e-ISSN 2477-6025
DOI: 10.21776/ub.jemis.2021.009.02.1

# NATURE INSPIRED METAHEURISTICS COMPARATIVE STUDY TO SOLVE TRAVELING SALESMAN PROBLEM

**Agung Chandra[1*], Aulia Naro[2]**
Program Studi Teknik Industri, Universitas Mercu Buana[1),2)]

***Abstract*** *There are numerous optimization method to solve the traveling salesman problem, TSP. One of methods is metaheuristics which is the state of the art algorithm that can solve the large and complex problem. The metaheuristics method is categorized as an approximate method can produce near-optimal solution for complex problem and even optimal solution for small problem in far shorter time than exact method. During Covid19 pandemic, most companies are trying to run business in more efficient and effective way, no exception in transportation sector. In this research, three of well-known nature inspired population based metaheuristics algorithm: Ant Colony Optimization – ACO, Artificial Bee Colony – ABC and Particle Swarm Optimization – PSO are compared to solve the 29 destinations in F&B company by using Matlab program. The result of this study is ACO produces the shortest distance, 94 kilometers and is 12.77% more efficient than ABC and 20.21 more efficient than PSO methods; but in process time consideration, the ABC has the fastest time to reach the optimality than others eventhough ACO reach optimality at 276 iterations; ABC reach at 861 iterations, and PSO reach at 10,000 iterations. For the next research, these methods should be tested in larger example and compared with Exact algorithm.*

***Keywords**: TSP, ACO, ABC, PSO*

## 1. Introduction

In this covid19 pandemic era, every profit oriented organization is trying to undergo this era by reducing the unnecessary cost in every aspect such as overtime, administration cost, electricity, fuel, water, transportation cost without disturbing operational works. This phenomena happens when sales is going down and organization keeps struggling alive. Transportation's aspect is the only one example. Factors that involved in the transportation are distance and time. In general, shorter distance means shorter time and less fuel used by transporter. Less fuel means greener transportation. Searching the shortest distance from the point of origin to destination is the goal of traveling salesman problem, call in short is TSP. Each destination is only visited only once [1]. TSP is categorized as a NP hard because large and complex and also called combinatorial optimization problem [2]. The applications of TSP are in logistics, genetics, manufacturing, telecommunications, neuroscience, scheduling, order picking in warehouse, transportation, school bus routing, et cetera. In addition, it is important to note that TSP is the same with single vehicle routing problem (VRP), then, in literatures, both are used interchangeably [3].

One of the popular technique is Metaheuristics which an optimization method that can be used to find the approximate solution in a complex and large problems and finding the high quality solution to hard which practically relevant combinatorial optimization problem which often easy to state but very difficult to solve in a reasonable time. Metaheuristics, as a approximate method can give near-optimal solution for complex problems and even optimal solution for small problems. This is the reason why metaheuristics become state of the art optimization algorithms. On the other hand, exact methods can produce optimal solution but time-consuming process when the problem becomes larger. The metaheuristics method is useless when exact algorithm can give acceptable search time to solve the target resources and the problem is easy to solve.

*Corresponding author Email: agung.chandra@mercubuana.ac.id
Published online at http://Jemis.ub.ac.id
Copyright ©2021 JTI UB Publishing. All Right Reserved
Cite this Article As Chandra, Agung et.al. (2021). Nature Inspired Metaheuristics Comparative Study To Solve Traveling Salesman Problem. *Journal of Engineering and Management in Industrial System,* 9 (2), p1-9
Paper Submitted : 6th July 2022
Paper Published : 30th Nov 2022

e-ISSN 2477-6025
DOI: 10.21776/ub.jemis.2021.009.02.1

There are two types of metaheuristics method: one is single solution based and the other one is population based. Single solution based metaheuristics are exploitation oriented which they have power to intensify the search in local regions and population based metaheuristics are exploration oriented. Bee colonies, particle swarm intelligence, and ant colony optimization is categorized as Nature Inspired Population Metaheuristics. Ant Colony and Particle Swarm intelligence are the most successful swarm intelligence inspired optimization problems [4].

On the other hand, Bee Colony Optimization – BCO or Artificial Bee Colony – ABC optimization is based on the concept that can increase the efficiency of artificial bees and allows reaching goals that could not be reached only by individual actions. BCO also has the capability to intensify the search the solution space and become popular because of its simplicity [5]. The performance analysis of optimization methods proved that ABC algorithm only deviates 0.0197% compared to Best Known Solution (BKS) [6].

In this study, the three powerful nature inspired metaheuristics algorithms mentioned above, Ant Colony Optimization – ACO, Artificial Bee Colony – ABC, and Particle Swarm Optimization – PSO will be compared, then we choose the appropriate algorithms for getting the shortest distance. These algorithms will be tested in distribution problem of food and beverage company from distribution center called depot to 29 its branches in Jakarta that happens daily. This study will give the benefit for the company wants to support the leaning program during this pandemic and to pass the most difficult condition.

## 2. Methods

### 2.1 The Framework of Research

To solve the TSP and to get the shortest distance for 29 destinations and one distribution center, MATLAB 2015a, Intel Core i5 7200 U CPU 2.5 GHz, 32 bit ACPI x64 based PC was used and the inputs are: number of cities, city coordinates or distance matrix using Euclidean distance method, and the output is total distance in kilometer unit and total time needed to process the result. The coordinates – latitudes and longitudes are derived from Google maps and converted into radian coordinates for every destination. The code of MATLAB for every algorithm can be designed input both coordinates and distances. Below is the framework of research at figure 1, coordinates of destinations at table 1 and the distances at table 2:

e-ISSN 2477-6025
DOI: 10.21776/ub.jemis.2021.009.02.1

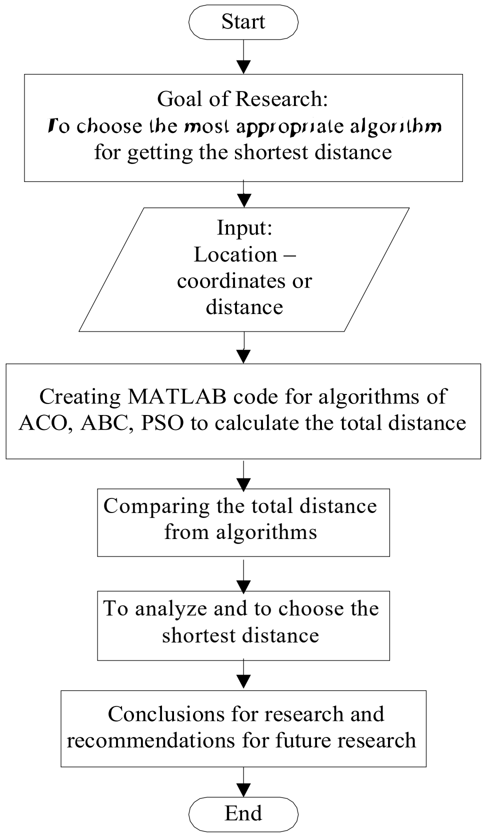


**Figure 1**. Framework of Research

**Table 1.** The Coordinates of Destinations

e-ISSN 2477-6025
DOI: 10.21776/ub.jemis.2021.009.02.1

**Table 1.** The Coordinates of Destinations

| No | City | Latitude (S) | Longitude (E) | X rad | Y rad | X (km) | Y (km) | X (peta) | Y (peta) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | **DCH** | -6.187692 | 106.773820 | -0.107995598 | 1.863554714 | -688.8181743 | 11886.13586 | 12.61 | 6.72 |
| 2 | **CG6** | -6.126652 | 106.713418 | -0.106930250 | 1.862500500 | -682.0231591 | 11879.41187 | 19.40 | 0.00 |
| 3 | **CPM** | -6.174611 | 106.790322 | -0.107767292 | 1.863842728 | -687.3619883 | 11887.97287 | 14.06 | 8.56 |
| 4 | **PRM** | -6.188177 | 106.73423 | -0.108004063 | 1.862863738 | -688.8721649 | 11881.72867 | 12.55 | 2.32 |
| 5 | **KTA** | -6.155008 | 106.817747 | -0.107425155 | 1.864321385 | -685.1797687 | 11891.02584 | 16.25 | 11.61 |
| 6 | **LMP** | -6.190000 | 106.738468 | -0.108035881 | 1.862937705 | -689.0751025 | 11882.20045 | 12.35 | 2.79 |
| 7 | **GMP** | -6.160906 | 106.818575 | -0.107528095 | 1.864335836 | -685.8363382 | 11891.11802 | 15.59 | 11.71 |
| 8 | **MTA** | -6.178593 | 106.792792 | -0.107836791 | 1.863885838 | -687.8052673 | 11888.24783 | 13.62 | 8.84 |
| 9 | **NSF** | -6.174581 | 106.789918 | -0.107766768 | 1.863835677 | -687.3586487 | 11887.9279 | 14.07 | 8.52 |
| 10 | **MDS** | -6.137537 | 106.831841 | -0.107120229 | 1.864567371 | -683.2348849 | 11892.5948 | 18.19 | 13.18 |
| 11 | **MOI** | -6.151022 | 106.892415 | -0.107355586 | 1.865624587 | -684.7360444 | 11899.33794 | 16.69 | 19.93 |
| 12 | **EMP** | -6.127337 | 106.790279 | -0.106942205 | 1.863841978 | -682.0994138 | 11887.96809 | 19.33 | 8.56 |
| 13 | **MKG** | -6.155546 | 106.908994 | -0.107434545 | 1.865913945 | -685.2396593 | 11901.18352 | 16.19 | 21.77 |
| 14 | **SDA** | -6.188272 | 106.824157 | -0.108005721 | 1.86443326 | -688.8827403 | 11891.73941 | 12.54 | 12.33 |
| 15 | **THC** | -6.188272 | 106.824157 | -0.108005721 | 1.86443326 | -688.8827403 | 11891.73941 | 12.54 | 12.33 |
| 16 | **GIF** | -6.194825 | 106.820171 | -0.108120093 | 1.864363691 | -689.6122248 | 11891.29568 | 11.81 | 11.88 |
| 17 | **ARK** | -6.300948 | 106.832048 | -0.109972289 | 1.864570984 | -701.425911 | 11892.61784 | 0.00 | 13.21 |
| 18 | **MLW** | -6.245714 | 106.803225 | -0.109008273 | 1.864067928 | -695.2772237 | 11889.40924 | 6.15 | 10.00 |
| 19 | **AMB** | -6.223353 | 106.826548 | -0.108618 | 1.864474991 | -692.7879816 | 11892.00558 | 8.64 | 12.59 |
| 20 | **LSA** | -6.223669 | 106.822842 | -0.108623516 | 1.864410309 | -692.823159 | 11891.59302 | 8.60 | 12.18 |
| 21 | **PI1** | -6.264837 | 106.784378 | -0.109342033 | 1.863738986 | -697.4060093 | 11887.31118 | 4.02 | 7.90 |
| 22 | **PFS** | -6.221231 | 106.833505 | -0.108580964 | 1.864596414 | -692.5517591 | 11892.78003 | 8.87 | 13.37 |
| 23 | **MKK** | -6.225016 | 106.841196 | -0.108647025 | 1.864730647 | -692.9731079 | 11893.6362 | 8.45 | 14.22 |
| 24 | **SCY** | -6.227516 | 106.797286 | -0.108690658 | 1.863964273 | -693.2514097 | 11888.74811 | 8.17 | 9.34 |
| 25 | **PSN** | -6.225312 | 106.799024 | -0.108652191 | 1.863994607 | -693.0060589 | 11888.94158 | 8.42 | 9.53 |
| 26 | **ECS** | -6.226045 | 106.811319 | -0.108664985 | 1.864209195 | -693.0876569 | 11890.31027 | 8.34 | 10.90 |
| 27 | **SPN** | -6.242589 | 106.844486 | -0.108953732 | 1.864788068 | -694.9293465 | 11894.00245 | 6.50 | 14.59 |
| 28 | **MBC** | -6.223893 | 106.87782 | -0.108627425 | 1.865369856 | -692.8480948 | 11897.71321 | 8.58 | 18.30 |
| 29 | **AMC** | -6.172308 | 106.95204 | -0.107727097 | 1.86666524 | -687.1056167 | 11905.97543 | 14.32 | 26.56 |

**Table 2**. The Distance Matrix

| | DCH | CG6 | CPM | PRM | KTA | LMP | GMP | MTA | NSF | MDS | MOI | EMP | MKG | SDA | THC | GIF | ARK | MLW | AMB | LSA | PI1 | PFS | MKK | SCY | PSN | ECS | SPN | MBC | AMC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **DCH** | 0 | 10 | 3 | 5 | 7 | 4 | 6 | 3 | 3 | 9 | 14 | 7 | 16 | 6 | 5 | 6 | 15 | 8 | 8 | 7 | 9 | 8 | 9 | 6 | 6 | 6 | 10 | 13 | 20 |
| **CG6** | 10 | 0 | 11 | 8 | 12 | 8 | 13 | 11 | 11 | 14 | 20 | 9 | 22 | 15 | 14 | 15 | 24 | 17 | 17 | 17 | 18 | 17 | 18 | 15 | 15 | 16 | 20 | 22 | 27 |
| **CPM** | 3 | 11 | 0 | 7 | 4 | 6 | 4 | 1 | 1 | 7 | 12 | 6 | 14 | 5 | 4 | 4 | 15 | 9 | 7 | 7 | 11 | 8 | 8 | 6 | 6 | 7 | 10 | 12 | 18 |
| **PRM** | 5 | 8 | 7 | 0 | 10 | 1 | 10 | 7 | 7 | 13 | 18 | 10 | 20 | 10 | 10 | 10 | 17 | 10 | 11 | 11 | 11 | 12 | 13 | 9 | 9 | 10 | 14 | 17 | 25 |
| **KTA** | 7 | 12 | 4 | 10 | 0 | 10 | 1 | 4 | 4 | 3 | 9 | 5 | 11 | 4 | 5 | 5 | 17 | 11 | 8 | 8 | 13 | 8 | 9 | 9 | 9 | 8 | 11 | 11 | 15 |
| **LMP** | 4 | 8 | 6 | 1 | 10 | 0 | 10 | 7 | 6 | 12 | 18 | 10 | 20 | 10 | 9 | 10 | 17 | 10 | 11 | 11 | 11 | 12 | 13 | 9 | 9 | 10 | 14 | 16 | 24 |
| **GMP** | 6 | 13 | 4 | 10 | 1 | 10 | 0 | 4 | 4 | 3 | 9 | 5 | 11 | 4 | 5 | 5 | 16 | 10 | 8 | 8 | 13 | 7 | 8 | 8 | 8 | 8 | 10 | 10 | 15 |
| **MTA** | 3 | 11 | 1 | 7 | 4 | 7 | 4 | 0 | 1 | 3 | 9 | 5 | 11 | 4 | 4 | 4 | 16 | 10 | 8 | 8 | 13 | 7 | 8 | 8 | 8 | 8 | 10 | 10 | 15 |
| **NSF** | 3 | 11 | 1 | 7 | 4 | 6 | 4 | 1 | 0 | 7 | 12 | 6 | 14 | 5 | 4 | 4 | 15 | 9 | 7 | 7 | 11 | 8 | 8 | 6 | 6 | 7 | 10 | 12 | 18 |
| **MDS** | 9 | 14 | 7 | 13 | 3 | 12 | 3 | 3 | 7 | 0 | 7 | 5 | 9 | 6 | 7 | 7 | 19 | 13 | 10 | 10 | 16 | 10 | 10 | 11 | 11 | 11 | 12 | 11 | 14 |
| **MOI** | 14 | 20 | 12 | 18 | 9 | 18 | 9 | 9 | 12 | 7 | 0 | 12 | 2 | 9 | 10 | 10 | 18 | 15 | 11 | 12 | 18 | 11 | 10 | 14 | 14 | 13 | 12 | 9 | 8 |
| **EMP** | 7 | 9 | 6 | 10 | 5 | 10 | 5 | 5 | 6 | 5 | 12 | 0 | 14 | 8 | 9 | 9 | 20 | 14 | 12 | 12 | 16 | 12 | 13 | 12 | 11 | 12 | 15 | 15 | 19 |
| **MKG** | 16 | 22 | 14 | 20 | 11 | 20 | 11 | 11 | 14 | 9 | 2 | 8 | 0 | 11 | 12 | 11 | 19 | 16 | 12 | 13 | 19 | 12 | 11 | 15 | 15 | 14 | 13 | 9 | 6 |
| **SDA** | 6 | 15 | 5 | 10 | 4 | 10 | 4 | 4 | 5 | 6 | 9 | 9 | 11 | 0 | 2 | 1 | 13 | 7 | 4 | 4 | 10 | 4 | 5 | 6 | 5 | 5 | 7 | 8 | 15 |
| **THC** | 5 | 14 | 4 | 10 | 5 | 9 | 5 | 4 | 4 | 7 | 10 | 9 | 12 | 2 | 0 | 1 | 12 | 6 | 4 | 4 | 9 | 4 | 5 | 5 | 4 | 4 | 7 | 8 | 16 |
| **GIF** | 6 | 15 | 4 | 10 | 5 | 10 | 5 | 4 | 4 | 7 | 10 | 20 | 11 | 1 | 1 | 0 | 12 | 6 | 4 | 4 | 9 | 4 | 5 | 5 | 5 | 4 | 6 | 8 | 15 |
| **ARK** | 15 | 24 | 15 | 17 | 17 | 17 | 16 | 16 | 15 | 19 | 18 | 14 | 19 | 13 | 12 | 12 | 0 | 7 | 9 | 9 | 7 | 9 | 9 | 10 | 10 | 9 | 7 | 10 | 20 |
| **MLW** | 8 | 17 | 9 | 10 | 11 | 10 | 10 | 10 | 9 | 13 | 15 | 12 | 16 | 7 | 6 | 6 | 7 | 0 | 4 | 4 | 3 | 5 | 5 | 3 | 3 | 3 | 5 | 9 | 19 |
| **AMB** | 8 | 17 | 7 | 11 | 8 | 11 | 8 | 8 | 7 | 10 | 11 | 12 | 12 | 4 | 4 | 4 | 9 | 4 | 0 | 1 | 7 | 1 | 2 | 4 | 4 | 2 | 3 | 6 | 16 |
| **LSA** | 7 | 17 | 7 | 11 | 8 | 11 | 8 | 8 | 7 | 10 | 12 | 16 | 13 | 4 | 4 | 4 | 9 | 4 | 1 | 0 | 7 | 2 | 3 | 3 | 3 | 2 | 4 | 7 | 16 |
| **PI1** | 9 | 18 | 11 | 11 | 13 | 11 | 13 | 13 | 11 | 16 | 18 | 12 | 19 | 10 | 9 | 9 | 7 | 3 | 7 | 7 | 0 | 8 | 8 | 5 | 5 | 6 | 8 | 12 | 22 |
| **PFS** | 8 | 17 | 8 | 12 | 8 | 12 | 7 | 7 | 8 | 10 | 11 | 13 | 12 | 4 | 4 | 4 | 9 | 5 | 1 | 2 | 8 | 0 | 1 | 5 | 4 | 3 | 3 | 5 | 15 |
| **MKK** | 9 | 18 | 8 | 13 | 9 | 13 | 8 | 8 | 8 | 10 | 10 | 12 | 11 | 5 | 5 | 5 | 9 | 5 | 2 | 3 | 8 | 1 | 0 | 5 | 5 | 4 | 2 | 5 | 14 |
| **SCY** | 6 | 15 | 6 | 9 | 9 | 9 | 8 | 8 | 6 | 11 | 14 | 11 | 15 | 6 | 5 | 5 | 10 | 3 | 4 | 3 | 5 | 5 | 5 | 0 | 1 | 2 | 6 | 9 | 19 |
| **PSN** | 6 | 15 | 6 | 9 | 9 | 9 | 8 | 8 | 6 | 11 | 14 | 12 | 15 | 5 | 4 | 5 | 10 | 3 | 4 | 3 | 5 | 4 | 5 | 1 | 0 | 2 | 6 | 9 | 18 |
| **ECS** | 6 | 16 | 7 | 10 | 8 | 10 | 8 | 8 | 7 | 11 | 13 | 15 | 14 | 5 | 4 | 4 | 9 | 3 | 2 | 2 | 6 | 3 | 4 | 2 | 2 | 0 | 5 | 8 | 17 |
| **SPN** | 10 | 20 | 10 | 14 | 11 | 14 | 10 | 10 | 10 | 12 | 12 | 15 | 13 | 7 | 7 | 6 | 7 | 5 | 3 | 4 | 8 | 3 | 2 | 6 | 6 | 5 | 0 | 5 | 15 |
| **MBC** | 13 | 22 | 12 | 17 | 11 | 16 | 10 | 10 | 12 | 11 | 9 | 19 | 9 | 8 | 8 | 8 | 10 | 9 | 6 | 7 | 12 | 5 | 5 | 9 | 9 | 8 | 5 | 0 | 11 |
| **AMC** | 20 | 27 | 18 | 25 | 15 | 24 | 15 | 15 | 18 | 14 | 8 | 19 | 6 | 15 | 16 | 15 | 20 | 19 | 16 | 16 | 22 | 15 | 14 | 19 | 18 | 17 | 15 | 11 | 0 |

The distance matrix in table 2 is assumed to be symmetric TSP that means the go and back distance is the same.

e-ISSN 2477-6025
DOI: 10.21776/ub.jemis.2021.009.02.1

### 2.2 Ant Colony Optimization – ACO

ACO was formalized as a metaheuristic in 1999 and has general characteristics in common with different algorithms such as population based incremental learning [7], mutual information maximizing input clustering (MIMIC) [8], cross entropy [9], stochastic gradient descent [10], and estimation of distribution algorithm [11]. ACO algorithms have been tested on a large number of academic problems such traveling salesman as well as assignment, scheduling, subset and constraint satisfaction problems, and world class performance has been achieved and with dozens of application areas. ACO algorithms are easily applied, easily understandable, and not obscured by many technicalities and is a standard test bed for new algorithmic ideas. ACO for TSP was also found as the most efficient ones for variety of problems. In TSP, tours are constructed by applying the following procedure to each ant: (1) choose a start city at which the ant is positioned according to some criterion; (2) use pheromone ($\eta_{ij} = 1/d_{ij}$) and heuristic values to probabilistically construct a tour by iteratively adding cities that the and has not visited yet until all cities have been visited and (3) go back to the initial city. After all ants have finished their tour, they may deposit pheromone on the tours they have followed. The following is a high-level view of an ACO algorithm for the TSP [12]:

```
Procedure ACOforTSP
 InitializeData
 While (not terminate) do
        ConstructSolutions
        LocalSearch
        UpdateStatistics
        UpdatePheromoneTrails
 End-while
End procedure
```

**Figure 2**. High Level View of An ACO Algorithm for The TSP
Source: Dorigo, M. & Stutzle, T: 2004

### 2.3 Artificial Bee Colony Optimization – ABC

ABC was proposed by Dervis Karaboga for numerical function optimization in 2005 and as a well known techniques for solving the continuous problems and can be applied successfully in discrete problems such as TSP [13, 14, 15]. ABC became a very popular method and attracted the interest of many researchers because its algorithm is simple, easy, very fast, and can be effectively applied to the combinatorial optimization problems. ABC has three essential components: food sources, employed bees and unemployed bees. There are two types of unemployed bees, onlooker bees and scout bees. This method was inspired by honey bee colonies and based on observing the nourishment behavior of honey bees [16]. A short algorithm is at figure 2 [17].

```
Initialize population
Repeat
Place the employed bees on their food sources and determine their nectar amounts
Calculate the probability value of the sources with which they are preferred by the onlooker bees
Place the onlooker bees on the food sources depending on their nectar amounts
Stop the exploitation process of the sources exhausted by the bees
Send the scouts to the search area for discovering new food sources randomly
Memorize the best food source found so far
Until requirements are met
```

**Figure** 3. Short Algorithm

Then, in this research, the parameter level and chosen values for ABC algorithm is following:

**Table 3.** Parameter and Selected Value

| Parameters | Selected Value |
|---|---|
| Food number / food source positions | 29 |
| Employed bee number | 15 |
| Onlooker bee number | 15 |
| Limit | 100 |
| Iteration number | 10,000 |

There are three manipulating operators that used: swap, insertion and reversion for exchanging between two positions.
The general steps as follow:

| DCH | CG6 | CPM | PRM | KTA | Total distance |
|---|---|---|---|---|---|
| 0 | 10 | 11 | 7 | 10 | 39 |

Fitness is 0.025
Now, employed bees evaluate every possible solution and communicate to onlooker bee. Then we use the local search by exchanging two positions randomly, for example between CPM and PRM, now the distance becomes:

| DCH | CG6 | PRM | CPM | KTA | Total distance |
|---|---|---|---|---|---|
| 0 | 10 | 8 | 7 | 4 | 29 |

Fitness is 0.033

e-ISSN 2477-6025
DOI: 10.21776/ub.jemis.2021.009.02.1

Now, a food source with a higher fitness value will have more chances to be selected, then this solution will record as the new food source. The process can be summarized with this:

1. Choose a food source
2. Record food source in the memory
3. Apply local search
4. Evaluate founded food source
5. If the new food source has better quality then record the new best solution; otherwise continue with the search
6. Go back to step 3 until the maximum number of iterations is reached

### 2.4 Particle Swarm Optimization – PSO

PSO is a stochastic optimization method or a biological inspired or natured inspired computational search and optimization method developed by Eberhart and Kennedy in 1995 and categorized as a population methods metaheuristics. It draws the origin of the ecosystem, specifically the social behavior of animals living in swarms, such as schools of fish and grouped flights of birds [18].

Discrete PSO is not as powerful as some specific algorithms, but can easily be modified for any discrete / combinatorial problem. The basic principle is very simple. A set of moving particles (the swarm) is initially thrown inside the search space. Each particle has the following criterions [19], [20]:

a. It has a position and a velocity
b. It knows its position, and the objective function value for this position
c. It knows its neighbors, best previous position and objective position function value
d. It remembers its best previous position

Swarm size S equal to N-1 which in this paper N is number of cities = 29 cities and S = 28. The algorithm as follows:

$$\begin{cases} v_{t+1} = c_1 v_t + c_2 (p_{i,t} - x_t) + c_3 (p_{g,t} - x_t) \\ x_{t+1} = x_t + v_{t+1} \end{cases} \quad (1)$$

Where

- $v_t$ = velocity at time step t
- $x_t$ = position at time step t
- $p_{i,t}$ = best previous position at time step t
- $p_{g,t}$ = best neighbour's previous best at time step t (or best neighbor)
- $c_1, c_2, c_3$ = social / cognitive confidence coefficient time step is also recognized as iteration or step

And the distance between two positions is defined by:

$d(x_1, x_2) = |x_1 - x_2|$ ; after that, we have distance matrix

In this research of traveling salesman problem, the truck of every optimization method visits every destination only once and will be back to the initial point. The tour is assumed to be symmetric which there is no difference in costs between the forward route or backward route.

## 3. Results and Discussion

After running each algorithm in MATLAB program for twenty times, because of similarities for each program run, the results are averaged and shown in table 2. In this program, ones can input coordinates of cities or distance between two locations in distance matrix.

e-ISSN 2477-6025
DOI: 10.21776/ub.jemis.2021.009.02.1

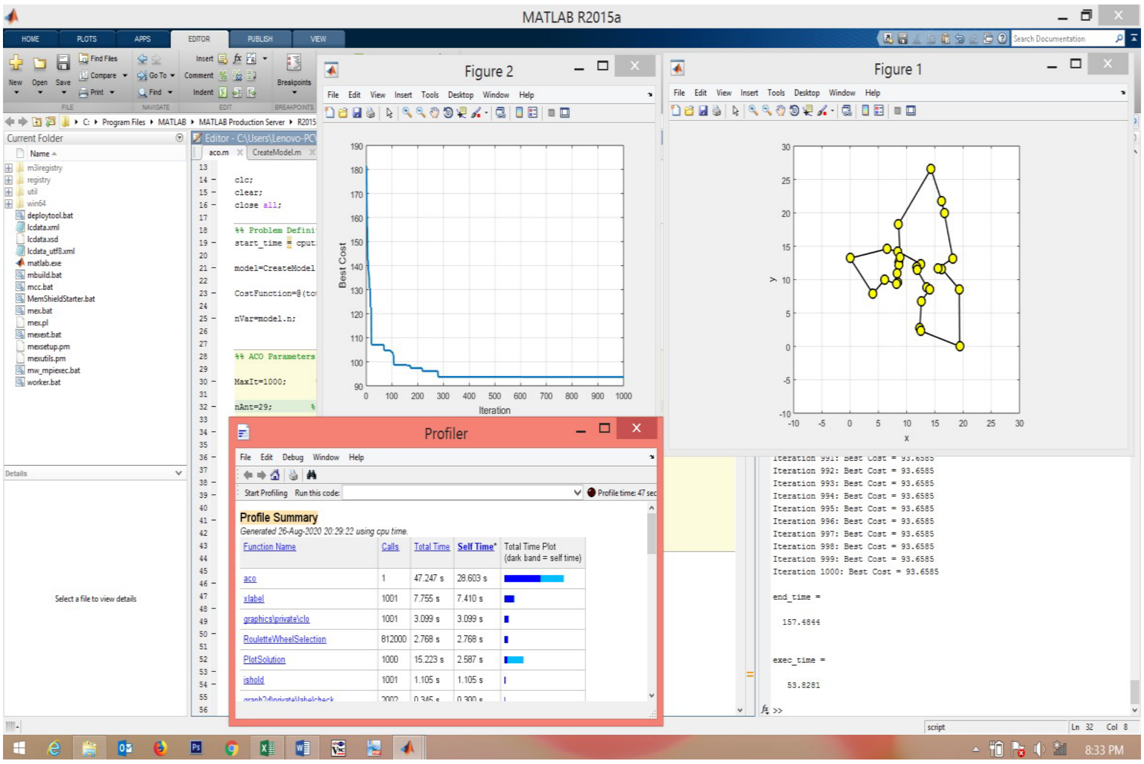

**Figure 4**. Ant Colony Optimization by Matlab

Figure 3 shows us that ACO has reached the optimality point starting at 276th iteration. It means from 276 to 1000 or more iterations, there is no better distance than 94. The figure in the graph section also tells us about the sequence from first city to other cities and back to first city and all only visited once.

The ABC also reached the optimality starting from 861st iteration, and after this iteration, it means there is no better result than 106 and can be seen in figure 5.

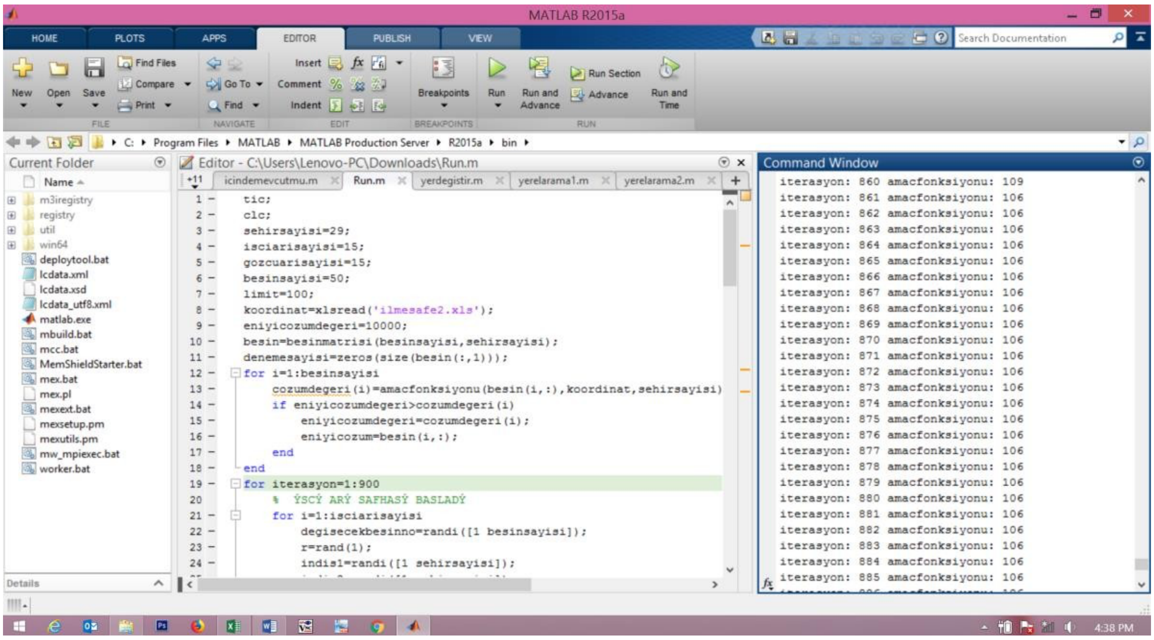

**Figure 5.** Artificial Bee Colony Optimization by Matlab

e-ISSN 2477-6025
DOI: 10.21776/ub.jemis.2021.009.02.1

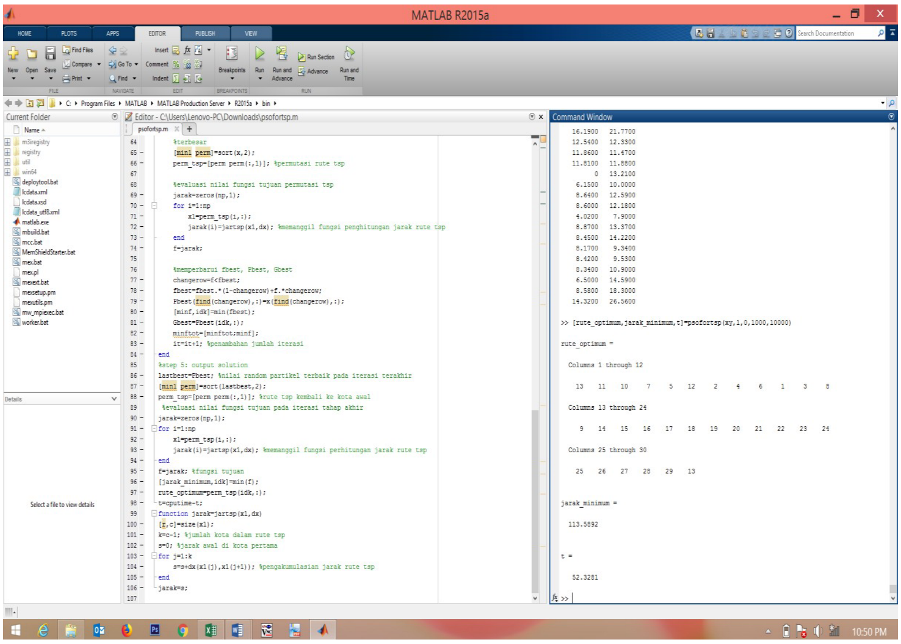


**Figure 6.** Particle Swarm Optimization

In Figure 6, it also shows the sequence of tour, distance and process time. The sequence, distance and process time. After running all matlab code, now the summary is at table 2:

**Table 3.** Summary of distance, time, and iteration

| Type of Algorithm | Distance (Kilometer) | Matlab Time (seconds) | Optimal iteration |
|---|---|---|---|
| ACO | 94 | 8.24 | 276 |
| ABC | 106 | 5.97 | 861 |
| PSO | 113 | 52.33 | 10,000 |

From Table 3, the shortest distance is reached by ACO algorithm, 94 kilometer which 12.77% shorter than ABC and 20.21% shorter than PSO, and the second place is ABC algorithm which is 6.60% shorter than PSO, it means the sequence starting from the shortest distance is ACO, ABC, and PSO.

The sequence of destinations based on ACO algorithm is: DCH – LMP – PRM – CG6 – EMP – GMP – KTA – MDS – MOI – MKG – AMC – MBC – MKK – AMB – PFS – ECS – PSN – SCY – MLW – PI1 – ARK – SPN – SDA – THC – GIF – NSF – CPM – MTA – DCH.

But from the Matlab time, the sequence fromthe fastest is ABC, ACO, and PSO. There is no algorithm that wins both distance and time. Forthe sake of the profit oriented company, one must carefully decide which method is more appropriate. If there is no traffic jam, of course, shorter distance is more beneficial. Shorter distance means more efficient fuel. But if there is traffic jam, shorter time is more beneficial. For the next research, more places or destinations are needed to convince the best algorithm especially in distance in order a company can reduce cost and continue running the business. These algorithms are also needed to compare with exact method such as Branch and Bound algorithm, then we will know how large the deviation is.

## 4. Conclusion

The Ant Colony Optimization – ACO method produce the shortest route in distance, 94 kilometers compared to two other nature inspired population metaheuristics, ABC and PSO. The ACO is more efficient around 12.77% than ABC and 20.21% than PSO. For process – time considerations, the ABC gives the faster time than ACO and PSO, which leads around 27.5% than ACO. It is recommended that the food and beverage company use this ACO method for fuel efficiency considerations and to reduce carbondioxide ($CO_2$).

**Acknowledgment**

Special thanks go to Maria Khristina (Indonesia), Muhlis Ozdemir (Istanbul University – School of Business, Department of Quantitative Method), and S. Mustapha Kalami Heris (member of Yarpiz) for supporting Matlab code. The last but not the least, thanks also go to Universitas Mercu Buana, Jakarta to encourage me in funds and supports.

e-ISSN 2477-6025
DOI: 10.21776/ub.jemis.2021.009.02.1

**References**

[1] Bonyadi M.R., Azghadi M.R., Shah- Hosseini H. Population Based Optimization Algorithm for Solving the Traveling Salesman Problem. In: Greco F. (eds.). *Traveling Salesman Problem.* In- Tech. Croatia; 2008. p.001-034.
[2] Saud S., Kodaz H., Babaoglu I. Solving Traveling Salesman Problem by Using Optimization Algorithm. The 9th International Conference on Advances in Information Technology: IAIT Conference Proceedings volume 2017.
[3] Brucato, C. Thesis: The Traveling Salesman Problem. University of Pittsburgh. 2013
[4] Talbi E.G. Metaheuristics: From Design to Implementation. New Jersey: John Wiley and Sons; 2009.
[5] Teodorovic D., Selmic M., Davidovic T. Bee Colony Optimization Part II: The Application Survey. Yugoslav Journal of Operations Research; 2015: 25 (2): 185-
[6] 219. Available from: DOI: 10.2298/YJOR131029020T.
[7] Chandra, A., and Natalia, C. Performance Analysis of Optimization Methods for Solving Traveling Salesman Problem. Innovative Technologies and Scientific Solutions for Industries; 2021: 1(15): 69-75. DOI: 10.30837/ITSSI.2021.15.069.
[8] Baluja S., Caruana R. Removing the genetics from the standard genetic algorithm. In: Prieditis A., Russell S. (eds). Proceedings of the Twelfth International Conference on Machine Learning (ML- 95); Palo Alto, California; 1995.
[9] De Bonet J.S., Isbell C.L., Viola P. MIMIC: Finding optima by estimating probability densities. In: Mozer M.C.,Jordan M.I., Petsche T. Advances in Neural Information Processing System; 1997: Vol 9: 424-431.
[10] Larranaga P., Lozano J.A. Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Genetic Algorithmsand Evolutionary Computation. Dordrecht, Netherlands, Kluwer Acedmic Publishers; 2001.
[11] Robbins H., Monroe H. A Stochastic approximation method. Annals of Mathematics and Statistics; 1951: 22: 400-407.
[12] Rubinstein R.Y. The cross entropy method for combinatorial and continuous optimization. Methodology and Computing in Applied Probability; 1999: 1: 1(2): 127-190.
[13] Dorigo M., Stutzle T. Ant Colony Optimization. Massachussetts Institute of Technology; 2004.
[14] Khan, I., and Kuma Maiti, M. A Swap Sequence based Artificial Bee Colony algorithm for Traveling Salesman Problem. Swarm and Evolutionary Computation; 2019: 44 (February): 428–438. DOI: 10.1016/j.swevo.2018.05.006.
[15] Choong, S.S., Wong, L.P., and Lim, C.P. An Artificial Bee Colony algorithm with a Modified Choice Function for the Traveling Salesman Problem. Swarm and Evolutionary Computation; 2018: 44 (February): 622–635. DOI:10.1016/j.swevo.2018.08.004.
[16] Masutti, T., and de Castro, L. Bee-Inspired Algorithm Applied to Vehicle Routing Problems: A Survey and a Proposal. Mathematical Problems in Engineering; 2017: 1–20. DOI:10.1155/2017/3046830.
[17] Onder E, Ozdemir M., Yildirim B.F. Combinatorial Optimization Using Artificial Bee Colony Algorithm and Particle Swarm Optimization. KAU IIBF Dergisi; 2013: 4(6): 59-70.
[18] Boussaid I., Lepagnot J., and Siarry P. A Survey On Optimization Metaheuristics. Information Sciences; 2013: 237: 82 – 117.
[19] Gharib A., Benhra J., and Chaouqi M. A Performance Comparison of Particle Swarm Optimization and Genetic Algorithm Applied to Traveling Salesman Problem. International Journal of Computer Applications; 2015: Vol.130 no.15: 35 – 39.
[20] Clerc M. Standard Particle Swarm Optimization From 2006 to 2011. 2012. Available from: http://www.mauriceclerc.net. [Accessed 26th August 2020].
[21] Clerc, M. 2000. Discrete Particle Swarm Optimization: Illustrated by the Traveling Salesman Problem. Available from: http://www.mauriceclerc.net. [Accessed26th August 2020].