Journal of Computing Research and Innovation (JCRINN) Vol. 7 No.2 (2022) (pp121-131)
https://jcrinn.com : eISSN: 2600-8793 / doi: 10.24191/jcrinn.v7i2.295
https://doi.org/10.24191/jcrinn.v7i2.295

Solving the Travelling Salesman Problem by Using Artificial Bee Colony Algorithm

Siti Hafawati Jamaluddin[1]*, Noor Ainul Hayati Mohd Naziri[2], Norwaziah Mahmud[3], Nur Syuhada Muhammat Pazil[4]
[1,2,3]Faculty of Computer and Mathematical Sciences, University Teknologi MARA Cawangan Perlis, Kampus Arau, Malaysia
[4]Faculty of Computer and Mathematical Sciences, University Teknologi MARA Cawangan Melaka, Kampus Jasin, Malaysia

*Corresponding author: *hafawati832@uitm.edu.my*
Received Date: 15 July 2022
Accepted Date: 5 August 2022
Revised Date: 20 August 2022
Published Date: 1 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:

$$min \sum_{i=1}^{n} \sum_{j\neq i, j=1}^{n} c_{ij} x_{ij} : \qquad 0 \le x_{ij} \le 1 \qquad i, j = 1, \ldots, n; \qquad (1)$$

Subject to:

$$\sum_{i=1, i\neq j}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n; \qquad (2)$$

$$\sum_{j=1, j\neq i}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n; \qquad (3)$$

$$\sum_{i\in Q}^{n} \sum_{j=Q}^{n} x_{ij} \le |Q| - 1 \qquad \square Q \square \{1, \ldots, n\}, |Q| \ge 2 \qquad (4)$$

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 city once 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 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 and scout bees move arbitrarily inside the chosen area.

All employed bees produced the initial sources of food. The steps have to be repeated until they meet all the requirements. In the first step, employed bees must find and determine the closest food source in their memory. After that, the food source has to evaluate because the employed bees should dance in the hive to tell the onlooker bees about the food source. Then, depending on the dance by employed bees, onlooker bees will choose one of their sources. 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.

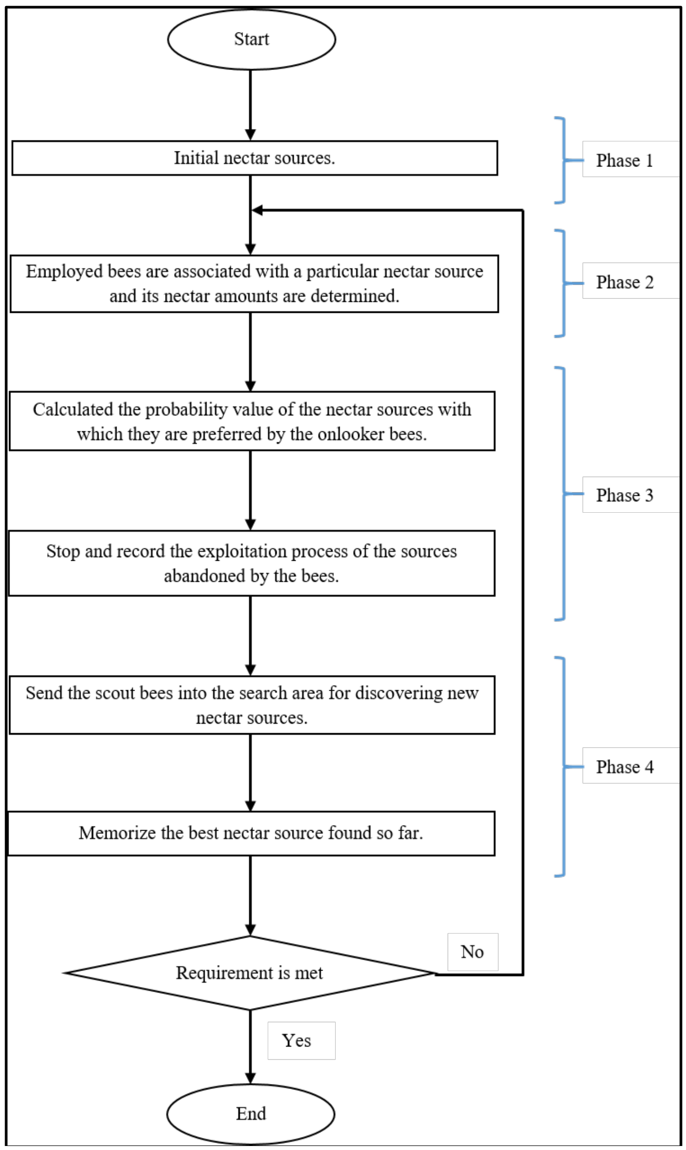

Figure 1: Flowchart of the ABC Algorithm
(Source: Guo, Li, Tang & Li, 2017)

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:

$$x_{ij} = x_j^{\max} + \text{rand}(0,1)\,(x_j^{\max} - x_j^{\min}) \tag{5}$$

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 $x_j^{\max}$ and $x_j^{\min}$ are respectively to the upper and lower bounds for the dimension j^{th}.

Consider the optimisation problem as $Minimize\ f(x) = x_1^2 + x_2^2$ with the range $-5 \le x_1, x_2 \le 5$ 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).

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

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:

$$fit_i = \begin{cases} \dfrac{1}{(1 + OBJ(x_i))} & if \quad OBJ(x_i) \ge 0 \\ 1 + abs(OBJ(x_i)) & if \quad OBJ(x_i) \le 0 \end{cases} \tag{6}$$

where fit_i is the value of fitness, $OBJ(x_i)$ is the objective value of the solution x_i and $abs(OBJ(x_i))$ is the absolute value of $OBJ(x_i)$.

Since all *f(x)* are positive, then the $fit_i = \dfrac{1}{(1 + OBJ(x_i))}$. So the initial fitness values are 0.1045, 0.3047 and 0.4764.

Journal of Computing Research and Innovation (JCRINN) Vol. 7 No.2 (2022) (pp121-131)
https://jcrinn.com : eISSN: 2600-8793 / doi: 10.24191/jcrinn.v7i2.295
https://doi.org/10.24191/jcrinn.v7i2.295

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:

$$x'_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \tag{7}$$

where k is different from i, $k \ni \{1,2,\ldots,SN/2\}$, $j \ni \{1,2,\ldots,D\}$ are randomly chosen indexes and φ_{ij} 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 φ_{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):

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN/2} fit_n} \tag{8}$$

where fit_i is the fitness value of the i^{th} source of food, x_i. The higher 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_i. Like the employed bees phase, a greedy selection is performed according to their fitness values between x_i and x_k.

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:

$$x'_{ij} = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}) \quad (9)$$

where i is the index of the employed bees whose "trail" values reach the "limit" value firstly, $j \ni \{1,2,\ldots,D\}$, rand(0,1) is a uniform random number in the range [0,1] and $x_j^{\max}$ and $x_j^{\min}$ 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

Journal of Computing Research and Innovation (JCRINN) Vol. 7 No.2 (2022) (pp121-131)
https://jcrinn.com : eISSN: 2600-8793 / doi: 10.24191/jcrinn.v7i2.295
https://doi.org/10.24191/jcrinn.v7i2.295

13	970	1340
14	510	700
15	750	900
16	1280	1200
17	230	590
18	460	860
19	1040	950
20	590	1390
21	830	1770
22	490	500
23	1840	1240
24	1260	1500
25	1280	790
26	490	2130
27	1460	1420
28	1260	1910
29	360	1980

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	0	107	241	190	124	80	316	76	152	157	283	133	113	297	228	129	348	276	188	150	65	341	184	67	221	169	108	45	167
2	107	0	148	137	88	127	336	183	134	95	254	180	101	234	175	176	265	199	182	67	42	278	271	146	251	105	191	139	79
3	241	148	0	374	171	259	509	317	217	232	491	312	280	391	412	349	422	356	355	204	182	435	417	292	424	116	337	273	77
4	190	137	374	0	202	234	222	192	248	42	117	287	79	107	38	121	152	86	68	70	137	151	239	135	137	242	165	228	205
5	124	88	171	202	0	61	392	202	46	160	319	112	163	322	240	232	314	287	238	155	65	366	300	175	307	57	220	121	97
6	80	127	259	234	61	0	386	141	72	167	351	55	157	331	272	226	362	296	232	164	85	375	249	147	301	118	188	60	185
7	316	336	509	222	392	386	0	233	438	254	202	439	235	254	210	187	313	266	154	282	321	298	168	249	95	437	190	314	435
8	76	183	317	192	202	141	233	0	213	188	272	193	131	302	233	98	344	289	177	216	141	346	108	57	190	245	43	81	243
9	152	134	217	248	46	72	438	213	0	206	365	89	209	368	286	278	360	333	284	201	111	412	321	221	353	72	266	132	111
10	157	95	232	42	160	167	254	188	206	0	159	220	57	149	80	132	193	127	100	28	95	193	241	131	169	200	161	189	163
11	283	254	491	117	319	351	202	272	365	159	0	404	176	106	79	161	165	141	95	187	254	103	279	215	117	359	216	308	322
12	133	180	312	287	112	55	439	193	89	220	404	0	210	384	325	279	415	349	285	217	138	428	310	200	354	169	241	112	238
13	113	101	280	79	163	157	235	131	209	57	176	210	0	186	117	75	231	165	81	85	92	230	184	74	150	208	104	158	206
14	297	234	391	107	322	331	254	302	368	149	106	384	186	0	69	191	59	35	125	167	255	44	309	245	169	327	246	335	288
15	228	175	412	38	240	272	210	233	286	80	79	325	117	69	0	122	122	56	56	108	175	113	240	176	125	280	177	266	243
16	129	176	349	121	232	226	187	98	278	132	161	279	75	191	122	0	244	178	66	160	161	235	118	62	92	277	55	155	275
17	348	265	422	152	314	362	313	344	360	193	165	415	231	59	122	244	0	66	178	198	286	77	362	287	228	358	299	380	319
18	276	199	356	86	287	296	266	289	333	127	141	349	165	35	56	178	66	0	112	132	220	79	296	232	181	292	233	314	253
19	188	182	355	68	238	232	154	177	284	100	95	285	81	125	56	66	178	112	0	128	167	169	179	120	69	283	121	213	281
20	150	67	204	70	155	164	282	216	201	28	187	217	85	167	108	160	198	132	128	0	88	211	269	159	197	172	189	182	135
21	65	42	182	137	65	85	321	141	111	95	254	138	92	255	175	161	286	220	167	88	0	299	229	104	236	110	149	97	108
22	341	278	435	151	366	375	298	346	412	193	103	428	230	44	113	235	77	79	169	211	299	0	353	289	213	371	290	379	332
23	184	271	417	239	300	249	168	108	321	241	279	310	184	309	240	118	362	296	179	269	229	353	0	121	162	345	80	189	342
24	67	146	292	135	175	147	249	57	221	131	215	200	74	245	176	62	287	232	120	159	104	289	121	0	154	220	41	93	218
25	221	251	424	137	307	301	95	190	353	169	117	354	150	169	125	92	228	181	69	197	236	213	162	154	0	352	147	247	350
26	169	105	116	242	57	118	437	245	72	200	359	169	208	327	280	277	358	292	283	172	110	371	345	220	352	0	265	178	39
27	108	191	337	165	220	188	190	43	266	161	216	241	104	246	177	55	299	233	121	189	149	290	80	41	147	265	0	124	263
28	45	139	273	228	121	60	314	81	132	189	308	112	158	335	266	155	380	314	213	182	97	379	189	93	247	178	124	0	199
29	167	79	77	205	97	185	435	243	111	163	322	238	206	288	243	275	319	253	281	135	108	332	342	218	350	39	263	199	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	Node	
	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

1	1150	1760

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.

ACKNOWLEDGMENTS

The authors express their gratitude to the colleagues and research team members for their collaboration and discussions, which have improved methodologies presented in this work.

CONFLICT OF INTERESTS DECLARATION

The authors declare no conflict of interests regarding the publication of this article.

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.