Ant Colony Optimization (ACO) Algorithm for Determining The Nearest Route Search in Distribution of Light Food Production

H Fahmi2, M Zarlis1, E B Nababan1, and P Sihombing1

1Computer Science Doctoral Degree, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara
2Faculty of Computer Science and Information Technology, Universitas Sumatera Utara

*E-mail: h.fahmizuhri@gmail.com

Abstract. At this time the development of technology that forces everything must move fast, the efficiency of time becomes a measure of the success of a distribution or production. the continuity of production and the success of production itself is related to distribution. When the distribution fails, production will fail. Indonesian people especially in North Sumatra often consume snacks when free time and snacks are the most consumed food by the Indonesian population until now the production of snacks in Indonesia rose to 50% in 2019, causing the food production to have a concept in timely distribution without experiencing delays in distributing goods that have been produced. One of the expeditions in Indonesia experienced problems in terms of late distribution of goods up to 15-20%, this would have a bad impact on the industry in the future, so to overcome the problems above the application of the ant colony optimization algorithm can help the above problems by placing 4 ants and the ants will spread from point A to point H. In the results of this study, there are 4 ants that move together where the best results are obtained by ant 2, closest to the point H, where the value obtained reaches 0,00015

1. Introduction
With the increasing market demand in the field of food production, causing the food production industry to try to provide the best service to consumers both in serving the food production process and the process of food distribution so as not to be hampered and arrive at the place according to the desired time [1]. The production process will not run smoothly without a good distribution, because both production and distribution must run linearly so that all activities related to production and also the distribution on target and also time efficiency which is the most important part in the distribution of food production [5]. One of the snack industries in North Sumatra has a problem in the distribution of snacks causing problems in food delivery, including distance and also the best route so that food production shipments can reach consumers on time. In a previous study conducted by Soltys stated that the application of the greedy algorithm was able to overcome the problem of distribution in the industrial environment due to the nature of the greedy algorithm namely the greedy algorithm due to the greedy nature of the greedy algorithm causing the distribution to be more optimal [6]. However, the weakness
of the greedy algorithm itself is not thinking about the consequences that occur in the future because the concept of the greedy algorithm is only focused on optimal results and does not think about the consequences so the application of the ant colony Optimization (ACO) algorithm is needed because it has a good effect on determining the nearest route in distribution snack food production. The ant colony optimization algorithm is inspired by organized ant behavior, which roams around its nest to search for food. The ant colony optimization algorithm approach models the concepts of foraging, net building, division of labor, cooperative support, self-assembly and real ant burial organization for a meta-heuristic approach[2].

2. Methodology
2.1 Ant Colony Optimization (ACO)
ACO is a development of Ant Colony. Informally, ACO works as follows: the first time, a number of m ants are placed at a number of n points based on some initialization rules (for example, randomly). Each ant makes a tour (that is, a possible evacuation route solution) by applying a status transition rule repeatedly. While building the tour, each ant also modifies the number of pheromones on the edges it visits by applying the local pheromone renewal rules mentioned earlier. ACO was formulated into a meta-heuristic computing approach by Marco Dorigo in 1999 [3]. When ants find roots from (source) to food (destination), ants communicate with other ants by storing pheromone traces (chemicals) as they walk along their path. This form of indirect communication is called Stigmergy. As more ants take a certain path, pheromone concentrations increase along that pathway. Pheromones along the road gradually evaporate, reducing their concentration on the road. Among the many paths between nests and food ants, choose a single optimal path based on maximum pheromone concentration along the path and several heuristic functions [4]. The ACO algorithm is defined as follows:

Algorithm 1: ANT Colony based Optimization
Input: An instance x of a combine optimization problem While termination condition not met do
Schedule activities
Ant based solution construction ( )
Pheromone update ( )
Daemon actions ( )
End scheduled activities
S_{best} → best solution in the population of solutions End while
Output: S_{best} candidate to optimal solution for x

Build solutions by selecting a subset of component sets. The solution starts with an empty partial solution and then at each step of construction a viable component is added. The selection of appropriate components is then made to choose the equation for the path algorithm the ant uses. Update Pheromone serves two tasks: To increase the value of good component pheromones, and to reduce the value of bad component pheromones. Pheromone reduction is achieved through evaporation. Many different algorithms have been proposed with different pheromone update equations. Daemon actions are usually used to perform centralized actions that cannot be carried out by a single ant and which may be a special problem. This action decides to save extra pheromones on the solution components that are included in the best solution. Researchers took samples of distributing their products to many cities. The city closest to the position of the cottage industry is illustrated by the city of Lubuk Pakam. the cities that often ask for our products are Tebing Tinggi, Pematang Siantar, Medan, Binjai, Kabanjahe, Sidikalang and Aceh. In table 1 is the distribution path with the following nodes,
Table 1. Name and City Code

| No | City            | Code |
|----|----------------|------|
| 1  | Lubuk Pakam    | A    |
| 2  | Tebing Tinggi | B    |
| 3  | Pematang Siantar | C   |
| 4  | Medan          | D    |
| 5  | Binjai         | E    |
| 6  | Kabanjahe      | F    |
| 7  | Sidikalang     | G    |
| 8  | NAD            | H    |

Table 2. Distance between cities

| No | Code City | Distance |
|----|-----------|----------|
| 1  | A - B     | 40 km    |
| 2  | A - C     | 69 km    |
| 3  | A - D     | 23 km    |
| 4  | A - F     | 65 km    |
| 5  | B - C     | 46 km    |
| 6  | C - F     | 100 km   |
| 7  | F - G     | 76 km    |
| 8  | G - H     | 386 km   |
| 9  | D - E     | 42 km    |
| 10 | D - F     | 75 km    |
| 11 | E - H     | 399 km   |

3. Result and Discuss

At this stage, it will explain about determining the best route based on the lowest total Distance in Kilo Meters (KM) in the distribution of snacks production from the bottom of the pakam to Aceh, which will be solved using the Ant Colony Optimization (ACO) algorithm. The calculation of the ACO algorithm can be divided into several stages such as the following:

a. Identification $d_{ij}$

$d_{ij}$ is the distance from node $i$ to node $j$. In this study, variables $d_{ij}$ replaced with $c_{ij}$. The total distance from node $I$ to node $J$. Following data $c_{ij}$ in the form of matrix can be seen in table 3.

Table 3. Matrix total distance between food production distribution nodes

|   | A  | B  | C  | D  | E  | F  | G  | H  |
|---|----|----|----|----|----|----|----|----|
| A | 0  | 40 | 69 | 23 | 0  | 65 | 0  | 0  |
| B | 0  | 0  | 46 | 0  | 0  | 0  | 0  | 0  |
| C | 0  | 0  | 0  | 0  | 0  | 100| 0  | 0  |
| D | 0  | 0  | 0  | 42 | 75 | 0  | 0  | 0  |
| E | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 399|
| F | 0  | 0  | 0  | 0  | 0  | 0  | 76 | 0  |
| G | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 386|
| H | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
Note:
Row nodes and column nodes can be i or j nodes initial parameter initialization for Pheromone at t-time

The parameters $\alpha$, $\beta$, $\rho$ follow predetermined values, while the pheromone is symbolized $(\tau_{ij})$ at the t-time, first given an initial value of 1 which will later experience changes with the pheromone update at the 7th stage. Here are the parameter values:

$(\tau_{ij})(t)=0.01$
$\alpha=0.1$
$\beta=1$
$\rho=0.5$
$m=5$
$n=8$

The parameters $\alpha$, $\beta$ and $\tau_{ij}$ used in stage 4, whereas $\rho$ used in step 7

b. Determine the number of ants
At this stage is determining the number of ants to be placed at each node to distribute snacks. The number of ants is symbolized by (m) and given a value of 8 according to the number of city nodes.

c. Make a tabulation
Ants that have been placed on each node start to distribute snack food production and produce a path or route. The number of tabulation will be proportional to the number of ants, then the tabulation will number 8. Steps to determine the tabulation:

   a. Put ants on node i for example on the 1st tabulation by the 1st ant. At first the ant is placed at node 1, then node i is the same as node 1. Then the ant will go to node j.

   b. Determine node j
Node j is the node between nodes 1 through 8 at the snack distribution node. The way to determine this is based on the greatest probability value, to find out then the formula is used:

   \[ \text{Invers distance} = \eta_{ij} = \frac{1}{c_{ij}}, \text{ and} \]

   \[ \text{Probability} = \frac{k \tau_{ij}(t)^{\alpha} \eta_{ij}^{\beta}}{\sum k \tau_{ij}(t)^{\alpha} \eta_{ij}^{\beta}} \]

$\sum \tau_{ij}(t)^{\alpha} \eta_{ij}^{\beta}$ is the number of values $[\tau_{ij}]^\alpha[\eta_{ij}]^\beta$ from all nodes i to j and k are ants to k it can be seen in table 4 Visibility between points.

|   | A   | B   | C   | D   | E   | F   | G   | H   |
|---|-----|-----|-----|-----|-----|-----|-----|-----|
| A | 0   | 0.025 | 0.014 | 0.043 | 0   | 0.015 | 0   | 0   |
| B | 0   | 0   | 0.021 | 0   | 0   | 0   | 0   | 0   |
| C | 0   | 0   | 0   | 0   | 0.01 | 0   | 0   | 0   |
| D | 0   | 0   | 0   | 0   | 0.023 | 0.013 | 0   | 0   |
| E | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0.002 |
| F | 0   | 0   | 0   | 0   | 0   | 0   | 0.013 | 0   |
| G | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0.002 |
| H | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
Four ants start the journey from point A, three ants visit the point according to the existing route to the point H, and not all points will be missed. At each step the ant will:

- Choose a point visited randomly
- Take notes of points visited in memory

The first point visited by the ant is the departure point, which is point A. The first point will be stored in the memory of each ant, then the ant will visit the next point. The process carried out by ants in the first iteration is: Visibility measure, $\eta (r,)$. Can be calculated by the equation:

$$\sum [\tau (r, u)] [\eta (r, s)] \beta = (0.01 \times 0.025) + (0.01 \times 0.014) + (0.01 \times 0.043) + (0.01 \times 0.015)$$

$$= 0.00025 + 0.00014 + 0.00043 + 0.00015$$

$$= 0.00097$$

a. Visiting the 2nd point

In table 5 below we can see the probability of a second ant visit

| Ant | Nodes | A  | B  | C  | D  | E  | F  | G  | H  | J  | Route  |
|-----|-------|----|----|----|----|----|----|----|----|----|--------|
| 1   | A     | 0  | 0,025 | 0,014 | 0,043 | 0 | 0,015 | 0 | 0 | B  | [a, b] |
| 2   | A     | 0  | 0,025 | 0,014 | 0,043 | 0 | 0,015 | 0 | 0 | C  | [a, c] |
| 3   | A     | 0  | 0,025 | 0,014 | 0,043 | 0 | 0,015 | 0 | 0 | D  | [a, d] |
| 4   | A     | 0  | 0,025 | 0,014 | 0,043 | 0 | 0,015 | 0 | 0 | F  | [a, f] |

The probability of a point is zero if the point already exists in memory or there is no connecting path. In table 5 it can be seen that the ant chooses the point with the greatest probability, but some ants actually choose the point with a small probability, this shows that the ant basically chooses a point randomly.

Information:

pheromone evaporation constant. This value is useful in order to avoid pheromone build up indefinitely considering the number of pheromones will continue to increase each time iteration.

$$(\eta, L_{nn})^1$$ where $L_{nn}$ expressing the distance between points r to s, $\eta$ is the number of nodes

$\tau (A, B) = (1-0.5) \times (0.01) + (0.025) = 0.0255$

$\tau (A, C) = (1-0.5) \times (0.01) + (0.014) = 0.019$

$\tau (A, D) = (1-0.5) \times (0.01) + (0.043) = 0.048$

$\tau (A, F) = (1-0.5) \times (0.01) + (0.015) = 0.02$

Calculate visibility measure $\eta (r, s)$

| Ant 1 | $\sum [\tau (r, u)] [\eta (r, s)] \beta$ = 0.0255 * 0.021 = 0.0005355 |
|---|---|
| Ant 2 | $\sum [\tau (r, u)] [\eta (r, s)] \beta$ = 0.019 * 0.01 = 0.00019 |
| Ant 3 | $\sum [\tau (r, u)] [\eta (r, s)] \beta$ = 0.048 * 0.013 = 0.000624 |
| Ant 4 | $\sum [\tau (r, u)] [\eta (r, s)] \beta$ = 0.02 * 0.002 = 0.00004 |
b. Visiting the 3rd point

In table 6 we can see the probability of a third ant visit

| Ant | Nodes | A     | B       | C | D | E | F | G | H | j | Route  |
|-----|-------|-------|---------|---|---|---|---|---|---|---|--------|
| 1   | B     | 0.025 | 0.000535 | - | - | 0 | - | - | - | B      | [a,b,e] |
| 2   | C     | 0.014 | -       | 0.0019 | - | 0 | - | - | - | C      | [a,c,f] |
| 3   | D     | 0.043 | -       | -    | 0 | - | 0 | - | 0.000624 | D      | [a,d,g] |
| 4   | F     | 0.015 | -       | -    | - | 0 | - | - | 0.0004 | F      | [a,e,h] |

Calculating local updates
\[
\begin{align*}
\tau(B, E) &= (1-0.5) \times (0.01) + 0.021 = 0.026 \\
\tau(C, F) &= (1-0.5) \times (0.01) + 0.01 = 0.015 \\
\tau(D, G) &= (1-0.5) \times (0.01) + 0.013 = 0.018 \\
\tau(E, H) &= (1-0.5) \times (0.01) + 0.002 = 0.007
\end{align*}
\]

Calculate visibility measure \( \eta(r, s) \)

Ant 1
\[
\sum [\tau(r, u)] \cdot [\eta(r, s)] \cdot \beta = 0.026 \times 0.021 = 0.000546
\]

Ant 2
\[
\sum [\tau(r, u)] \cdot [\eta(r, s)] \cdot \beta = 0.015 \times 0.01 = 0.00015
\]

Ant 3
\[
\sum [\tau(r, u)] \cdot [\eta(r, s)] \cdot \beta = 0.018 \times 0.013 = 0.000234
\]

Ant 4
\[
\sum [\tau(r, u)] \cdot [\eta(r, s)] \cdot \beta = 0.007 \times 0.002 = 0.000014
\]

Until this process the ants S1, S2, S3 and S4 make it to the destination point, which is point H, but the ants S2 are more quickly visited because of the closest ants find.

4. Conclusion

From the results of calculations that have been done with the application of the ant colony optimization algorithm, it is found that the ants move faster than ants 1.3 and 4. so that the AH point gets the calculation result of 0.00015, so the distribution of food production is more optimal and the problem of accumulation of goods does not occur due to the route chosen by applying the ant colony optimization algorithm is the best route.

References

[1] Armstrong, C. M., Niinimäki, K., Kujala, S., Karell, E., & Lang, C. (2015). Sustainable product-service systems for clothing: Exploring consumer perceptions of consumption alternatives in Finland. *Journal of Cleaner Production*. https://doi.org/10.1016/j.jclepro.2014.01.046

[2] Dharmendra Sutariya, P. K. (2014). *A SURVEY OF ANT COLONY BASED ROUTING ALGORITHMS FOR MANET*. *European Scientific Journal*.

[3] Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: A new meta-heuristic. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*. https://doi.org/10.1109/CEC.1999.782657
[4] Mangai, S., Tamilarasi, A., & Venkatesh, C. (2008). Dynamic core multicast routing protocol implementation using ANT colony optimization in ad hoc wireless networks. In Proceedings of the 2008 International Conference on Computing, Communication and Networking, ICCCN 2008. https://doi.org/10.1109/ICCCNET.2008.4787711

[5] Sassanelli, C., Terzi, S., & Pinna, C. (2018). A Model to Classify Manufacturing Archetypes for Distributed Production. 2018 IEEE International Conference on Engineering, Technology and Innovation, ICE/ITMC 2018 - Proceedings, 1–9. https://doi.org/10.1109/ICE.2018.8436366

[6] Soltys, M. (2012). Greedy Algorithms. In An Introduction to the Analysis of Algorithms. https://doi.org/10.1142/9789814271424_0002

[7] A. C. Nusantara, E. Purwanti, and S. Soelistiono, “Classification of digital mammogram based on nearest-neighbor method for breast cancer detection,” Int. J. Technol., vol. 7, no. 1, pp. 71–77, 2016, doi: 10.14716/ijtech.v7i1.1393.