Solving the Travelling Salesman Problem by Using Artificial Bee Colony Algorithm

Siti Hafawati Jamaluddin¹*, Noor Ainul Hayati Mohd Naziri², Norwaziah Mahmud³, Nur Syuhada Muhammat Pazil⁴

¹,²,³Faculty of Computer and Mathematical Sciences, University Teknologi MARA Cawangan Perlis, Kampus Arau, Malaysia
⁴Faculty of Computer and Mathematical Sciences, University Teknologi MARA Cawangan Melaka, Kampus Jasin, Malaysia

Corresponding author: *hafawati832@uitm.edu.my
Received Date: 15 August 2022
Accepted Date: 5 September 2022
Published Date: 30 September 2022

HIGHLIGHTS

- The aim of find the minimum cost of time or distance for Travelling Salesman Problem (TSP)
- Use of artificial bee colony algorithm to determine the minimum cost of time or distance
- Use of secondary data which consists of 29 cities in Bavaria to implement the algorithm
- The shortest distance obtained was 3974km

ABSTRACT

Travelling Salesman Problem (TSP) is a list of cities that must visit all cities that start and end in the same city to find the minimum cost of time or distance. The Artificial Bee Colony (ABC) algorithm was used in this study to resolve the TSP. ABC algorithms is an optimisation technique that simulates the foraging behaviour of honey bees and has been successfully applied to various practical issues. ABC algorithm has three types of bees that are used by bees, onlooker bees, and scout bees. In Bavaria from the Library of Traveling Salesman Problem, the distance from one city to another has been used to find the best solution for the shortest distance. The result shows that the best solution for the shortest distance that travellers have to travel in all the 29 cities in Bavaria is 3974km.

Keywords: Travelling Salesman Problem, Artificial Bee Colony Algorithm, Optimisation

INTRODUCTION

Irish and British mathematicians, W. R. Hamilton and Thomas Kirkman have introduced Traveling Salesman Problem (TSP). TSP is the classic algorithmic mathematical problem that focused on optimisation in the field of computer sciences and operations research to find a better solution for the problem that is the shortest, fastest and cheapest. The easiest way to convey the TSP is by describing the location as a set of nodes.

From the previous research, each author has many TSP definitions. The first definition stems from Kaspi, Zofi and Teller (2019). TSP is defined as a list of cities that a salesperson has to visit all the cities that start and end in the same city to find the minimum cost, time or distance. The travelling salesman problem
is defined as containing n for cities (vertices), and m for edges between 2 vertices is weight as travelling
time or distance.

Moreover, the generalised TSP covers all variants of TSP, a set of cities that includes a depot and a subset. It covers some of the customers’ satisfaction and the main objective is to find the shortest total distance travelled by the salesman (Pandiri & Singh, 2019). However, the article by Khan and Maiti (2019) said that TSP is the standard combinatorial discrete optimisation problem consisting of a set of N cities (vertices). So the objective of the problem is to find the shortest path that starts and visit all the vertices once and return to the starting vertex.

There are several solutions to solve the TSP but the result is approximate and not always optimal. The TSP can use the optimisation algorithm method to solve the problem faced by the salesman experiencing the problem, where the route distance that the salesman has to use to distribute the product from the original place and visited all the cities before returning to the original place. Besides that, the salesman also has to consider the cost of travel while distributing the product. In conclusion, the salesman must find the shortest distance to travel all of the city once and back to the place of origin.

The TSP can be formulated as an integer linear program. Miller-Tuckerm-Zemlin (MTZ) and Dantzig-Fulkerson-Johnson (DFJ) have proposed the formulation for TSP. The MTZ formulation is still useful in certain settings even though the MTZ formulation is not stronger than the DFJ formulation. Hence, in this study, DFJ formulation is used. The formulation can be defined as follows:

\[
\min \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij}; \quad 0 \leq x_{ij} \leq 1 \quad i, j=1,\ldots,n;
\]  

Subject to:

\[
\sum_{i=1, i \neq j}^{n} x_{ij} = 1 \quad j = 1,\ldots,n; \quad (2)
\]

\[
\sum_{j=1, j \neq i}^{n} x_{ij} = 1 \quad i = 1,\ldots,n; \quad (3)
\]

\[
\sum_{i \in Q} \sum_{j=Q}^{n} x_{ij} \leq |Q| - 1 \quad \forall \ Q \subseteq \{1,\ldots,n\}, |Q| \geq 2 \quad (4)
\]

where \( x_{ij} \) is the path from city i to city j, is the distance from city i to city j and Q is the number of edges
between the nodes.

Based on the above equation, the shortest distance will be determined for the traveller to travel all of the
one and back to the origin place.

**RELATED WORKS**

In the report by O’Neil and Hoffman (2019), TSP investigates the shortest path for pickup and delivery
such as meal delivery and ride-sharing is meaningful in on-demand last-mile logistics. Using low-width

---

Copyright © 2022 UiTM Press. This is an open access article licensed under CC BY-SA
https://creativecommons.org/licenses/by-sa/4.0/
decision diagrams in assignment problem assumptions duals as primal heuristics can find a good result within scrupulous time budgets.

In the expert opinions of Akhand, Ayon, Shahriyar Siddique and Adeli (2019), TSP represents by all spider monkeys wherever Swap Operator (SO) and Swap Sequence (SS) primarily based operations are utilised in separate spider monkey optimisation that allows interaction among monkeys is getting the best result of TSP. An example of SOs is a global leader, local leader or randomly chosen from the spider monkey group that conveyance regarding the exploitation of the involvement of other members. The result of the experiment exposes the effectiveness of discrete optimisation of spider monkey for solving TSP.

Based on a basic genetic algorithm, combining crossover and dynamic mutation is an improved strategy that has been proposed to optimise mutation characters and gain population diversity. The advanced algorithm's convergence rate and best solution show that it is superior to the standard, prepared selection and adaptive crossover probability of genetic algorithm and a new method for TSP are provided (Xu, Pei & Zhu, 2018).

Based on the random method, to overcome the optimized locations of the exit door for a safer emergency evacuation, the ABC algorithm was proposed by Khamis et al. (2019). As a result, ABC has fewer control parameters to be tuned to find the foremost optimal locations of the exit door. In representing the group of dynamics, it used a crowd evacuation model supported by the Social Force Model (SFM) and became the premise for the optimizer of cost functions. By minimizing crowd evacuation times and increasing the number of individuals being evacuated, the optimum locations of exit doors for a multi-room situation can improve evacuation potency. It is additionally clearly incontestable that the optimized design style is impressive in rising the evacuation efficiency underneath the various desired speeds of the group to evacuate.

**ARTIFICIAL BEE COLONY ALGORITHM**

Based on the research done by Zuloaga and Moser (2017), Artificial Bee Colony (ABC) algorithm is one of the groups of "swarm intelligence" which refers to the collective behaviour of the decentralised and self-organised system, commonly composed of agents that follow uncomplicated rules where the communications lead to the evolution of intelligent behaviours.

In 2005, based on the honey bee swarm's intelligent foraging behaviour, Dervis Karaboğa introduced the ABC algorithm. As claimed by Mridula, Rahman and Ameer (2018), the colony of bees is divided into three types with different method that approaches the food: the employed bees, onlooker bees and scout bees. Employed bees have visited food beforehand and moved the honey bees to the source of food while onlooker bees are waiting at the area to decide the food source. The bees evaluated its nectar amount after choosing a neighbour around that. They are determined and replace the sources of food with the new sources of food discovered by scout bees. The best sources of food are found (Guo, Li, Tang & Li, 2017).
Food sources, employed bees and unemployed bees are the parts of bee honey’s destructive behaviour and the behaviour patterns are unification and abandonment of source of food (Lvshan, Dongzhi & Weiyu, 2017). The distance between the honeycomb and the sources of food, the element of honey and the complication of takeout honey are the factors that affect the value of the food source. Each employed bee has a one-to-one concurrence with its matching food source where it can share information obtained with other bees. Scrutinising and utilising the food source is the main task for unemployed bees. After deciding whether to recruit or give up, detecting bees must analyse the new food source.

The four phases of the ABC algorithm are illustrated in Figure 1.

**Figure 1:** Flowchart of the ABC Algorithm
(Source: Guo, Li, Tang & Li, 2017)
Based on Figure 1, there are four phases in the standard ABC algorithm. The phases are the initialisation of the parameters and population phase, employed bees phase, onlooker bees phase and scout bees phase. All the flow will repeat until all requirements are met. The elaboration of these phases for ABC is described below:

**Initialisation of The Parameters and Population**

$x_{ij}$ represents a possible solution randomly produced by a group of food sources $SN/2$ where $SN$ is the colony’s population size. The following equation is:

$$x_{ij} = x_{j}^{\text{max}} + \text{rand}(0,1) \left( x_{j}^{\text{max}} - x_{j}^{\text{min}} \right)$$  \hspace{1cm} (5)

where $i = 1,2,\ldots,SN/2$ and $D$ denoted the number of problem dimension, rand(0,1) is a uniform random number in the range [0,1] and $x_{j}^{\text{max}}$ and $x_{j}^{\text{min}}$ are respectively to the upper and lower bounds for the dimension $j^{\text{th}}$.

Consider the optimisation problem as $\text{Minimize } f(x) = x_1^2 + x_2^2$ with the range $-5 \leq x_1, x_2 \leq 5$ and parameters of ABC Algorithm are set as the colony size equal to 6, dimension of the problem equal to 2 and limit for scout is colony size multiple with dimension and divided into 2 which is equal to 6. Then, initialise the position of 3 food source $SN/2$ of employed bees randomly using uniform distribution in range (-5, 5).

$$x = \begin{bmatrix} 1.4112 & -2.5644 \\ 0.4756 & 1.4338 \\ -0.1824 & -1.0323 \end{bmatrix}$$

So, the values of $f(x)$ are 8.5678, 2.2820 and 1.0990 for 3 food sources.

Then, the solution is calculated by the following equation:

$$\text{fit}_i = \begin{cases} \frac{1}{\left(1 + \text{OBJ}(x_i)\right)} & \text{if } \text{OBJ}(x_i) \geq 0 \\ 1 + \text{abs} (\text{OBJ}(x_i)) & \text{if } \text{OBJ}(x_i) \leq 0 \end{cases}$$  \hspace{1cm} (6)

where $\text{fit}_i$ is the value of fitness, $\text{OBJ}(x_i)$ is the objective value of the solution $x_i$ and $\text{abs}(\text{OBJ}(x_i))$ is the absolute value of $\text{OBJ}(x_i)$.

Since all $f(x)$ are positive, then the $\text{fit}_i = \frac{1}{\left(1 + \text{OBJ}(x_i)\right)}$. So the initial fitness values are 0.1045, 0.3047 and 0.4764.
Employed Bee Phase
In this phase, the employed bees will search for food sources throughout the whole space. Every source of food, $x_i$ is allocated to one and only employed bee and a new food source, $x'_i$ is generated for each employed bee in the neighbourhood of the food sources $x_i$ and $x_k$ its present position as follows:

$$x'_i = x_i + \varphi_i (x_j - x_k) \quad (7)$$

where $k$ is different from $i$, $k \in \{1, 2, ..., SN/2\}$, $j \in \{1, 2, ..., D\}$ are randomly chosen indexes and $\varphi_i$ is a uniformly distributed real number between $[-1, 1]$. A greedy selection is performed according to their fitness values between $x_i$ and $x_k$. Employed bees will share the information with onlooker bees about the nectar amount of food sources after completing the search.

For example, if $k=1$ and $j=0$, which is a randomly selected index. For the first food source, let $\varphi_{ij} = 0.8050$ which be $\varphi_{ij}$ a randomly produced number in the range $[-1, 1]$. So, $x'_1 = 2.1644$ and $-2.5644$.

After that, calculate $f(x'_1)$ using formula $f(x)$ and fitness of $x'_1$, which is

$$f(x'_1) = 11.2610$$ and fitness value is $0.0816$. Then apply greedy selection to compare the value of fitness previous and new one. Since $0.0816<0.1045$, the solution for the first food source could not be improved. Increase its trial counter. Apply the calculation for a second and third food source. The new values of $f(x)$ are $8.5678$, $2.2820$ and $1.0714$ and the fitness values for 3 food sources are $0.1045$, $0.3047$ and $0.4828$.

Onlooker Bee Phase
Onlooker bees will evaluate the information of food sourced from employed bees and choose a source of food, $x_i$, based on its probability value $P_i$ as calculated in Eq(8):

$$P_i = \frac{\text{fit}_i}{\sum_{n=1}^{SN/2} \text{fit}_n} \quad (8)$$

where $\text{fit}_i$ is the fitness value of the $i^{th}$ source of food, $x_i$. The higher $\text{fit}_i$, the largest chance source of food to be selected. When the food source, $x_i$ has been selected, one onlooker bee updates $x'_ij$ by using Eq. (7) and if the new food source $x'_ij$ has equal or better fitness value than $x_i$, a new member of the population is replaced by $x'_ij$. Like the employed bees phase, a greedy selection is performed according to their fitness values between $x_i$ and $x'_ij$.

Since the new values of $f(x)$ are $8.5678$, $2.2820$ and $1.0714$ and the fitness values for 3 food sources are $0.1045$, $0.3047$ and $0.4828$. Then, the probability values are $0.1172$, $0.3416$ and $0.5412$. Then, onlooker bee updates using equation employed bee. Replace the new food source with better fitness value and apply greedy selection. The new values of $f(x)$ are $8.5678$, $2.0855$ and $1.0669$ and the fitness values for 3 food sources are $0.1045$, $0.3241$ and $0.4838$. 
Scout Bee Phase
The number of scout bees is not defined beforehand in the colony. If the source of food, \( x_i \) cannot be improved through a predetermined number of trial “limit”, the source of food, \( x_i \) is to be abandoned and the corresponding employed bee becomes a scout. A new source of food, \( x'_i \) produced by the scout randomly as follows:
\[
x'_{ij} = x_{ij}^{\text{min}} + \text{rand}(0,1)(x_{ij}^{\text{max}} - x_{ij}^{\text{min}})
\]
(9)
where \( i \) is the index of the employed bees whose “trail” values reach the “limit” value firstly, \( j \in \{1,2,\ldots,D\} \), \( \text{rand}(0,1) \) is a uniform random number in the range \([0,1]\) and \( x_{ij}^{\text{max}} \) and \( x_{ij}^{\text{min}} \) are respectively to the upper and lower bounds for the dimension \( j \)th.

The phase from 3.2 to 3.4 will repeat until the termination criterion is satisfied.

RESULTS AND DISCUSSION
To solve the TSP, the secondary data is used to implement. The data is adopted from the Library of Traveling Salesman Problem (TSPLIB) from Zuse Institute Berlin. The data set consists of 29 cities in Bavaria with a distance from one city to another city as shown in Table 1 and Table 2. In Table 1 below, \( X \) is the node of cities for the \( x \)-axis and \( Y \) is the node for the \( y \)-axis. Based on the node value in Table 1, the distance between these two nodes for 29 cities is obtained as in Table 2.

Table 1: The Nodes of 29 Cities

| City | Node | X     | Y     |
|------|------|-------|-------|
| 1    |      | 1150  | 1760  |
| 2    |      | 630   | 1660  |
| 3    |      | 40    | 2090  |
| 4    |      | 750   | 1100  |
| 5    |      | 750   | 2030  |
| 6    |      | 1030  | 2070  |
| 7    |      | 1650  | 650   |
| 8    |      | 1490  | 1630  |
| 9    |      | 790   | 2260  |
| 10   |      | 710   | 1310  |
| 11   |      | 840   | 550   |
| 12   |      | 1170  | 2300  |
Table 2: Distance between 29 Cities

| Cities | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 2      | 60 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 3      |   | 50 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 4      |   |   | 40 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 5      |   |   |   | 30 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 6      |   |   |   |   | 20 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 7      |   |   |   |   |   | 10 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 8      |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 9      |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 10     |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 11     |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 12     |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 13     |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 14     |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 15     |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 16     |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 17     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 18     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 19     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 20     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 21     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 22     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 23     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 24     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 25     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |   |
| 26     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |   |
| 27     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |   |
| 28     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |   |
| 29     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | 0  |   |   |   |   |   |   |   |   |

The ABC algorithm is used to get the optimum route for 29 cities with the shortest distance. Table 3 below shows the optimum route of 29 cities with a distance of 3974km. Based on the table below, the optimum route will start with city 1, then city 8, city 4, 17, 18, 9, 3, 5, 12, 26, 29, 28, 25, 14, 22, 20, 15, 10, 6, 2, 13, 21, 27, 7, 23, 16, 26, 19, 11 and end with city 1 back.
Table 3: The Optimum Route of 29 Cities

| City | X   | Y   |
|------|-----|-----|
| 1    | 1150| 1760|
| 8    | 1490| 1630|
| 4    | 750 | 1100|
| 17   | 230 | 590 |
| 18   | 460 | 860 |
| 9    | 790 | 2260|
| 3    | 40  | 2090|
| 5    | 750 | 2030|
| 12   | 1170| 2300|
| 26   | 490 | 2130|
| 29   | 360 | 1980|
| 28   | 1260| 1910|
| 25   | 1280| 790 |
| 14   | 510 | 700 |
| 22   | 490 | 500 |
| 20   | 590 | 1390|
| 15   | 750 | 900 |
| 10   | 710 | 1310|
| 6    | 1030| 2070|
| 2    | 630 | 1660|
| 13   | 970 | 1340|
| 21   | 830 | 1770|
| 27   | 1460| 1420|
| 7    | 1650| 650 |
| 23   | 1840| 1240|
| 16   | 1280| 1200|
| 24   | 1260| 1500|
| 19   | 1040| 950 |
| 11   | 840 | 550 |
CONCLUSION

The Artificial Bee Colony (ABC) algorithm is a technique of optimisation that simulates honey bees’ foraging behaviour and has been successfully applied to various practical issues. A set of honey bees can perform tasks successfully through social cooperation called a swarm. In the ABC algorithm, there are three types of bees: employed bees, onlooker bees, and scout bees. The employed bees will search for food around the sources of food in the memory and all the collected information will be shared with the onlooker bees. The onlooker bees tend to select the good food sources found by the employed bees. The food sources with higher quality and the best fitness value will have more chance for the onlooker bees to select the good source of food than the lower quality. The scout bees are translated from a few employed bees, which abandon their food sources and search for new ones.

In this study, the ABC algorithm has been used to solve the Travelling Salesman Problem (TSP) for the data collected from Library of Traveling Salesman Problem (TSPLIB) from Zuse Institute Berlin. The data set consists of 29 cities in Bavaria with nodes for every city and a distance from one city to another city. MATLAB software version R2015a is used to solve the TSP Traveller has to visit all 29 cities and return to the first city, City 1. The result shows that the best solution for the shortest distance travellers has to travel in all 29 cities is 3974km.

REFERENCES

Akhand, M. A. H., Ayon, S. I., Shahriyar, S. A., Siddique, N., & Adeli, H. (2019). Discrete Spider Monkey Optimization for Traveling Salesman Problem. *Applied Soft Computing*, 105887.

Guo, Y., Li, X., Tang, Y., & Li, J. (2017). Heuristic artificial bee colony algorithm for uncovering community in complex networks. *Mathematical Problems in Engineering*, 2017.

Kaspi, M., Zofi, M., & Teller, R. (2019). Maximising the Profit per Unit Time for the Travelling Salesman Problem. *Computers & Industrial Engineering*.

Khamis, N., Selamat, H., Ismail, F. S., Lutfy, O. F., Haniff, M. F., & Nordin, I. N. A. M. (2019). Optimised exit door locations for a safer emergency evacuation using crowd evacuation model and artificial bee colony optimisation. *Chaos, Solitons & Fractals*, 109505.

Khan, I., & Maiti, M. K. (2019). A swap sequence based artificial bee colony algorithm for traveling salesman problem. *Swarm and evolutionary computation*, 44, 428-438.

Lvshan, Y., Dongzhi, Y., & Weiyu, Y. (2017, November). Artificial bee colony algorithm with genetic algorithm for job shop scheduling problem. In *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)* (pp. 433-438). IEEE.

Mridula, K. M., Rahman, N., & Ameer, P. M. (2018). Sound velocity profile estimation using ray tracing and nature inspired meta-heuristic algorithms in underwater sensor networks. *IET Communications*, 13(5), 528-538.
O’Neil, R. J., & Hoffman, K. (2019). Decision diagrams for solving traveling salesman problems with pickup and delivery in real time. *Operations Research Letters, 47*(3), 197-201.

Pandiri, V., & Singh, A. (2018). A hyper-heuristic based artificial bee colony algorithm for k-Interconnected multi-depot multi-traveling salesman problem. *Information Sciences, 463*, 261-281.

Xu, J., Pei, L., & Zhu, R. Z. (2018). Application of a genetic algorithm with random crossover and dynamic mutation on the travelling salesman problem. *Procedia computer science, 131*, 937-945.

Zuloaga, M. S., & Moser, B. R. (2017, July). Optimising resource allocation in a portfolio of projects related to technology infusion using heuristic and meta-heuristic methods. In *2017 Portland International Conference on Management of Engineering and Technology (PICMET)* (pp. 1-23). IEEE.