BAREKENG: Jurnal Ilmu Matematika dan Terapan
June 2022 Volume 16 Issue 2 Page 651–662
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol16iss2pp651–662

OPTIMIZING THE PROCESS OF PICK-UP AND DELIVERY WITH TIME WINDOWS USING ANT COLONY AND TABU SEARCH ALGORITHMS

Imas Saumi Amalia [1*], Toni Bakhtiar[2], Jaharuddin[3]

[1,2,3]*Department of Mathematics, Faculty of Mathematics and Natural Science, IPB University*
Meranti St., Kampus IPB Dramaga, Bogor 16680, Indonesia

Corresponding author's e-mail: [1] imassaumi@apps.ipb.ac.id*

Abstract. *The provision of goods shuttle services sometimes faces several constraints, such as the limitation on the number of vehicles, vehicle capacity, and service time, or the vehicle used has single transport access. To avoid losses, a strategy is needed in determining the optimal route and policy for arranging goods in the vehicle especially if there are two types of goods involved. Traveling Salesman Problem and Pick-up and Delivery with Handling Costs and Time Windows (TSPPDHTW) is a model of an optimization problem that aims to minimize the total travel and goods handling costs in the goods pick-up and delivery with the constraints previously mentioned. Solving that model using the exact method requires a very long computation time so it's not effective to be implemented in real-life. This study aims to develop a (meta)heuristic based on Ant Colony Optimization (ACO) and Tabu Search (TS) to be ACOTS to solve TSPPDHTW with reasonable computation time. The development is carried out by adding functions of clustering, evaluating constraints, cutting tours, arranging of goods, and evaluating moves on the TS, as well as modifying transition rules. The result has a deviation of about 22% and 99.99% less computational time than the exact method.*

Keywords: *ACOTS, handling costs, pick-up delivery, travel costs, TSPPDHTW*

Article info:

Submitted: 7th March 2022 *Accepted: 4th May 2022*

How to cite this article:
I. S. Amalia, T. Bakhtiar and Jaharuddin, "OPTIMIZING THE PROCESS OF PICK-UP AND DELIVERY WITH TIME WINDOWS USING ANT COLONY AND TABU SEARCH ALGORITHMS", *BAREKENG: J. Il. Mat. & Ter.*, vol. 16, iss. 2, pp. 651-662, June, 2022.

1. INTRODUCTION

The provision of goods shuttle services is an alternative for companies to improve their service for customers. Although providing pick-up and delivery services can help companies to expand their market, in practice, the companies often face various constraints, such as time limits for each customer (time windows), limitations on vehicle capacity, and limitations on the type of vehicle used. If there is more than one type of good involved in the process, it is important to pay attention to the arrangement of goods in vehicles that have single transportation access because it can affect the time of the process of unloading goods during service at the customer's location. Single transport access for unloading and loading goods on vehicles may cause of additional time to handle goods that are blocked by other goods when being unloaded. Therefore, so that the delivery process can be carried out effectively, in addition to determining the best route, a strategy for preparing goods in the vehicle is needed so that is can minimize the total travel and handling time. In [1] this can be categorized as solving the Traveling Salesman Problem and Pick-up and Delivery with Handling Costs and Time Windows (TSPPDHTW).

The TSPPDHTW case has been modeled by [1] and solved using the exact method. However, the search for solutions using exact methods has not succeeded in showing the efficiency model to be applied in real life. The exact method for this problem requires a very long computation time and increases exponentially, so that it cannot be used for big data. The TSPPDHTW or generally TSP is one of the classical NP-hard problems in combinatorial optimization, and it needs faster algorithms to solve so that it can be applied in the real-life [2]. In this study, the case of TSPPDHTW will be solved using the metaheuristic method. Metaheuristics are one method that can produce near-optimal results in a computationally efficient manner. This method is extensively used in a combination of two or more metaheuristics called hybrid algorithms [3].

Ant Colony Optimization (ACO) and Tabu Search (TS) are metaheuristics that are quite often combined by researchers to solve combinatorial optimization problems. ACO is an algorithm that is inspired by the natural ability of ants to communicate with their fellow colonies by giving pheromones to the paths they travel when they find food [4]. While TS is an algorithm that performs searches starting from the initial solution and then explores the solution space through a move by paying attention to search history [5]. De La Cruz *et al*. [6] used ACO and TS for *Heterogeneous Vehicle Routing Problems with Time Windows and Multiple Products* (HVRPTWMP). Hanif [7] combined TS with ACO to solve the TSP case and succeeded with better efficiency than TS itself or the Genetic Algorithm. Dewantoro *et al*. [8] have also proven that the use of the hybrid of TS and ACO algorithms provides optimal solutions more effectively in terms of time and quantity. In a recent study, Diallo *et al*. [9] used ACO and TS to solve *Virtual Network Embedding* (VNE). In this paper, ACO and TS were used to solve the TSPPDHTW problem that had been modeled in previous studies.

2. RESEARCH METHODS

This study uses the model and data in [1]. There is a bicycle shuttle case involving 14 bicycle shops in Bogor as a sample of customers and 1 as a depot. In this case, the goods sent or goods of type I are new bicycles or bicycles that have been repaired, while the goods of type II, or goods that are picked up, are damaged bicycles that will be repaired. Travel time data between stores was obtained using the help of Google maps, which was taken in 2020. The data on the number of requests and time windows at each store used hypothetical data. The method used is a metaheuristic based on ACO and TS, hereinafter referred to as ACOTS. The development of ACOTS and obtaining the solution was carried out with the help of OCTAVE software with device specifications A-9 memory AMD Radeon VGA 4GB.

2.1 *TSPPDHTW*

TSPPDHTW is the model of the pickup and delivery of goods that has to meet several constraints. here are limitations on vehicle capacity, on transport access of the vehicle, and on the time customers can be served, which is known as time windows. Some of the assumptions used in modeling this problem are:

1) The vehicle route starts from the depot and returns to the depot.
2) Each customer is only visited exactly once.

3) The number of vehicles used is one, and it has single transportation access.
4) The vehicle has a limited capacity of Q, which is represented as a collection of partitions in the form of space with a capacity of one unit of goods and has a position number of 1,2,3,...,Q in each partition. Each position can only be occupied by one item. Position number 1 is the position at the rear of the vehicle, or the position closest to transportation access, which means that goods in this position will receive first handling.
5) There are two types of goods involved in the process, namely goods that will be delivered to the customer (goods type I) and goods that will be picked up from the customer (goods type II).
6) The length of time for unloading or loading each unit of goods from or into the vehicle is the same.

The purpose of this model is to minimize the total time required to carry out the process of pickup and delivery for all customers, which consists of the total travel time and handling time. Handling time is defined as the time required to unload goods that are blocking the process of unloading other goods or the time required to rearrange goods for easy unloading at the customer's next location. The following is the model of TSPPDHTW, which is written mathematically.

Sets:

V	:	the set of customer that will be visited including the depot,
V_c	:	the set of customer that will be visited,
E	:	the set of side or route from customer i to customer j, $\forall i,j \in V, i \neq j$.

Indices:

i,j	:	the index for the depot (i = 1) and customers $(1,2,3,\dots,\lvert V\rvert)$,
i,j	:	the index for the customers $(2,3,4,\dots,\lvert V_c\rvert)$,
k	:	the index for the position of the goods in the vehicle $(1,2,3,\dots,Q)$,

Parameters:

c_{ij}	:	the travel time from customer location i to customer location j,
α_i	:	the number of goods that must be delivered to customer i, or the demand of goods type I by customer i
β_i	:	the number of goods that must be picked up from customer i or the demand of goods type II by customer i
Q	:	the vehicle capacity
h_a	:	the time it takes to unload each type I item from the vehicle,
h_b	:	the time required to load each type II item into the vehicle,
l_i	:	the earliest time service to customer i can be done,
u_i	:	the latest time for customer service i may be done.

Decision variables:

$$x_{ij} = \begin{cases} 1, \text{if customer } j \text{ is visited after customer } i \\ \quad 0, \text{others.} \end{cases}$$

$$a_{ij}^{k} = \begin{cases} 1, \text{if position } k \text{ is occupied by goods of type I when the} \\ \quad \text{vehicle passes through the route } (i,j) \\ 0, \text{others.} \end{cases}$$

$$b_{ij}^{k} = \begin{cases} 1, \text{if position } k \text{ is occupied by goods of type II when the} \\ \quad \text{vehicle passes through the route } (i,j) \\ 0, \text{others.} \end{cases}$$

$$r_{i}^{k} = \begin{cases} 1, \text{if position } k \text{ is involved in handling process} \\ \quad \text{when the vehicle is at the costumer } i'\text{s location} \\ 0, \text{others.} \end{cases}$$

v_i^k	:	length of handling time for position k at customer i location,
s_i	:	the time when the service of customer i is started.

Objective Function

The objective function of this problem is to minimize the total time of travel and handling required to transport goods to all customers, written mathematically as follows.

$$\min Z = \sum_{(i,j)\in E, i\neq j} c_{ij}x_{ij} + 2\left[\sum_{i\in V_c}\sum_{k\in Q} v_i^k - \sum_{i\in V_c} h_\alpha \alpha_i\right] \tag{1}$$

Constraints
The following constraints must be met.

$$\sum_{j\in V, j\neq i} x_{ij} = 1, i \in V. \tag{2}$$

$$\sum_{i\in V, j\neq i} x_{ij} = 1, j \in V. \tag{3}$$

$$\sum_{j\in V}\sum_{k=1}^{Q}(a_{ji}^{k} - a_{ij}^{k}) = \alpha_i, i \in V. \tag{4}$$

$$\sum_{j\in V}\sum_{k=1}^{Q}(b_{ij}^{k} - b_{ji}^{k}) = \beta_i, i \in V. \tag{5}$$

$$r_i^k \leq r_i^{k-1}, i \in V_c, k \in \{2, \dots, Q\}. \tag{6}$$

$$a_{ij}^k + b_{ij}^k \leq x_{ij}, (i,j) \in E,\ k \in \{1, \dots, Q\}. \tag{7}$$

$$a_{ij}^{k-1} + b_{ij}^{k-1} \leq a_{ij}^k + b_{ij}^k, (i,j) \in E,\ k \in \{1, \dots, Q\} \tag{8}$$

$$\sum_{j\in V, j\neq i} a_{ij}^k - \sum_{j\in V, j\neq i} a_{ji}^k \leq r_i^k, i \in V_c, k \in \{1, \dots, Q\}. \tag{9}$$

$$\sum_{j\in V, j\neq i} a_{ji}^k - \sum_{j\in V, j\neq i} a_{ij}^k \leq r_i^k, i \in V_c, k \in \{1, \dots, Q\}. \tag{10}$$

$$\sum_{j\in V, j\neq i} b_{ij}^k - \sum_{j\in V, j\neq i} b_{ji}^k \leq r_i^k, i \in V_c, k \in \{1, \dots, Q\}. \tag{11}$$

$$\sum_{j\in V, j\neq i} b_{ji}^k - \sum_{j\in V, j\neq i} b_{ij}^k \leq r_i^k, i \in V_c, k \in \{1, \dots, Q\}. \tag{12}$$

$$v_i^k = \textstyle\sum_{j\in V, j\neq i}(h_a a_{ji}^k + h_b b_{ji}^k) r_i^k, i \in V_c,\ i \neq j, k \in \{1, \dots, Q\}. \tag{13}$$

$$l_i \leq s_i \leq u_i, i \in V_c. \tag{14}$$

$$s_i + \textstyle\sum_{k=1}^{Q} v_i^k \leq u_i, i \in V_c. \tag{15}$$

$$s_i \geq l_i, i \in V_c. \tag{16}$$

$$s_i + 2\textstyle\sum_{k=1}^{Q} v_i^k + h_b\beta_i - h_a\alpha_i + c_{ij} - M\big(1 - x_{ij}\big) \leq s_j, \quad i \in V_c, j \in V_c,\ i \neq j, \tag{17}$$

$$u_i \geq 1 + (|V| - 2)x_{i1}, \forall i \in V, i > 1. \tag{18}$$

$$u_j \geq |V| - 1 - (|V| - 2)x_{1j}, \forall j \in V, j > 1. \tag{19}$$

$$u_j \geq u_i + x_{ij} - (|V| - 2)\big(1 - x_{ij}\big) + (|V| - 3)x_{ji}, \quad \forall i, j \in V, j \neq 1. \tag{20}$$

$$x_{ij}, a_{ij}^k, b_{ij}^k \in \{0,1\}, k \in \{1, \dots, Q\}. \tag{21}$$

$$r_i^k \in \{0,1\}, i \in V_c, k \in \{1, \dots, Q\}. \tag{22}$$

$$v_i^k \geq 0, i \in V_c, k \in \{1, \dots, Q\}. \tag{23}$$

$$s_i \geq 0, i \in V_c, k \in \{1, \dots, Q\}. \tag{24}$$

Equations (2) and (3) guarantee that each customer only visits once. Equations (4) and (5) are the flow of goods constraints. Equation (6) – (8) guarantee regularity in the position of goods in the vehicle. Equations (9) – (12) guarantee that any goods that are not involved in handling will not experience a change in position. Equation (13) calculates the handling time at each customer location at position. Equations (14) – (17) guarantee that the time window constraint is met. Equation (18) – (20) are the sub-tour elimination constraints that guarantee the formation of a complete route. Equations (21) through (24) state the characteristics of each of the variables involved [1].

2.1 *Ant Colony Optimization*

Artificial ants or ant agents at ACO will find solutions to the problems based on stochastic procedures according to pheromone concentration and heuristic information. k^{th} ant will choose the next node from the set of nodes that hasn't been chosen yet at time t by calculating states transition probability, $p_{ij}^k(t)$ [10].

$$p_{ij}^k(t) = \begin{cases} \dfrac{\left(\tau_{ij}(t)\right)^{\alpha} \cdot \left(\eta_{ij}(t)\right)^{\beta}}{\sum_{j \notin tabu_\epsilon}\left(\left(\tau_{ij}(t)\right)^{\alpha} \cdot \left(\eta_{ij}(t)\right)^{\beta}\right)}, & \text{if the } k \ ant \text{ is allowed to road on (i, j)} \\ 0 \quad , & \text{otherwise} \end{cases} \tag{25}$$

$\tau_{ij}(t\ )$ denotes the amount of pheromones at (i, j) at time t, η_{ij} denotes the inverse of edge weight $(i, j), \alpha$ is the heuristic factor, β is the visibility heuristic factor, and $tabu_\epsilon$ is the tabu list that records the nodes that k^{th} ant has visited. The ants will use the Roulette Wheel Selection (RW) algorithm by considering that transition probability value to choose the next step in building the route [11]. After all the ants have successfully established the route, the pheromone value is updated using the following equation [12].

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k(t) \tag{26}$$

$\tau_{ij}(t)$ denotes the amount of pheromone on the road (i, j) at t time t, ρ shows the rate of evaporation of pheromone, and m is the number of ants. Each ant gives the amount of pheromone on the road which has been passed at time t following this equation.

$$\Delta\tau_{ij}^k(t) = \begin{cases} \dfrac{Q}{L^k}, & \text{if } \ k^{th} \text{ant roads on } \ (i,j) \\ 0 & \text{, otherwise} \end{cases} \tag{27}$$

While Q shows the concentration of the enhancement of pheromone, in [13] it equals 1.

2.2 *Tabu Search*

Tabu Search (TS) was first introduced by Glover and McMillan in 1986 [14]. TS works on the principle of local search. In each iteration, TS will look for a neighborhood solution to find the best solution. TS uses memory to prevent getting stuck or cycling during the solution search process. The neighborhood of a solution is the set of all solutions resulting from a move. The move is a method used by TS to move from one solution to another based on certain rules. Some types of moves that can be carried out in TS are:

(1) Insert vertex forward: takes a certain node from the route that has been formed and then inserts it in front of other nodes.
(2) Insert vertex backward: takes a certain node from the route that has been formed and then inserts it behind another node.
(3) Exchange the vertex: swap the position of the node to i and node to j.
(4) 2-opt: rearrangement between nodes i and j by reversing the order of their positions

A move is considered tabu if it is selected to be applied to the current solution. Moves that are members of the tabu list are maintained for a number of iterations (tenure). For local search-based metaheuristics such as TS, the initial solution plays an important role in determining the quality of the solution [15–17]. A simple heuristic algorithm can be used to generate an initial solution for TS. Then, in each iteration, the neighborhood solution that gives the best solution is chosen. This step was repeated until the termination criteria were met [18].

3. RESULTS AND DISCUSSION

The ACOTS algorithm was developed based on the main principles of two algorithms, namely ACO and TS. In this case, ACO and TS synergize to find a better approximation solution than when each of these algorithms worked alone. ACO will provide a good initial solution for TS, while TS will conduct deeper exploration to find a better solution than the solution obtained by ACO.

TSPPDHTW is a problem that has several constraints, such as limitations on vehicle capacity and transportation access, and time windows of customers. In addition to considering the travel time in determining the route, TSPPDHTW also considers the policy of arranging goods in the vehicle. The goods arrangement policy will affect the service time, which is limited by the customer's time window. Thus, the complexity of the problem increases so that it is necessary to modify the ACO and TS algorithms to properly solve the case of TSPPDHTW.

3.1. *Modification of ACO and TS to Resolve the TSPPDHTW*

The main working principle of ACO includes determining the next node by calculating the transition probability and generating random values, as well as updating pheromone values in the ant trails. Modifications are made by adding clustering functions, evaluating constraints, cutting tours, and arranging goods in the vehicle. The following is an explanation of the performance of each function on the ACO.

3.1.1 *Clustering*

This function aims to group all customers who will be visited based on the proximity of their time windows into several clusters (only one cluster may be formed). One customer with another is said to have close time windows if the time windows of the two customers coincide. Thus, each cluster will have several customers, each of which has time windows that intersect with each other. This function can avoid the formation of infeasible tours due to errors in prioritizing customers based on the time window. Thus, the feasible solution will be easier to find.

3.1.2. *Evaluation Constraints*

The constraints that must be met in the case of TSPPDHTW are vehicle capacity and the time window of customers. When ACO wants to choose the next customer, each customer who has not been selected will be evaluated by considering the number of requests for goods of type I (d_a_j) and II (d_b_j). A customer meets the capacity constraint if the number of requests for type II goods (goods to be picked up or loaded into the vehicle) from the customer does not exceed the space in the vehicle when the vehicle arrives at its location. This condition can be represented as the following equation.

$$total\_demand\_2 + d\_b_j \leq total\_demand\_1 + d\_a_j \tag{28}$$

The $total\_demannd\_1$ represents the accumulated number of type I goods that have been unloaded from the vehicle (have been delivered) before customer j is visited, while $total\_demand\_2$ represents the accumulated number of type II goods loaded into the vehicle before customer j is visited.

Customers who meet the capacity constraints will be evaluated for the second time based on their time windows. A customer meets the time window constraint if the vehicle arrives before the upper time of the customer’s time window and the remaining time is still sufficient to serve the customer. This condition can be represented as the following equation

$$s_i + st_i + ht_i + c_{ij} + st_j + ht_j \leq u_j \tag{29}$$

The s variable represents the start time of service, st represents the length of service time, ht represents the additional time required for handling, u represents the upper limit of the time windows of the customer, and c_{ij} represents the travel time from i to j where i is the last customer visited and j as a candidate customers to be visited next. Customers who meet both constraints will be calculated to have the transition probability so that it can be determined which customers will be visited next.

3.1.3. *Next Customer Determination*

After calculating the cumulative probability value of customers who meet the constraints and generating random values, the customer who will be selected as the next customer is the first customer who has a cumulative probability value greater than the random value (Roulette Wheel Selection Algorithm). To solve the case of TSPPSDHTW, which has time window constraints, to facilitate the search for a feasible route, this step is modified so that the customer who will be selected as the next customer is the customer who has a cumulative probability value that is greater than the random value and the smallest initial time window limit. This aims to avoid the customer whose large initial time window limit is chosen early so that the other customers can not be visited because the time has passed the other time window deadline (can not be served).

3.1.4. *Cutting Tour*

Cutting Tour is a function that helps ants find a route, especially when the ants are stuck at a node. An ant is trapped at a certain node if at that node the ant has no customers that can be visited next because there are no customers who meet the two constraints so that the route formation cannot be continued. When that happens, the Cutting Tour function will cut the tour that has been formed. Each cut causes the ant to have to take one step back to the previous node or customer, so that the customer where the ant was trapped returns to the set of customers that have not been selected. This will help the ants in finding a feasible route so that all customers can be visited.

3.1.5 *Item arrangement*

Another thing that is important in the TSPPDHTW case is the policy on the arrangement of goods in the vehicle. The right policy will increase the effectiveness of customer service. The policy of arranging goods used in this study is to arrange type I goods of the next customer in the rearmost position of the vehicle, followed by type II goods that have been taken, then type I goods that have not been delivered are in the front position. So that, when the vehicle arrives at the customer's location, a number of type I goods can be immediately unloaded, so there is no additional handling time needed because there are no items that are blocking the handling operation. This policy only requires the additional handling time for moving the type I goods of the next customer to the rearmost position of the vehicle.

3.1.6 *Local Search of TS*

The main working principle of TS is seeking a better solution based on an initial solution by doing the move. There are three types of moves carried out in this study, namely swap, reversion, and insertion, which are the result of modification move in [16]. These moves are described in Figure 1.

1	**2**	3	4	**5**
1	**5**	3	4	**2**

(a)

1	**2**	3	4	**5**
1	**5**	**4**	**3**	**2**

(b)

1	**2**	3	4	**5**
1	**3**	4	5	**2**

(c)

1	**2**	3	4	**5**
1	**5**	**2**	3	4

(d)

Figure 1. Swap (a), reversion (b), insert backward (c), and insert forward (d)

In general, TS will directly choose the move that results in a better solution. In solving the TSPPSDHTW case, TS was modified by adding a function that can evaluate the move to be made based on capacity constraints and time windows. The route formed after the move will be evaluated to see if the route is still feasible or not. In the end, this function will determine if a move is allowed on the route that was formed at that time.

The flowchart of the ACOTS Algorithm can be seen in Figure 2. The chart marked with the red box shows the processes added as a result of the modifications made to the basic algorithm.

3.2. *Simulation on ACO Parameters*

ACO has several parameters that affect ACO's ability to find optimal solutions and its convergence speed [10, 13]. Determination of the value of this parameter depends on the problem at hand. Therefore, before the ACOTS Algorithm is applied to solve the TSPPDHTW case, a parameter simulation is carried out first to find out the suitable parameters which are expected to provide a good solution. The ACO parameters that are simulated in this study include $\alpha, \beta\ m$, and the number of iterations. In each scenario, the simulation was carried out 10 times. The simulation was conducted by observing the effect of changes in parameter values on the value of the objective function, which is the sum of total travel costs (times) and handling costs (times), as well as the influence on the computing time required.

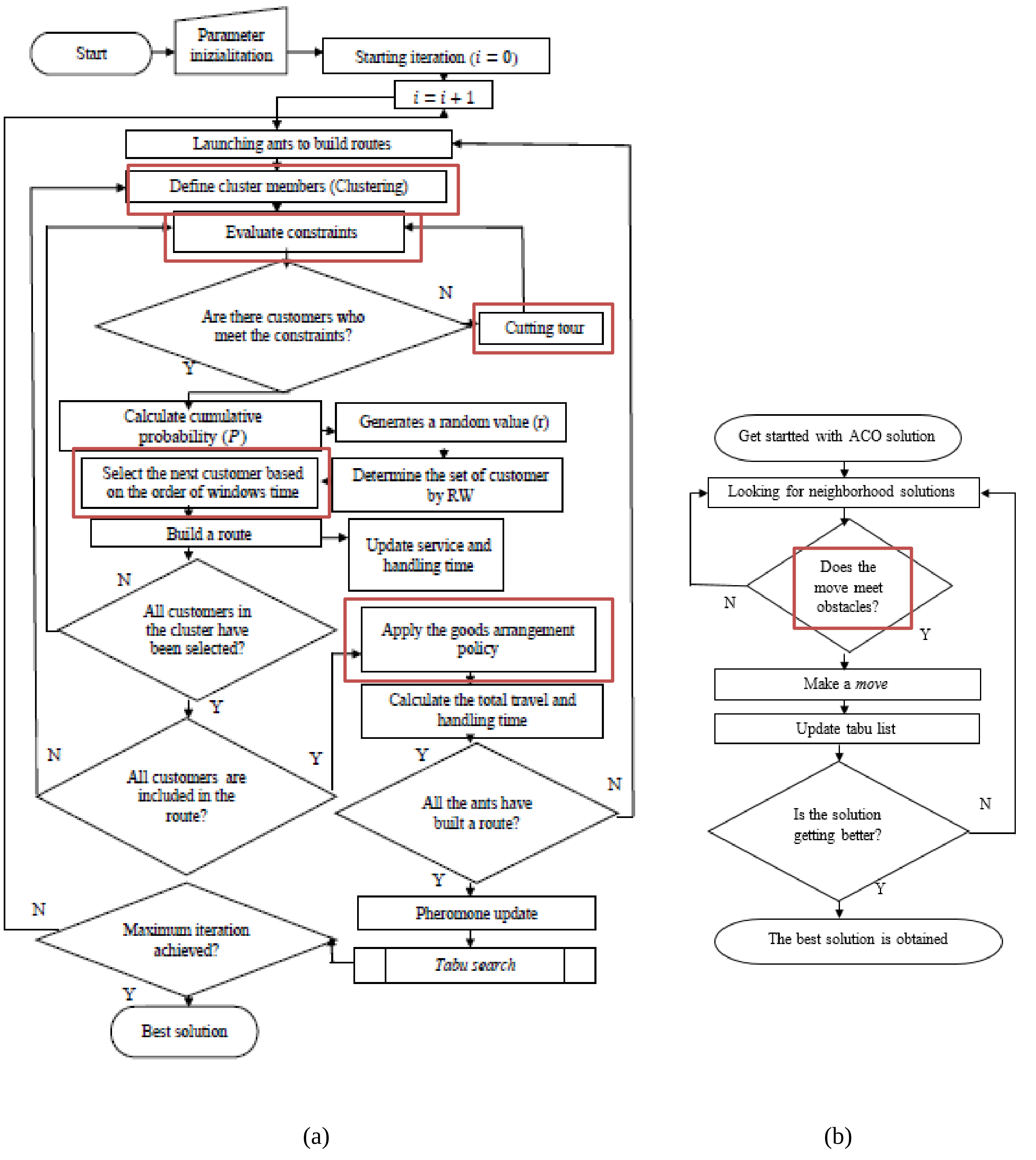
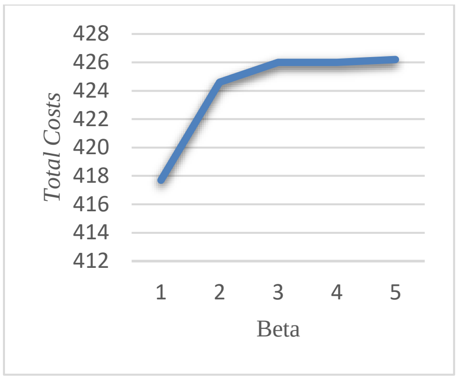
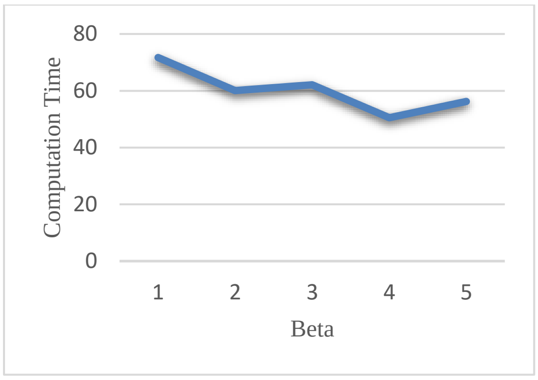

Start
Parameter inizialitation
Starting iteration ($i = 0$)
$i = i + 1$
Launching ants to build routes
Define cluster members (Clustering)
Evaluate constraints
Are there customers who meet the constraints?
N
Y
Cutting tour
Calculate cumulative probability (P)
Generates a random value (r)
Select the next customer based on the order of windows time
Determine the set of customer by RW
Build a route
Update service and handling time
All customers in the cluster have been selected?
Apply the goods arrangement policy
Calculate the total travel and handling time
All customers are included in the route?
All the ants have built a route?
Pheromone update
Maximum iteration achieved?
Tabu search
Best solution
(a)
Get startted with ACO solution
Looking for neighborhood solutions
Does the move meet obstacles?
Make a move
Update tabu list
Is the solution getting better?
The best solution is obtained
(b)

Figure 2. Flow Chart of ACOTS (a) and TS (b)

Simulation on parameter α does not show a significant pattern or trend either towards the value of the objective function or computation time. In contrast to the parameter β, in this parameter simulation, the larger the value of β causes the total costs to be greater and then tends to be constant while the computation time is quite fluctuating with a decreasing pattern. This can be seen in Figure 3.

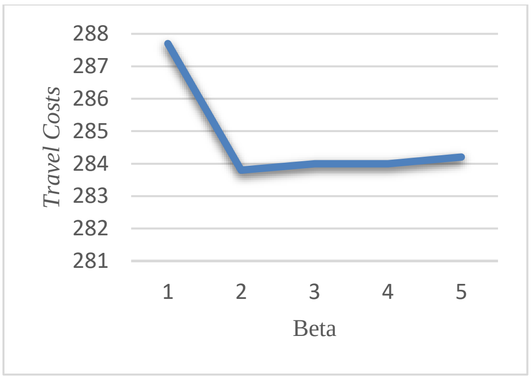


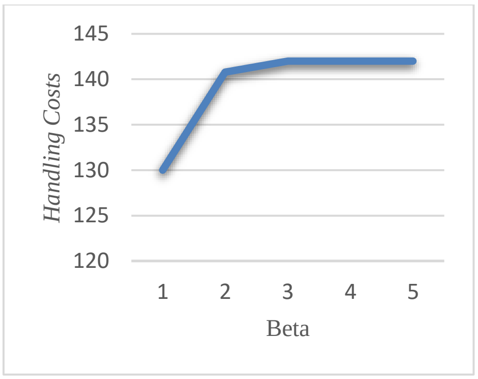


(a) (b)

Figure 3. The effect of β on the objective function (a) and computational time (b)

However, if it is plotted against travel costs, an increase β causes the value of travel costs to decrease but handling costs to increase. This simulation shows that β as a distance control factor, it is quite influential in determining the minimum route. But, considering that the arrangement of goods is not regulated by ACO's performance, ACO's desire to choose the minimum route gives the implication of the increased handling costs. This can be seen in Figure 4.





(a) (b)

Figure 4. The effect of β on travel costs (a) and handling costs (b)

Tabel 1 shows that in this simulation, the best value for total costs at ρ=0.5, m = 15, and 5 iterations, is obtained by α=3, and β=1. But in Table 2, it can be seen that with $m = 5$, α=5, and β=1 the total costs are smaller. This shows that the number of ants does not have a significant effect on the resulting solution in this case. Table 3 shows that ACOTS provides a better solution than ACO working alone without TS, although the computation time is slightly longer. In this case, the total cost is in minutes and the computation time is in seconds.

Table 1. The result of parameter simulation on α dan β

α	β	ρ	m	ACO Iteration	TS Iteration	Total Costs	Computation Time
1	1	0,50	15	5	5	417,7	71,7
1	2	0,50	15	5	5	424,6	60,1
1	3	0,50	15	5	5	426,00	62,1
1	4	0,50	15	5	5	426,00	50,5
1	5	0,50	15	5	5	426,20	56,2
2	1	0,50	15	5	5	417	75,4
3	**1**	**0,50**	**15**	**5**	**5**	**415,6**	**68,8**
4	1	0,50	15	5	5	419,2	56,9
5	1	0,50	15	5	5	418,8	63,9

Tabel 2. The result of parameter simulation on α and the number of ants

α	β	ρ	m	ACO Iteration	TS Iteration	Total Costs	Computation Time
1	1	0,50	5	5	5	419,6	50,8
1	1	0,50	25	5	5	415,20	65,2
5	**1**	**0,50**	**5**	**5**	**5**	**412,30**	**55,1**
5	1	0,50	25	5	5	414,90	75,6
10	1	0,50	5	5	5	414,50	51,5
10	1	0,50	25	5	5	416,40	81,4

Table 3. The result differences between ACO and ACOTS solutions

Iteration	Total Costs		Computation Time	
	ACO	ACOTS	ACO	ACOTS
5	425	412,30	6,93	55,1
10	426,6	417,90	12,4	205,9
15	424,8	420,90	17,6	354,5
20	424,3	417,00	26,8	836,2

3.3. Completing the Case Study of TSPPDHTW using the ACOTS Algorithm

The ACOTS algorithm was then applied to the TSPPDHTW case as a continuation of [1] to solve the TSPPDHTW using the exact method with a computation time of about 134 hours with device specification Core I3 memory and 8GB VGA Intel Grafis. In this study, the case was solved using the heuristic method through the ACOTS Algorithm to accelerate the acquisition of a solution. Based on the simulations, the TSPPDHTW case is quite well solved using the ACOTS Algorithm with parameters $\alpha = 5$, $\beta = 1$, ρ=0,5,$m = 5$ with 5 iterations each for ACO and TS. The best solution generated from the ACOTS Algorithm with a computation time of 61 seconds is presented in Table 4.

Table 4. The result of TSPPDHTW by ACOTS

Customer (i)	Name of Shop	Route	Travel Time (minutes)	Service Time	Time Windows
13	Oeyand	1-13	28	08.00	08.00-12.00
5	Koto Jambak	13-5	1	08.02	07.00-12.00
10	Harapan Kencana	5-10	31	08.35	08.00-09.00
2	Focus	10-2	14	08.56	06.00-10.00
4	Infinite Bike	2-4	23	09.29	08.00-10.00
14	Kiki Jaya	4-14	29	12.00	12.00-14.00
8	Central Sepeda	14-8	38	13.00	13.00-16.00
7	Bike Colony Terasutra	8-7	10	13.29	09.00-14.00
3	Rodalink Bogor	7-3	3	13.49	11.00-15.00
6	Family	3-6	26	14.34	12.00-16.00
12	Berkah	6-12	23	15.20	13.00-17.00
9	Unitide Bike	12-9	26	16.00	16.00-17.00
15	Sinar Harapan	9-15	17	19.00	19.00-22.00
11	Bogor Mountain Biking	15-11	8	19.10	06.00-22.00
1	Semeru Bike Shop (depot)	11-1	9	-	

The route recommendation as a result of a heuristic solution is Semeru Bike Shop as a depot (1) – Oeyand (customer 13) – Koto Jambak (customer 5) - Harapan Kencana (customer 10) - Focus (customer 2) – Infinite Bike (customer 4) - Bike Kiki (customer 14) – Central Sepeda (Customer 8) – Bike Colony Terasutra (customer 7) – Rodalink Bogor (customer 3) – Family (customer 6) - Berkah (customer 12) - United Bike (customer 9) - Sinar Harapan (customer 15) – Bogor Mountain Biking (customer 11) and back to the depot. The total travel time for this route is 286 minutes. This value has a difference of 6 minutes from the total travel time on the optimal route. The illustration of the policy on the arrangement of goods in the vehicle according to the route formed can be seen in Table 5. The blue box shows type I goods, while the black box shows type II goods. The total handling time required with this policy is 122 minutes, which has a difference of 66 minutes from the total handling time required with the optimal policy. The total time needed to pick up and deliver the goods to all customers with this route and goods arrangement policy is 408 minutes. This value has a deviation of about 22% with the exact method, which has an objective function value of 334 minutes. However, the ACOTS algorithm in solving this case has a much faster computation time than using the exact method. The ACOTS algorithm can save up to 99.99% of the time.

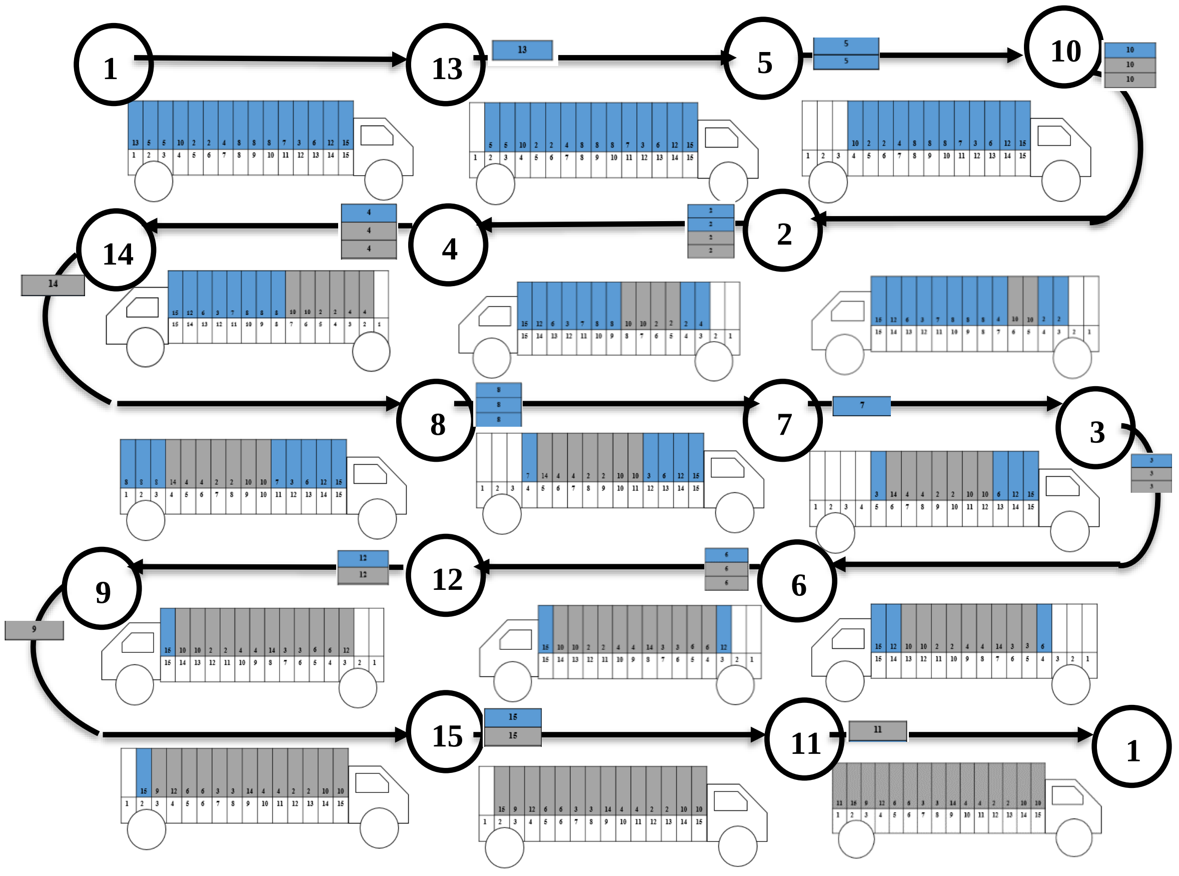


Figure 5. The illustration of goods arrangement in the vehicle

4. CONCLUSIONS

The ACO and TS algorithms can be developed into a new algorithm (called ACOTS) which can solve combinatorial cases such as TSPPDHTW. Based on the simulation, the suitable parameters for this case are $\alpha = 5, \beta = 1, \rho = 0{,}5$ with 5 ants and 5 iterations. In this study, the TSPPDHTW case study in the previous research can be completed using the ACOTS Algorithm with an objective function value of 408 minutes and a computation time of 61 seconds. Although the objective value has a fairly large deviation from the exact method, which is about 22%, the ACOTS algorithm can save up to 99.99% of the time in finding a solution. This shows that ACOTS can be an alternative method for completing the TSPPDHTW model in a computationally efficient manner so that it can be implemented in real life.

REFERENCES

[1] I. S. Amalia, T. Bakhtiar, and Jaharuddin, "Minimizing the handling time in VRP with pickup-and-delivery and time windows," *J. Phys. Conf. Ser.*, vol. 1863, no. 1, 2021, doi: 10.1088/1742-6596/1863/1/012002.
[2] P. Guo, M. Hou, and L. Ye, "MEATSP: A Membrane Evolutionary Algorithm for Solving TSP," *IEEE Access*, vol. 8, pp. 199081–199096, 2020, doi: 10.1109/ACCESS.2020.3035058.
[3] K. E. Adetunji, I. W. Hofsajer, A. M. Abu-Mahfouz, and L. Cheng, "A Review of Metaheuristic Techniques for Optimal

Integration of Electrical Units in Distribution Networks," *IEEE Access*, vol. 9, pp. 5046–5068, 2021, doi: 10.1109/ACCESS.2020.3048438.

[4] J. Kozak, *Studies in Computational Intelligence 781: Decision Tree and Ensemble Learning Based on Ant Colony Optimization*. 2019. [Online]. Available: http://www.springer.com/series/7092

[5] M. S. Umam, M. Mustafid, and S. Suryono, "A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem," *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2021, doi: 10.1016/j.jksuci.2021.08.025.

[6] J. J. De La Cruz, C. D. Paternina-Arboleda, V. Cantillo, and J. R. Montoya-Torres, "A two-pheromone trail ant colony system - Tabu search approach for the heterogeneous vehicle routing problem with time windows and multiple products," *J. Heuristics*, vol. 19, no. 2, pp. 233–252, 2013, doi: 10.1007/s10732-011-9184-0.

[7] S. Hanif, "RESEARCH PAPERS TRAVELLING SALESMEN PROBLEM SOLUTION WITH ANT-COLONY," vol. 14, no. 2, pp. 1–9, 2020.

[8] R. W. Dewantoro, P. Sihombing, and Sutarman, "The Combination of Ant Colony Optimization (ACO) and Tabu Search (TS) Algorithm to Solve the Traveling Salesman Problem (TSP)," *2019 3rd Int. Conf. Electr. Telecommun. Comput. Eng. ELTICOM 2019 - Proc.*, pp. 160–164, 2019, doi: 10.1109/ELTICOM47379.2019.8943832.

[9] M. Diallo, A. Quintero, and S. Pierre, "An Efficient Approach Based on Ant Colony Optimization and Tabu Search for a Resource Embedding across Multiple Cloud Providers," *IEEE Trans. Cloud Comput.*, vol. 9, no. 3, pp. 896–909, 2021, doi: 10.1109/TCC.2019.2904227.

[10] Q. Li, W. Tu, and L. Zhuo, "Reliable rescue routing optimization for urban emergency logistics under travel time uncertainty," *ISPRS Int. J. Geo-Information*, vol. 7, no. 2, 2018, doi: 10.3390/ijgi7020077.

[11] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Phys. A Stat. Mech. its Appl.*, vol. 391, no. 6, pp. 2193–2196, 2012, doi: 10.1016/j.physa.2011.12.004.

[12] Z. Zhang, Z. Xu, S. Luan, X. Li, and Y. Sun, "Opposition-based ant colony optimization algorithm for the traveling salesman problem," *Mathematics*, vol. 8, no. 10, pp. 1–16, 2020, doi: 10.3390/MATH8101650.

[13] W. Deng, J. Xu, Y. Song, and H. Zhao, "An effective improved co-evolution ant colony optimisation algorithm with multi-strategies and its application," *Int. J. Bio-Inspired Comput.*, vol. 16, no. 3, pp. 158–170, 2020, doi: 10.1504/IJBIC.2020.111267.

[14] C. K. Teoh, A. Wibowo, and M. S. Ngadiman, "Review of state of the art for metaheuristic techniques in Academic Scheduling Problems," *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 1–21, 2015, doi: 10.1007/s10462-013-9399-6.

[15] S. A. Fahmy, B. A. Alahlani, and T. F. Abdelmazuid, "A tabu search approach for designing shopping centers," *2017 9th IEEE-GCC Conf. Exhib. GCCCE 2017*, 2018, doi: 10.1109/IEEEGCC.2017.8447954.

[16] L. Pan, X. Liu, Y. Xia, and L. N. Xing, "Tabu Search Algorithm for the Bike Sharing Rebalancing Problem," *IEEE Access*, vol. 8, pp. 144543–144556, 2020, doi: 10.1109/ACCESS.2020.3011844.

[17] S. Basu, "Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey," *Am. J. Oper. Res.*, vol. 02, no. 02, pp. 163–173, 2012, doi: 10.4236/ajor.2012.22019.

[18] M. Gmira, M. Gendreau, A. Lodi, and J. Y. Potvin, "Tabu search for the time-dependent vehicle routing problem with time windows on a road network," *Eur. J. Oper. Res.*, vol. 288, no. 1, pp. 129–140, 2021, doi: 10.1016/j.ejor.2020.05.041.