.2 summarises ten research papers on the MDS problem from various angles. The papers are numbered 1 (2010), 1 (2013), 1 (2016), 2 (2018), 1 (2019), 2 (2020), and 2 (2021). (2021). As a result, 80% of the papers are from the last four years, while the remaining 20% are from older papers. MDS was mentioned in the titles of all papers for various reasons. However, there is agreement among authors that the MDS problem was approached differently, with each containing a different or specific algorithm. This implies that the MDS problem, as an idea, can lead to a variety of applications and purposes. The table has six columns: author/year/title, objective, methodology, gaps, results, and contribution. The goal of this paper is to either compromise two algorithms of $O(a^2)$ and $O(\log n)$ (Lenzen & Wattenhofer 2010). This approach an easy method by a complex two-step ACO algorithm (Jovanovic & Tuba 2013), to dedicate central and distributed approximation algorithms Mahmood et al.

(2016), to ease the probabilistic wireless sensor (Boria et al. 2018), to compare the performance The methodology used by all papers is nearly identical, which is to run the new algorithm through a process. The gap is the most important aspect of the table. The difficulty of using the non-deterministic polynomial to solve the MDS problem was the source of the gap in 2010. (Lenzen & Wattenhofer 2010). However, in 2013, a trial to solve that problem was carried out by implementing the greedy algorithm (Jovanovic & Tuba 2013), which was then followed by the objective function exponential time $O(k^n)$ algorithms (Mahmood et al. 2016). Other gaps that have emerged since 2016 include the use of MDS in natural wireless sensors (Boria et al. 2018), optimization problems (Li et al. 2018), finding the smallest number of nodes (Abed & Rais 2019), searching for new ideas (Cai et al. 2021), and dealing with graph theory (ALOFAIRI et al. 2021). The results appear to be successful in all cases, such as minimising the time required to run the test (Lenzen & Wattenhofer 2010) or using polynomial (Boria et al. 2018), or dealing with large amounts of data (ALOFAIRI et al. 2021). As a result, these research papers have made the MDS problem more efficient and produced better results.

As conclusion for MDS, Table 2.2 summarizes ten research papers published between 2010 (Lenzen & Wattenhofer 2010) and 2021(ALOFAIRI et al. 2021). The papers discuss the design and implanting dedicated central distribution for approximation, presenting swarm intelligence behaviour, and employ heuristic approach. The methodology was about implementing various algorithms, integrating different strategies, constructing novel local search framer. The results were denoted to show the performance of several algorithms and compare the results. The paper have mainly contributed to highlight the missing trading-off in previous work, applying partitioning scheme, and to handle the dynamic of the network. It is important to embark on the gaps mentioned in these articles. The show solving the non-deterministic polynomial time, showing the importance of optimization problem with several applications.

Table 2.2 Literature review of MDS problem

#	Author/Year Title	Objective	Methodology	Gaps	Results	Contribution
1	Lenzen & Wattenhofer (2010). Minimum dominating set approximation in graphs of bounded arboricity	To compromise two algorithms $\mathcal{O}(a^2)$ and $\mathcal{O}(\log n)$ to generalize the problem on specific graphs.	Our first algorithm employs a forest decomposition, achieving a factor $\mathcal{O}(a^2)$ approximation in randomized time $\mathcal{O}(\log n)$.	Non-deterministic Polynomial time (NP)-hard to solve the minimum dominating set problem.	• Decomposing $O(a^2)$ with approximation in randomized time $\mathcal{O}(\log n)$. • Approximation ratio of $\mathcal{O}(a \log\Delta)$, where Δ is the maximum degree.	Trade-off between running time and approximation ratio, that is, for any parameter $a \geq 2$, we can obtain an $O(a \log\Delta)$.
2	Jovanovic & Tuba (2013). Ant colony optimization algorithm with pheromone correction strategy for the minimum connected dominating set problem	To approach an easy solution without utilising the previously developed complex two-step ACO algorithm.	Implemented a one-step ACO algorithm based on a known simple greedy algorithm that has a significant drawback of being easily trapped in local optima.	The MCDSP is being used in mobile ad hoc networks (MANETs) and sensor grids.	Comparing the new approach to the existing algorithms using standard benchmark data.	By incorporating a pheromone correction strategy and paying close attention to the ACO algorithm's initial condition, this negative effect can be avoided.
3	Mahmood et al. (2016). Membrane computing to enhance time efficiency of minimum dominating set	To design and implement dedicated central and distributed approximation algorithms for restricted graph classes.	Using the algorithmic tile self-assembly model, to solve MDS problem in $O(n^2)$ steps.	In recent research exponential time $O(k^n)$ algorithms are used for some graph classes for solving the MDS problem.	• The MDS problem has been solved in $O(n^2)$ steps. • In the area of membrane computing, P systems introduce two levels of parallelism.	Introduces an algorithm based on the parallelism feature of the P systems model for solving the MDS problem in linear time $O(n)$.

… to be continued

… continuation

4	Boria et al. (2018). The probabilistic minimum dominating set problem	Present a natural wireless sensor network problem with a probabilistic MDS model.	Demonstrate that the objective function calculation for this general probability problem is #P-complete.	The natural wireless sensor network problem.	• The objective function (#P-complete) was introduce a restricted version of probabilistic MDS. • Objective function performed in polynomial.	Studying the complexity of this restricted version in graphs mainly in trees and paths with some approximation results.
5	Li et al. (2018). An Efficient Local Search Algorithm for the Minimum k-Dominating Set Problem	On classical instances, compare the performance of VSCC2 to that of the classic GRASP algorithm and the well-known commercial solver CPLEX.	Integration of the MKDSCC2 strategy and scoring mechanism. Propose a strategy for vertex selection.	This is a significant NP-hard combinatorial optimization problem with a wide variety of applications.	A fast local search algorithm (VSCC2) is used to implement a two-level configuration checking strategy, a scoring mechanism, and a vertex selection strategy.	In terms of solution quality and computation time, the VSCC2 algorithm is extremely competitive.
6	Abed & Rais (2019). Solving the Minimum Dominating Set Problem of Partitioned Graphs Using a Hybrid Bat Algorithm	Presenting the swarm intelligence behaviour to deal with population bat algorithm (BA) to find the smallest set of nodes that dominate the graph.	Proposing method partitions to reduce the computational time of finding the MDS solution by analyzing performance of the hybrid algorithm.	MDS finds the minimum number of nodes that have connections to all other nodes in a given graph.	The Simulated annealing (SA) algorithm balances exploitation and exploration for a best possible solution.	The gained results showed significant speed up when the partitioning scheme was applied.
7	Cai et al. (2021). Two-goal Local Search and Inference Rules for	To employ heuristic approaches (FastDS) to obtain good solutions	• Using a novel local search framework. • constructing	Developing an efficient local search algorithm for MDS with two ideas.	• FastDS obtains the best performance. • obtains better	FastDS is evaluated on 4 standard benchmarks and 3 massive graphs

… to be continued

… continuation

	Minimum Dominating Set	within reasonable time.	procedure with inference rules.		solutions algorithms on massive graphs.	benchmarks
8	Alipour et al. (2020). On Distributed Algorithms for Minimum Dominating Set problem, from theory to application	To propose a distributed algorithm for the MDS problem.	Implementing an algorithm on massive social networks and compare the results with the state of the art algorithms using dynamic networks by adding/deleting edges and vertices.	A new algorithm on massive social networks with the state of the art algorithms.methodology.	• Solving the distance MDS • Studying experimentally the efficiency of the proposed algorithm. • Our proposed algorithm is fast and easy to implement,	Reasonable solutions to use it in distributed model practically.
9	Alipour & Salari (2021)). On distributed algorithms for minimum dominating set problem and beyond	Studying MDS problem and the minimum total MTDS problem.	Presenting a distributed randomized algorithm to solve MDS and MTDS problems using our theoretical results.	The minimum dominating set (MDS) & MTDS	Computing approximately MDS and MTDS to achieve an upper bound on the size of MDS of a graph.	The new algorithms handle the dynamic of the network under constraints in choosing the elements of MDS.
10	ALOFAIRI et al. (2021). Quality Evaluation Measures of Genetic Algorithm and Integer Linear Programming for Minimum Dominating Set Problem	To compare the performance of two approaches to solving the MDS problem (ILP and HGA).	• Utilizing the available benchmark test. • Proposing three measures to evaluate the quality of the obtained solution.	Due to the inadequacy of previous methods, numerous methods have been developed to solve the MDS problem and generate distinct solutions for the same graph.	The experimental results indicate that the ILP-MDS outperforms the HGA-MDS in terms of calculating the domination number, determining the optimal solution, and manipulating large data graphs.	The average of the node's degree and between centrality is calculated using the close performance of two methods.

2.4 SUMMARY

In terms of definition and related topics, this chapter has outlined the foundations and principles of MDS. This chapter thoroughly discussed the definition, approach, and heuristics of MDS. The use of approximate algorithms is frequently used to solve the MDS. Hence, this chapter has reviewed works that used this class of algorithms to solve the MDS. Also, this chapter has focused on studies that concerns the ABC algorithm due to its importance in various optimization fields. Additionally, this chapter has presented summarization through tabular forms to ease the reading of the reviewed works

CHAPTER III

METHODOLOGY

3.1 INTRODUCTION

This chapter describes the research method used to achieve the objectives of this study. This chapter discusses the stages of this research and the proposed approach for resolving the MDS. The technique used in this study and the phases of the proposed work have been described. The proposed method then shows algorithmic steps for reducing the DS of a given graph.

This chapter is divided into seven sections. Section 3.2 describes the methodology used to conduct the research for this study. Sections 3.3–3.6 go over the steps of the research methodology. Finally, Section 3.7 summarises the contents of this chapter.

3.2 RESEARCH METHODOLOGY

This section goes over the research methods that were used in this study. In computer science, research approaches are typically classified into four categories Johnson (2006). Demonstration proof, empiricism, mathematical proof, and hermeneutics are a few examples. Furthermore, numerous studies make use of a variety of research methodologies. Empiricism proofs are used because mathematical proofs are difficult or impossible to implement in practice (Bartz-Beielstein et al. 2010). As a result, this paper takes an experimental approach to create a solver for minimising the dominant set for a specific graph.

In practice, the research approach for this study is divided into five phases, as shown in Figure 3.1. This graphic summarises the research design used in this study.

The details of the study phases will be discussed in greater depth in the sections that follow.

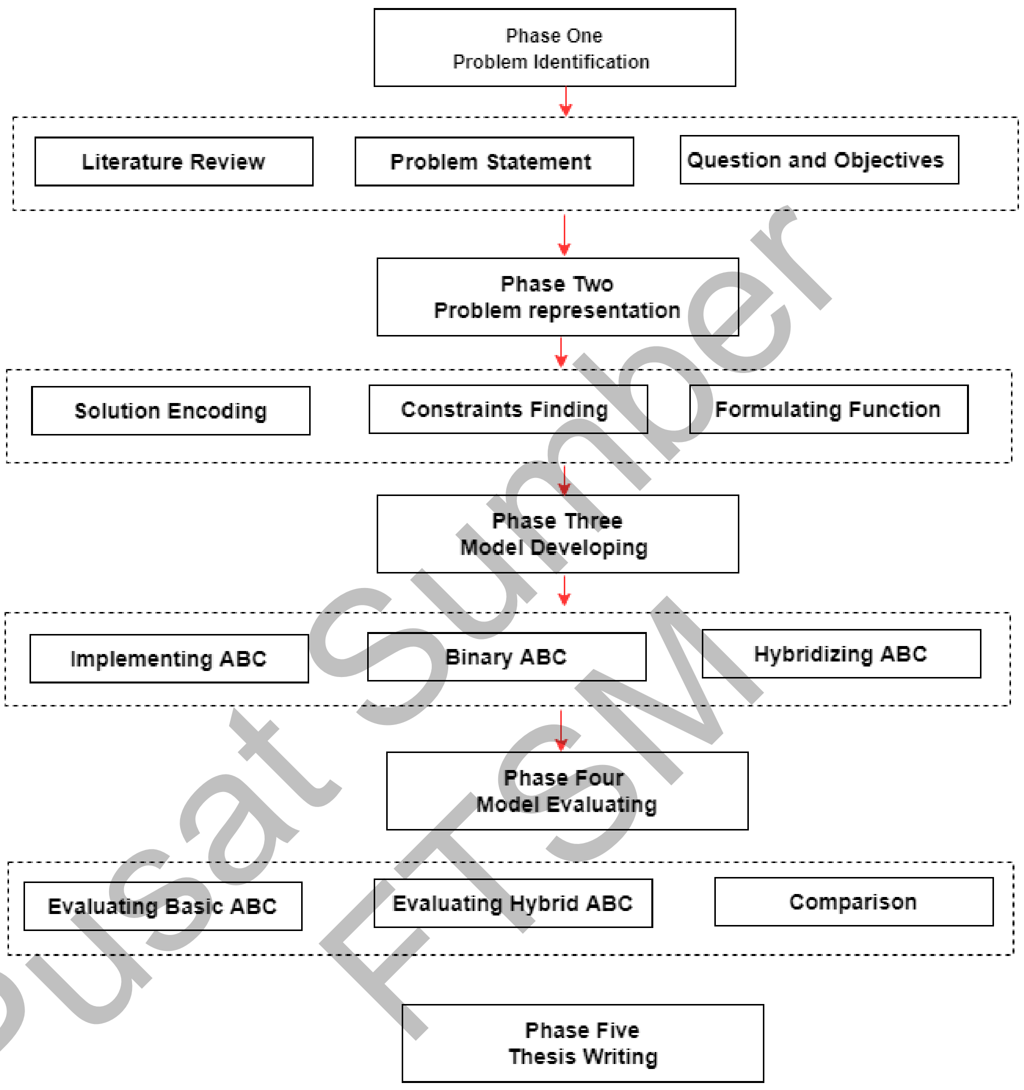

Figure 3.1 The proposed research methodology

3.3 PROBLEM IDENTIFICATION

The first step in this investigation was to identify the problem by evaluating existing *MDS* approaches. To that end, we posed the following research questions in this study: How can the artificial bee colony's exploration and exploitation operations be balanced in order to find the smallest DS for large graph instances? Furthermore How effective is artificial bee colony hybridization when the MDS problem is taken into account?

This research aims to optimise the artificial bee colony's exploitation method to achieve a balance between exploration and exploitation. The second goal is to evaluate the proposed method (Hybrid artificial bee colony algorithm) and compare it to previous work.

3.4 PROBLEM REPRESENTATION

The second section of this study is devoted to formulating the DS problem. This problem entails encoding the solution to the problem, which will be used as input for the search algorithm. This phase also establishes the constraints that the output of the search process must meet. Finally, the second phase establishes the fitness function used to evaluate the solutions discovered during the search operation. As a result, the output of this step is an optimization model that must be solved by reducing the DS to the smallest possible set while remaining constrained by the given constraints.

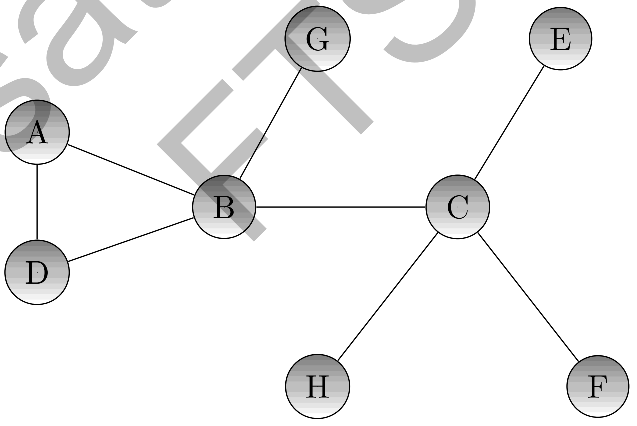

Figure 3.2 An example of the MDS

Based on the definition of the DS, a given graph G is divided into two subsets: DS and NDS. As a result, a binary representation is a standard encoding strategy for this problem, with ones denoting the DS and zeros denoting the NDS. Consider using a simple graph, such as the one shown in Figure 3.2, to demonstrate this representation.

Each node is represented by a single bit in binary encoding, arranged alphabetically. Thus, for the above graph, a random MDS could be 10100010, implying that the MDS is $\{A, C, G\}$. The ideal MDS for this case, however, has only two nodes, B and C, which the binary string $x = 01100000$ can represent. The MDS has a capacity of two nodes, with the remaining nodes connected to the MDS. Figure 3.2 and 3.3's binary representations are found in Table 3.1.

Table 3.1 Binary representation for Figure 3.2

Random MDS of Figure 3.2							
A	B	C	D	E	F	G	H
1	0	1	0	0	0	1	0

Optimal $\boldsymbol{MDS}$ of the graph in Figure 3.2							
A	B	C	D	E	F	G	H
0	1	1	0	0	0	0	0

Thus, in order to evaluate an MDS solution, we must count the ones (see Equation 3.1) and guarantee that nodes are connected to the MDS, as illustrated in Equation (3.2):

$$f(x) = \sum_{i=1}^{n} x_i \qquad \ldots(3.1)$$

$$s.t. \sum_{j=1}^{n} A_{ij} x_j \geq 1 \qquad \ldots(3.2)$$

where n is the graph order, and A is the adjacency matrix, which is as follows:

$$A = \begin{cases} 1 & if\ i\ and\ j\ are\ adjacent \\ 0 & otherwise \end{cases}$$

3.5 THE PROPOSED METHOD

The proposed method starts by reading the MDS model from the previous phase. After that, the proposed ABC algorithm treats encoded solutions as a food source for artificial bees to investigate. The positions of the food sources are then updated for a predetermined number of cycles. Each cycle, the local search is launched with a predetermined probability known as the local search rate (l_r). These steps are depicted in Figure 3.3. The ABC method typically works with continuous variables, requiring the use of an intermediary function to convert the ABC food source to an MDS solution. This study uses the angular modular function to deal with continuous variables. This section goes over each of these steps.

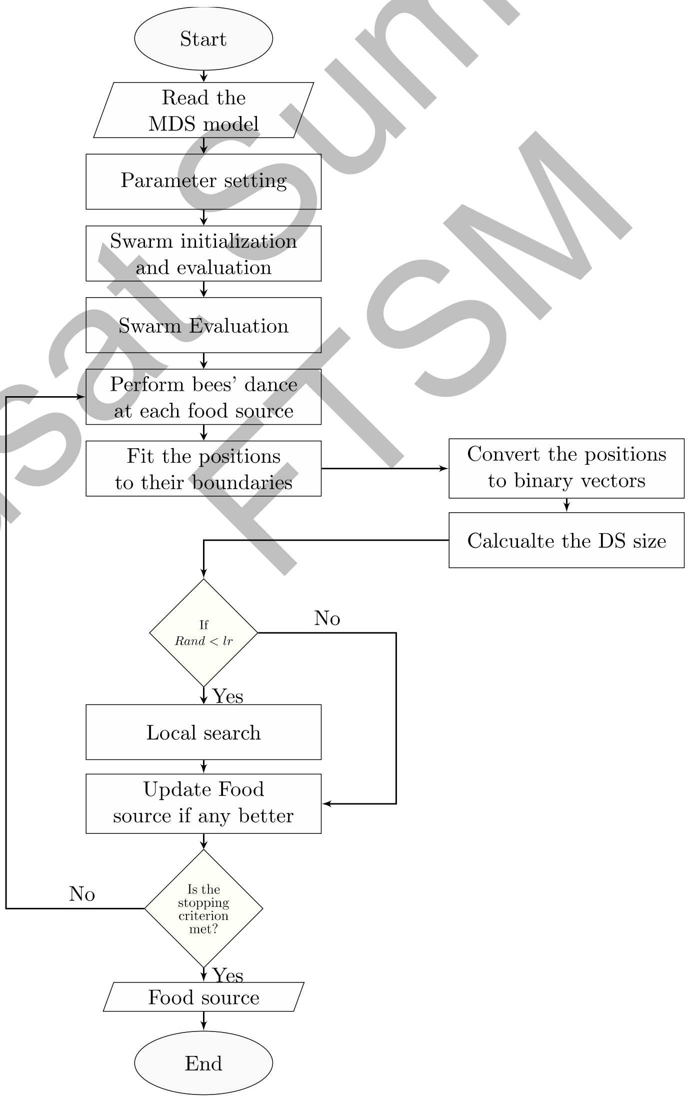
Start
Read the MDS model
Parameter setting
Swarm initialization and evaluation
Swarm Evaluation
Perform bees' dance at each food source
Fit the positions to their boundaries
Convert the positions to binary vectors
Calcualte the DS size
If Rand < lr
No
Yes
Local search
Update Food source if any better
Is the stopping criterion met?
No
Yes
Food source
End

Figure 3.3 The workflow of the proposed method

3.5.1 ABC Algorithm

The search for the best source is delegated to three groups of bees that comprise the ABC algorithm. The first group is made up of bystanders, who gather in the dance area to choose a food source. The second group consists of employed bees; a bee returns to a previously visited food source. The third group is the scouts, and each member conducts a random search. Because each employed bee is assigned to a single food source, the total number of food sources near the hive equals the total number of employed bees. When a food source is depleted, the hired bee is transformed into a scout. The procedures used by these groups to find the best food source are depicted in Algorithm 1.

The initialization step of the ABC algorithm involves the bees randomly selecting a set of food sources and quantifying their nectar production (see Lines 1-3 in Algorithm 1). When these bees return to their hive, they will share nectar source information with the bees in the dance area. Following the dissemination of knowledge, hired bees to return to previously visited food sources that are still fresh in their memory. The hired bees will then choose new food sources based on visual cues in close proximity to the current one. When employed bees return to the hive's dance area, they share nectar information with viewers, allowing them to select their preferred food source. An observer would choose a food source based on the amount of nectar shared by the employed bees, with a high probability of choosing a nectar-rich food source. Equation 3.3 calculates the nectar amount by dividing the probability of selecting a food source by the total amount of nectar produced by other bees. When a food source runs out, the bees abandon it and select a new one randomly (as shown in Line 17 of Algorithm 1).

Algorithm 1: ABC algorithm

```
   Input: $MDS$ Model
1  Initialize food sources (Foods) randomly $x_i$ (i=1,2, ..., PS)
2  Deploy colony of bees, Employees, Onlookers, and Scouts
3  Evaluate food sources
4  repeat
       /* Employeed bees' phase*/
5      foreach $x \in Foods$ do
6          $j \leftarrow random(0, N)$, where $N$ is the graph order
7          $y \leftarrow$ random Food source, where $x \neq y$
8          Generate new food source based on $x_j$ and $y_j$ using
             Equation 3.4
9          Update $x$ if it is inferior to the new food source
       /* Onlookers bees' phase*/
10     $OP \leftarrow$ Onlooers probability using Equation 3.3
11     for $i = 1$ to $PS$ do
12         if $random(0, 1) \leq OP_i$ then
13             $j \leftarrow random(0, N)$
14             $k \leftarrow random(0, PS)$
15             Generate new food source based on $Foods_{ij}$ and
                 $Foods_{kj}$ using Equation 3.4
16             Update $Foods_i$ if it is inferior to the new food
                 source
       /* Scout bees' phase*/
17     Send an employed bee to a random food source when it
         acrosses its limit of trials
18 until $Max_{iterations}$ is reached;
   Output: FoodSource
```

The ABC algorithm considers a solution to the MDS problem to be the location of a food source. The quality of the solution is proportional to the nectar content of the food source. The population's total number of solutions equals the number of onlookers or employed bees. The ABC method starts by randomly selecting solutions from a uniform distribution. Each solution/food source $x_i(i = 1,2,3,..PS)$ is an N-dimensional vector, with PS denoting population size and N denoting graph order. Following initialization, the employed, observer, and scout bees seek the population of solutions regularly. The search process ends when the specified maximum number of MC cycles is reached. Each cycle, an artificially employed bee probabilistically alters the solution or food source it memorises to find a new food source. This novel food source's nectar content and fitness value are being assessed. If the nectar content of the new food source is superior to that of the old, the bee will prefer the new over the old. Because the amount of nectar in the MDS is proportional to its size, an artificial bee will replace an older MDS with a smaller one. If the new MDS is larger than the old, the bee will keep the old. When the artificially employed bees have finished their cycle, they distribute the new MDSs to the artificial observers, who choose one based on its size. A bystander will change the new MDS in the same way that employed bees do.

An onlooker will choose a food source probabilistically, where a food source i is associated with a selection probability pi which is calculated as follows:

$$p_i = \frac{f_i}{\sum_{j=1}^{PS} f_i} \qquad \ldots(3.3)$$

where f_i is the fitness value of solution or food source i evaluated, and PS is the population size which is equal to the number of food sources and the number of employed bees (BN). In the MDS, f_i is the DS size of the solution i. The artificial bees in the ABC algorithm produces a new food source (solution) from previous ones as follows:

$$v_{ij} = X_{ij} + \phi_{ij}\left(X_{ij} + X_{kj}\right) \qquad \ldots(3.4)$$

where $k \in \{1, 2, 3, \ldots, PS\}$ and $j \in \{1, 2, 3, \ldots, N\}$ are randomly chosen indices, and $k \neq i. \phi$ is a random number between [-1, 1] that controls the production a new source food around X_ij.

The new source food produced by Equation 3.4 is evaluated by the artificial bee and compared to X_i to replace it if the new on carries smaller MDS. This process repeated till a predefined maximum number of cycles (MC) is reached.

3.5.2 Conversion to Binary

It is critical to remember that the ABC algorithm works with continuous variables. On the other hand, the proposed method uses a function known as angular modulation to convert these variables to binary solutions. The initial population of the ABC (Food sources) is uniformly distributed across the time range [0, 1]. This is maintained by inspecting the variable boundaries throughout the search process. Thus, the round function can be used to convert variables with values less than 0.5 to 0 and variables with values greater than 0.5 to 1. A sigmoid function can also be used to complete this work. On the other hand, these functions reduce variables with smaller values to zero, even if they have the potential to be in the dominating set. As shown in Equation (3.5), we used the angular modulation function to obtain the binary solution:

$$g(x) = \sin(2\pi(x-a) \times b \times \cos(2\pi(x-a) \times c)) + d \qquad \ldots(3.5)$$

where a, b, c, and d are coefficients that determine the shape of the function. That is, a controls the shift over the horizontal axis of the function, b indicates the maximum frequency of the sin function, c and d are the frequency of the cos function and the vertical shift, respectively. In this study, we used the default values of this function, which are a = 0, b = 1, c = 1, and d = 0. The input of this function x is a decision variable from a food source of which is uniformly distributed. The number these variables are determined by the graph order to find the MDS in the original problem space.

3.5.3 Hybridized ABC with Local Search Algorithm

At each stage of the ABC algorithm, an artificial bee selects a food source to change, allowing the system to explore a large area of the search space. On the other hand, ABC falls short of the goal of hastening the search for a promising solution. As a result, we incorporate a local search algorithm to improve the ABC's exploitation process. This study employs a technique known as late acceptance hill-climbing (LAHC) in its local search algorithm. This algorithm starts with a predefined solution or food generated by the ABC.

Algorithm 2: Late Acceptance Hill Climbing (LAHC)

```
1  $X \leftarrow$ Initial MDS solution obtained from the ABC algorithm
2  $L \leftarrow$ length of the list
3  for i=1 to L do
4      $f_i = f(X)$                          ▷ Initialize the fitness list.
5  $X^* = X$                                 ▷ Memorize the best solution.
6  for i=1 to Max_terations do
7      $X'$=move($X$)   ▷ Move from the current solution to a new one.
8      $v = i \bmod L$
9      if $f(X') \leq f_v || f(X') \leq f(X)$ then
10         $X = X'$                          ▷ Accept the new solution.
11         if $X' < X^*$ then
12             $X^* = X'$
13     $f_v = f(X)$      ▷ Insert the current cost to the fitness list.
   Output: $X^*$
```

The primary steps of the LAHC are to obtain the initial solution and iteratively modify it to a new state. The latter will be used as a starting point based on predefined criteria. As shown in Line 9 of Algorithm 2, these criteria determine whether a new solution can be used as a new starting point. The first requirement is that the new solution outperforms the existing one. The second criterion determines whether the new solution is better than the solution from the previous cycle. This procedure is repeated until a predetermined maximum number of repetitions is reached.

The ABC algorithm will use the LAHC output as a new food source. This food source is distributed for harvesting to other artificial bees. The ABC algorithm's exploitation process is consolidated in this procedure. However, as shown in Algorithm 3, the LAHC is only called when a predefined probability, denoted lr, is met. This strategy strikes a balance between experimenting with and exploiting the ABC algorithm.

Algorithm 3 depicts the hybrid binary ABC steps (HBABC). This algorithm differs significantly from the original ABC in three ways. The first difference is in solution encoding; the HBABC uses the binary conversion function shown in Equation 3.5. (see line 9 of Algorithm 3). The second distinction can be found in Algorithm 3, line 11, where the HBABC modifies the probability function of the standard ABC (as shown in Equation 3.6)

$$p_i = 0.9 \times \frac{f_i}{\sum_{j=1}^{PS} f_j} + 0.1 \qquad \ldots(3.6)$$

As demonstrated in previous experiments, this equation positively affects the ABC's exploratory process (S.-C. Huang 2015). This function is unique to the spectator phase, as it necessitates the selection of a food source. This ability allows observer bees to investigate food sources with a low chance of success. Food sources with a high nectar content, on the other hand, will continue to be associated with a high probability of selection.

HBABC's search strategy is now focused on exploring and exploiting the issue space, which is the third improvement. Lines 20-24 of Algorithm 3 show the steps involved in activating the local search algorithm, which is controlled by the l_r parameter. These procedures entail picking a random food source and using the LAHC to expand the search radius. If the obtained food source has a higher nectar content, the associated MDS is lower than the previous one.

Algorithm 3: Hybrid binary ABC algorithm

```
   Input: $MDS$ Model
1  Initialize food sources (Foods) randomly $x_i$ (i=1,2, ..., PS)
2  Deploy colony of bees, Employees, Onlookers, and Scouts
3  Evaluate food sources
4  repeat
       /* Employeed bees' phase*/
5      foreach $x \in Foods$ do
6          $j \leftarrow random(0, N)$, where $N$ is the graph order
7          $y \leftarrow$ random Food source, where $x \neq y$
8          Generate new food source based on $x_j$ and $y_j$ using
             Equation 3.4
9          Convert the new food source to binary using
             Equation 3.5
10         Update $x$ if it is inferior to the new food source
       /* Onlookers bees' phase*/
11     $OP \leftarrow$ Onlooers probability using Equation 3.6
12     for $i = 1$ to $PS$ do
13         if $random(0, 1) \leq OP_i$ then
14             $j \leftarrow random(0, N)$
15             $k \leftarrow random(0, PS)$
16             Generate new food source based on $Foods_{ij}$ and
                 $Foods_{kj}$ using Equation 3.4
17             Convert the new food source to binary using
                 Equation 3.5
18             Update $Foods_i$ if it is inferior to the new food
                 source
       /* Scout bees' phase*/
19     Send an employed bee to a random food source when it
         acrosses its limit of trials
20     if $random(0, 1) \leq lr$ then
21         $k \leftarrow random(0, PS)$
22         $x = LAHC(Foods_k)$                    ▷ call Algorithm 2
23     if $f(x) \leq f(Foods_k)$ then
24         $Foods_k = x$
25 until $Max_{iterations}$ is reached;
   Output: best FoodSource
```

3.6 MODEL EVALUATION

The fourth component of this study's proposed technique, as shown in Figure 3.1, is to evaluate the proposed algorithm. This step is used to assess the performance of the proposed method to include evaluating the basic ABC concerning the *MDS* problem. After that, the binary version of the ABC is examined by performing numerous ABCs.

This includes a comparison of three binary functions: angle modulation, round function, and sigmoid function. The performance of the hybrid ABC is then compared to that of the basic ABC. This test aims to show how the suggested local search affects the ABC algorithm.

3.7 SUMMARY

This chapter discussed the approach used in this study to find a solution to the MDS problem. Each section of this chapter describes the stages of our process. This includes the representation of the addressed problem, which establishes the objective function for evaluating the quality of the generated MDS solution. The constraints of the problem and the encoding of the MDS solution are also described. Following that, the proposed strategy for resolving the MDS problem is thoroughly detailed. This includes illustrating the flow of the algorithms used and providing pseudocode for each algorithm. That is, when necessary, the algorithmic procedures are explained mathematically. Finally, the proposed method's evaluation is discussed. As a result, it is easier to follow along with the results, which will be provided in the following chapter.

CHAPTER IV

RESULTS ANALYSIS AND DISCUSSION

4.1 INTRODUCTION

This chapter presents the findings of the experiments carried out in this study which includes a description of the evaluation criteria and the datasets. Because the proposed method in this study is based on a hybrid algorithm, we present the results of each method to demonstrate the impact of hybridization. The chapter provides the ABC algorithm's results, and its hybrid variant. This chapter is organized into five sections including the current one. Section 4.2 shows the experimental design used in this study, which illustrates the experiments conducted in this study. Section 4.3 describes the dataset used in this study and the evaluation metrics. Section 4.4 is dedicated to analysing the components of the proposed algorithm. Then, Section 4.5 presents a comparison to related works. Finally, Section 5 gives a summary of the chapter.

4.2 EXPERIMENTAL DESIGN

This section explains the experiments carried out in this study. These include a series of experiments designed to demonstrate the effectiveness of each component of the proposed method. Figure 4.1 depicts the three main experiments carried out in this study. The first and second experiments use the ABC and LAHC algorithms to solve the MDS. The first experiment demonstrates which binary conversion function is best for the ABC. The second is devoted to demonstrating the effect of the local search algorithm on the ABC. The third experiment compares the hybrid binary ABC to the related works. . The hybrid ABC is compared to the basic ABC to show the impact of the local search algorithm on the ABC.

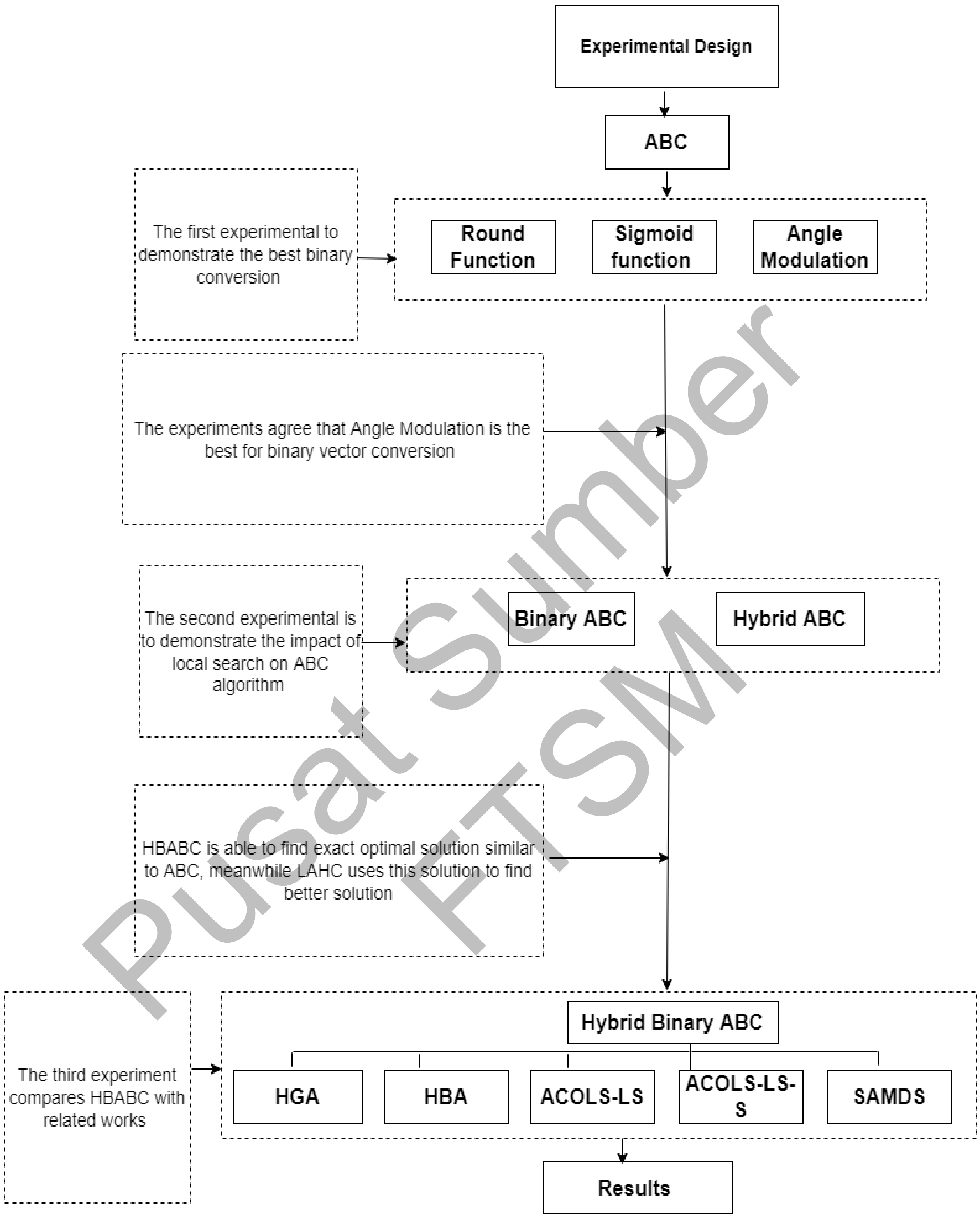
Experimental Design
ABC
The first experimental to demonstrate the best binary conversion
Round Function
Sigmoid function
Angle Modulation
The experiments agree that Angle Modulation is the best for binary vector conversion
The second experimental is to demonstrate the impact of local search on ABC algorithm
Binary ABC
Hybrid ABC
HBABC is able to find exact optimal solution similar to ABC, meanwhile LAHC uses this solution to find better solution
The third experiment compares HBABC with related works
Hybrid Binary ABC
HGA
HBA
ACOLS-LS
ACOLS-LS-S
SAMDS
Results

Figure 4.1 Experimental design of the proposed method

4.3 EVALUATION: DATASETS AND METRICS

4.3.1 Datasets

This section describes the dataset used in this study to evaluate the method. In this study, the proposed method for the MDS problem is evaluated using a dataset of unit disc graphs (UDG), as demonstrated in previous studies (Pinacho-DaviDSon & Blum 2020). UDG is defined as equal-sized intersection graphs in the plane to provide a graph-theoretic model for broadcast networks (cellular networks) and computational geometry problems. There are four regions or spaces in the space. The points are evenly distributed on a plane with a predefined area. UDGs are used to simulate WSN networks in which each node is a wireless sensor network. If the transmission ranges of two nodes overlap, they are connected. As a result, this data has three predefined parameters: the number of nodes, the plane's area, and the transmission range. UDGs are made up of fourteen networks based on these parameters, as shown in Table 4.1 These networks had previously been used in MDS research (Abed & Rais 2017, 2019; Chalupa 2018; Hedar & Ismail 2012; Potluri & Singh 2011). As previously stated, UDGs is used to simulate WSN networks, so the range is an indicator of graph density. In other words, an instance with a large range would have more edges. Table 4.1 divides the networks into two categories based on the sequence of the ranges within each network. The six networks (Net-01, Net-03, Net-07, Net-12, Net-13, and Net-14) have the same range sequence. This range is either 150, 200, or 250 at each step of 50 and will be referred to as identical networks. The remaining networks in the same table are not identical in the range within nor in the step, and they will be referred to as non-identical networks.

Table 4.1 Unit disk graph (UDG) dataset

Network	# Nodes	Node's range	Area
Net-01	50	150, 200, 250	1000X1000
Net-02	80	60, 90, 120	400X400
Net-03	100	150, 200, 250	1000X1000
Net-04	100	80, 90, 100, 110, 120	600X600
Net-05	200	70, 80, 90, 100, 110, 120	700X700
Net-06	200	100, 110, 120, 130, 140, 150, 160	1000X1000
Net-07	250	150, 200, 250	1000X1000
Net-08	250	130, 140, 150, 160	1500X1500
Net-09	300	180, 190, 200, 210, 220	2000X2000
Net-10	350	200, 210, 220, 230	2500X2500
Net-11	400	220, 230, 240	3000X3000
Net-12	500	150, 200, 250	2000X2000
Net-13	750	150, 200, 250	2000X2000
Net-14	1000	150, 200, 250	2000X2000

4.3.2 Evaluation metrics

This study's proposed algorithm is based on stochastic operators, so statistical metrics are used for evaluation (Bäck et al. 1995). These metrics necessitate multiple algorithms runs, i.e., 30 runs for each graph instance. Then, we calculate the mean and standard deviations for each instance and identify the minimum value, as shown below.

- **The minimum run value:** Out of any given number of runs, only the smallest value is considered which represents the best solution obtained by the algorithm.

- **Mean:** For the thirty runs, the average value is depicted in Equation 4.1. The mean is defined as the sum of all values in terms of the fitness function ($f(x_i)$) divided by the total number of runs (N). The mean value shows average solution that can be obtained from the algorithm.

$$Mean = {}^{1}/_{N} \times \sum_{i=1}^{N} f(x_i) \qquad \ldots(4.1)$$

- **Standard Deviation (SD):** SD measures the variation of the algorithm in terms of given values (x_i) and the mean (x') as shown in Equation 4.2. Very small SD indicates more stability of a function than high SD values (Karras et al. 2017). The formula of the SD is shown in Equation 4.2.

$$SD = \sqrt{\frac{1}{N-1} \times \sum_{i=1}^{N} (x_i - x')^2} \qquad \ldots(4.2)$$

4.4 RESULTS ANALYSIS

4.4.1 Results of the binary Conversion Functions

This section displays the ABC algorithm's performance about the binary function, which represents the best solution for the MDS problem. The standard ABC algorithm was designed to solve continuous problems; however, the binary functions adapt the ABC solution to the MDS problem. Figure 4.2, Figure 4.3, Figure 4.4 depicts the performance of the proposed ABC for three typical networks that use a binary conversion function with round, Sigmoid, and angle modulation functions. The networks were chosen at random from Table 4.1 . These nets are Net 1 (range 150, 200, 250), Net 8 (range 130, 140, 150, and 160), and Net 11 (range 220, 230, and 240). The goal of examining these functions is to demonstrate how to convert the ABC algorithm's source foods into binary vectors. Despite the different ranges used in the proposed nets, all nets agreed that the angle modulation function is superior to the round and Sigmoid functions in displaying the fewest nodes, which reflects the best dominating set. The presented result agrees with Yarkin & Coon (2021). The DS size has been measured and tabulated as a result of this experiment. The binary conversion affects the algorithm's performance because it distinguishes the dominant nodes from the others. This result contributes to the formula of this function. This function denotes variables even if their values are less than 0.5 because the round function ignored these variables

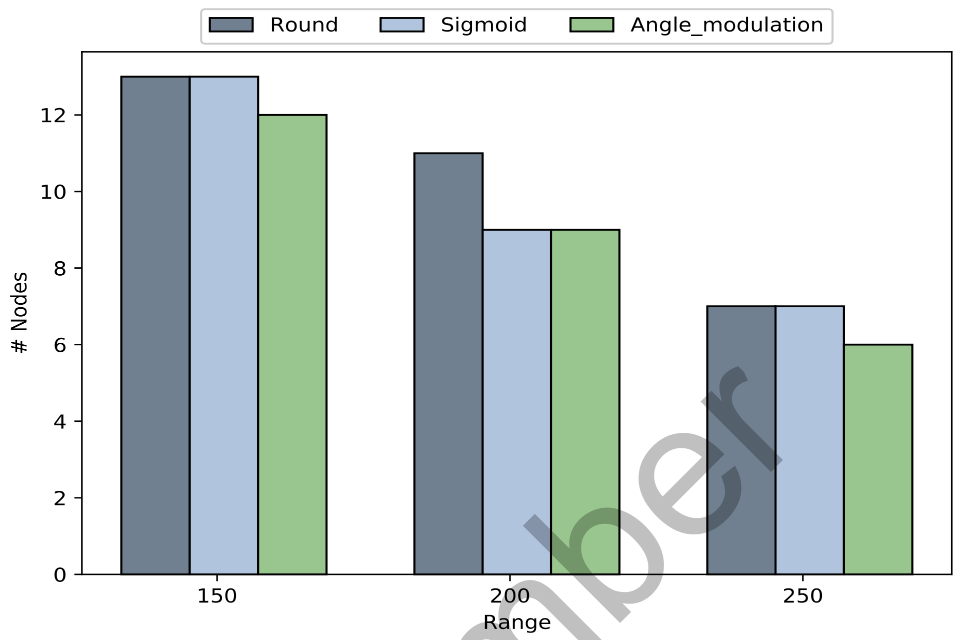

Figure 4.2 The round, sigmoid, and angle modulation functions performance of for (a) Net 1

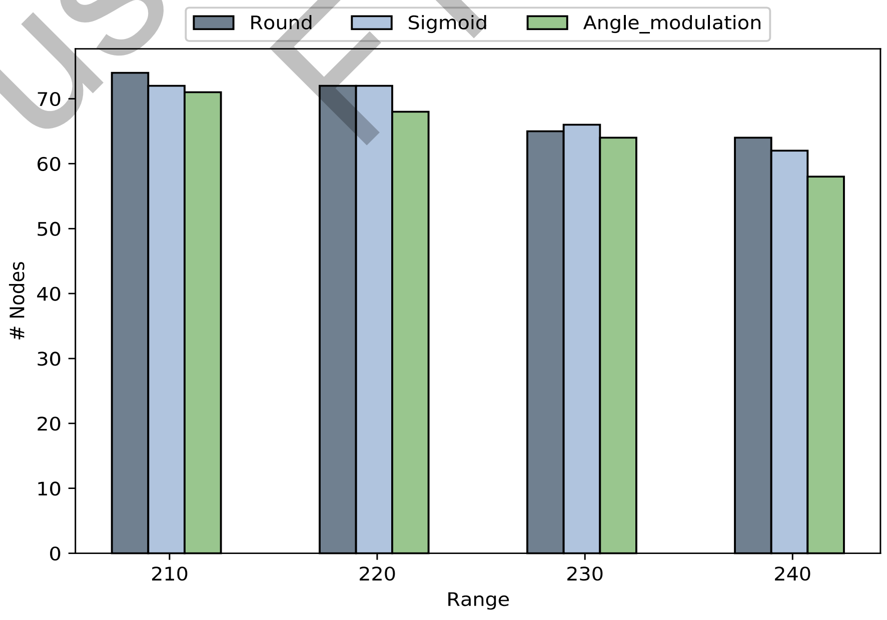

Figure 4.3 The round, sigmoid, and angle modulation functions performance of for (b) Net 8

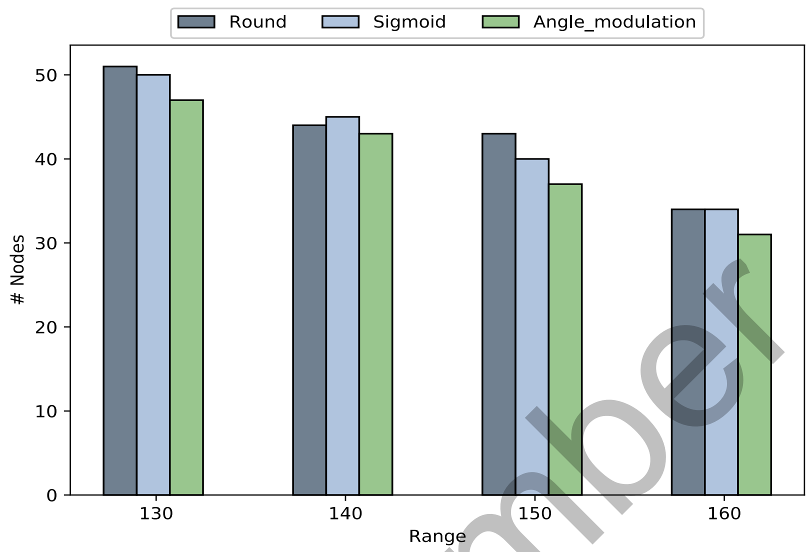

Figure 4.4 The round, sigmoid, and angle modulation functions performance of for (c) Net 11

4.4.2 Results of the hybrid binary ABC

The impact of the local search algorithm on the ABC is depicted in this section. As shown in Figure 4.5, Figure 4.6, Figure 4.7, the proposed hybrid ABC (HABC) was used to minimize the MDS size by demonstrating the algorithm convergence (fitness) over 100 iterations (Liu et al. 2013). The experiment is carried out on Net 1 because it has the lowest order among the other instances, with only 50 nodes. Net 1 has three instances of varying ranges, so the experiment was carried out for all of them. This demonstrates how the algorithm behaves when more solutions are available (denser graph). HABC provides better solutions throughout the search process for the lowest range of 150 (Figure 4.5). ABC performs better in the early stages of the search process for the higher density (range) of 250. However, even at a late stage in the search process, HABC exploited suitable solutions provided by the LAHC, as shown in Figure 4.7. As a result, the HABC was able to find the exact optimal solution as the ABC during previous iterations. The LAHC used this solution to find better solutions in subsequent iterations.

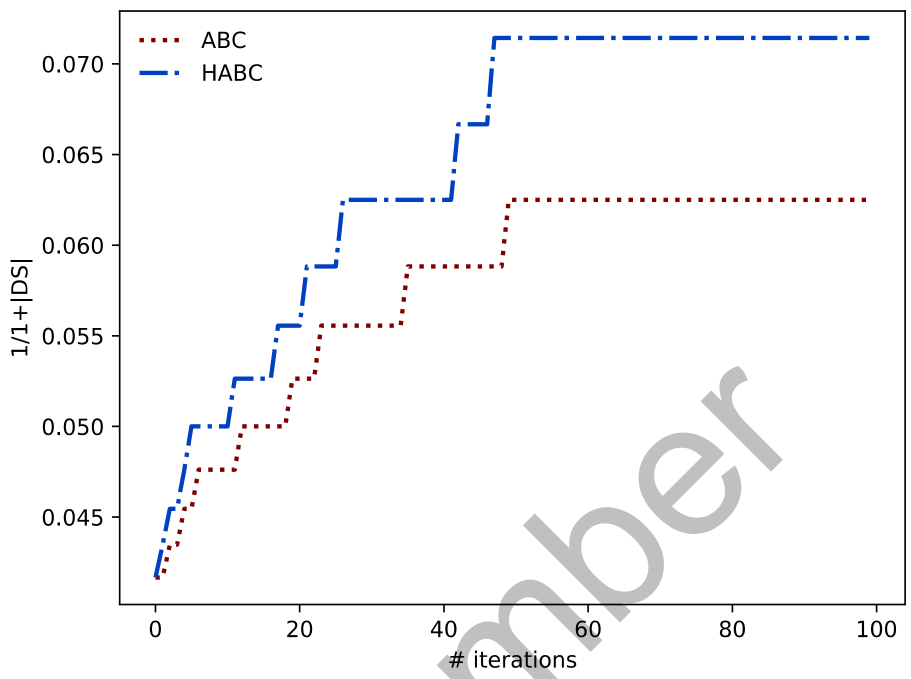

Figure 4.5 Analysis of Hybrid ABC algorithm at range of (a) 150

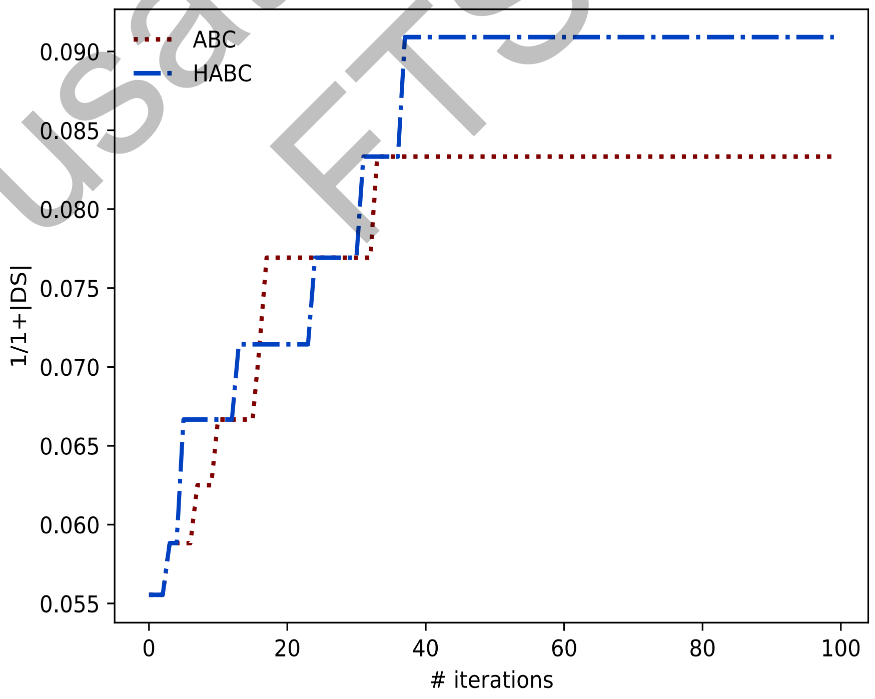

Figure 4.6 Analysis of Hybrid ABC algorithm at range of (b) 200

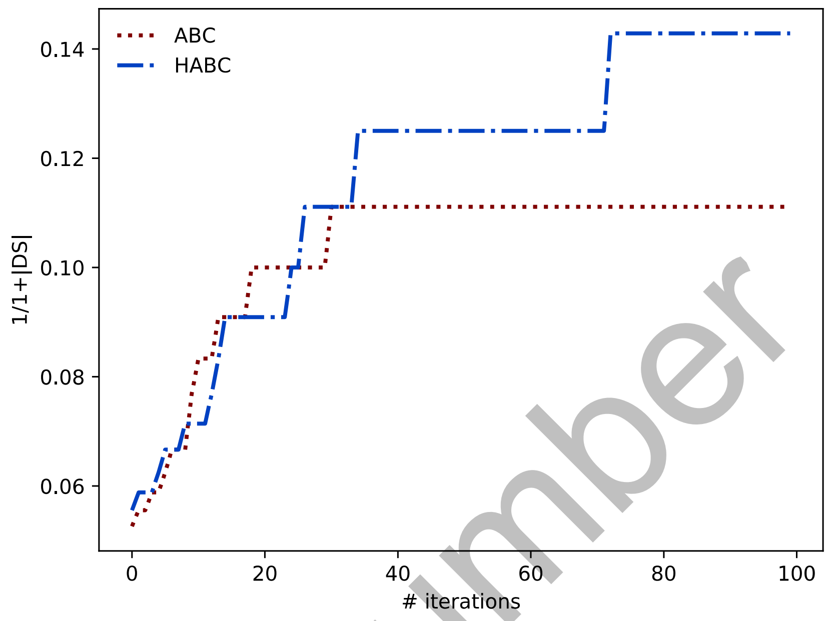

Figure 4.7 Analysis of Hybrid ABC algorithm at range of (c) 250

4.4.3 Comparison to the Related Works based on identical Range Networks

This section compares the currently proposed method to related works using two meta-heuristic algorithms, namely the hybrid bat (BA) algorithm (HBA) (Abed & Rais 2019), the hybrid genetic algorithm (HGA), and the simulated annealing MDS (SAMDS) (Hedar & Ismail 2012). The comparison is based on the data in Table 4.2. The comparison is carried out by selecting nets with ranges that are similar to each other at 150, 200, and 250. Net 1, Net 3, Net 7, Net 12, Net 13, and Net 14 are the names of these nets. HBAB represents the current work, while the algorithms HGA (Hedar & Ismail 2012), ACOLS (Potluri & Singh 2011), and ACO-LS-S (Chalupa 2018) were used for comparison.

Table 4.2 displays the HBABC results for thirty runs, which are reported in terms of the minimum value (min), the mean of the runs (Mean), and the standard deviation (SD). However, in previous works, the SAMDS and HGA were only run for twenty runs. Despite this, our proposed method produced an average MDS size close to the minimum. That is, our proposed algorithm outperforms others in terms of stability. This is due to the ABC algorithm's exploration capability and the LAHC's exploitation procedure. The latter intensifies the search for a suitable solution, allowing the algorithm to produce roughly the same solution on each run.

Table 4.2 Comparison of identical networks with same range of 150, 200, and 250

Graph Instances		HBABC			HGA	ACOLS-LS	ACO-LS-S
Network	Range	Min	Mean	SD	Mean	Mean	Mean
Net-01	150	12	12.1	0.30	12.9	12.9	14.6
	200	9	9.1	0.37	9.4	9.4	10.1
	250	6	6.2	0.4	6.9	6.9	7.1
Net-03	150	18	18.1	0.3	17	17	17.4
	200	10	10.1	0.3	10.4	10.4	10.7
	250	8	8.1	0.37	7.5	7.6	7.5
Net-07	150	19	19.1	0.37	18.7	18.1	18
	200	11	11.3	0.47	11.4	11	11
	250	8	8.2	0.42	8	8	8
Net-12	150	62	62.4	0.49	67.3	64.5	63.8
	200	38	38.3	0.45	41.4	39.8	38.6
	250	25	25.4	0.49	27.9	26.8	25.8
Net-13	150	65	65.3	0.45	72.9	68.7	65.2
	200	39	39.4	0.49	43.9	41.3	38.9
	250	26	26.4	0.48	28.7	27.3	26.3
Net-14	150	66	67.4	0.76	74.8	70.3	66.7
	200	40	41.3	0.74	44.8	42.5	40.4
	250	27	28.5	0.76	29.8	28.2	27

The normalized MDS for three selected Nets were plotted using the three algorithms, namely the current work (HBABC) and comparative works described by HGA, ACOLS-LS, and ACO-LS-S as shown in Figure 4.8, Figure 4.9 Figure 4.10. These Figures shows the performance of the HBABC in comparison to other algorithms when nodes' ranges are identical for all networks. For Net-14, which is the highest graph order, the HBABC has outperformed all methods when node's range is low. While it was inferior to ACO-LS-S when nodes' ranges were high. This is mainly contributed to density of the graph, where the ACO-LS-S is highly dependent on the node's degree as a heuristic function. Also, the same behaviour can be observed for Net-3, and Net-7. However, the HBABC was able to outperform all other methods for other networks. The overall observation shows that HBABC is better in seven of the 18 cases, while it is worse in five of the 18 cases. ACO-LS-S appears to behave similarly to the proposed algorithm HBABC.

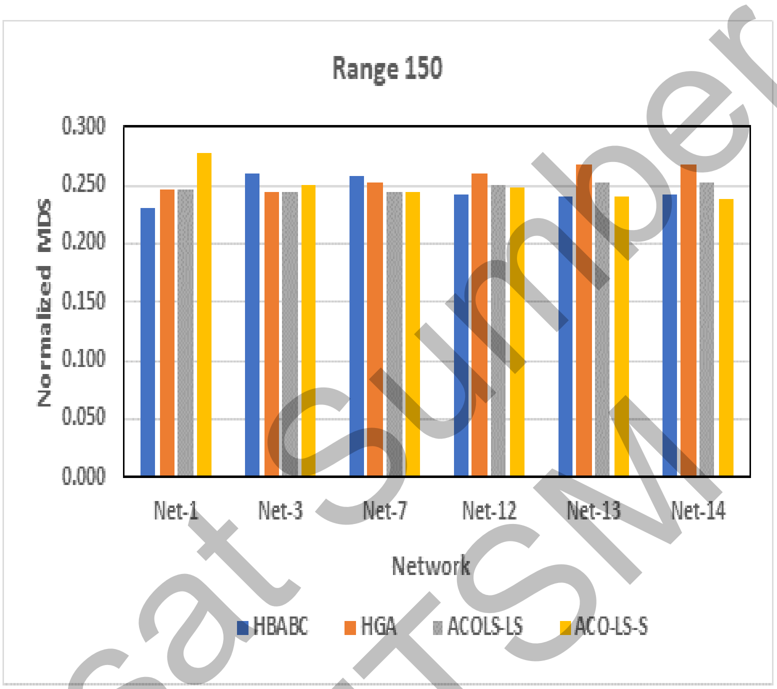

Figure 4.8 Network of Net-01, Net-03, Net-07, Net-12, Net-13, and Net-14 comparison using HBABC, HGA, ACOLS-LS, and ACOLS-LS-S at different range (a) 150

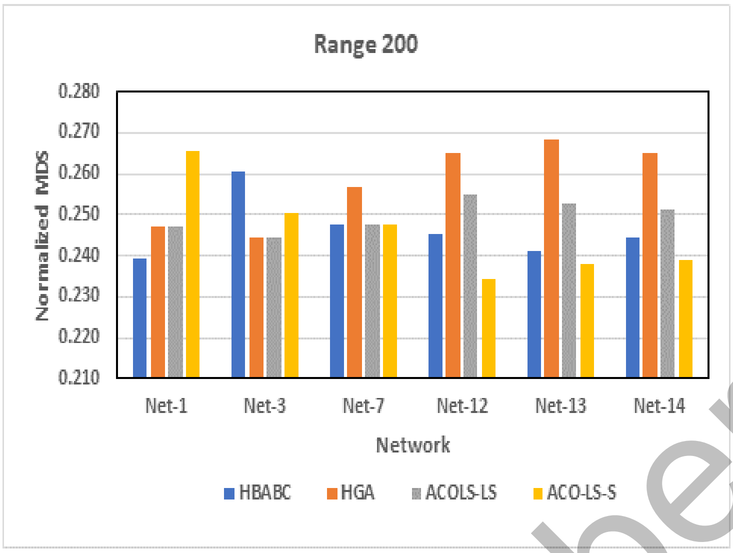

Figure 4.9 Network of Net-01, Net-03, Net-07, Net-12, Net-13, and Net-14 comparison using HBABC, HGA, ACOLS-LS, and ACOLS-LS-S at different range (b) 200

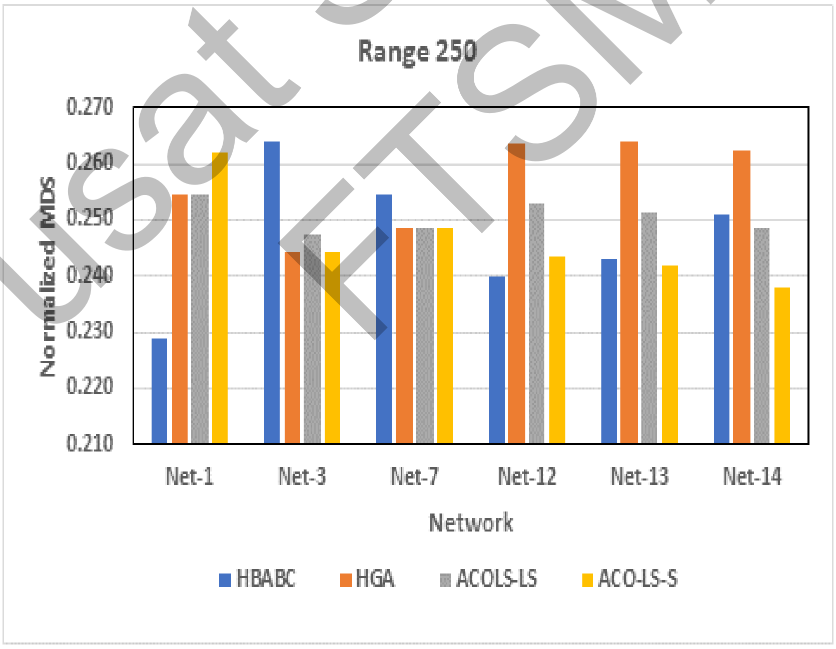

Figure 4.10 Network of Net-01, Net-03, Net-07, Net-12, Net-13, and Net-14 comparison using HBABC, HGA, ACOLS-LS, and ACOLS-LS-S at different range (c) 250

Another approach is to demonstrate the effect of detecting the actual mean of MDS recorded by HABC, HGA, ACOLS-LS, and ACOLS-LS-S with a range variation