DEVELOPMENT OF WHALES OPTIMIZATION ALGORITHM IN SOLVING PATH PLANNING IN PRINTED CIRCUIT BOARD: HOLE DRILLING PROCESS

KOK CHI ZHAO

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEVELOPMENT OF WHALES OPTIMIZATION ALGORITHM IN SOLVING PATH PLANNING IN PRINTED CIRCUIT BOARD: HOLE DRILLING PROCESS

KOK CHI ZHAO

A project report submitted in partial fulfillment of the requirements for the degree of Bachelor of Electronics Engineering Technology with Honours

Faculty of Electronics and Computer Technology and Engineering
Universiti Teknikal Malaysia Melaka

2025

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : DEVELOPMENT OF WHALES OPTIMIZATION ALGORITHM IN SOLVING PATH PLANNING IN PRINTED CIRCUIT BOARD: HOLE DRILLING PROCESS

Sesi Pengajian : 2024/2025

Saya KOK CHI ZHAO mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.

☐ **TIDAK TERHAD**

Disahkan oleh:

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: --

AMAR FAIZ BIN ZAINAL ABIDIN
Pensyarah
Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK)
Universiti Teknikal Malaysia Melaka (UTeM)

Tarikh : 23th Jan 2025

Tarikh : 23th Jan 2025

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled "Development Whales Optimization Algorithm in Solving Path Planning in Printed Circuit Board: Hole Drilling Process" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Student Name	:	KOK CHI ZHAO
Date	:	23th Jan 2025

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology with Honours.

Signature : ....

Supervisor Name : ENCIK AMAR FAIZ BIN ZAINAL ABIDIN

Date : 23 JAN 2025

DEDICATION

I dedicated myself to my beloved family—Father, Mother, and my dear eldest brother—I owe you everything. Your unwavering support, love, and sacrifices have shaped me into who I am today. Through life's ups and downs, you've been my pillars of strength, guiding me with wisdom and kindness. Each day, I cherish the moments we share, the laughter we create, and the bonds that tie us together. Thank you for being my rock, my confidant, and my forever home. Lastly, this dedication is also dedicated to my supervisor, Enick Amar Faiz Bin Zainal Abidin, who always helps me with positive vibes.

Thank you.

ABSTRACT

This thesis explores the application of the Whales Optimization Algorithm (WOA) to optimize the drilling path in Printed Circuit Board (PCB) manufacturing, a critical process in electronic device production. The PCBs require precise drilling for holes to ensure proper component placement and functionality. In traditional ways, problems often result in inefficient drilling sequences, increasing production time and costs. The WOA is inspired by humpback whales' hunting behaviour and addresses these inefficiencies by minimizing the total drilling path length. To minimize the path length by proposing models applied in WOA and mimicking the behaviour of the Whales, such as exploration and exploitation to discover the possibility of the optimal drilling path length. The research involves applying the WOA to the PCB drilling sequences process, with results showing significant improvements in production efficiency and cost-effectiveness. This study concludes the WOA offers a robust solution for PCB drilling optimization, suggesting further reseacrh could enhance the algorithm.

ABSTRAK

Tesis ini meneroka aplikasi Algoritma Pengoptimuman Paus (Whales Optimization Algorithm, WOA) untuk mengoptimumkan laluan penggerudian dalam pembuatan Litar Bercetak (Printed Circuit Board, PCB), yang merupakan proses kritikal dalam penghasilan peranti elektronik. PCB memerlukan penggerudian lubang yang tepat untuk memastikan penempatan komponen dan fungsi yang betul. Secara tradisional, masalah sering timbul dalam bentuk urutan penggerudian yang tidak efisien, yang meningkatkan masa dan kos pengeluaran. WOA yang diilhamkan oleh tingkah laku memburu ikan paus bongkok menangani ketidakefisienan ini dengan meminimumkan jumlah panjang laluan penggerudian. Untuk mencapai matlamat ini, model dicadangkan dalam WOA yang meniru tingkah laku paus, seperti eksplorasi dan eksploitasi, bagi menemui kemungkinan panjang laluan penggerudian yang optimum. Penyelidikan ini melibatkan penerapan WOA dalam proses urutan penggerudian PCB, dengan hasil yang menunjukkan peningkatan ketara dalam kecekapan pengeluaran dan keberkesanan kos. Kajian ini menyimpulkan bahawa WOA menawarkan penyelesaian yang kukuh untuk pengoptimuman penggerudian PCB, dan mencadangkan kajian lanjut untuk mempertingkatkan algoritma ini.

ACKNOWLEDGEMENTS

First, I sincerely thank my supervisor, Encik Amar Faiz Bin Zainal Abidin, from the Electronic and Computer Department at the University of Technical Malaysia, Malacca. His unwavering guidance, words of wisdom, and patience have been instrumental throughout this project. His mentorship has shaped my understanding and fuelled my determination.

I am equally indebted to Universiti Teknikal Malaysia Melaka (UTeM) for their financial support. Their backing enabled me to embark on this journey and achieve the project's goals. Additionally, I sincerely appreciate my classmates from BERZ, who generously shared insights and ideas, enriching the project's development.

My deepest gratitude goes to my parents and family members. Their love, encouragement, and prayers sustained me during the intense study period. Their unwavering belief in my abilities kept me motivated and focused. An honourable mention also goes to me, whose encouragement and understanding were invaluable. And to myself, thank you for your unwavering support and friendship.

Finally, I thank all the staff members at Campus Technology and my classmates. Their cooperation, camaraderie, and willingness to collaborate made this project a collective success. I also want to express my sincere appreciation to the faculty members and other individuals who contributed, even if not explicitly mentioned here.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF SYMBOLS

π	-	Pi for spiral logarithmic
$\sum$	-	Summation of Equation
x_k	-	Coordinate x
k	-	Number holes position
y_k	-	Coordinate y
$\overrightarrow{D}$	-	Distance of agent optimal position
$\overrightarrow{X}$	-	Position of agent
t	-	Number of iteration
$\overrightarrow{X_{rand}}$	-	Random position of agent
$\vec{A}$	-	Vector A
$\vec{C}$	-	Vector C
$\vec{r}$	-	Random number
p	-	Probability between Exploration and Exploitation
l	-	random number between [-1,1]
b	-	A constant for defining the shape of the logarithmic spiral

LIST OF ABBREVIATIONS

d	-	Distance

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter aims to deliver the structure and introduces the brief idea of this project. It will focus on the background of the Printed Circuit Boards drilling process, briefly describe the problem statement, state the objectives and scope of the project, and discuss the project's contribution.

1.2 Background

Printed circuit boards (PCBs) are a component of the world today; they play the role of the backbone for mounting electronic components and providing connections between components. A PCB is a flat board that offers electronic components and supports their electrical connection through the conductive route. The route is made of copper and etched on an insulation board. The PCB simplifies the complex wiring to make the electronic component more compact.

Manufacturing printed circuit boards involves several complex processes that require precision and attention to detail, such as fabrication, drilling, component placement drilling, welding, and testing. Drilling and component placement are particularly crucial to those processes. Drilling is essential for making vias and holes on the PCB and being able to place the component correctly to ensure the circuit works well. Component placement must be more cautious to reduce the heat and noise between components while testing. Hongyan Shi et al. (2022) said these processes are important to ensure the PCB circuit works functionally.

From all of the processes, drilling determines the whole PCB and whether the PCB can still function well. The parameters that need to be considered in path planning before the drilling process are optimization, accuracy, complexity management and cost efficiency. Path planning is optimizing the distance of machine travelling to ensure drilling on the right location to avoid multiple overlapping paths causing production line inefficiency to the manufacturer. As Najwa Wahida et al.(2017) highlighted, effective path planning can improve production lines and PCB's qualities.

1.3 Problem Statement

One of those common problems for drilling a Printed Circuit Board is time spent on point-to-point movement, especially when there are multiple holes to drill. According to Shafie, Naqiuddin and Nor Aiman (2021), optimising the drilling path is crucial for reducing the drilling time and improving productivity by comparing the traditional method with Swarm Intelligence (SI) techniques. Furthermore, N.Wahida et al. (2017) stated that about 70% of the machining time in multiple-hole drilling processes involves tool movement.

Manufacturers today are still concerned about the efficiency of the production of PCBs due to the time spent. Denish et al. (2020) stated a growing need for manufactured parts to increase production rates by optimising the length of the route travelled to drill. D.H. Al-Janan et al.(2016) acknowledge there are always possible ways to improve the efficiency of the drilling paths when the number of hole positions are well managed. To achieve the production goal, the manufacturer needs to enhance the optimization on route travelled around to drill the PCBs.

When manufacturers use the traditional ways for drilling, the cost will be a bottomless pit of expenses, and they need to be concerned about all the parameters of routing the PCB's hole. As Zheng et al. (2011) said, drilling is a particularly complicated machining process, and it becomes more complicated when it is on PCBs. "There are also many traditional drilling path optimization problems that are similar in terms of approaching in the larger Tralling Salesman Problem(TSP) literature", also stated by R.Dewil et al. (2018). The lack of systematic research into PCB drilling processes can cost an enormous of expenses and time even before starting the machining process.

1.4 Project Objective

The main aim of this project is to propose on Printed Circuit Board path palnning using Whales Algorithm Optimization by optimizing the distance of drilling holes with specific coordinate holes. In order to achieve the main objective, the sub-objectives will be the leading as:

a) To review existing computational algorithms and case studies that had been existing in previous literature reviews. Some explanations of the PCB drilling process and Manhattan and Euclidean distance will be elaborated for understanding in terms of applying them in Whales Optimization Algorithms. There will be a table to summarise the information about those algorithms and case studies in Chapter 2.

b) To apply the Whales Optimization Algorithm for optimizing the path planning in the PCB holes drilling process. The details of the algorithm and the adaption to PCB path planning will be explained in Chapter 3

c) To benchmark the Whales Optimization Algorithm (WOA) applied to PCB path Planning with previous literature on using other algorithms on path planning. The initial result of the WOA is discussed in Chapter 4.

1.5 Scope of Project

Scopes are the area to guarantee the project will be within its expected limit. Those scopes will be playing as an area to guarantee the project is heading in the right course with those objectives stated. In this project, the priority is using The Whales Optimization Algorithm for path planning 14 specific holes. Our primary aim is to find the optimized best route for overall distance to drill the particular coordinates of 14 holes. Due to the other, previous optimization algorithm literature for use as comparison and benchmarking with adapted model in this project was also using 14 coordinates holes.

The path planning will be simulated throughout by using Matlab via MATLAB R2023b. has the capabilities to develop a functional Whales Optimization Algorithm for route optimization. The simulation will plot out the optimized route with its total distance for analysis and benchmarking and to ensure the WOA on path planning is applicable, especially if it is routing in the x-axis and y-axis. Matlab is also a platform for this project that is capable of simulating WOA with built-in. The reason that Matlab is used in this project is that in terms of the origin of the Whales Optimization Algorithm is verified.

For path planning, the route is designated on the two-dimensional aspect; therefore, only the path routes through the x-axis and y-axis. Due to the validation result of WOA on path planning hence in this will also using previous literature review with routing in x-axis and y-axis for benchmarking. From calculating the distance of the path in terms of the x-axis and y-axis, the Manhattan distance formulae will be used in this adapted model as the fitness function. However, some algorithms plan the route to move diagonally by using Euclidean distance formulae. Still, the adapted model will be compared with other algorithms that also plotted the path planning in the 2D model, so there will be an imbalance when the comparison of results between other algorithms is done.

Time spent on drilling is inefficient, as stated in the problem statement. Therefore, the total distance of routing the drilling path will be the foremost important aspect to take concern of. By optimizing the total distance, the time spent on drilling will also decrease reasonably.

1.6 Project Contribution

The project will contribute to the industry of PCB manufacturing in terms of improving and enhancing the drilling process. By introducing a path planning algorithm, the project tends to find the time reduction on point-to-point movement while drilling. This improvement will be assist the overall production line of the manufacturing progress. The swarm-based optimization algorithm in this project will be expected to minimize the path planning that the drill bit will travel for drilling, which can reduce costs and increase production rates.

This project will also provide a method for manufacturers to manage the complexity of the PCB drilling, such as optimizing the sequences and hole's position to ensure the efficiency of drilling paths and reduce the path overlapping. To maintain the functionality of the PCB circuits, it is very important to take into account the calculated management of PCB drilling. Furthermore, the adapted algorithms also attempt to be cost-effective in reducing the extra expenses for adjusting parameters as the manufacturers use the traditional drilling method.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction to PCB Drilling Process

The PCB Drilling process is an advanced process that involves specialized machinery to create holes for component mounting and different connections. As Yi, (2024) stated that in the technology of machinery for drilling must have capabilities to handle different types of holes such as Plated Through holes (PTH) that can reach the entire PCB thickness and have conductor on the internal walls and Non-Plated Through Holes (NPTH) refers to no conductors on wall internal walls which are used for installing screws, spacers and supportive pin to hold the PCB.

The drilling process in the machinery starts with the manufacturer creating a detailed drilling map that consists of coordinated holes, the size of the holes and the placement of the PCB. These details will be assigned to the instrument to perform the drilling with a high percentage of accuracy. As the computer numerically controlled machine drilling PCB with the manufacturer designed the finalized PCB layout, including the coordination and size of holes then convert into a drill file for drilling operation but and before initialling the process of drilling the setup on CNC will involves selecting the suitable and specify materials drill bits to achieve high precision of the exact coordinate of PCB hole.

Drilling a printed circuit board is an essential task in terms of accuracy. Path Planning is one of the critical parts of the process, as it determines the most efficient route to drill across the PCB on the specific holes. According to S. NoorFaroque et al. (2015), an automated PCB drilling machine with efficient path planning can improve the stability and accuracy of the drilling process.

Optimizing the path planning for drilling PCBs can reduce the time and cost of the PCB manufacturing process. Several researchers, like N.W.Z Abidin et al. (2015), have used soft computing approaches for multi-hole drilling path optimization to solve the tool movement and switching time, improving the significant time spent drilling.

2.1.1 Manhattan Distance and Euclidean Distance

Distance calculation is a regular mathematical method for calculating the distance from one point to another point. When a clustered graph shows, the measurement of distance shows how close it will be due to the size of the numbers and groups, and it is similar. As the parameters measured are similar, distance measurement is the best method to assess clusters.

Multiple distance measures, such as Minkowski distance, are generalized from different distance measures like Euclidean distance, Manhattan distance, Chebyshev distance and Hamming distance. Two distances have been globally used from those measurements, such as Manhattan Distance and Euclidean Distance. For Euclidean, Distance is one of the commonly used measurements between two points. It is the straight line distance in dimensions in Figure 2.1.

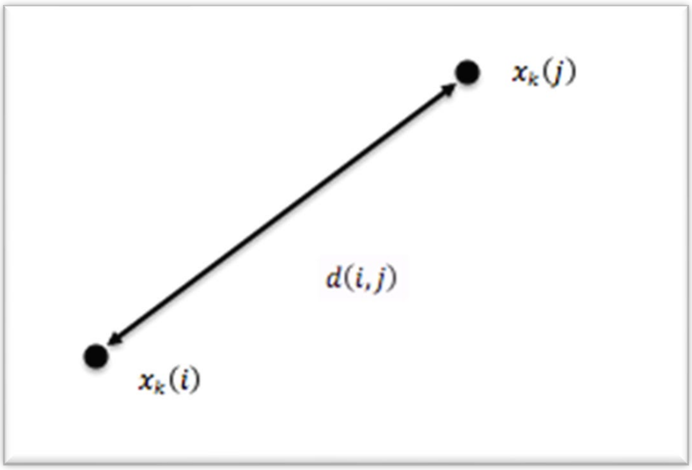

Figure 2-1: Euclidean Distance between two points (Sujan 2015)

Manhattan distance is based on the Manhattan network, which is similar to how streets and buildings are arranged in Manhattan, as stated by Dalfo et al. (2007). Manhattan distance is similar to Euclidean space in terms of two-point but with the x-axis and y-axis coordinate system. The pattern of the Manhattan network and real-life examples of orthogonal streets in Manhattan are in Figure 2.2.

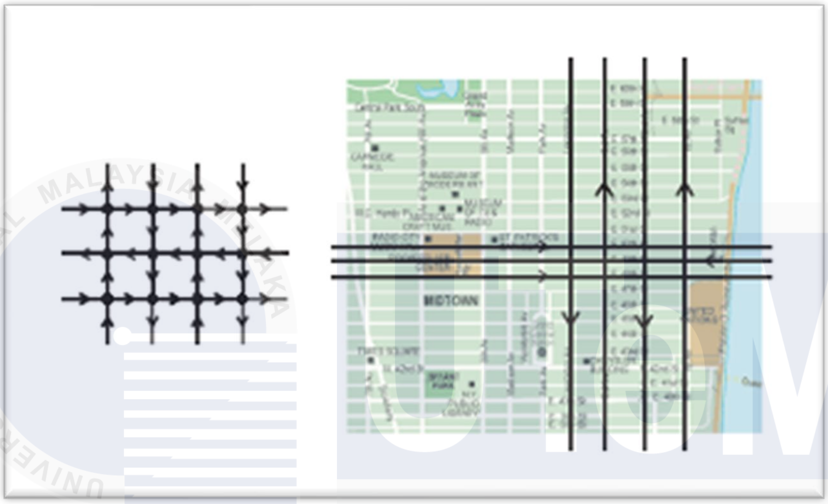

Figure 2-2: Pattern of Manhattan network and orthogonal streets in Manhattan Street

2.1.2 Understanding Manhattan Distance and Euclidean Distance Equation

In terms of the equation of Manhattan Distance and Euclidean Distance, there must be a coordinate of two points for calculating the distance. The Euclidean equation “d” is based on the Pythagorean Theorem, which is the longest side of the triangle between two coordinate points of kth and yet, the so-called hypotenuse, and the other two sides are called opposite and adjacent. The n value is the number of dimensions where the coordinates are positioned, and the k value is the number of coordinates in the dimension. The total distance will be obtained from the subtraction between the second and first coordinates. After that, the x and y values will be squared up as the Pythagorean Theorem states, then summed before the square root's function to get the coordinates' longest side. The Euclidean Distance [2.1] between x^{kth} and y^{kth} is the coordinate.

$$d(x,y) = (\sum_{k=1}^{n}(x_k - x_k)^2 + (y_k - y_k)^2)^{\frac{1}{2}} \qquad [2.1]$$

Equation 2-1: Euclidean Distance

There will also be the same as Euclidean Distance in terms of n and k for Manhattan Distance to calculate the distance between kth and yet, and it is the sum of the horizontal and vertical of the coordinates, but before the summation, there will be a magnitude to positively the subtracted value. The Manhattan Distance equation [2.2] is shown in Figure 2.4.

$$d(x,y) = \sum_{k=1}^{n}|(x_k - x_k)| + |(y_k - y_k)| \qquad [2.2]$$

Equation 2-2: Manhattan Equation

2.1.3 Illustration of Manhattan Distance and Euclidean Distance Equation

The working of Manhattan [2.1] and Euclidean [2.2] distance equation will be shown as below in Figure 2.3 with 4 different coordinates which is (2, 3), (5, 7), (8, 6) and (5, 2) as labelled in 1, 2, 3 and 4 respectivelly.

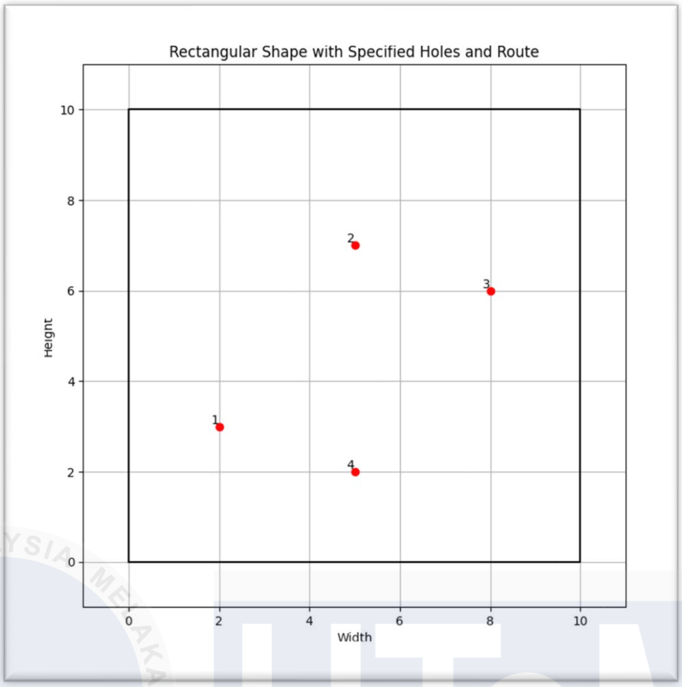

Figure 2-3: 10*10 Rectangular Shape with Specified Holes

The Table 2.1 using Manhattan Distance equation [2.2] with sequence from 1 - 2 - 3 - 4 according the graph above shows below:

$$d(x,y) = \sum_{k=1}^{n} |(x_k - x_k)| + |(y_k - y_k)|$$

Sequence of holes	Calculation
$distance\ 1\ to\ 2$	$(\lvert 5-2\rvert + \lvert 7-3\rvert) = 7$
$distance\ 2\ to\ 3$	$(\lvert 8-5\rvert + \lvert 6-7\rvert) = 4$
$distance\ 3\ to\ 4$	$(\lvert 5-8\rvert + \lvert 2-6\rvert) = 7$
$Total\ distance\ from\ 1 - 2 - 3 - 4$	$(7+4+7) = 18$

Table 2-1: Calculation Manhattan distance with sequence from 1- 2 – 3 – 4

The Table 2.2 of Euclidean distance equation [2.1] with sequence from 1 - 2 - 3 - 4 according the graph above shows below

$$d(x,y) = \left(\sum_{k=1}^{n}(x_k - x_k)^2 + (y_k - y_k)^2\right)^{\frac{1}{2}}$$

Sequence of holes	Calculation
$distance\ 1\ to\ 2$	$((5-2)^2 + (7-3)^2)^{\frac{1}{2}} = 5$
$distance\ 2\ to\ 3$	$((8-5)^2 + (6-7)^2)^{\frac{1}{2}} = 3.16$
$distance\ 3\ to\ 4$	$((5-8)^2 + (2-6)^2)^{\frac{1}{2}} = 5$
$Total\ distance\ from\ 1-2-3-4$	$(5 + 3.16 + 5) = 13.16$

Table 2-2: : Calculation Euclidean distance with sequence from 1 - 2 - 3 – 4

The Figure 2.4 shows the connected coordinates in Manhattan distance and Euclidean distance which blue dash line represent Manhattan distance that only connected through x -axis and y-axis and Euclidean Distance connected through the longest line as green line.

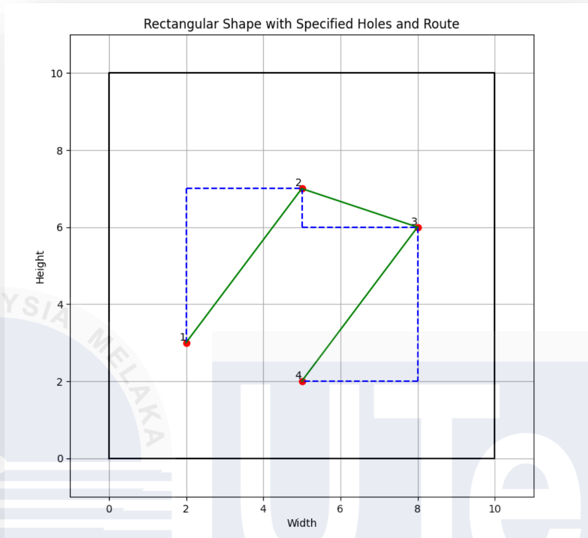

Figure 2-4 Connected Coordinates blue dash for Manhattan distance and Green line for Euclidean Distance

2.2 Whales Optimization Algorithm

Stochastic Algorithms developed on Path Planning

Algorithm	Inspiration	Author & Year
Global Convergence PSO (GCPSO)	Bird Flock	Zhu. 2006
Binary PSO	Bird Flock	Othman et al, 2011
Particle Swarm Optimization (PSO)	Bird Flock	Adam et al., 2010
Simulated Kalman Filter (SKF)	Kalman Filter	Nor Hidayati et al, 2016
Cuckoo Search Genetic Algorithm (CSGA)	Cuckoo	Wei Chen Esmonde et al.,2014
Ant Colony System Optimization (ACS)	Ants	Saealal et al., 2011
Travelled Salesman Problem (TSP)	Travelling Salesman	Eiichi Aoyama et al., 2004
Genetic	Natural Evolution	Denish Khatiwada et al, 2019
Modified Shuffled Frog Leaping (mSL)	Frog Leaping	A.M. Dalavi., 2019
Bat	Bat	Sunny Diyaley., 2019
Firefly	Fireflies	Asrani Lit., 2011
Gravitational Search Algorithm (GSA)	Gravitation	Omar et al, 2014

Table 2-3 : Algorithms applied on PCB Path Planning

Optimization Algorithm is mostly called as computational intelligent that planned to use for finding the best solution for its respective problem. The algorithm explore in it respective space to get the solution and by setting up the parameters to improve the objective function. There are various type of optimization algorithms that have been done and there are also have some different classification of algorithms such as deterministic algorithms and stochastic algorithms. A different stochastics algorithm on PCB path planning will be stated in Table 2.3. As whales optimization algorithm are inspired by humpback whales's bubble-net hunting prey behaviour and randomness of this behaviour makes it classfy as meta-heuristic algorithms. Whales Optimization Algorithm classification as computational intelligence which bound with nature-inspired algorithms for solving optimization and decision making problems that lays on biology-based CI (BbCI) stated Nadim Rana et al.,(2020). There will be graphs and charts to show the research of the whales optimization algorithms.

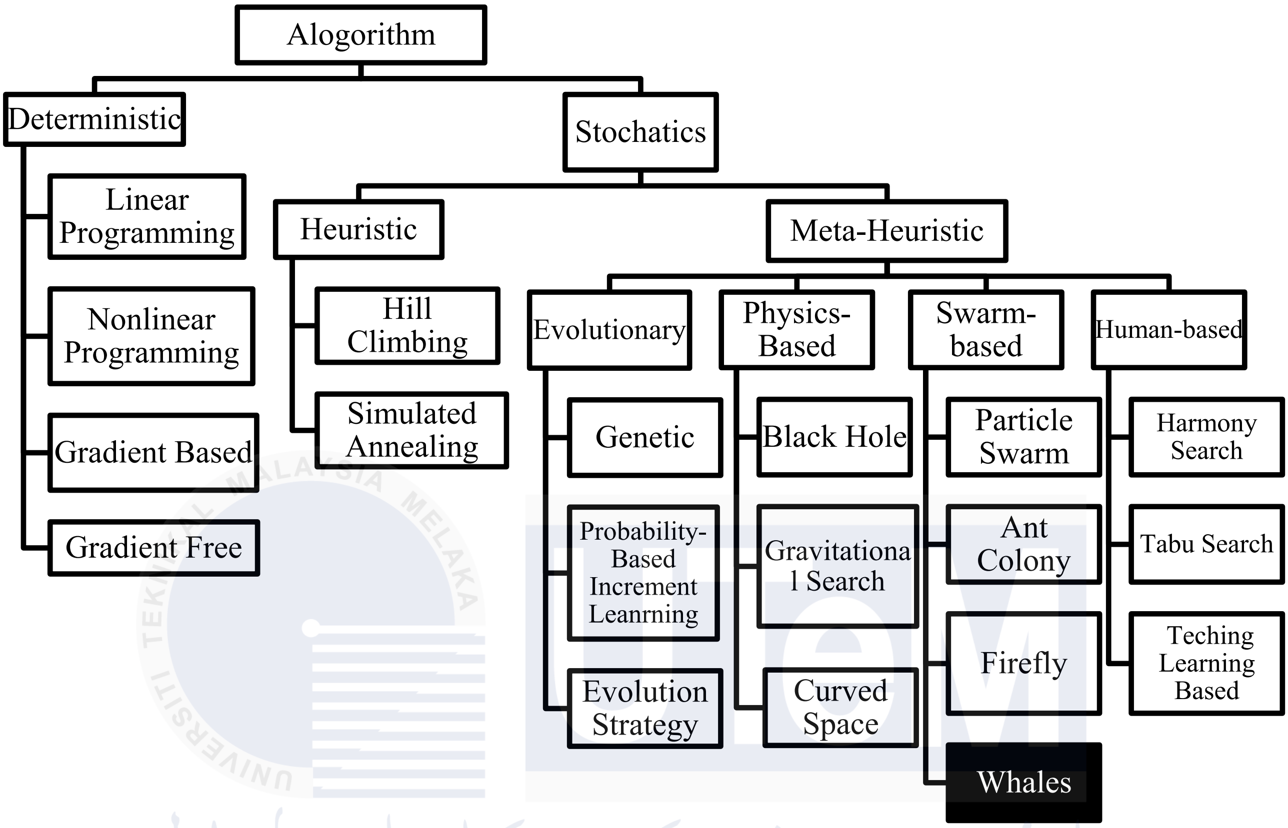

Figure 2-5: Classification of Algorithms

Whales Optimization Algorithm as one of the meta-heuristic swarm-based intelligent algorithm when compared with other swarm intelligence methods, it is simple to implement and robust which make WOA comparable with different nature inspired algorithms. The classification of WOA shown in Figure 2.5. In WOA, the population of humpback whales will search through a multi-dimensional space for food. The location of each agent are represented as different decision variables and the distance between agent with the food represents the value of objective cost. There are 3 operational processes that measured the location of the agent which are shrinking encircling prey, bubble-net attacking method (exploitation phase) and search for prey (exploration phase). The basic principle of the Bubble-net feeding behavior of humpback whales will show in Figure 2.6.

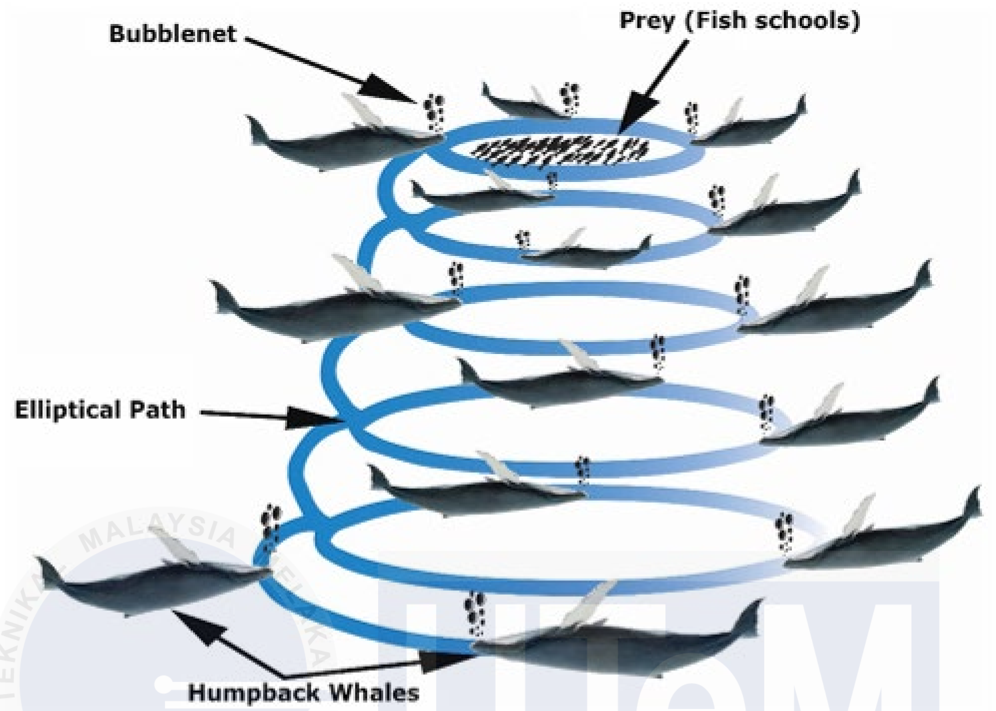

Figure 2-6 : Bubble-net feeding behavior of humpback whales

The pseudo code shows in Figure 2.7 represents the each Whales (X_i) for initial whales when they have recognize the proximate location of the prey, agents will start encircling the prey and calculate the agents optimal position following the equation [2.3] and the position of each agent in the search space will be calculated for finding the best position to update and applied on equation [4] which both equations will be in the while loop. The fitness of each agent will calculate using the objective function or fitness equation for it specific problem.

The fitness function is the goal to find the solution with highest or lowest fitness value depending on problems. X* is the best fitness from the initial population and represent a reference point for other agent during the optimization. There is a *t* indicates the current iteration for the number of find the best position. There will be a loop of each iteration through each agent in the population to get the best position. There are multiples parameter that used in this algorithm is playing a critical role in exploration and exploitation phases of the algorithm such as $\vec{a}, \vec{A}, \vec{C}, l\ and\ \ p$. The $\vec{a}$ where consists in equation 2.9 is linearly decreased from 2 to 0 over the course of iterations to controls the search radius of each agent, which a high value will allow a larger search jump and vice versa in both exploration and exploitation phase. $\vec{A}$ and $\vec{C}$ are coefficient vectors as their equation 2.9 and 2.10. The $\vec{A}$ is a random value representing [-1,1] as a range which will force the agent to move away from other agents to prevent matching up the same agent and getting the same position in exploration phase. For $\vec{A}$ in exploitation phase is also a random value in the interval [-a, a] where $\vec{a}$ is decreased from 2 to 0 over course of iterations. The $\vec{C}$ in equation 2.10 represent a number used between 2 and 0 determined by $\vec{r}$ is a random vector in between [0,1]. For 'l' is a random number between [-1,1] and is an element-by-element multiplication in equation 2.7. By acknowledging the humpback whales swim around the prey within shrinking circle and along a spiral-shaped path simultaneously. To simulate this behavior, there is a probability of 50% to choose between exploration or exploitation to update to position of agents during optimization and it represented as 'p' as stated in the pseudo code. The determination to decide between exploration and exploitation strategies, the random number of p is playing a critical parameter when the value of p is less than 0.5 and the exploration phase will be taking part. In the exploration phase, a parameter $|A|$ will be considered in between greater than 1 or less than 1. A random search agent will be chosen and update the position when $|A|$ is less than 1 by using the equation 2.4, while the best solution will be updated from the position of the agent when $|A|$ is greater than 1 using the equation 2.6 where X_{rand} is a random position vector (a random whale) chosen from the current population. While for random value p is greater than 0.5, the exploitation phase will be taking part to calculate and update the position between whales and prey by using equation 2.7 from mimic the helix shaped movement of the humpback whale.

$$\overrightarrow{D} = \left|\vec{C} \cdot \overrightarrow{X^*}(t) - \vec{X}(t)\right| \quad [2.3]$$

Equation 2-3: Distance of Initial agent optimal position

$$\overrightarrow{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D} \quad [2.4]$$

Equation 2-4: Position for Initial Exploration Phase

$$\overrightarrow{D} = \left|\vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X}\right| \quad [2.5]$$

Equation 2-5: Distance of agent optimal position

$$\overrightarrow{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot \vec{D} \quad [2.6]$$

Equation 2-6: Position for Exploration Phase

$$\vec{X}(t+1) = \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) \quad [2.7]$$

Equation 2-7: Spiral Equation for Exploitation Phase

$$\overrightarrow{D'} = \overrightarrow{X^*}(t) - \vec{X}(t) \quad [2.8]$$

Equation 2-8: Distance of the agent to prey

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad [2.9]$$

$$\vec{C} = 2 \cdot \vec{r} \quad [2.10]$$

Equation 2-9: vector A and C

```
Initialize the whales population Xi(i = 1, 2, ..., n)

Calculate the fitness of each search agent

X*=the best search agent

while(t < maximum number of iterations)

    for each search agent

        Update a, A, C, l, and p

          if1(p<0.5)

            if2(|A|< 1)

               Update the position of the current search agent by the Eq.[3]

            else if2(|A| >= 1)

                Select a random search agent ( Xrandom)

                Update the position of the current search agent by the Eq.[

            end if2

          elseif1(p >= 0.5)

             Update the position of the current searchby the Eq. (2.5)

          end if1

    end for

Check if any search agent goes beyond the search space and amend it

Calculate the fitness of each search agent

Update X* if there is a better solution

t=t+1

end while

return X*
```

Figure 2-7 : Pseudo code for Whales Algorithm

2.2.1 TagCloud of WOA applied

The keywords from Scopus has classified out that those application with WOA applied from 2016 – 2024 has been arrange as Tag Cloud for better visual for acknowledge the fields for WOA mostly applied in Figure 2.8. The Application has search according to the article that have been published in Scopus. The WOA is mimicking the hunting behaviors of the whales and find the optimal solution in various domain, from that the algorithms is mostly used as optimization algorithms, forecasting and supporting vector machines, and others such as PID controllers, Cluster Analysis and etc are those the least published articles among this Tag Cloud.

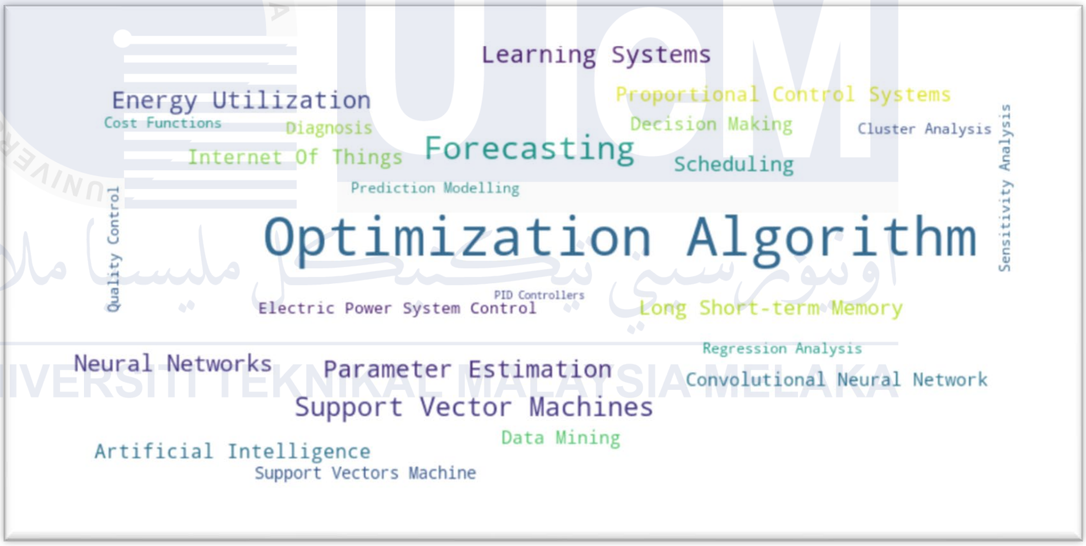

Figure 2-8: Tag Cloud for WOA Application

2.2.2 Research Paper Whales Optimization Algorithm by Countries

The research paper about Whales Optimization Algorithm that have published from 2016 – 2024 in Computer Science and Engineering fields, the Chinese is conquering following by Indians is the second Country. Mostly WOA research paper were published by Asia Countries due to concentrated efforts such as research trends and communities. The researchers from Asian countries are actively explore through novel optimization techniques, such as Particle Swarm Optimization(PSO), Ant Colony Optimization(ACO) and etc., including WOA. Others country which is the lowest contributing in this WOA are Americas, Europeans, and Africa Continent. The percentages of published WOA Article shows in Figure 2.9.

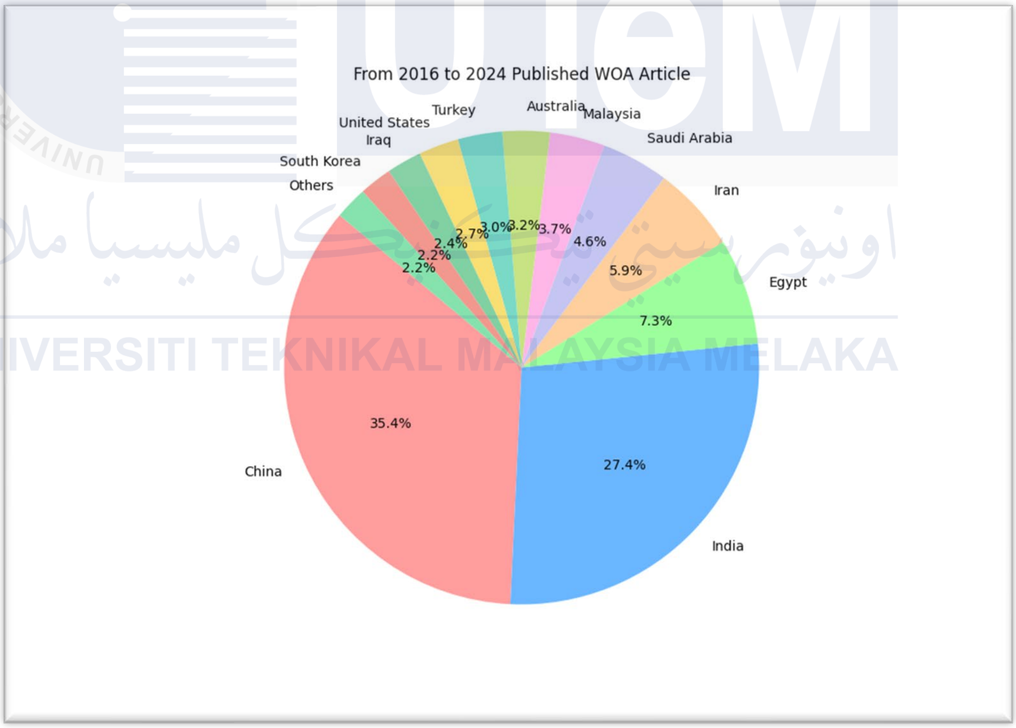

Figure 2-9: Research Paper by Countries

2.2.3 Research Paper that published

The WOA research paper was started publishing start from 2016 due to Whales behaviors were only discovered that it is able to apply as Optimization Algorithm. The trend were starting significantly by years after the algorithm were discovered. According to the charts, there will be over thousands of article publish due to there are over 500 article published since the starting of this year. The growth of WOA is shown in Figure 2.10.

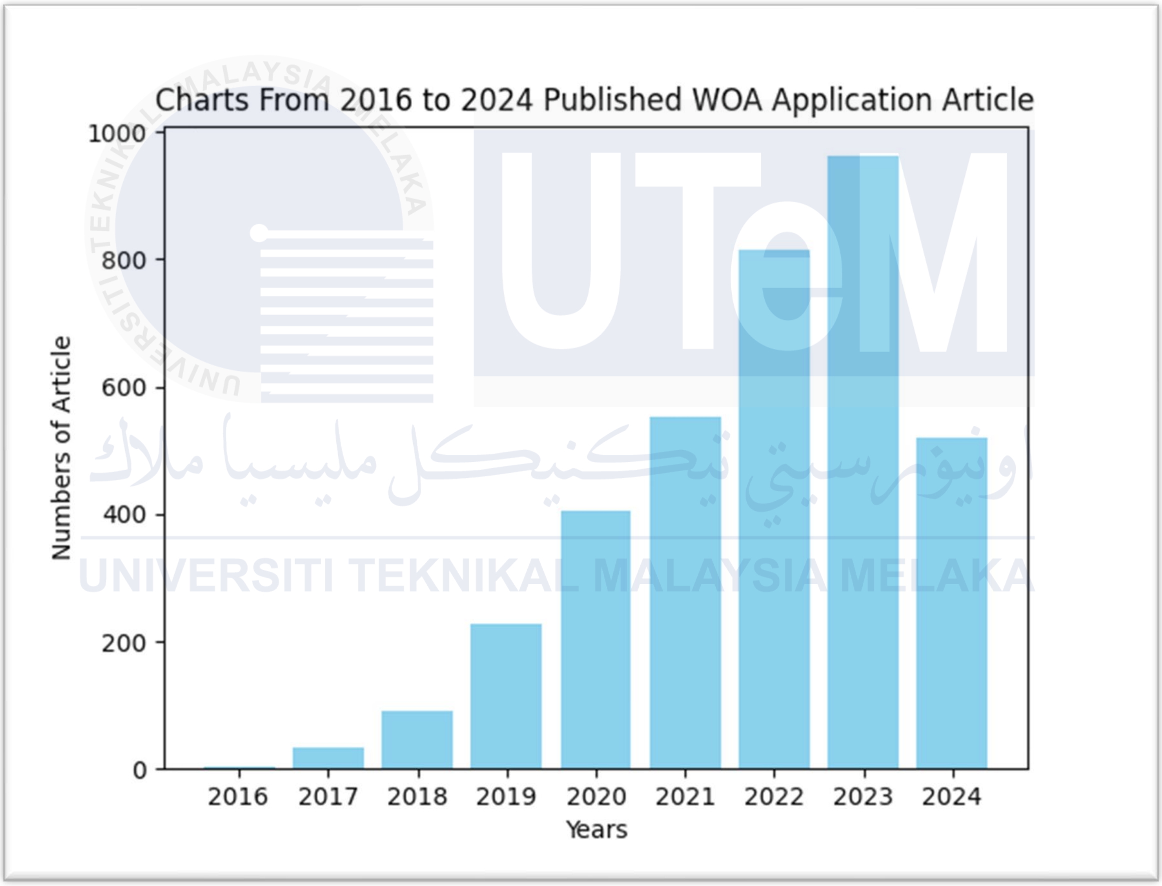

Figure 2-10: Growth of WOA research paper

2.3 Past Related Research Paper

The past research papers are focused on the others Optimization Algorithm for path planning. Therefore, the research are related to the working of optimization algorithms working on different situation such as fourteen holes that some researches have been used or hundreds of specific holes and its path planning. Those research are validate researches or article from authorized website such as Science Direct and Scopus.

2.3.1 Optimization of Drilling Process Using Non-Conventional Method

This research was published by H.Abdullah et al, (2020) in Malaysia, it is aimed at minimize the tool path length in the drilling process in order to decrease the drilling time. The optimization algorithm that this paper using is Ant Colony Optimization and Particle Swarm Optimization. Both of the algorithms are applied on minimize the drilling path for 158 holes in Solidwork Software shows in Figure 2.11. Their result shows that PSO and ACO can reduce the tool path length as it compared with other tool path length that produced by Mastercam software shows in Table 2.4.

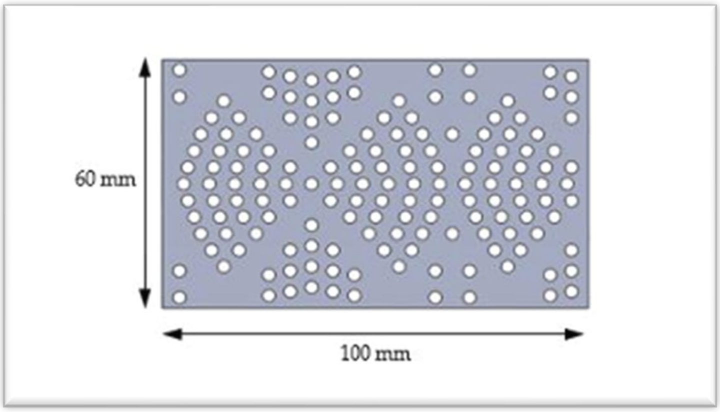

Figure 2-11: A rectangular workpiece with 158 holes

Method	Total path length, (mm)
ACO	970.4575
PSO	947.5632
GA	1108.1375
MasterCam	2707.529

Table 2-4: The Comparison total path length between 4 methods

2.3.2 Optimization of Drilling Path Planning for A Rectangular Matrix of holes Using Ant Colony Optimization

This research was published by A.T. Abbas et al, (2011) from Egypt. The paper applied the Ant Colony Optimization Algorithm (ACO) for path planning of a Computer Numerically Controlled (CNC) drilling on different rectangular matrices with different numbers of holes, and the parameters are shown in Table 2.5. This mainly solves the total drilling time, which is the travelling salesman problem. A Modified ACO algorithm is proposed in this research, and its comparison between modified ACO, basic ACO and Genetic Algorithms in terms of tool travel distance. The modified ACO is modified in terms of the initial pheromone matrix so that the agent will go from one space to another, which is called neighborhood space, without knowing the space is interconnected to improve its performance.

	Case Study		
	1	2	3
Layout of the matrix of holes	4 * 4	5 * 5	11 * 11
Total number of Holes	20	25	121
u-directional (P_u)	100.0mm	100.0mm	100.0mm
v-directional (P_v)	50.0mm	50.0mm	50.0mm
Lower Bound on optimum path length ($f_{L.B}$)	1300.0mm	1650.0mm	7050.0mm
Number of runs performed for each algorithm	50	50	50

Table 2-5 : Summary of the case study problems

2.3.3 Optimization of the multi-hole drilling path sequence for concentric circular patterns

This research paper was published by S.Diyaley et al, (2020) and originated from India. Optimizing path sequence is the primary task for facilities substantial reduction in tool travelling distance and drilling time. There will be comparison between multiples algorithms such as ant colony optimization, artificial bee colony algorithm, particle swarm optimization, firefly algorithm, differential evolution, and teaching learning-based optimization algorithm for determining the optimal path sequences with the traditional technique which is spiral path method in CNC operation due to this research is applied those algorithms on four and five consentric circular patterns of holes, and a heat exchanger tube sheet with two thousand and six hundreds of holes as three different case studies. The parameters of three case studies are shows below Figure 2.12 and Table 2.6 and 2.7.

Figure 2-12: A typical heat exchanger tube sheet

Iest problem	1	2
Number of concentric circles	4	5
Inner pitch circle diameter (mm)	30	50
Pitch circle diametric increment (mm)	15	50
Iotal number of holes	120	225
Length of the spiral path (mm)	718.310	2583.01

Table 2-6: Parameters for Case Studies 1 & 2

Iole diameter (mm)	4
Iole depth (mm)	3
iameter of first pitch circle (mm)	32
ncrement in pitch circle diameters (mm)	16
lumber of holes in the first pitch circle	8
ncrement in number of holes per pitch circle	8
lumber of concentric pitch circles	25
otal number of holes	2600
ength of the spiral path (mm)	17,897.79

Table 2-7: Parameters for Case Studies 3

2.3.4 Optimizing drilling conditions in printed circuit board by considering hole quality

This research was published by E.Aoyama et al, (2004) from Japan. It is focused on solving the movement time for drilling process on the printed circuit board. The method that was applying on this research are conventional and Tracel Saleman Problem (TSP) for certain case studies. The case studies for comparison between conventional method and Travel Saleman Problem are that a group of contained fifty holes and a combined four group that contain fifty holes per group shows in Figure 2.13 and 2.14 respectively. There is a third case study with a group of 20 holes arranged in random on the PCB. The comparison between those method are separated to four patterns which is the conventional method and TSP will applied to the each group continuouly and another is both method applied to all holes as a target. For the third case study were abit challenging due to those 20 holes are having randomly arranged in Figure 2.15 and will affect the shortening ratio therefore certain formulae is modified for TSP.

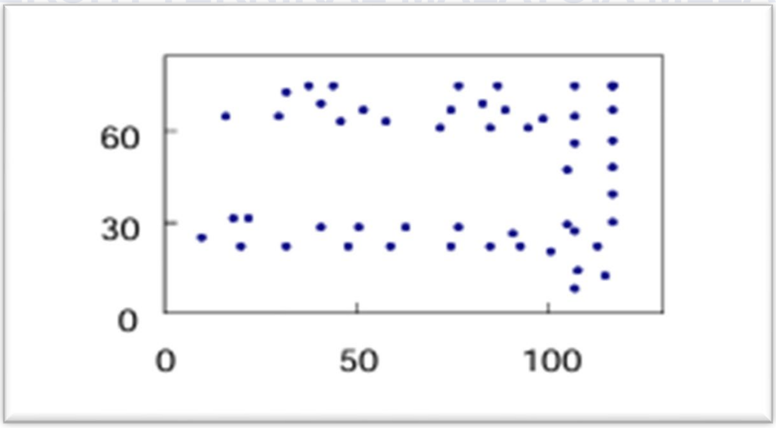

Figure 2-13: Case Study with 50 holes

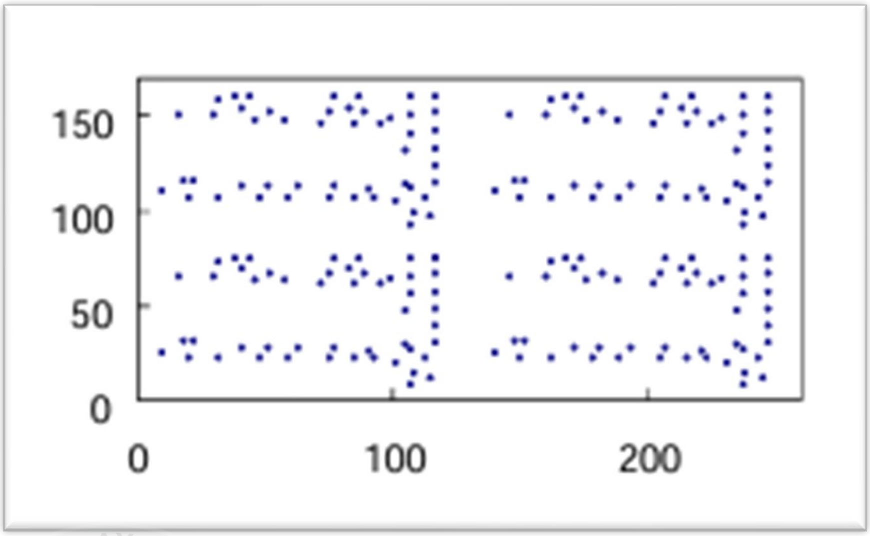

Figure 2-14: Case Study with 4 group of 50 holes combined

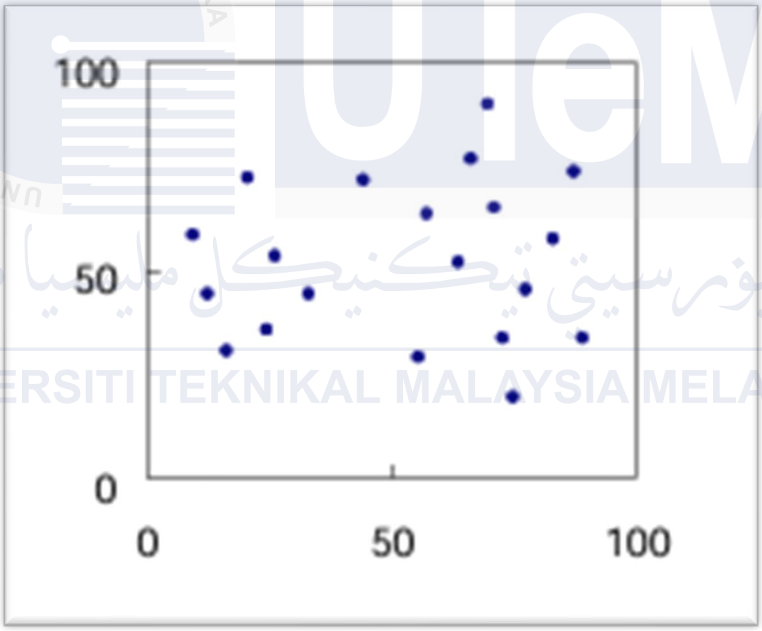

Figure 2-15: Case study with 20 holes

2.3.5 PCB Drill Path Optimization by Combinatorial Cuckoo Search Algorithm

This research was published by W.C.E Lim et al, (2014) from Malaysia. It is focused on solving the the drilling path optimization problem on the printed circuit board. For this paper, the proposed algorithm is the Combinatorial Cuckoo Search from the modified original Cuckoo Search Algorithm, where the agent can not only lay one egg at every net, but it can lay eggs at every nest for simulating existing better solutions for a better combinations. Furthermore, the modification from the original algorithm selects the best solution to pass on to the next generation to ensure the algorithm progresses properly. The third modification is considered to introduce a similarity-based mutation to explore promising areas. These modifications can ensure the improved efficiency of the algorithm. There are two case studies conducted where only 5 holes for different worktable movements from the BCS algorithm, shown in Figures 2.16 and 2.17. The worktable movements are separated to two movement which the drill will finish travelled x direction and then continue with the y direction and the for second movement is both direction are allowed to travel simultaneously.

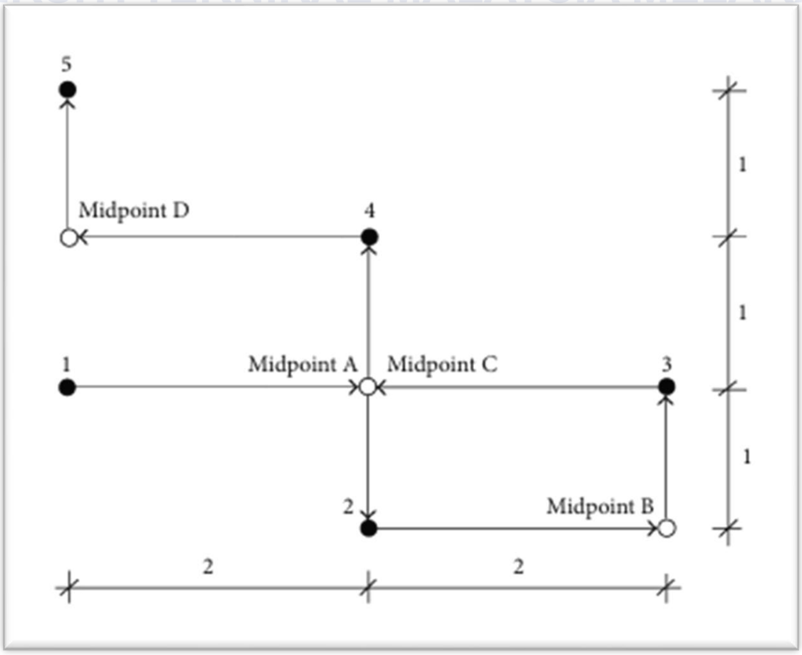

Figure 2-16: Worktable movement for Case Study 1

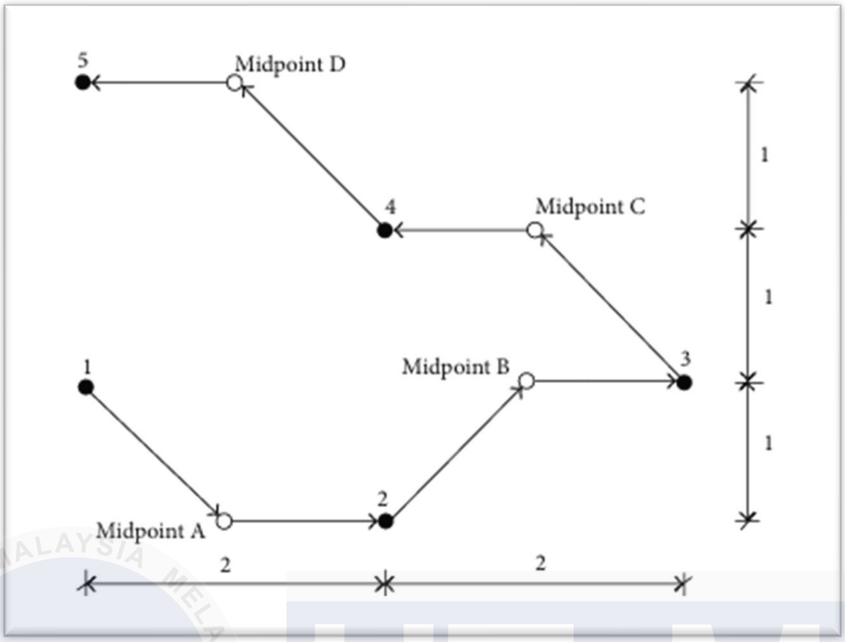

Figure 2-17: Worktable Movement for Case Study 2

2.3.6 Tool path optimization for drilling holes using Genetic Algorithm

This research was published by D.Khatiwada et al. (2020) and originated from Nepal. It attempts to optimize the drilling path in terms of drilling a large number of holes. For this paper, the proposed Genetic Algorithm is used to solve some case studies with a large number of holes, as shown in Figure 2.18. The case studies take the path obtained by commercial software NC plot for comparison with the optimized tool path using the Genetic Algorithm.

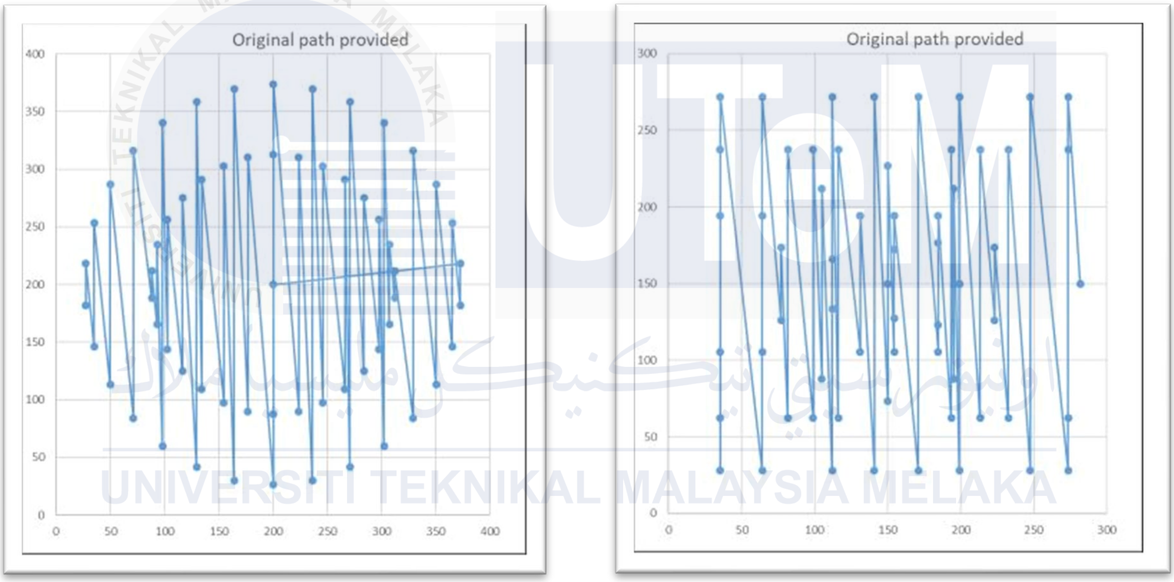

Figure 2-18: Raw Data of from Solidworks Model

2.3.7 Tool Path Optimization for Computer Numerical Control Machines based on Parallel ACO Algorithm

This research was published by N.M. Rodriguez et al, (2012) and originally from Mexico. The paper purpose to optimize the tool path for CNC machines drilling holes in PCB. By achiving the purpose, the optimization method is using Parallel Ant Colony Optimization Algorithm which the agent can explore different parts of problem space at the same time. There are three experiments that conduct using Parallel-ACO with having different holes and different parameters respectively in this article which are 10, 27 and 45 holes in Figure 2.8, 2.9 and 2.10.

PARAMETERS FOR EXPERIMENT I

Description	I-A	I-B	I-C
Num. Holes	10	10	10
Num. Ants	30	30	30
Iterations	30	30	30
Alpha	1	2	2.5
Beta	4	4.5	4

Table 2-8: Parameter For Experiment 1 with 10 holes

PARAMETERS FOR EXPERIMENT II

Description	2 – A	2 – B	2 – C
Num. Holes	27	27	27
Num. Ants	30	30	30
Iterations	30	30	30
Alpha	1	2	2.5
Beta	4	4.5	4

Table 2-9: Parameter For Experiement 2 with 72 holes

PARAMETERS FOR EXPERIMENT III

Description	3 – A	3 – B	3 – C
Num. Holes	45	45	45
Num. Ants	70	70	70
Iterations	70	70	70
Alpha	1	2	2.5
Beta	4	4.5	4

Table 2-10: Parameter For Experiment 3 with 45 holes

2.3.8 Solving Single Tool Hole Drilling Path Optimization Problems using Evolutionary Algorithm

This research was published by M.Madic et al, (2023) and originated from Serbia. The paper purpose to investigate whether there exist better hole drilling sequence with optimal drill travel path. To solve the travelling salesman problem, the method using in this paper is Evolutionary Alogorithm (EA). There are 3 case studies with diffirent of multiple holes applied to the EA that have been conducted in this paper. The number of holes from the case1, 2, and 3 are 15, 18 and 22 respectively in Figure 2.19, 2.20 and 2.21. Those optimal result from EA are compared with the result obtained using Feature CAM ommercial software.

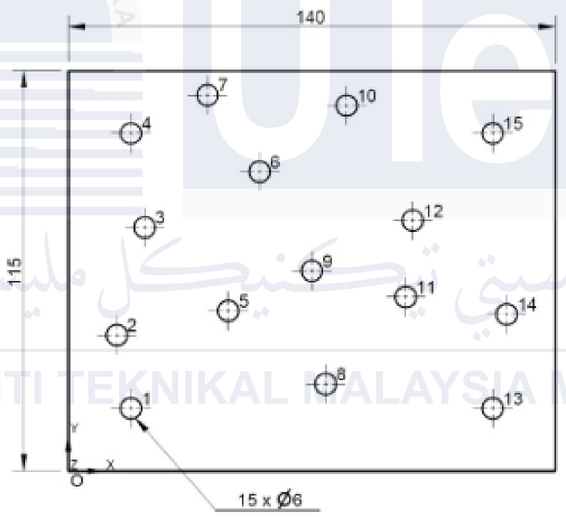

Figure 2-19: Placement of holes for Case Study 1 (15 holes)

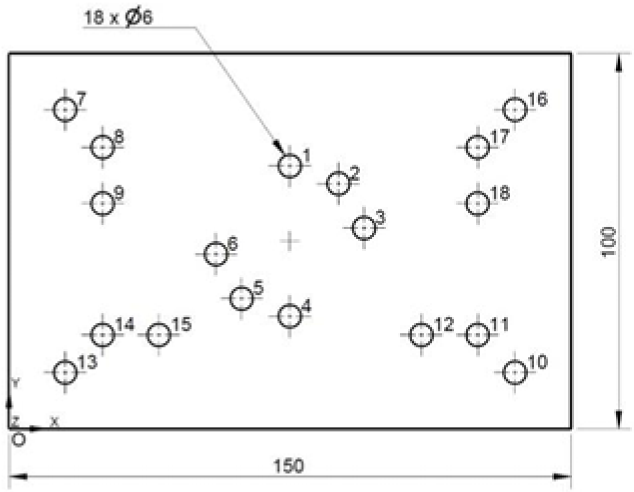

Figure 2-20: Placement of holes for Case Study 2 (18 holes)

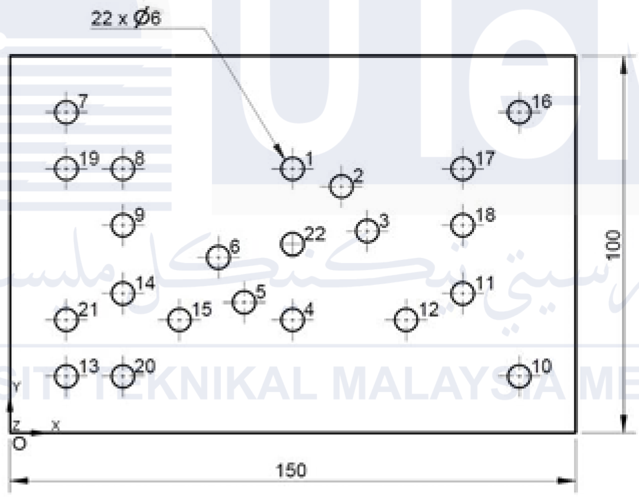

Figure 2-21: Placement of holes for Case Study 3 (22 holes)

2.3.9 Tool Path Optimization of Drilling Sequence in CNC Machine Using Genetic Algorithm

This research was published by N.K.A. Al-Sahib et al, (2014) and originally from Iraq. The paper determine to find the optimum shortest path to shorten the drilling time for the drilling of a given holes and able to reduce the drilling cost and improve computer numerical controlled machine efficiency. To solve the travelling salesman problem, the method using in this paper is Genetic Algorithm. There are two case studies based on Travel Salesman Problem which having 10 holes and 80 holes respectively shows in Figure 2.22 and 2.23. Those result from both case studies will be compered in terms of mechining time with the path obtained with the famous CAM software "ArtCAM", where the software can produce many paths using variuos sequencing holes.

n	x	y
1	10	80
2	390	50
3	80	20
4	350	150
5	40	120
6	300	10
7	150	40
8	100	240
9	240	80
10	200	200

Figure 2-22: Case Studies 1 with 10 holes

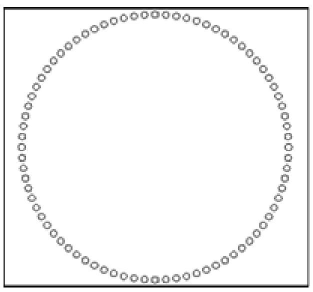

Figure 2-23: Case Study 2 with 80 holes

2.3.10 Tool Path Optimization For Hole Pattern making On Printed Circuit Boards By Combinational Of TSP and ACO

This research was published by V.H.Dao and T.K.Dao, (2018) originated from Vietnam. The paper propose to solve the non-productive time to reduce the air time of drilling. The research showed to solve the problem, the method are using the combination of Travelling Salesman Probelm and Ant Colony Optimization. The Method is applied on a complex Printed Circuit Board as an case study for showing the optimal path planning shows in Figure 2.24. The combination of TSP and ACO in this research is the TSP will provide the initial estimate of drilling path then the ACO will process the path from TSP to get the most optimal path.

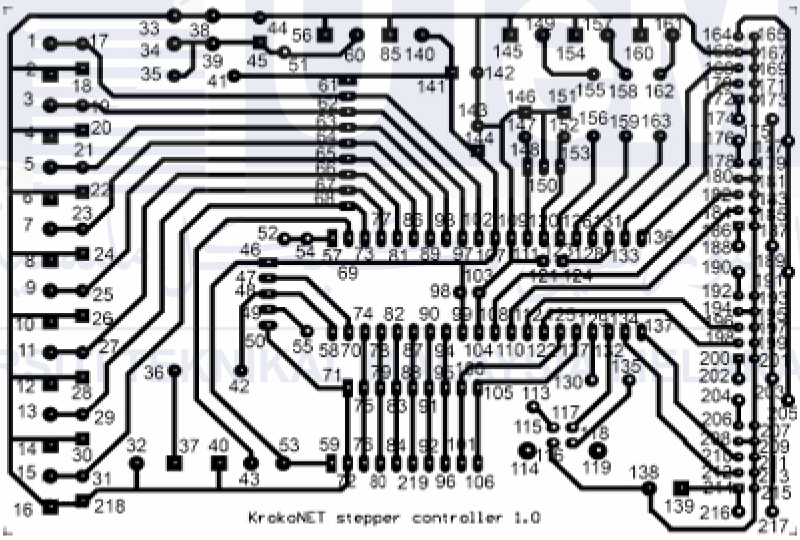

Figure 2-24: PCB with 219 specific holes

2.3.11 Multi-Hole Drilling Tool Path Planning and Cost Management through Hybrid SFLA-ACO Algorithm for Composites and Hybrid Materials

This research was published by N.Mehmood et al, (2022) and originated from Pakistans. The paper focused on consumption of time for optimization of the tool path. The method that used in this research is the hybridization of Shuffled Frog Leaping Algorithm (SFLA) and Ant Colony Optimization (ACO) metaheuristic algorithms to solve the tool path. The proposed method applied for the case study which have numbers of holes with 5, 10, 15, 20, and 25 and the complexity of the holes is increased exponentially with increasing number of holes as evident by the design space, which is 120, $3.6 * 10^6$, $1.3 * 10^{12}$, $2.4 * 10^8$, and $1.6 * 10^{25}$ respectively. The results using hyridization of SFLA-ACO for case study in this research are testbenched with the original SFLA, modified SFLA, Dynamic Programming(DP), and ACO and Immune-based Evolutionary Approach.

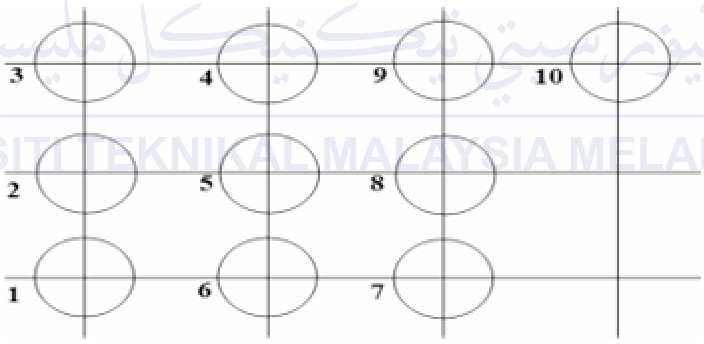

Figure 2-25: Holes locations for 10 holes

2.3.12 Determination of the optimal drill path sequence using bat algorithm and analysis of its optimization performance

This research was published by S.Diyaley et al, (2019) and originally from India. The paper focused on detemination of optimal drill path sequence for CNC machines. The algorithm that used in this article is called Bat Algorithm to determine the different case studies. Those case studies are different layouts consisting of 5 * 5, 7 * 7, 9 * 9, 11 * 11 matric holes in Table 2.11, and a 14 specific holes model in Figure 2.26. The results form those case studies are benchmarked between Genetic Algorithm, PSO Algorithm, ACO Algorithm and ABC Algortihm.

Drilling parameter	Test problem 1	2	3	4
Layout of the matrix holes	5 × 5	7 × 7	9 × 9	11 × 11
X-direction pitch (mm)	5	5	5	5
Y-direction pitch (mm)	5	5	5	5
Z-direction pitch (mm) (depth of drilling)	5	5	5	5
Hole diameter (mm)	0.5	0.5	0.5	0.5
Total number of holes	25	49	81	121
Approach (mm)	0.1	0.1	0.1	0.1
Overrun (mm)	0.1	0.1	0.1	0.1
Half drill point angle (°)	60	60	60	60
Drill speed (rpm)	35,000	35,000	35,000	35,000
Drill traversing speed (m/min)	15	15	15	15

Table 2-11: Parameters for case studies

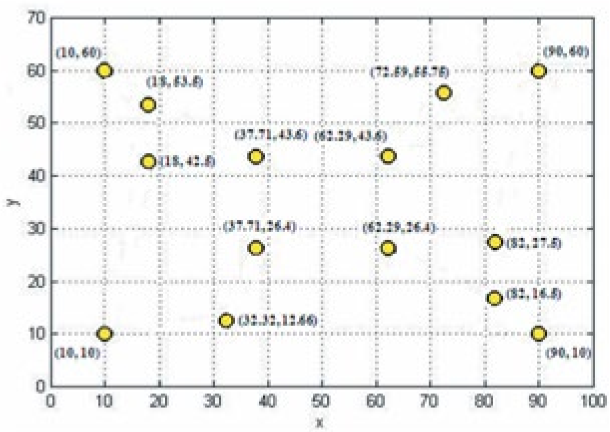

Figure 2-26: Schematic diagram of 14 hole model

2.3.13 Application of a hybridized cuckoo search-genetic algorithm to path optimization for PCB holes drilling process

This research was published by W.C.E. Lim et al, (2013) and originated from Malaysia. The paper is target to solve the path optimization problem for printed circuit board. The algorithm used in this article is hybridized cuckoo search-genetic algorithm which is the combination of cuckoo search algorithm and genetic algorithm that can improve the agent search space to get more solution. The method is applied on to two workpieces which is from others research that used PSO and Global Convergence Particle Swarm Optimition (GCPSO) for workpiece 1 that have 9 specific holes in Figure 2.27 and workpiece 2 is also taken from other reacrch that used Ant Colony System (ACS) in Figure 2.28. There is other case study consists multiple holes such as 30, 50, 75 ,100, and 200 of holes for comparison with original Cuckoo Search and Genetic Algorithm. For Workpieces 1 and 2 also have comparison in terms of polulation number applied, the minimum generation number ofr global convergence and average generation number for global convergence.

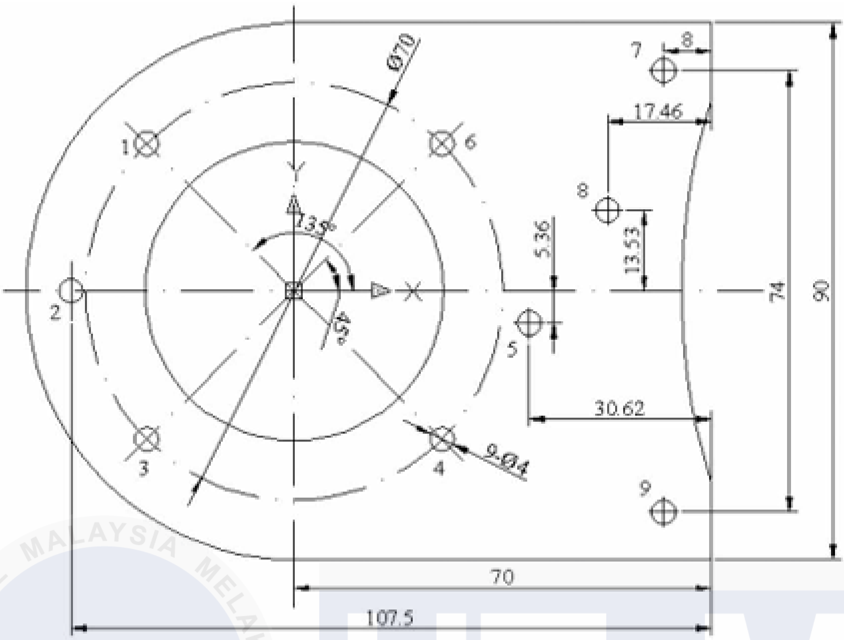

Figure 2-27: Workpiece 1

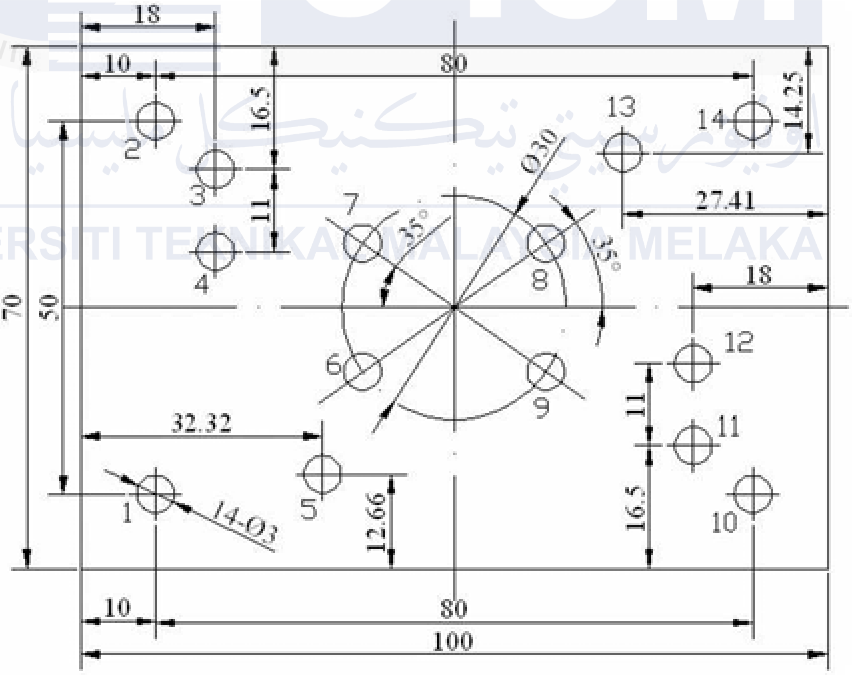

Figure 2-28: Workpiece 2

2.3.14 A Kalman Filter Approach to PCB Drill Path Optimization Problem

This research was published by N.H.A.Aziz et al, (2017) and originally from Malaysia. The paper is target to find the solution to solve when the number of holes for drilling increased and the number of possible solutions will also increase. The method in this paper using is Simulated Kalman Filter that inspired from Kalman Filter. There is a benchmarking between proposed proposed method SKF and Particle Swarm Optimization (PSO), Ant Colony System (ACS) and Cuckoo Search (CS) due to the case study for comparison is using the same PCB workpiece that consists 14-holes for drilling shows in Figure 2.41.

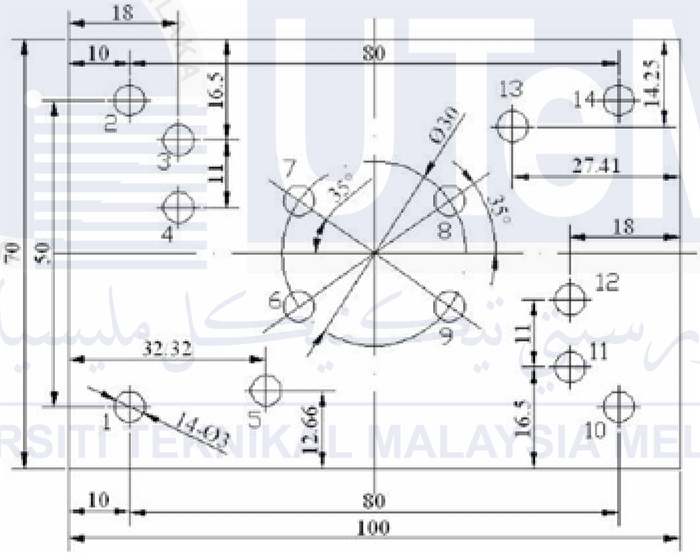

Figure 2-29: PCB workpiece with 14 Specified Holes

2.3.15 Simulation Approach of Cutting Tool Movement Using Articficial Intelligence Method

This research was published by H.Abdullah et al, (2015) and originally from Malaysia. The article targetting to solve the default tool path generated by CAD/CAM system which produced longer distance and increase the drilling time. The methods that used to solve those problems are Genetic Algorithm and Ant Colony Optimization. The Genetic Algorithm is used for optimizing the cutting tools movement and the Ant Colony Optimization to generate the shortest tool path. For case study in this paper is a PCB workpiece modal with consisting 76 holes shows in Figure 2.42 as the 3D model and Figure 2.43 in 2D view. The result outcome from GA and ACO are compared with MasterCAM software.

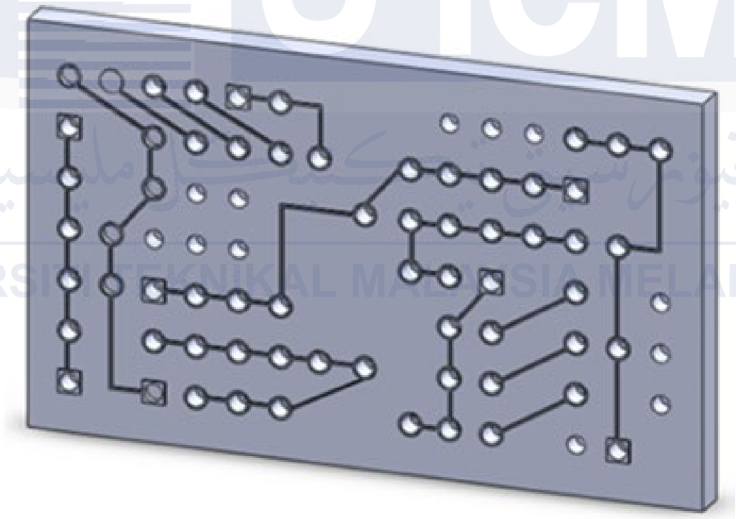

Figure 2-30: The 3-dimension PCB workpiece with 76 holes

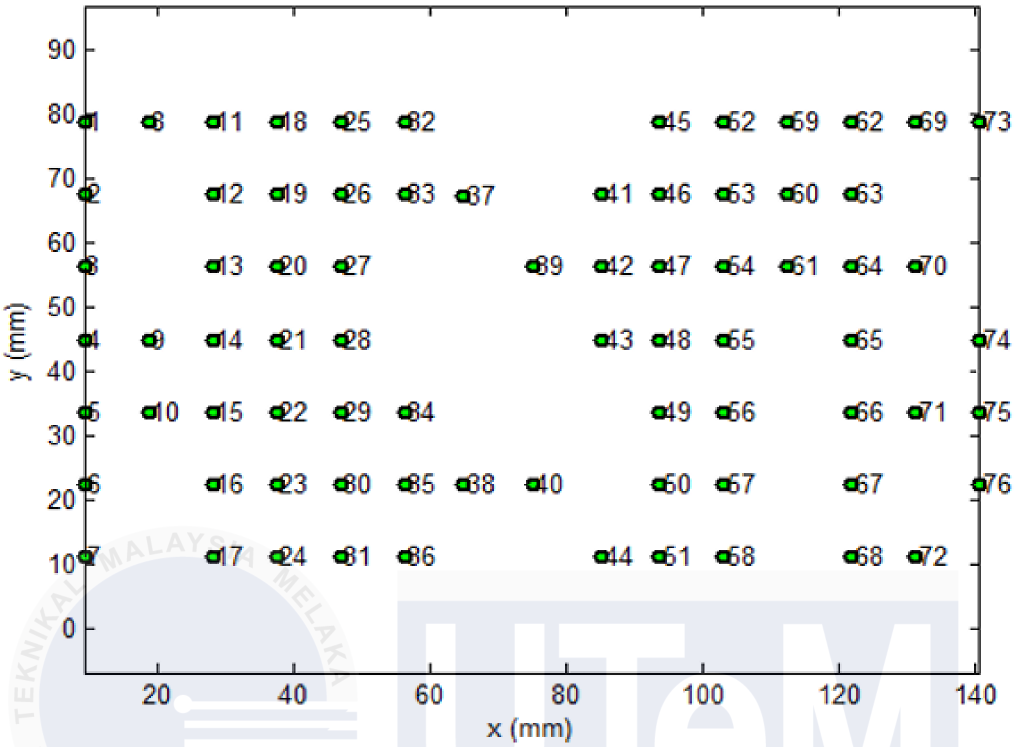

Figure 2-31: 2-Dimensions PCB Modal with 76 holes

2.4 Summary

Past research is mostly aimed at solving and finding the best path planning and optimization of the path when it comes to PCB drilling. Although the aims are the same, the methods applied to the problems are different in terms of the behaviours of the algorithm. Plenty of metaheuristic algorithms still have not been explored and applied to these problems. For example, a table has shown the algorithms that are used for path planning, and 9 algorithms are going to be used for benchmarking with the model and will be shown in Chapter 4. Therefore, the algorithms that have not been published and researched are because of the possibility of the algorithm is able to apply to finding the best path planning.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the explanation will roughly show the Whales Optimization Algorithm on adaptation of algorithm to PCB path planning along with the research had been done. The first pace that going in this project will be follow as understanding the method applied. After that comes with the overview combination of the method with the path planning. The process for this project will also be clarified in this chapter.

3.2 Project Overview

The overview for the project flowchart is shown in below Figure 3.1. For this semester, the Final Year Project will be conduct by proposing the purpose to carry out this project. The objectives and problems of this project will be determined after the proposed project. The limits of the project need to clarify in scope of the project with details. For analyzing the method applied on the project, literature studies and investigation have to be done in order to understand the requirement need in the project. In order to acknowledge to the development and stracture of the project, past researches have to be done as much as possible related to the project to assists in developing. The software acts as an important path for conducting this project, the functionality of the software indeed must be well-knowledge to let the project progress smoothly.

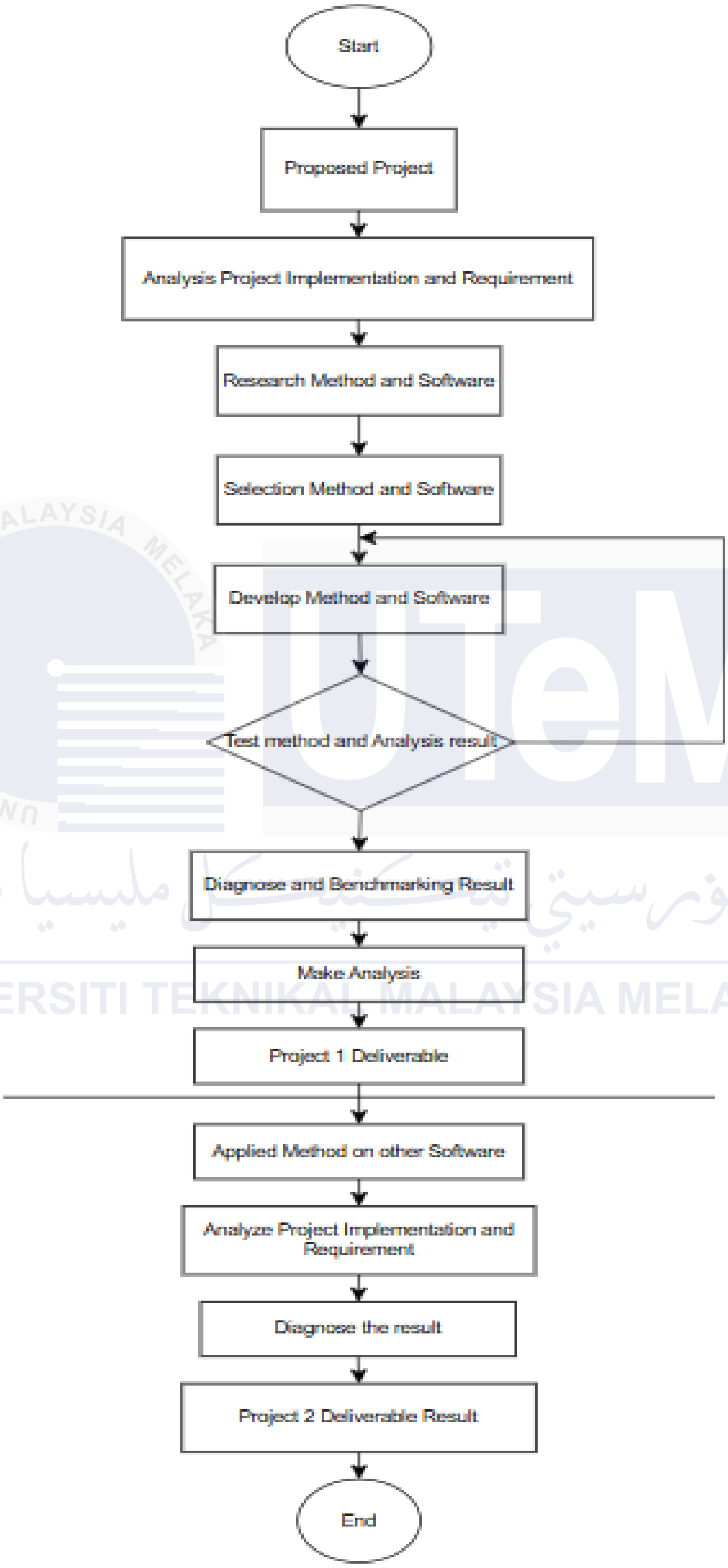

Figure 3-1 : Project Flowchart of Final Years Project 1 & 2

The development of the method used in this project will be conducted by taking the optimal result. Before taking the result, the function that is applied to the method is identified and ensure that the function is suitable for this method. The applied function must be combined with the method to get the optimal result. To get the optimal result, the method must be demonstrated to ensure the results are accurate. To safeguard a reliable result, understanding each program function of the project is a necessary process. After the understanding of the whole program functionality, the actual equation will be applied on the program for taking the actual optimal result. The testing and analysis of the optimal outcome will be followed. The result will be benchmarked with other different methods to compare the methods. After the comparison is finished, the outcome will be concluded and carried out in this first fiscal year project.

For planning the second fiscal year project, the method will focus on tuning in terms of parameters to get more accurate results on optimization. Understanding another function of the software will also be one of the processes to achieve before getting the best optimal result. After acknowledging the software, the demonstration will be conducted and improvising on the combination between the method and the software will also be developed simultaneously to ensure the result is benchmarkable. Following the processes will be the testing and benchmarking the optimal result between the other methods with this project's method.

The flowchart of this project is roughly shown in Figure 3.1, and the Gantt chart for final year projects 1 and 2 are also shown below in Table 3.1 and Table 3.2. The planning for the second fiscal year of the project is going through as planned according to what can be improved from the project's fiscal year. The Flowchart and Gantt chart shows the progress of the project and how it was carried out within the two semesters. The Gantt Chart shows the expected progress of the project as represented with red line box, while the actual progress is shaded sky blue for Table 3.1 and dark purple for Table 3.2.

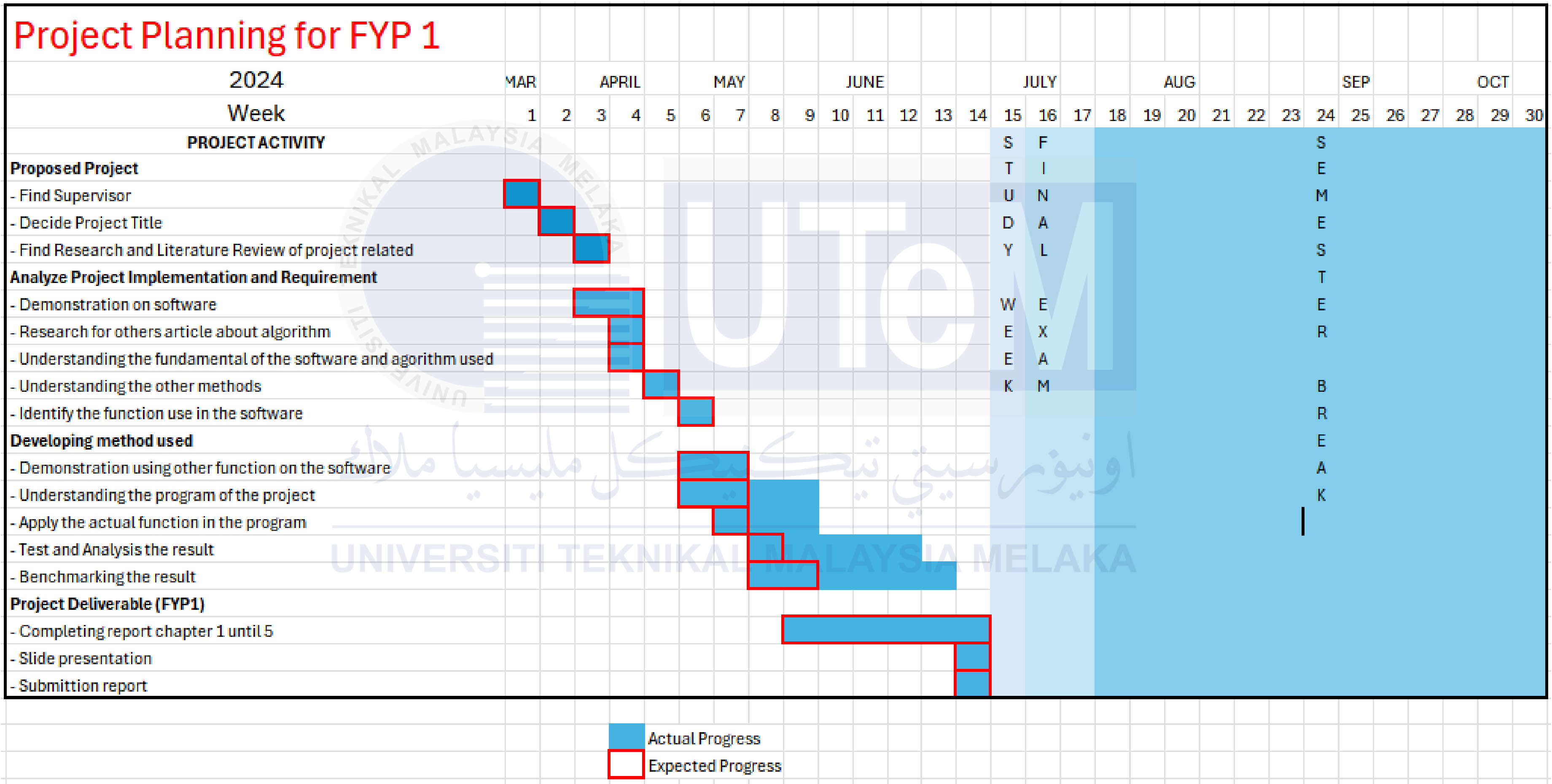

Table 3-1: Gantt Chart of Final Years Project 1

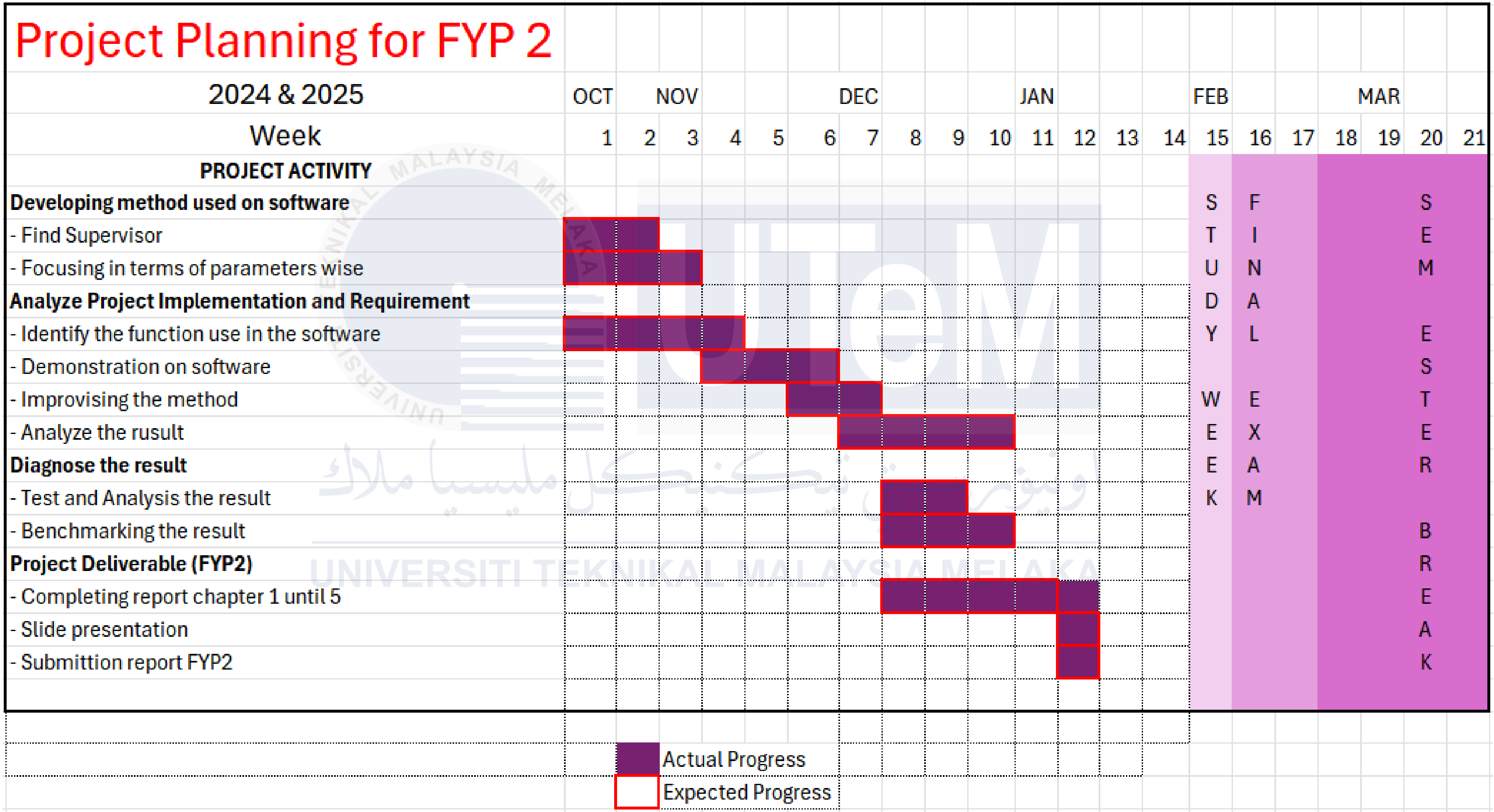

Table 3-2: Gantt Chart of Final Years Project

3.3 Whales Optimization Algorithm on Printed Circuit Board

Hole No.	Coordinate (x,y)	Hole No.	Coordinate (x,y)	Hole No.	Coordinate (x,y)	Hole No.	Coordinate (x,y)
1	(10,10)	5	(32.32,12.66)	9	(62.3,26.4)	13	(72.59,55.75)
2	(10,60)	6	(37.7, 26.4)	10	(90, 10)	14	(90, 60)
3	(18, 53.6)	7	(37.7, 43.6)	11	(82, 16.5)		
4	(18, 42.5)	8	(62.3, 43.6)	12	(82, 27.5)		

Table 3-3 : Coordinates for 14 Holes

The model will be applied with the Whales Optimization Algorithm using Zhu's (2006) 14-hole drilling problem in Figure 3.2. The problem has been chosen due to its complexity, and multiples research papers have been published using this as a case study with different optimization algorithms in Hidayati et al (2016), Lim et al (2013), Daadoo et al (2017), Dalavi et al. (2019), Diyaley et al. (2019), Asrani et al (2011) and Omar et al. (2014). The Printed Circuit boards are assumed to have a length of 100mm and a width of 70mm. The objective of this problem is to drill 14 holes scattered around the PCB. These holes are assumed to be the same size. The coordinates for each hole in the Printed Circuit Board model are stated in Table 3.3. The targeted optimal result is 280mm as the best optimal distance for benchmarking with other existing algorithms and the path planning is connected in a specific route shown in Figure 3.3 from Zhu's (2006) where the path can be either the path having sequence 2–3 4–7–8–13–14–10–11–12–9–6–5–1 or 1–5–6–9–12–11–10 14–13–8–7–4–3–2 where the optimal result is 280mm.

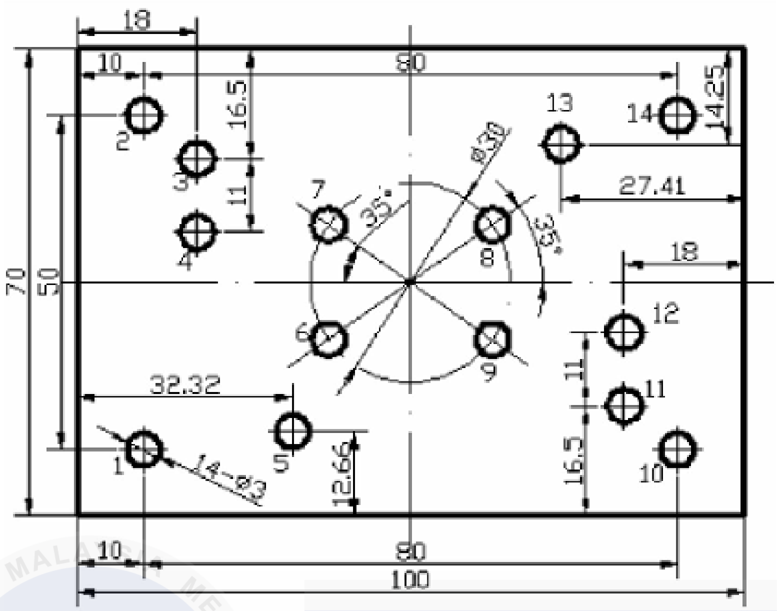

Figure 3-2: Case Study with 14 Holes

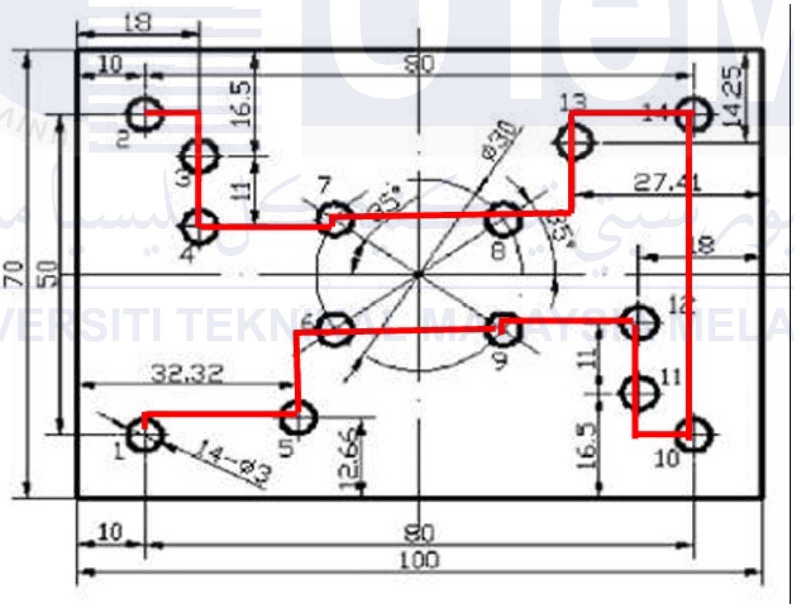

Figure 3-3: Optimal Solution For Case Study

The model will be applied with the Whales Optimization Algorithm to test with larger-size problems compared with Lim et al. 2013, which are 50-hole case studies and 100-hole case studies, shown in Tables 3.4 and 3.5. The result will be benchmarked with Lim's previous case study, which has a Cuckoo Search Algorithm, a Genetic Algorithm, and a Hybrid Cuckoo Search with a Genetic Algorithm.

3.4 Modeling in Whales Optimization Algorithm

The combination of path planning with the Whales Optimization Algorithm is related to the applied fitness function. The fitness function in this project uses the Manhattan Distance to calculate the distance between the holes, and the total distance will be the solution. The solution will be optimized to get to the best solution according to the iteration and the number of Whales. The beginning of the process will initialize the agents, and the total distance as the solution will be randomly calculated from the 14 holes using the equation (Manhattan distance), and each agent will calculate and find the best solution multiple times according to the iteration—the parameters used in S. Mirjalili and A. Lewis (2016) applied this optimization process because most of the Whales Optimization Algorithm is caused by its parameters. The flowchart of the modeling is shown in Figure 3.4.

Possible Holes Route	Total Distance (mm)
Distance 1-2-3	11
Distance 1-3-2	13
Distance 2-1-3	16

Table 3-4 : Possible Holes Route with Different Total Distance

Following the flowchart, the modal will initialize the population as Xi and Fitness Function Calculation of the initial whale solutions. Examples shows in Figure 3.4 shows the 3 coordinates and the 3-hole model and possible route solution will be like $X_1 = \{(2,3), (5,7), (8,6)\}$, $X_2 = \{(2,3), (8,6), (5,7)\}$, $X_3 = \{(5,7), (2,3), (8,6)\}$, stated in Table 3.4 with repective order of drilling holes. Then the algorithm calculates the fitness value of each possible route by computing the total drolling path length. The fitness function is defined using the Manhattna distance formula stated in Chapter 2 due to beachmark with other papers that used the equivilant fitness function.

After the algorithm begin, the iterative process will starting to run from 0 to 10000 by setting the iteration counter $t = 0$. At each iteration, t is increamented by 1and the process continues until a proedefined stopping condition, such as a maximum number of itreations 10000 which has set or the convergence of the best solution is satisfied in its iteration. In each iteration, the parameters such *a, A, C, l,* and *p* are updated to guide the exploration and exploitation process. The parameter *a* decreases linearly with each iteration stated in Chapter 2, where value a from 2 to 0 for iterations to control the search radius of each whale, which a high value will allow a larger search jump and vice versa in both exploration and exploitation phase. Parameter *A* is a random value representing [-1,1] as a range that will force the agent to move away from other agents to prevent matching up with the same agent and getting the same position in the exploration phase. For *A* in the exploitation phase, there is also a random value in the interval [-*a, a*] where a is decreased from 2 to 0 throughout iterations. The parameter *C* represents a number between 2 and 0, determined by r, a random vector in between [0,1]. For parameter *l* is a random number between [-1,1] and is an element-by-element multiplication in the equation stated in Chapter 2. By recognizing humpback whales swim in a spiral pattern and in a diminishing circle around their prey.

Depending on the randomness of parameter *p* determines the phase in which the whales engage. Suppose the value *p* is smaller and value *A* is greater than or equal to 1; the whales will enter the Exploration Phase, and the solution X_i allows the algorithm to search for new regions in the boundary space by moving the whale to a position relative to the randomly selected whale X_{rand}. The new position calculation is stated in Chapter 2 as Position for Initial Exploration Phase. For instance, the X_{rand} = {(2,3), (5,7), (8,6)} and X_i

= {(5,7), (2,3), (8,6)}, and the terms A = 1.5 and C = 1.2, there will be an updated and calculated position for each coordinate and a new fitness function is formed. If the value A requirement is smaller than 1, the starting fitness value will be updated as a solution. In the exploitation phase, when the value p is greater than 0.5, the whales are focused on exploiting the prey. In this phase, the value of Xi is updated relative to the current best solution as X* stated in Chapter 2 as the Spiral Equation for Exploitation Phase formula. As the example model, if $X^* = X_2$ = {(5,7), (2,3), (8,6)} and X_i = {(2,3), (8,6), (5,7)}, the coordinates of X_i are updated iteratively to approach X_i.

After updating the position, the algorithm ensures that all coordinates remain within the predefined grid boundaries, such as in the example modal; the initialization function generates search agents' positions within the range [-100000, 100000] for each dimension. This creates enormous search space and allows for extensive exploration. Since only 3 coordinates and 3 possible optimal routes are shown in Table 3.4, the result will be less different due to very few dimensions. If any whales are searched outside the boundary, their position will be corrected, and their fitness function will be recalculated. The new position will be updated until the iteration is shown in Figure 3.4.

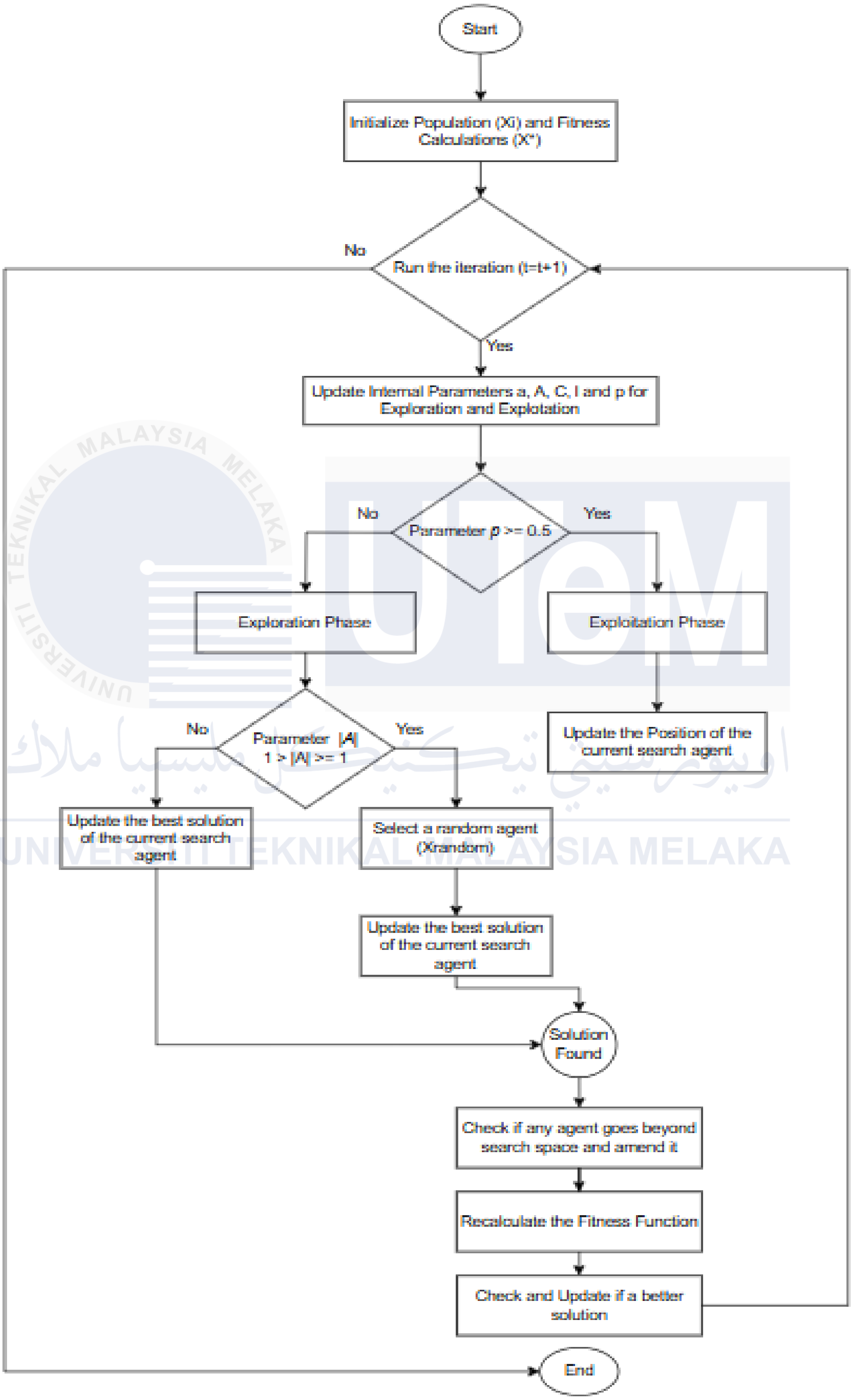

Figure 3-4: Flowchart of the modeling

Number of Holes	Coordinate hole (x,y)
1	(2,3)
2	(5,7)
3	(8,6)

Table 3-5 : Coordinates of 3 Holes

	Hole 1	Hole 2	Hole 3
Vote	-15.1652	-10.1193	36.8372
Arranged	36.8372	-10.1193	-15.1652
Sequence	3	2	1

Table 3-6 : Step of Arranging the Vote

The Whales Optimization Algorithm's visualization is proposed for continuous optimization problems like Path Planning for Printed Circuit Board. The population of humpback whales can search through a multi-dimensional search space to maximize the sequence requirement for a problem. In the path planning optimization problem, the sequence value represents the number of holes required. An example of a 3-hole printed circuit model is shown in Figure 3.5 and Table 3.7 for each coordinate. 6 possible path plans can be routed in the x-axis and y-axis, and the path ends where the last three holes are entered. There are 3 different paths (1-2-3), (1-3-2), and (2-1-3), and Manhattan Distance calculates the total distance as the whale's fitness function. For example, the boundaries are set from -100000 to 100000, and certain votes are generated randomly according to the number of dimensions. The number of dimensions, *dim* is according to the number of holes shown in Figure 3.5. For instance, Figure 3.6 has 3 holes with 3 *dim* and will generate 3 numbers within the boundary range. Table 3.8 shows the steps when the votes are generated from the boundaries, and it will be arranged from the greatest to the smallest to acknowledge the sequences of the vote and place the sequence following the original vote shown in the MATLAB coding in Figure 3.6 and others coding will be in the Appedix . The agents will start encircling within their search space to adapt to the case study and try to find the prey. When the agent has found the prey(solution), the agent starts the exploitation phase as it will update its position, as shown in Table 3.4. Due to the other three possible routes, (2-3-1), (3-1-2), and (3-2-1), they have the same total distance as others in Table 3.4. For this example, the (1-2-3) is the best result found by the fitness function applied on each agent and will be updated as the best solution.

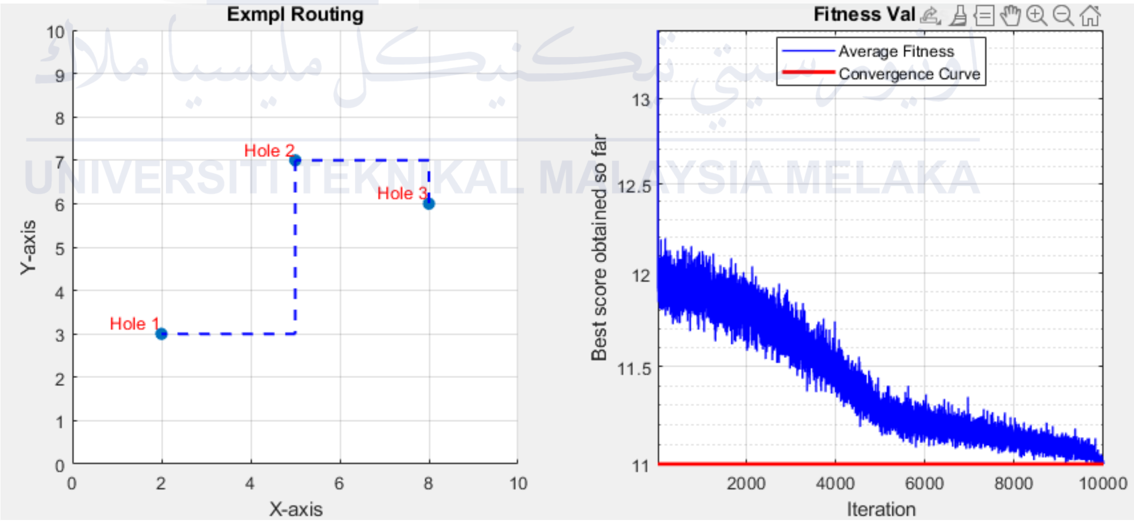

Figure 3-5: 3 holes model in 10 * 10 size

```
    case 'Exmpl'
        fobj = @Exmpl;
        lb=-100000;
        ub= 100000;
        dim=3;

end

end

%EXMPL
function total_distance = Exmpl(vote)

[arranged, sequence]  = sort(vote, 'descend');

    hole_positions=[2, 3; 5, 7; 8,6];

total_distance = Manhattan_Distance_Calculation_Fucntion(hole_positions, sequence);
end
```

Figure 3-6 : Setting and Coding vote for Exmpl

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

The purpose of this chapter is to explain the result during the performance and implementation of WOA on finding the optimal result of multiple case studies with others algorithm. The result is analyzed in terms of efficiency and the stability of the algorithm applied on different case studies.

4.2 Parameter

Settings and Parameters	Standard WOA
Number of Whales, *n*	500
Number of Iteration	10000
A	[-1 ~ 1] (force agent to move away from a reference whale)
C	[0, 2]
b	1
l	[-1, 1]
p	0.5

Table 4-1 : Settings and Parameters for Whales Optimization Algorithm

The Whales Optimization Algorithm is implemented to solve Zhu, 2006)14-hole drilling problem. The complexity of the problem is stated as 14 holes with a specific coordinate, which are scattered around the PCB. The result of this project uses the different values of agent and iteration to match and compare with the case study of Aziz et al. (2016), but the same parameters are used in S. Mirjalili and A. Lewis 2016 paper. The number of whales, iterations, and parameters are shown in Table 4.1. According to Table 4.1, the value has been set for benchmarking with other algorithms, which also applied the same number of agents and iterations. Other parameters such as A, C, b, l, and *p* values are followed according to the paper and can be used in this drilling path case. Parameter *A* helps to balance exploration and exploitation. Parameter *C* introduces randomness in scaling the distance between the whale and the best solution. Parameter *b* is a constant that defines the logarithmic spiral's shape. *l* is a random number in the range [-1,1] to add randomness to the spiral movement in the exploitation phase. Value *p* is the parameter used to choose between exploitation and exploration.

Study	Zhu's (PSO)	Othman	Adam
Number of Agents n	100	50	50
Number of Iterations t	10000	2500	5000
Number of Computations	50	50	50
Inertia Weight w	0.0, 0.5, 1.0	0.9 - 0.4	0.9 - 0.4
Cognitive Component c1	Not available	1.42	1.42
Social Component c2	Not available	1.42	1.42
Randomizer r1,r2	Random number [0,1]	Not applicable	Random number [0,1]

Table 4-2 : Parameter for PSOs'

Study	SAEALAL (ACS)	Lim (CSGA)	Aziz (SKF)	Omar (GSA)	Asrani (FA)
Number of Agents n	25	50	50	50	50
Number of Iterations t	2500	4000	1000	2500	10000
Number of Computations	50	50	50	50	50
DECISION FACTOR, r_0	0.5	Not applicable	Not applicable	Not applicable	Not applicable
Local evaporation factor, p_1	0.3	Not applicable	Not applicable	Not applicable	Not applicable
Global evaporation factor, p_2	0.3	Not applicable	Not applicable	Not applicable	Not applicable
Discovery Rate of Alien eggs, p_a	Not applicable	0..1667	Not applicable	Not applicable	Not applicable
Crossover Rate, *ac*	Not applicable	0.2143	Not applicable	Not applicable	Not applicable
Mutation Rate, *ah*	Not applicable	1	Not applicable	Not applicable	Not applicable
Measurement Noise (R)	Not applicable	Not applicable	0.5	Not applicable	Not applicable
Process Noise, Q	Not applicable	Not applicable	0.1	Not applicable	Not applicable
Initial Estimate Error, P	Not applicable	Not applicable	1000	Not applicable	Not applicable
Gravitational Constant, G_0	Not applicable	Not applicable	Not applicable	100	Not applicable
Alpha, α	Not applicable	Not applicable	Not applicable	20	Not applicable
Epsilon, *e*	Not applicable	Not applicable	Not applicable	0.01	Not applicable
Attractiveness β_0	Not applicable	Not applicable	Not applicable	Not applicable	1
Randomization Parameter α	Not applicable	Not applicable	Not applicable	Not applicable	1
Absorption Coefficient γ	Not applicable	Not applicable	Not applicable	Not applicable	1

Table 4-3 : Others' Parameter

Table 4.2 and 4.3 compares several algorithms applied to the 14-hole PCB drilling optimization problem, focusing on the Particle Swarm Optimizations (PSO), Ant Colony System (ACS), Cuckoo Search-Genetic Algorithm (CSGA), Simulated Kalman Filter (SKF), Gravitational Search Algorithm (GSA) and Firefly Algorithm (FA). Each algorithm is tuned with specific parameters to optimize the drilling path. The parameters are followed by the case study from Aziz et al. SKF Algorithm has used PSO, ACS, CSGA, and SKF itself. For GSA and FA is according to the original paper.

4.3 Implementation and Results

Performance Indicators	Basic PSO (ω=1.0)	BPSO (ω=0.9-0.4)	GC PSO (ω=0.9-0.4)	ACS	CS	SKF	GSA	FA	Our
The least iteration number during global convergence	93	71	7	193	23	27	87	22	209
The average iteration number during global convergence	847	783	353	1,037	429	73	632.36	1652.4	506.2
The least number of solutions searched	13,950	3550	350	4,825	2,300	2,700	4350	1100	10,450
The average number of solutions searched	127,050	39,150	17,650	25,925	42,900	7,300	31,618	82,620	253,100
The least search ratio (%)	3.20e-5	8.14e-6	8.03e-7	1.11e-5	5.28e-6	6.19e-6	9.98e-6	2.52e-6	2.39-e-5
The average search ratio (%)	2.91e-4	8.98e-5	4.05e-5	5.95e-5	9.84e-5	1.67e-5	7.25e-5	1.89e-4	5.8e-4
Average fitness after computing 50 computations	289.6	296	292.29	283.6	291.3	291.6	280.5	288.2	293.6

Table 4-4 : Performance Comparison

The algorithm performance comparison differing the result between the published results obtained by Zhu's (2006) Basic Particle Swarm Optimization Algorithm, Othman et al's (2011) Global Convergence PSO, Adam et al's (2010) Global convergence PSO with decreasing inertia weight, Saealal et al's (2013) Ant Colony System, Lim et al's (2014) Cuckoo Search, Aziz et al's (2016) Simulated Kalman Filter, Omar et al's (2014) Gravitational Search Algorithm and Lit et al's (2011) Firefly Algorithm is shown in Table 4.2. The comparison shows the least iteration number during global convergence, the average iteration number during global convergence, the least number of solutions searched, the least search ratio, and the average search ratio. Other performance indicators, such as the lowest and average iteration numbers, have been added compared to the usual performance comparison. Other added performance indicators are the least number of solutions searched, the average number of solutions, and their respective ratio due to the published paper using different numbers of agents in the population.

To get the search ratio, Aziz et al. (2016) SKF algorithm came up with an equation in Equation 4.1, which shows the total Searched Solution divided by Solution Space. The total searched solution is given by the number of iterations required, multiplied by the number of agents used, and the solution space is 14! divided 2 due to the total optimal sequence being 280mm in 2 sequences, which is 2-3-4-7-8-13-14-10-11-12-9-6-5-1 and 10-11-12-9-6-5-1-2-3-4-7-8-13-14. In total, there are 43 589,000,000 solutions in solution space. The search ratio is split out into 2, which is the least and average search ratio.

$$Search\ ratio = \frac{Total\ Searched\ Solution}{Solution\ Space} \qquad [4.1]$$

Equation 4-1 : Search Solution

From the implementation, WOA obtained one of the optimal solutions of 280 mm given by the optimal sequence of 10-11-12-9-6-5-1-2-3-4-7 8-13-14. Table 4.2 shows the WOA have a moderate performance in finding the optimal result compared to those algorithms. Table 4.2 shows the least iteration number during global convergence measures the minimum number of iterations required by each algorithm to converge to a global solution during its best run. A lower value which means a faster convergence rate. From those algorithms, GC PSO (ω = 0.9-0.4) from Zhu's 2006 shows the fastest and most efficient algorithm, which only needs 7 iterations. Followed by Cuckcco Search (CS) and Firefly(FA), both showing similar good performance with 23 and 22 iterations, respectively. However, our's is the slowest, which took 209 iterations to get the best solution. The average iteration number during global convergence for WOA is 506.2. Among other algorithms, WOA, which is higher than many competing algorithms, is because the adaptive algorithm has to balance between exploration and exploitation shifts depending on the situation it is facing. This shows the benefit of finding global optimal results. The WOA evaluates 10450 solutions, which is significantly higher than other algorithms; this reflects WOA's emphasis on solution exploration, ensuring a wide range of solutions is considered before focusing on promising areas of the search space. For the average number of solutions searched by WOA is 253100 in Table 4.2. This is due to how WOA mimics whales' hunting behaviours, where the results are cross-checking through spiralling and encircling techniques. The WOA relies on a balance of stochastic exploration and deterministic exploitation, causing the lowest search ratio to be 2.39e-5, shown in Table 4.2. For the average search ratio, WOA performs at one of the highest among other algorithms, with 5.8e-4. This indicates that WOA explores an enormous search space across each run, enhancing the robustness in finding high-quality solutions.

Case Study Problems	PCB with 50 holes	PCB with 100 holes
CS (Average)	79.270mm	156.298mm
GA (Average)	70.479mm	109.576mm
CSGA (Average)	69.412mm	97.783mm
WOA (Average)	142.841mm	343.645mm
CS (Best)	75.465mm	144.880mm
GA (Best)	68.605mm	106.651mm
CSGA (Best)	68.299mm	94.693mm
WOA (Best)	92.469mm	188.378mm

Table 4-5 : Comparison of WOA with CS, GA, and CSGA for study problem

The results are shown in Table 4.3 in two case studies: 50 holes and 100 holes from Lim et al. (2016) with CS, GA, and CSGA. The implementation of both case studies found the average and best optimal route. The result shows that WOA is not suitable for a large-scale problem. For example, the case study, which was 100 holes, shows an enormous scale of difference in WOA.

4.4 Analyzing the result

The global convergence curve is a tool used to understand how WOA operates. It displays the combination pattern of fitness across from 0 to 10000 iterations affected by the convergence curve. This is to observe the stabilization of the modal in terms of average fitness and convergence curve. Figure 4.1 shows the best optimal route is 280; the hole sequence is 10-11-12-9-6-5-1-2-3-4-7-8-13-14. and graph of the convergence curve and average fitness. The balancing of WOA can be observed from the right panel graph where the convergence curve and the fitness curve are synchronized when the convergence has a steep drop. The fitness curve also has a steep drop until the convergence has become linear. The average fitness shows a gradual improvement to discover the optimal result as the algorithm balances exploration and exploitation.

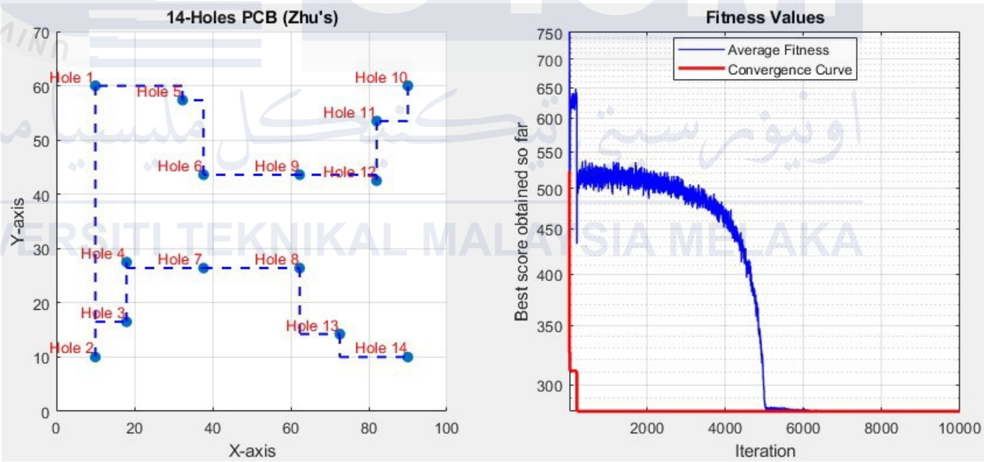

Figure 4-1 :The Route and Global Convergence for Best Result

Figure 4.2 showcases the Whale Optimization Algorithm (WOA) on the worst optimal result among the 50 runs. The left panel displays the optimized path on 309.5mm, and the hole sequence is 6-5-1-2-3-4-7-8-9-13-14-12-11-10. The right panel highlights the convergence, with an initial rapid drop in the best fitness value. The average fitness was rebalancing the exploitation and exploration behavior and instantly climbed up to almost 650mm, while the convergence discovered the optimal result after 2000 iterations.

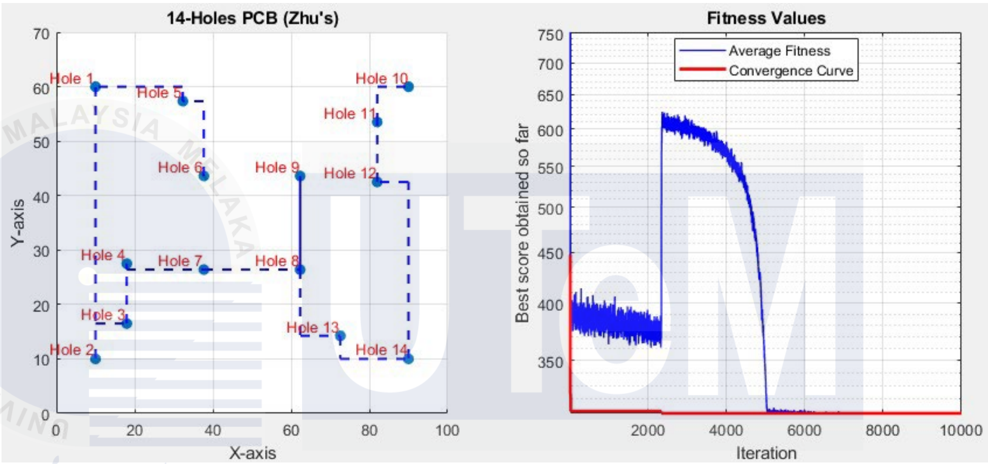

Figure 4-2 : The Route and Global Convergence for Worst Result

Figures 4.3 and 4.4 are the best and worst for the case study 50 holes Lim et al. (2013). The left panel in Figure 4.3 shows the most optimized routing path for all 50 holes, achieving the shortest possible distance, which is (92.469mm) and the right panel displays the convergence curve, where the best fitness value rapidly decreases and stabilizers after around 2000 iteration, indicating efficient convergence to the global optimum. Figure 4.4 on the left panel is the worst result (190.6mm), and the route will also be more complex compared to Figure 4.3. The right panel of Figure 4.4 shows a slower convergence with fluctuations in average fitness values, showing difficulty in consistently refining the solution.

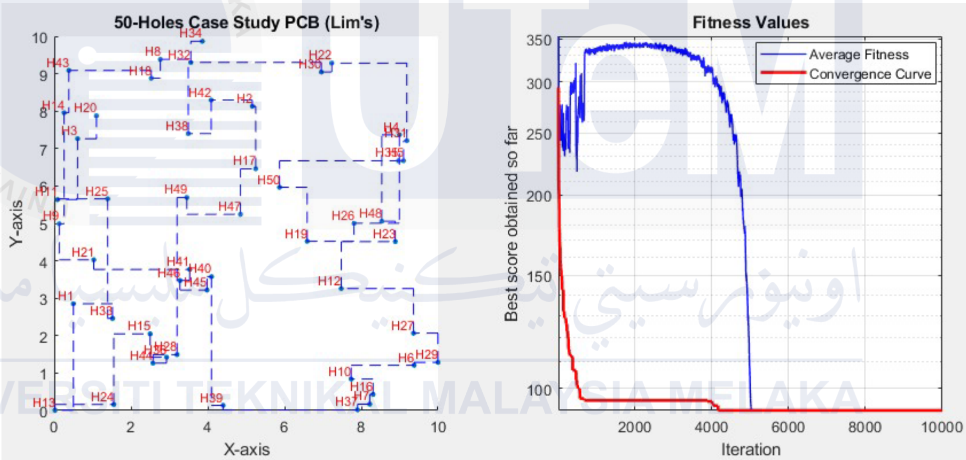

Figure 4-3 : The Route and Global Convergence for Best Result

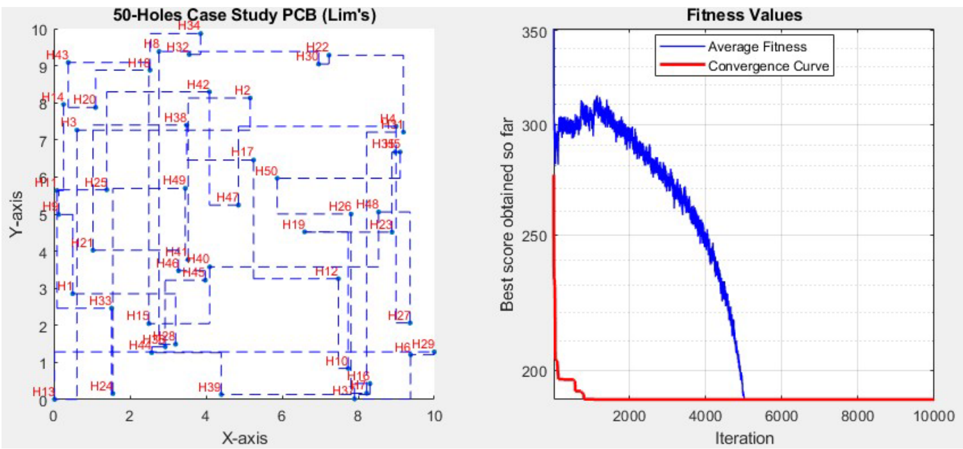

Figure 4-4 : The Route and Global Convergence for Worst Result

The results shown in Figures 4.5 and 4.6 are from one of the Lim et al. (2013) case studies, which contain 100 holes. Figure 4.5 shows the best result from these 10000 iterations with 500 whales, where the optimal routing is 228.84mm on approximately 4000 iterations. Figure 4.6 illustrates the worst result, 407.372mm to finish the route and slower, fluctuating convergence requiring more iterations to stabilize.

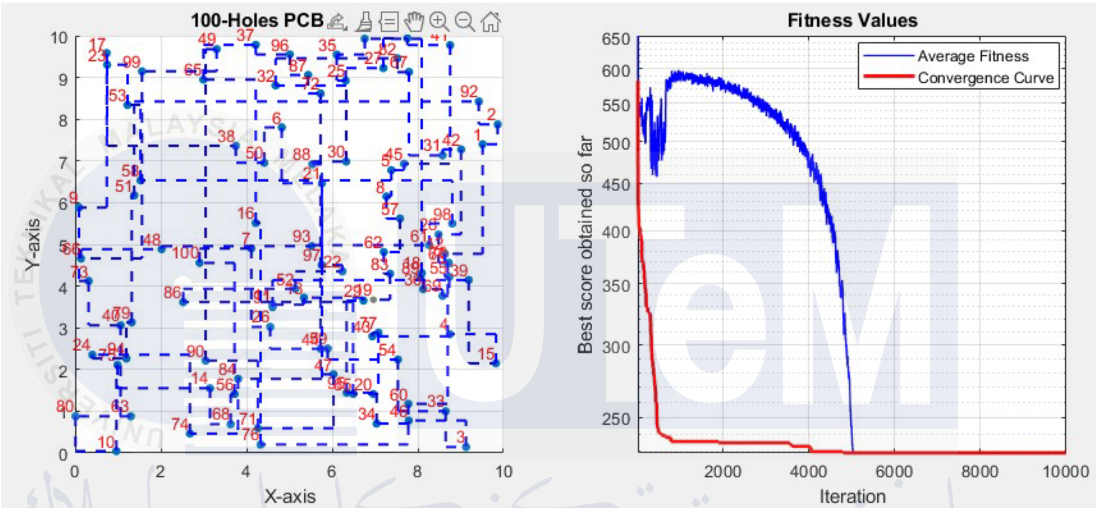

Figure 4-5 : The Route and Global Convergence for Best Result

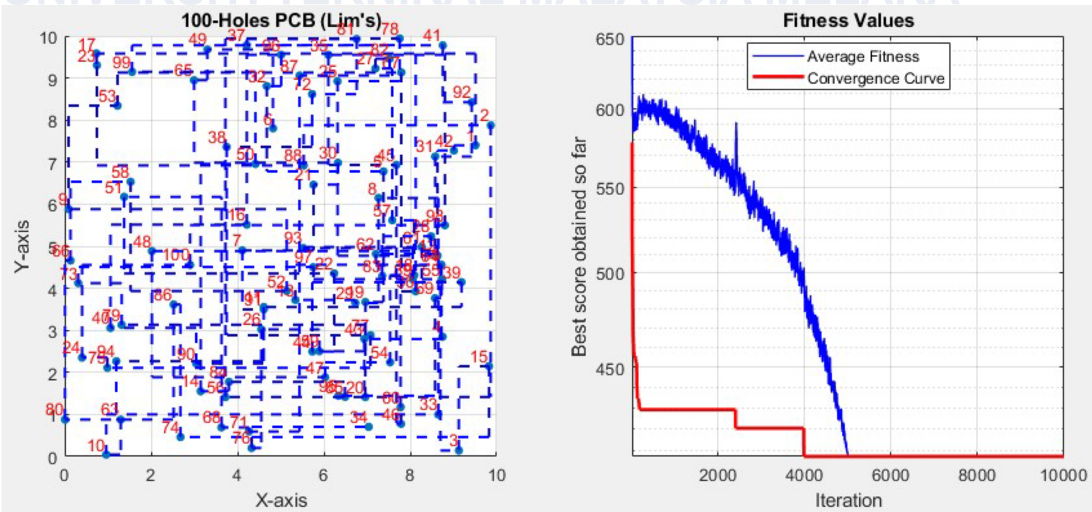

Figure 4-6 : The Route and Global Convergence for Worst Result

4.5 Implementation on Parameter b

In the Whale Optimization Algorithm, parameter *b* enables the variable to discover the suitable parameter to modify the WOA to get the optimal time costing to find the optimal result. Parameter *b* in WOA is used to manipulate the spiral updating position during bubble-net hunting simulation, such as the tightness and shape of the logarithmic spiral that models the whale's movement. The method used to find the suitable parameter value was set as 50 runs a row, and the average result and the standard deviation after finishing running were observed. This method was applied to the case study 14 holes. The parameter b has been set from 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, and 100, and the boundaries have been set from 100, 500, 1000, 5000, 10000, 50000, 100000, and 500000 in other words the range between -50 to 50 for 100 etc. shown in Table 4.4 and Table 4.5.

Parameter b \ Boundaries	100	500	1000	5000	10000	50000	100000	500000
0.01	8.9	7.01	9.14	8	9.16	8.12	9	8.5
0.05	7.73	7.88	8.67	8.29	8.67	6.78	9.04	8.64
0.1	8.46	8.3	7.53	8.07	8.46	8	9.07	8.62
0.5	8.5	8.14	7.84	8.5	7.7	8.15	8.47	7.56
1	8.19	7.23	7.86	7.7	8.1	6.5	7.6	8.2
5	5.23	5.5	5.35	5.54	5.3	6	6.88	5.62
10	5.86	6.72	6.51	6.29	4.88	6.3	6.16	6.34
50	7.18	6.42	7	6.3	7.95	6.3	7.2	5.7
100	6.96	6.05	6.5	7.76	6.08	7.8	4.95	6

Table 4-6 : Comparison Standard Deviation for Parameter *b*

Parameter b \ Boundaries	100	500	1000	5000	10000	50000	100000	500000
0.01	293.3mm	291.4mm	293.42mm	292.4mm	293.2mm	293.73mm	292.37mm	293.3mm
0.05	293.96mm	292.75mm	293.65mm	292.54mm	293.65mm	293mm	293.76mm	292.56mm
0.1	292.88mm	295.38mm	293.71mm	291.78mm	298.88mm	292.6mm	294.11mm	293.54mm
0.5	291.48mm	292.19mm	293.14mm	292.47mm	294.18mm	292.74mm	292.23mm	291.32mm
1	292.97mm	292.99mm	294.13mm	293.64mm	293.79mm	289.85mm	291.45mm	292.45mm
5	289.91mm	289.03mm	292.71mm	289.03mm	292.56mm	290.83mm	289.3mm	291.55mm
10	291.13mm	290.4mm	289.25mm	292.66mm	289.1mm	292.66mm	290.52mm	291mm
50	292.1mm	293.03mm	293.86mm	293.65mm	292.8mm	293.65mm	291.83mm	294.23mm
100	293.49mm	291.66mm	291.64mm	294.8mm	292.55mm	294.77mm	294.13mm	291.64mm

Table 4-7 : Comparison Average Fitness for Parameter *b*

4.5.1 Analysis 3D and 2D Graph

Tables 4.4 and 4.5 show the general comparison, and it would be difficult to observe. Therefore, to know the difference between the value of Average Fitness and the Standard Deviation, a 3D and a 2D graph will be appropriate to observe, as shown in Figure 4.7 and Figure 4.8, respectively. Figure 4.7 shows the 3D view of Table 4.4 to observe the suitable parameter that can enhance the performance of WOA. The 3D graph has Parameters B on the z-axis, Standard Deviation on the y-axis, and the x-axis is the Boundaries. The range of Standard Deviation has been arranged into multiple colors, shown in Figures 4.7 and 4.8, respectively, to recognize the better and worse parameters for WOA. In Figure 4.8, a top view of the 3D view made a clear difference in parameter *b*. Parameter ***b*** in 5 is the most suitable parameter in this case study, and the worst parameter b, which is 0.01-0.5, is shown in the Graphs. This result indicates that parameters b from 0.01 to 0.5 have produced a tighter spiral modal that causes the Whales not to be able to balance the behaviors between exploitation and exploration. The best parameter to enhance WOA performance is value 5 for parameter b, which is the green color area, as shown in Figure 4.8. Parameter *b* value 5 has the best standard deviation value for almost all set up boundaries.

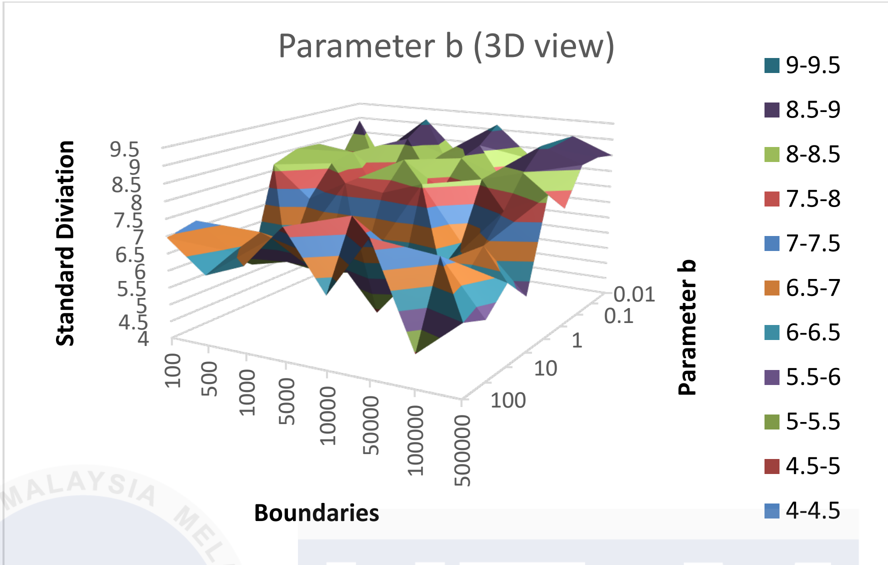

Figure 4-7 : 3D View of Standard Deviation

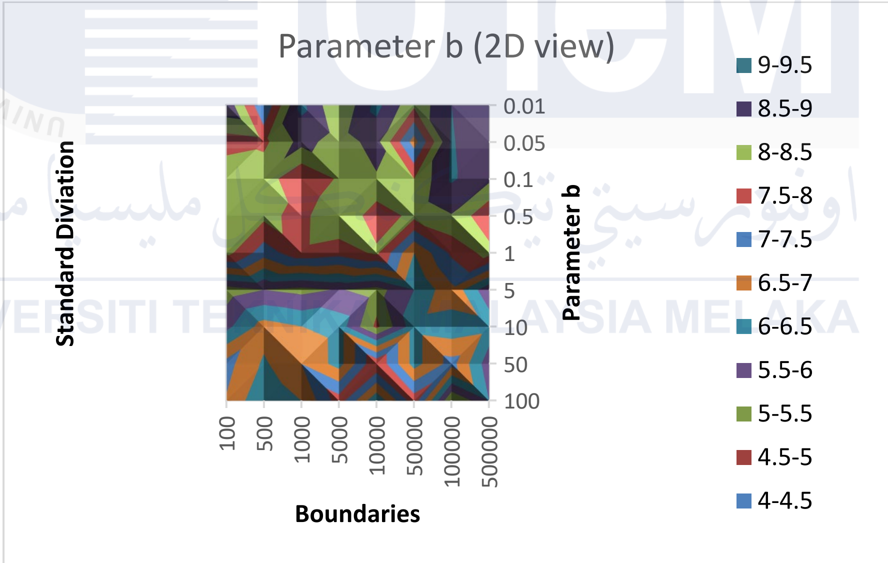

Figure 4-8 : 2D View of Standard Deviation

Tables 4.9 and 4.10 show the Average Fitness in 3D and 2D graphs for Parameter *b*. This graph shows which Parameter consistently gets the optimal result, and the value approaching 280mm is considered to have multiple optimal results in 50 runs. From Figure 4.9, the highest value is located on parameter b 0.01. Most results are between 290mm and 295mm, mainly covering parameters b 0.01, 0.05, 0.1, 0.5, 50, and 100, which show that these parameters are unsuitable for WOA to generate these sizes of spiral for finding the optimal result due to the unable to balance the exploitation and exploration in terms of the relationship of the shape of spiral and boundaries that have set up. The most critical hit is on parameter *b* 0.1 in 10000 boundaries, where the value has hit 298.88mm on average, as shown in Figure 4.9. Parameters *b* 5 and 10 have multiple average fitness values close to 280 in 50 runs, 289.03 on the average fitness observed from the 2D graph shown in Figure 4.10.

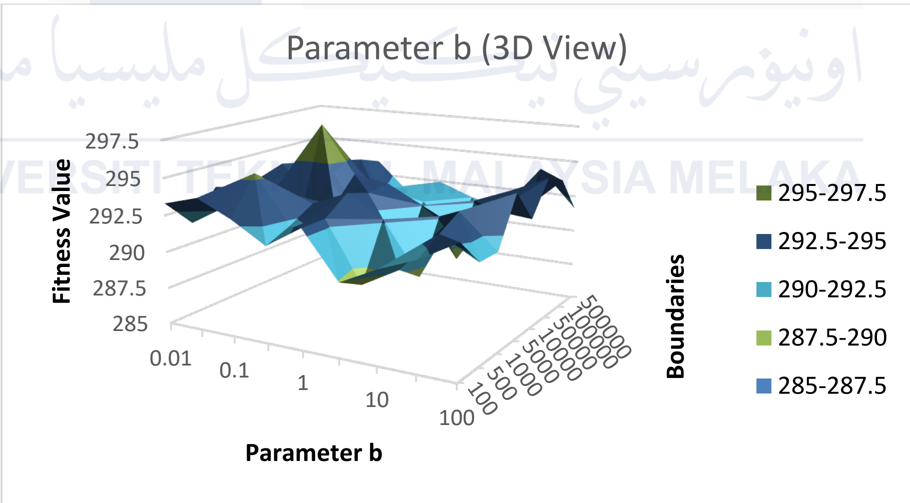

Figure 4-9 : 3D View of Average Fitness

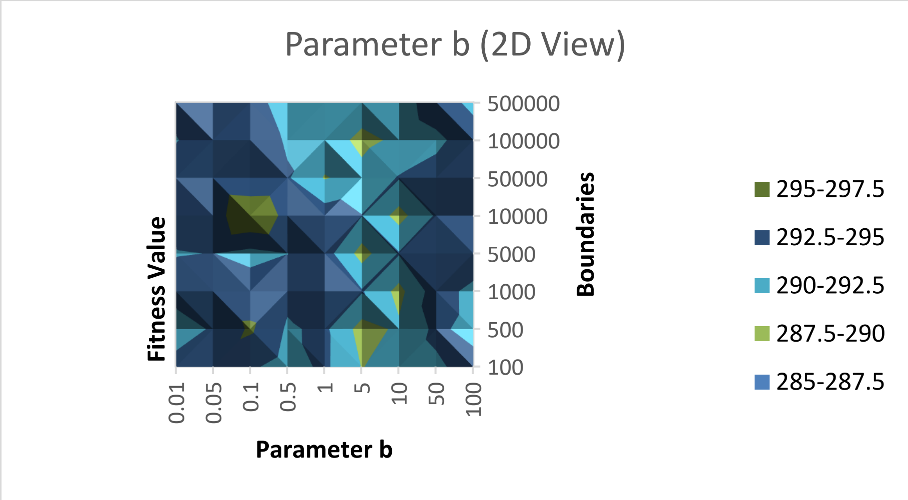

Figure 4-10 : 2D View of Average Fitness

CHAPTER 5

CONCLUSION

5.1 Introduction

Chapter 5 summarizes all the primary research points of this study and concludes crucial information and observations made during this project. These findings focus on the project's objective, which is to research the pass algorithm used for the case study stated in Chapter 4 and determine the algorithmic parameter to select to construct the better result of finding the optimal result. The parameter has been tested to guarantee optimal fitness in multiple case studies.

5.2 Achievement of Project

The objective of this project was to review the existing computational algorithms and their case studies that have been done in previous literature reviews with different cases. The Explanation of the Whales Optimization Algorithm is applied with Manhattan Distance as the Fitness Function for benchmarking with other Algorithms such as Basic PSO, Binary PSO, Global Convergence PSO, Ant Colony System, Firefly, Simulated Kalman Filter, Cuckoo Search, Gravitational Search, and Firefly which these Algorithms are also applied Manhattan Distance Equation as their fitness function. A table summarises the algorithms applied to optimize the drilling path, and a few use the Manhattan Distance Equation, where a table summarizes the algorithms in Chapter 4. The table also shows the benchmark values, such as the least iteration number, the average iteration number, the least number of solutions searched, etc. It proves that WOA is an average algorithm for solving 14 holes with the default parameters settings from the origin paper due to its balancing between exploitation and exploration to get a better and more convincing result.

The Modeling of the Whale Optimization Algorithm has shown in Chapter 3, arranging a sequence of the vole to get a particular fitness value to search for the optimal fitness value.

To prove the balancing of the Whale Optimization Algorithm, the table and graph show the changes in manipulating parameter b that will affect the shape of the spiral during the exploitation phase. The tables in Chapter 4 show the standard deviation and average fitness of 9 different values of parameter b with 8 different boundaries for the search space, which shows that the best parameter b value for the 14-hole case study is 5.

5.3 Future work

Recommendations for further research are important to achieve a better result and analysis. A good-performance laptop is recommended to use for this project because to it time spent collecting data is enormous. To get a better analysis of this modal of algorithm tunning in terms of its parameters is also an objective to make the modal more efficient on getting the optimal result for those types of case study.

REFERENCES

[1] Adam, A., Abidin, A. F. Z., Ibrahim, Z., Husain, A. R., Yusof, Z. M., & Ibrahim, I. (2010). A particle swarm optimization approach to robotic drill route optimization. *2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*. https://doi.org/10.1109/ams.2010.25

[2] Abdullah, H., Zakaria, M. S., Talib, N., Kiow, L. W., & Saleh, A. (2020). Optimization of drilling process using Non-Conventional method. *International Journal of Mechanical Engineering and Robotics Research*, 1233–1239.https://doi.org/10.18178/ijmerr.9.9.1233-1239

[3] Abbas, A. T., Aly, M. F., & Hamza, K. (2011). Optimum drilling path planning for a rectangular matrix of holes using ant colony optimisation. *International Journal of Production Research*, *49*(19), 5877–5891. https://doi.org/10.1080/00207543.2010.507608

[4] Diyaley, S., Aditya, A., & Chakraborty, S. (2020). Optimization of the multi-hole drilling path sequence for concentric circular patterns. *OPSEARCH/Opsearch*, *57*(3), 746–764. https://doi.org/10.1007/s12597-020-00441-0

[5] Lim, W. C. E., Kanagaraj, G., & Ponnambalam, S. (2014). PCB drill path optimization by combinatorial Cuckoo search Algorithm. *the Scientific World Journal/TheScientificWorldjournal*, *2014*, 1–11. https://doi.org/10.1155/2014/264518

[6] Al-Sahib, N. K. A., & Abdulrazzaq, H. F. (2014). Tool Path Optimization of Drilling Sequence in CNC Machine Using Genetic Algorithm. *Innovative Systems Design and Engineering*, *5*(1), 15–26. https://www.iiste.org/Journals/index.php/ISDE/article/download/10481/10676

[7] Mehmood, N., Umer, M., & Asgher, U. (2022). Multi-Hole Drilling Tool Path Planning and Cost Management through Hybrid SFLA-ACO Algorithm for Composites and Hybrid Materials. *Journal of Composites Science*, *6*(12), 364. https://doi.org/10.3390/jcs6120364

[8] Aziz, N. H. A., Aziz, N. a. A., Ibrahim, Z., Razali, S., Abas, K. H., & Mohamad, M. S. (2016, December 1). *A Kalman Filter approach to PCB drill path optimization problem*. https://doi.org/10.1109/spc.2016.7920699

[9] Kanagaraj, G., Ponnambalam, S., & Lim, W. (2014). Application of a hybridized cuckoo search-genetic algorithm to path optimization for PCB holes drilling process. *Computer & Indutrial Engineering*. https://doi.org/10.1109/coase.2014.6899353

[10] Daadoo, N. M., Eleyan, D., Tarapiah, S., Atalla, S., & Eleyan, A. (2018). Computer Numerical Control-PCB Drilling Machine with Efficient Path Planning – Case Study 2. *Automatic Control and Computer Sciences*, *52*(5), 451–463. https://doi.org/10.3103/s0146411618020050

[11] Dalavi, A., Pawar, P., & Singh, T. (2020). Determination of optimal tool path in drilling operation using Modified Shuffled Frog Leaping Algorithm. *International Journal for Engineering Modelling*, *32*(2–4). https://doi.org/10.31534/engmod.2019.2-4.ri.01v

[12] Diyaley, S., Biswas, A. B., & Chakraborty, S. (2019). Determination of the optimal drill path sequence using bat algorithm and analysis of its optimization performance. *Journal of Industrial and Production Engineering*, *36*(2), 97–112. https://doi.org/10.1080/21681015.2019.1585974

[13] Zhu, G. Y. (2006). Drilling Path Optimization Based on Swarm Intelligent Algorithm. *International Conference on Robotics and Biomimetics*. https://doi.org/10.1109/robio.2006.340357

[14] Rana, N., Latiff, M. S. A., Abdulhamid, S. M., & Chiroma, H. (2020). Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments. *Neural Computing & Applications*, *32*(20), 16245–16277. https://doi.org/10.1007/s00521-020-04849-z

[15] Mirjalili, S., & Lewis, A. (2016). The whale Optimization Algorithm. *Advances in Engineering Software*, *95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[16] Dahal, S. (2015). *Effect of different distance measures in result of cluster analysis*. https://aaltodoc.aalto.fi/handle/123456789/17787?show=full

[17] Dalfó, C., Comellas, F., & Fiol, M. (2007). The multidimensional Manhattan network. *Electronic Notes in Discrete Mathematics*, *29*, 383–387. https://doi.org/10.1016/j.endm.2007.07.063

[18] Yang, X. (2010). Engineering Optimization: An Introduction with Metaheuristic Applications. http://ci.nii.ac.jp/ncid/BB03412264

[19] Abidin, N. W. Z., Rashid, M. F. F. A., & Mohamed, N. M. Z. N. (2017). A review of multi-holes drilling path optimization using soft computing approaches. *Archives of Computational Methods in Engineering*, *26*(1), 107–118. https://doi.org/10.1007/s11831-017-9228-1

[20] Dewil, R., Küçükoğlu, İ., Luteyn, C., & Cattrysse, D. (2018). A critical review of multi-hole drilling path optimization. *Archives of Computational Methods in Engineering*, *26*(2), 449–459. https://doi.org/10.1007/s11831-018-9251-x

[21] Al-Janan, D. H., & Liu, T. (2016). Path optimization of CNC PCB drilling using hybrid Taguchi genetic algorithm. *Kybernetes*, *45*(1), 107–125. https://doi.org/10.1108/k-03-2015-0069

APPENDICES

Appendix A   Part of MATLAB Coding

```
% This function initialize the first population of search
agents
function Positions=initialization(SearchAgents_no,dim,ub,lb)
Boundary_no= size(ub,2); % numnber of boundaries
% If the boundaries of all variables are equal and user
enter a signle
% number for both ub and lb
if Boundary_no==1
    Positions=rand(SearchAgents_no,dim).*(ub-lb)+lb;
end
% If each variable has a different lb and ub
if Boundary_no>1
    for i=1:dim
        ub_i=ub(i);
        lb_i=lb(i);
        Positions(:,i)=rand(SearchAgents_no,1).*(ub_i-
lb_i)+lb_i;
    end
end
```

Coding 1 : WOA Initialization Program

```
% This function containts full information and implementations of the
benchmark
% functions in Table 1, Table 2, and Table 3 in the paper
% lb is the lower bound: lb=[lb_1,lb_2,...,lb_d]
% up is the uppper bound: ub=[ub_1,ub_2,...,ub_d]
% dim is the number of variables (dimension of the problem)
function [lb,ub,dim,fobj] = Get_Functions_details(F)
switch F
    case 'PCB14'
        fobj = @F1;
        lb=-100000;
        ub= 100000;
        dim=14;

    case 'Exmpl'
        fobj = @Exmpl;
        lb=-100000;
        ub= 100000;
        dim=3;

    case 'PCB50'
        fobj = @F2;
        lb=-100000;
        ub= 100000;
        dim=50;

    case 'PCB100'
        fobj = @F3;
        lb=-100000;
        ub= 100000;
        dim=100;
end

% F1
function total_distance = F1(vote)

[arranged, sequence]  = sort(vote, 'descend');

    hole_positions=[10, 60; 10, 10; 18, 16.5; 18, 27.5; 32.32, 57.34; 37.7,
43.6; 37.7, 26.4;
                    62.3, 26.4; 62.3, 43.6; 90, 60; 82, 53.5; 82, 42.5;
72.59, 14.25; 90, 10];

total_distance = Manhattan_Distance_Calculation_Fucntion(hole_positions,
sequence);
end
```

Coding 2 : Get Functions Detials Program (Start)

```
% F2
function total_distance = F2(vote)

[arranged, sequence]  = sort(vote, 'descend');

       hole_positions=[0.499, 2.852; 5.162, 8.129; 0.610, 7.264; 8.993,
   7.366; 9.100, 6.674;
                     9.375, 1.209; 8.220, 0.170; 2.765, 9.382; 0.131, 4.988;
   7.735, 0.839;
                     0.086, 5.636; 7.477, 3.258; 0.020, 0.007; 0.258, 7.957;
   2.500, 2.048;
                     8.307, 0.432; 5.250, 6.458; 2.530, 8.880; 6.591, 4.521;
   1.100, 7.875;
                     1.031, 4.026; 7.231, 9.284; 8.886, 4.514; 1.552, 0.162;
   1.393, 5.655;
                     7.811, 5.002; 9.361, 2.065; 3.201, 1.493; 9.997, 1.286;
   6.958, 9.044;
                     9.188, 7.210; 3.558, 9.303; 1.520, 2.461; 3.859, 9.867;
   8.968, 6.672;
                     2.927, 1.423; 7.902, 0.017; 3.494, 7.399; 4.401, 0.138;
   4.097, 3.576;
                     3.530, 3.771; 4.089, 8.296; 0.382, 9.086; 2.573, 1.264;
   3.981, 3.218;
                     3.278, 3.473; 4.852, 5.242; 8.533, 5.056; 3.455, 5.692;
   5.871, 5.965];

total_distance = Manhattan_Distance_Calculation_Fucntion(hole_positions,
sequence);

end
```

Coding 3 : Get Functions Detials Program (Cont...)

```
% F3
function total_distance = F3(vote)

[arranged, sequence]  = sort(vote, 'descend');

hole_positions = [9.510, 7.400; 9.863, 7.875;  9.125, 0.143; 8.747, 2.847;
7.380, 6.779;
4.822, 7.804; 4.111, 4.902; 7.266, 6.148; 0.090, 5.883; 0.963, 0.046;
4.608, 3.555; 8.631, 4.775; 5.340, 3.723; 3.148, 1.552; 9.823, 2.148;
4.215, 5.512; 0.737, 9.583; 8.091, 4.312; 6.962, 3.673; 6.959, 1.413;
5.761, 6.463; 6.238, 4.353; 0.747, 9.305; 0.402, 2.353; 6.316, 8.925;
4.553, 3.023; 7.194, 9.226; 8.483, 5.236; 6.729, 3.651; 6.327, 6.986;
8.577, 7.133; 4.678, 8.808; 8.648, 0.999; 7.037, 0.705; 6.105, 9.550;
8.126, 3.928; 4.216, 9.783; 3.749, 7.362; 9.185, 4.146; 1.061, 3.058;
8.753, 9.777; 9.009, 7.276; 6.938, 2.797; 5.732, 2.498; 7.679, 6.933;
7.787, 0.777; 6.030, 1.885; 2.019, 4.886; 3.304, 9.679; 4.415, 6.957;
1.377, 6.174; 5.141, 3.935; 1.223, 8.342; 7.534, 2.238; 8.731, 4.216;
3.721, 1.414; 7.584, 5.618; 1.527, 6.534; 5.897, 2.504; 7.777, 1.176;
8.264, 4.981; 7.205, 4.815; 1.298, 0.877; 8.698, 4.549; 2.995, 8.948;
0.136, 4.661; 7.792, 9.131; 3.629, 0.691; 8.573, 3.765; 8.723, 4.558;
4.266, 0.596; 5.728, 8.617; 0.314, 4.123; 2.680, 0.463; 0.989, 2.108;
4.330, 0.203; 7.080, 2.884; 7.753, 9.938; 1.320, 3.131; 0.008, 0.877;
6.760, 9.928; 7.523, 9.458; 7.351, 4.296; 3.803, 1.776; 6.490, 1.420;
2.522, 3.615; 5.438, 9.060; 5.535, 6.921; 8.056, 4.216; 3.046, 2.213;
4.609, 3.497; 9.423, 8.424; 5.515, 4.957; 1.194, 2.265; 6.337, 1.445;
5.010, 9.552; 5.756, 4.504; 8.801, 5.501; 1.560, 9.147; 2.904, 4.556];

total_distance = Manhattan_Distance_Calculation_Fucntion(hole_positions,
sequence);

end
```


Coding 4 : Get Functions Detials Program (End)

```
function total_distance =
Manhattan_Distance_Calculation_Fucntion(hole_positions, sequence)
    total_distance = 0;
    for i = 1:length(sequence) - 1
        x1 = hole_positions(sequence(i), 1);
        y1 = hole_positions(sequence(i), 2);
        x2 = hole_positions(sequence(i + 1), 1);
        y2 = hole_positions(sequence(i + 1), 2);
        distance = abs(x1 - x2) + abs(y1 - y2);
        total_distance = total_distance + distance;
    end
end
```

Coding 5 : Manhattan Distance Calculation Function

```
% The Whale Optimization Algorithm
function [Leader_score,Leader_pos,Convergence_curve,
AverageFitness_curve]=WOA(SearchAgents_no,Max_iter,lb,ub,dim,fobj)
% initialize position vector and score for the leader
Leader_pos=zeros(1,dim);
Leader_score=inf; %change this to -inf for maximization problems
%Initialize the positions of search agents
Positions=initialization(SearchAgents_no,dim,ub,lb);
Convergence_curve=zeros(1,Max_iter);
AverageFitness_curve=zeros(1,Max_iter);
t=0;% Loop counter
% Main loop
while t<Max_iter
    AverageFitness = 0;
    for i=1:size(Positions,1)

        % Return back the search agents that go beyond the boundaries of the
search space
        Flag4ub=Positions(i,:)>ub;
        Flag4lb=Positions(i,:)<lb;

Positions(i,:)=(Positions(i,:).*(~(Flag4ub+Flag4lb)))+ub.*Flag4ub+lb.*Flag4l
b;

        % Calculate objective function for each search agent
        fitness=fobj(Positions(i,:));
        AverageFitness = AverageFitness  + fitness;
        % Update the leader
        if fitness<Leader_score % Change this to > for maximization problem
            Leader_score=fitness; % Update alpha
            Leader_pos=Positions(i,:);
        end

    end
```

Coding 6 : Whale Optimization Algorithm(Start)

```
    a=2-t*((2)/Max_iter); % a decreases linearly fron 2 to 0 in Eq. (2.3)

    % a2 linearly dicreases from -1 to -2 to calculate t in Eq. (3.12)
    a2=-1+t*((-1)/Max_iter);

    % Update the Position of search agents
    for i=1:size(Positions,1)
        r1=rand(); % r1 is a random number in [0,1]
        r2=rand(); % r2 is a random number in [0,1]

        A=2*a*r1-a;  % Eq. (2.3) in the paper
        C=2*r2;      % Eq. (2.4) in the paper


        b=1;               %  parameters in Eq. (2.5)
        l=(a2-1)*rand+1;   %  parameters in Eq. (2.5)

        p = rand();        % p in Eq. (2.6)

        for j=1:size(Positions,2)

            if p<0.5
                if abs(A)>=1
                    rand_leader_index = floor(SearchAgents_no*rand()+1);
                    X_rand = Positions(rand_leader_index, :);
                    D_X_rand=abs(C*X_rand(j)-Positions(i,j)); % Eq. (2.7)
                    Positions(i,j)=X_rand(j)-A*D_X_rand;      % Eq. (2.8)

                elseif abs(A)<1
                    D_Leader=abs(C*Leader_pos(j)-Positions(i,j)); % Eq.
(2.1)
                    Positions(i,j)=Leader_pos(j)-A*D_Leader;      % Eq.
(2.2)
                end

            elseif p>=0.5

                distance2Leader=abs(Leader_pos(j)-Positions(i,j));
                % Eq. (2.5)

Positions(i,j)=distance2Leader*exp(b.*l).*cos(l.*2*pi)+Leader_pos(j);

            end

        end
    end
    t=t+1;
    Convergence_curve(t)=Leader_score;
    AverageFitness_curve(t)=AverageFitness/size(Positions,1);
    [t Leader_score];
end
```

Coding 7 : Whale Optimization Algorithm(End)

```
clear all
clc

SearchAgents_no = 500;
Function_name = 'PCB14';
Max_iteration = 10000;
num_runs = 10;


all_best_scores = zeros(1, num_runs);
all_best_solutions = zeros(num_runs, 14);

for run = 1:num_runs

    [lb, ub, dim, fobj] = Get_Functions_details(Function_name);

    [Best_score, Best_pos, WOA_cg_curve, WOA_af_curve] = WOA(SearchAgents_no,
    Max_iteration, lb, ub, dim, fobj);

    figure('Position', [269, 240, 1000, 400])

    subplot(1, 2, 1);
    vote = Best_pos;
    [arranged, sequence] = sort(vote, 'descend');

    all_best_scores(run) = Best_score;
    all_best_solutions(run, :) = Best_pos;

    disp(['Run ' num2str(run) ':']);
    disp(['Best solution obtained by WOA is : ', num2str(sequence)]);
    disp(['Best optimal value of the objective function found by WOA is : ',
num2str(Best_score)]);

    plotRouting(sequence,Function_name);

    subplot(1, 2, 2);
    semilogy(WOA_af_curve, 'Color', 'b', LineWidth = 1);
    hold on;
    semilogy(WOA_cg_curve, 'Color', 'r', LineWidth = 2);
```

Coding 8 : Main Program call (Start)

```
    title('Fitness Values');
    xlabel('Iteration');
    ylabel('Best score obtained so far');
    axis tight;
    grid on;
    box on;

    legend({'Average Fitness', 'Convergence Curve'}, 'Location', 'best');

    hold off;

    % % Save each figure to a file (optional)
    % saveas(gcf, ['WOA_0.05_5000_Run_' num2str(run) '.fig']);
    %
    % % Save results to a .mat file for each run
    % save(['WOA_0.05_5000_Run_' num2str(run) '.mat'], 'Best_score',
'Best_pos', 'WOA_cg_curve',
'WOA_af_curve','Max_iteration','SearchAgents_no');

end

display(['The best solution obtained by WOA is : ', num2str(sequence)]);
display(['The best optimal value of the objective function found by WOA is :
', num2str(Best_score)]);
```

Coding 9 : Main program (End)