Shortest Path Search Simulation on Busway Line using Ant Algorithm

K Pramono, K Wijaya, W Cuosman, D Hartanto, A Dharma, Sumita Wardani
Universitas Prima Indonesia

*wilbertcuosmaaan@gmail.com

Abstract. Busway lane must be planned carefully, so that it can reach a number of points in an area efficiently. Problem that occurs is how to find the shortest route in a busway lane, this problem can be solved by using Ant algorithm. Naturally, ant colony will be able to find the shortest path on their way from nest to food sources by using pheromone. The Ant algorithm procedure to find shortest path in this research is using 100 ants to track all the path in a busway lane, drop pheromone on each path, the shorter a path is, the more pheromone gets accumulated on the path, and pheromone evaporates in each iteration. After going through a number of iterations, the pheromone will accumulate on the shortest path and all ants will follow the shortest path. Application can be used to find the shortest path from a busway lane planning in Medan using Ant algorithm. For a maximum of 30 terminals, ant algorithm requires only about 28 seconds to find the shortest route.

1. Introduction
In this current globalization era, human activity has increased. The activity can be facilitated by good transportation. Land transportation is the most widely used of the existing transportation. One of the land transportation media is by bus. The lane that is passed by bus (busway) must be well planned so that it can reach a number of locations in an area. Problem that occurs is how to determine the shortest route in a busway lane that has been determined in an area. This problem includes the search for the shortest path in the field of Algorithm and Programming. One of the algorithm that can be used to find the shortest route is Ant algorithm.

Ant algorithm is a fairly competitive algorithm as an optimization technique for the shortest route problems with various factors involved, such as cost and path length [1]. Ant algorithms are also used to increase efficiency in information retrieval and reduce overhead costs [2]. The ant algorithm is applied to the search for the shortest path as in [3] and [4]. Several variations of the ant algorithm are also discussed in [5]. Another implementation is the problem of choosing the best short-range combination in transparent optical networks [6]. The ant algorithm is adopted by the behavior of ant colonies known as the ant system. Naturally, an ant colony will be able to find the shortest path on the way from the nest to food sources. Ant colonies can find the shortest route between the nest and the food source based on footprints on the trajectory that has been passed [7]. Ants leave pheromones on the path they pass through and ants can sense pheromones left by other ants to guide them to the shortest path [8]. When ants trace out a path from their nest to a food source, ant drop pheromone on that path, the shorter a path is, the more pheromone gets accumulated on the path [9]. The more ants that pass through a track, the more obvious the footprints will be. This causes the trajectory the ants
traverse in small amounts, the longer the distance, will decrease the density of ants that pass through it, or it will not be passed at all. Conversely, the trajectory of ants in large quantities will increase the density of ants that pass through it or even all ants will pass through the trajectory. Detecting the amount of density is done by using pheromone chemicals released by ants.

The ant algorithm is chosen in this research because this algorithm is able to find the shortest path and always find a solution that is close to optimal for all the shortest distance problems [10]. Research in [11] concludes that ant algorithm is the best solution for the shortest path problem in Ad hoc networks. Finding the shortest path in a network using the Ant algorithm, is the most practical way among similar methods according to research in [12], while research in [13] concludes that Ant algorithm is the currently best known metaheuristic for the all-pairs shortest path problem from a theoretical perspective. Simulation in [14] and [15] shows how the ants dynamically find different solutions based on the distances between the machines and the money level, minimizing always the traveling time [14] and mobile agent can achieve better efficiency and shorter time to complete the task by using ant algorithm [15]. Based on the description above, this research will be made in the form of a thesis with the title "Shortest Path Search Simulation on Busway Line using Ant Algorithm".

2. Research Methodology
The design of this shortest path search simulation software using Ant algorithm on Busway line uses waterfall system development method. Waterfall is a model called classic life cycle, where it implies a systematic and sequential approach to software development.

2.1. Data Collection
Data collection is done using the Library Research method, which is done by collecting reference from electronic journals, such as scientific journals and articles, and sources from print media, such as "Static and Dynamic Shortest Path Finding in Mobile Network by using Ant Colony Optimization" [2], "Ants find the shortest path" [4], "Ant-net: An Adaptive Routing Algorithm" [8], "Secure Route Selection in Manet Using Ant Colony Optimization" [11] and "Using Ant Colony Optimization for Shortest Path Problems" [12].

2.2. Analysis
The analysis will be focused in the process of finding shortest path using Ant algorithm. For example, there is a route consists of 4 (four) points of busway stop, as shown in figure 1 below.

```
Figure 1. Example of A Busway Route
```

Destination of each point can be seen in table 1 below.

| Distance | A  | B   | C   | D   |
|----------|----|-----|-----|-----|
| A        | -  | 196 | 303 | 322 |
| B        | 196| -   | 120 | 172 |
| C        | 303| 120 | -   | 80  |
| D        | 322| 172 | 80  | -   |
Calculate the visibility ($\eta_{ij}$), which can be counted by the inverse of path distance from point i to point j. The distance from A to B is 196, therefore the visibility is $1/196 = 0.0051$, as shown in table 2 below.

| Visibility | A    | B    | C    | D    |
|------------|------|------|------|------|
| A          | -    | 0.0051 | 0.0033 | 0.0031 |
| B          | 0.0051 | -    | 0.0083 | 0.0058 |
| C          | 0.0033 | 0.0083 | -    | 0.0125 |
| D          | 0.0031 | 0.0058 | 0.0125 | -    |

The procedure to find shortest path by ant algorithm is as follows:

- **Number of ants** in this example = 5 (reduced from 100, to simplify the analysis)
  The number of ants used in the research = 100
  The more ants used, the more accurate path searching process will be, but the process will take longer time. Conversely, low amount of ants used will affect the accuracy of the shortest path found.

- **Value of $\alpha$ = 1**
  $\alpha$ variable is the constant to control the pheromone intensity. The higher value of $\alpha$, the more influence of pheromones level on the chance of ants to choose a path.

- **Value of $\beta$ = 1**
  $\beta$ variable is terminal visibility, which is calculated based on the formula $(1 / \text{path distance from point i to point j})$. The higher the $\beta$ value, the more influential of visibility on the chances of ants to choose a path.

- **Value of $\rho$ = 0.1**
  $\rho$ variable is an index of evaporation of pheromones after ants completing all terminal points. The higher the value of $\rho$, the more traces of pheromones are lost in one iteration.

- **All path are given pheromone initial values = 1**, as shown in table 3 below.

| Pheromone | A | B | C | D |
|-----------|---|---|---|---|
| A         | - | 1 | 1 | 1 |
| B         | 1 | - | 1 | 1 |
| C         | 1 | 1 | - | 1 |
| D         | 1 | 1 | 1 | - |

- **For each ant**, select the starting point terminal randomly, as shown in table 4 below.

| Ant | Starting Point |
|-----|----------------|
| A1  | B              |
| A2  | D              |
| A3  | A              |
| A4  | D              |
| A5  | C              |

- **For each ant**, 
  - Look for paths to other terminals randomly using probability in the following equation 1.
\[ p_{ij}^{k}(t) = \frac{\tau_{ij}(t)^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{r \in I_{j}} \tau_{ir}(t)^{\alpha} \cdot \eta_{ir}^{\beta}} \]  

(1)

information:
\begin{align*}
    p_{ij} & = \text{probability / chance to choose a path from point } i \text{ to point } j \\
    k & = \text{ant index} \\
    t & = \text{iteration of } t \\
    \tau_{ij} & = \text{pheromone level on the path from point } i \text{ to point } j \\
    \alpha & = \text{constant to control pheromone intensity} \\
    \eta_{ij} & = \text{visibility = (1 / path distance from point } i \text{ to point } j) \\
    \beta & = \text{control visibility constant} \\
\end{align*}

- Ant A1 calculates probability to choose next point from B,
- Calculate \[ \sum_{r \in J_{i}} \tau_{ir}(t)^{\alpha} \cdot \eta_{ir}^{\beta} = (1)^{1} \cdot (0.0051)^{1} + (1)^{1} \cdot (0.0083)^{1} + (1)^{1} \cdot (0.0058)^{1} \]
- \[ \sum_{r \in J_{i}} \tau_{ir}(t)^{\alpha} \cdot \eta_{ir}^{\beta} = 0.0192 \]
- Calculate chance for each point from B,
  - Point A \[ p = (1)^{1} \cdot (0.0051)^{1} / 0.0192 = 0.2656 \text{ (Cumulative = 0.2656)} \]
  - Point C \[ p = (1)^{1} \cdot (0.0083)^{1} / 0.0192 = 0.4323 \text{ (Cumulative = 0.6979)} \]
  - Point D \[ p = (1)^{1} \cdot (0.0058)^{1} / 0.0192 = 0.3021 \text{ (Cumulative = 1)} \]
- Generate a random number from 0 to 1, for example the random number is 0.7871, so next point to visit is point D (random number < cumulative).
- Do the same process for each ant, and the results of selecting second point can be seen in table 5.

### Table 5. Choosing Next Point

| Ant | Start | Probability Value | Random Number | Next Point | Route |
|-----|-------|-------------------|---------------|------------|-------|
| A1  | B     | 0.2656            | -             | 0.6979     | 1     | 0.7871 | D | B → D |
| A2  | D     | 0.2656            | 0.4159        | 1          | -     | 0.5187 | C | D → C |
| A3  | A     | -                 | 0.4335        | 0.7305     | 1.0001| 0.2393 | B | A → B |
| A4  | D     | 0.2656            | 0.4159        | 1          | -     | 0.1987 | A | D → A |
| A5  | C     | 0.2656            | 0.4813        | -          | 1     | 0.3761 | B | C → B |

- Do the same process for third point and the complete route of each ant can be shown in table 6.

### Table 6. First Route Completion for Each Ant

| Ant | Start | Probability Value | Random Number | Next Point | Route |
|-----|-------|-------------------|---------------|------------|-------|
| A1  | B → D | C                 | 1             | -          | -     | 0.1987 | A | B → D → C → A |
| A2  | D → C | B                 | 1             | -          | -     | 0.7197 | A | D → C → B → A |
| A3  | A → B | D                 | -             | 1          | -     | 0.9164 | C | A → B → D → C |
| A4  | D → A | B                 | -             | 1          | -     | 0.0761 | C | D → A → B → C |
| A5  | C → B | A                 | -             | -          | 1     | 0.3980 | D | C → B → A → D |

- When all ants have visited all the terminal points, exit loop and continue next step below.
Pheromone Adding

When ants return from food source to nest, there is an addition in the pheromones level by ant, using the formula in equation 2 below.

\[
\tau_{ij}(t) = \tau_{ij}(t) + \Delta \tau_{ij}(t)
\]

(2)

information:
- \(\tau_{ij}\) = pheromone level on the path from point \(i\) to point \(j\)
- \(t\) = iteration of \(t\)
- \(\Delta \tau_{ij}\) = number of pheromones added to the path from point \(i\) to point \(j\)

The number of pheromones additions (\(\Delta \tau_{ij}\)) can be calculated by equation 3.

\[
\Delta \tau_{ij}(t) = \left(\frac{1}{\text{Length}(t)}\right)
\]

(3)

information:
- \(\Delta \tau_{ij}\) = number of pheromones added to the path from point \(i\) to point \(j\)
- \(t\) = iteration of \(t\)
- \(\text{Length}(t)\) = the length of the path ant passes when returning to nest

○ Total length of each ant and level of pheromone added can be seen in table 7.

| Ant | Route | Total Length | Pheromone Added (\(\Delta \tau_{ij}\)) = \(1/\text{Length}\) |
|-----|-------|--------------|--------------------------------------------------|
| A1  | B → D → C → A | 751         | 0.0013                                           |
| A2  | D → C → B → A | 718         | 0.0014                                           |
| A3  | A → B → D → C | 751         | 0.0013                                           |
| A4  | D → A → B → C | 718         | 0.0014                                           |
| A5  | C → B → A → D | 718         | 0.0014                                           |

○ Update pheromone level as follows,

Ant-1 (B → D → C → A) updates pheromone on each road:
- Route B → D:
  \((B, D) = (D, B) = 1 + 0.0013 = 1.0013\)
- Route D → C:
  \((D, C) = (C, D) = 1 + 0.0013 = 1.0013\)
- Route C → A:
  \((C, A) = (A, C) = 1 + 0.0013 = 1.0013\)
- Route A → B:
  \((A, B) = (B, A) = 1 + 0.0013 = 1.0013\)

Repeat pheromone update until Ant-5

○ Pheromone level on each road after added by ants, can be seen in table 8 below.

| Pheromone | A     | B     | C     | D     |
|-----------|-------|-------|-------|-------|
| A         | 1.0068| -     | 1.0026| 1.0042|
| B         | 1.0068| -     | 1.0042| 1.0026|
| C         | 1.0026| 1.0042| -     | 1.0068|
| D         | 1.0042| 1.0026| 1.0068| -     |
Pheromone Evaporation
After an iteration ends, reduce the value of the pheromone by using the formula in equation 4.

\[ \tau_{ij}(t) = (1 - \rho)\tau_{ij}(t) \]  

**information:**

- \( \tau_{ij} \) = pheromone level on the path from point \( i \) to point \( j \)
- \( t \) = iteration of \( t \)
- \( \rho \) = pheromone evaporation index, determined at 0.1

- As evaporation index \( \rho \) determined at 0.1, so the evaporation will decrease pheromone by 10% as follows:
  - \((A, B) = (1-0.1)*1.0068 = 0.9061\)
  - \((A, C) = (1-0.1)*1.0026 = 0.9023\)
- Repeat the evaporation on each path until
  - \((D, C) = (1-0.1)*1.0068 = 0.9061\)

- Pheromone on each road after evaporation, can be seen in table 9 below.

| Pheromone | A    | B    | C    | D    |
|-----------|------|------|------|------|
| A         | -    | 0.9061 | 0.9023 | 0.9038 |
| B         | 0.9061 | -    | 0.9038 | 0.9023 |
| C         | 0.9023 | 0.9038 | -    | 0.9061 |
| D         | 0.9038 | 0.9023 | 0.9061 | -    |

- Repeat until all ants follow the same path.
- Take the shortest path by taking a route that has the most pheromone value from each point. If we take table 9 as solution, we can see that point A has the largest pheromone value to point B (A→B), point B has the the largest pheromone value to point C (A→B→C), point C has the the largest pheromone value to point D (A→B→C→D), so the shortest path will be A→B→C→D→A.

2.3. System Design
Software design is done by prototyping or design the main display of software. Form display in the software is described simply by using Microsoft Visio software. System design also includes system modeling using Unified Modeling Language.

2.4. System Development
Software coding is done using Microsoft Visual Basic.Net 2012 programming language. The components used in making this software are standard components of VB.Net, without any additional components from outside.

2.5. Testing
Testing is done to fix any errors that might occur in the system during the system development process. Errors or bugs from software can be reduced as much as possible from this testing. Software testing is done by testing the shortest path searching process at several input bus terminal stops.

3. Result and Discussion
3.1. Result
The following will discuss the implementation of the shortest path search simulation application using the Ant algorithm. Display of simulation form in running mode can be seen in figure 2. Display of Simulation form to search for the shortest path. To determine the bus terminal points, click on the map.
using mouse. For example, there are 11 (eleven) bus terminal points randomly assigned to the Simulation form, as shown in figure 2 below.

As shown in figure 2, simulation form has a toolbar at the top, which consists of several buttons as follows:

- "Start Search" button, functions to start the shortest path searching using Ant algorithm.
- "Stop Search" button, functions to stop the application searching for shortest path.
- "Delete All Terminals" button, functions to delete all bus terminal points that have been entered in the Simulation form through the mouse click event.
- "Show Distance" button, functions to display (or not display) the coordinates of terminals and distances between terminal points calculated using Euclidean distance.
- "Show Map" button, functions to show or hide maps of Medan city in the application.
- "About" button, functions to display About form that contains the thesis researcher and brief information about the application.
- "Exit" button, functions to close the application.

The shortest path found by Ant algorithm is illustrated in the form of a red lines, as shown in the following figure 3.
Results of application testing can be seen in table 10 below.

| No | Points / Terminals | Run Time (seconds) | Shortest Path |
|----|--------------------|--------------------|---------------|
| 1  | 5                  | 3.54               | 1287.99       |
| 2  | 10                 | 5.67               | 1847.51       |
| 3  | 15                 | 6.72               | 2302.56       |
| 4  | 20                 | 10.03              | 2839.1        |
| 5  | 25                 | 22                 | 3532.62       |
| 6  | 30                 | 28.53              | 4002.97       |

3.2. Discussion
Based on the application testing, it can be seen that Ant algorithm is able to find the shortest route from a given number of terminal position. For a maximum of 30 terminals, Ant algorithm requires only about 28 seconds to find the shortest route. This is certainly much faster to search for shortest path, than when using a brute force attack scheme. The execution time is proportional to the input number of points. The more points or terminals, the longer it takes for Ant algorithm to find the shortest route.

4. Conclusions and Suggestions
4.1. Conclusions
After completing the design of shortest path search simulation application using Ant algorithm on the Busway line, we can conclude that application can be used to find the shortest path from a busway lane planning in Medan using Ant algorithm for a maximum 30 terminals with searching run time around 28.53 seconds, depending on the terminal points and computer specification used to run the application.

4.2. Suggestions
Some suggestions that can be given for further development of this application are as follows:
- This simulation application can be developed by integrating the application with Google Maps, so that the application can take coordinates directly from Google Maps and calculate the distance between bus terminals based on distance from Google Maps.
- Application can be developed by adding several Ant algorithm variations, such as: Rank Based Ant System (ASrank) or Max-Min Ant System (MMAS), so that the strengths and the weakness of each algorithms can be compared.

References
[1] ANGUS, D., *Solving a unique Shortest Path problem using Ant Colony Optimisation*, IEEE, 2012.
[2] BORKAR, and KUMAR, *Static and Dynamic Shortest Path Finding in Mobile Network by using Ant Colony Optimization*, S.R.P.C.E. Nagpur, India, 2017.
[3] GLABOWSKI, et. al. *Shortest Path Problem Solving Based on Ant Colony Optimization Metaheuristic*, Poznan University of Technology, Poland, 2014.
[4] JAYADEVA, et.al, *Ants Find the Shortest Path: A Mathematical Proof*, Indian Institute of Technology Delhi, India, 2013.
[5] NAYAK, N. et.al., *Comparison of different Ant based techniques for identification of shortest path in Distributed Network*, C.S.E, T.I.T/ R.G.P.V., Bhopal, India, 2013.
[6] BRASILEIRO, I. et al, *Ant Colony Optimization Applied to the Problem of Choosing the Best Combination among M Combinations of Shortest paths in Transparent Optical Networks*, Federal University of Piauí, Terezina, Piauí, Brazil, 2016.
[7] DORIGO, M. and GAMBARDELLA, L.M. *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*, Université Libre de Bruxelles, Belgium,
2008.

[8] CHANDANA, M. and THAKUR, S., *Ant-net: An Adaptive Routing Algorithm*, Amity University, India, 2013.

[9] GURAV, A.A. and NENE, M.J. *Optimal Path Identification using Ant Colony Optimization in Wireless Sensor Network*, Department of Applied Mathematics, Defence Institute of Advance Technology, Pune, India, 2013.

[10] DOERR, B. *Ants Easily Solve Stochastic Shortest Path Problems*, Max Planck Institute for Informatics Saarbrücken, Germany, 2012.

[11] MAHESHWARI, S. and BHARDWAJ, M, *Secure Route Selection in Manet Using Ant Colony Optimization*, SRM University, NCR Campus, Modinagar, Ghaziabad, India, 2015.

[12] SAEEDI, S. and MAHDAVI, I, *Using Ant Colony Optimization for Shortest Path Problem*, Mazandaran University of Science and Technology, Babol, Iran, 2016.

[13] HOROBA, C. and SUDHOLT, D. *Running Time Analysis of ACO Systems for Shortest Path Problems*, Technische Universit¨at Dortmund, Germany, 2014.

[14] SILVA, C.A. and RUNKLER, T.A, *Ant Colony Optimization for dynamic Traveling Salesman Problems*, Siemens AG, Munich, 2010.

[15] ZHISUI, Z. *Ant Colony Algorithm Based on Path Planning for Mobile Agent Migration*, Department of Mathematics and Computer Science Tongling University, Tongling, China, 2011.