Jurnal Teknik Industri
Vol. 23, No. 2, August 2022, pp. 105-120
ISSN : 1978-1431 print | 2527-4112 online

An Improved Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows

M. F. Ibrahim [a*], M. I. Mustofa [a], P. Meilanitasari [a], S. U. Wijaya [a]
[a] Logistics Engineering Department, Universitas Internasional Semen Indonesia, Gresik, Indonesia
* Corresponding author: muhammad.ibrahim@uisi.ac.id

ARTICLE INFO

Article history
Received May 27, 2022
Revised August 13, 2022
Accepted August 28, 2022
Available Online August 31, 2022

Keywords
Ant Colony Optimization
VRPTW
Distribution
Logistic
Supply Chain

ABSTRACT

Distribution plays an important role in the supply chain system. One of the critical problems in distribution is the vehicle routing problem. This research proposes the Improved Ant Colony Optimization (IACO) algorithm to solve the Vehicle Routing Problem with Time Windows (VRPTW). The main objective is to minimize the total vehicle mileage by considering the vehicle capacity and customer time windows. The proposed IACO algorithm is inspired by the conventional Ant Colony Optimization (ACO) algorithm by adding local search and mutation processes. Numerical experiments were conducted to test that the routes generated did not violate the customer's time window constraints. In addition, this study also compares the proposed IACO algorithm routes with other metaheuristic algorithms, namely ACO classic and Tabu Search. In addition, this investigation was carried out by experimenting with the number of iterations. The results of numerical experiments prove that the proposed IACO algorithm can minimize the total vehicle mileage without violating capacity constraints and time windows.

1. Introduction

In recent years, the logistics business has grown significantly worldwide [1, 2]. Increasing customer demand and demand fulfillment are challenges to logistics activities [3, 4]. Distribution activities are important in moving goods from producers to consumers on time, in the right quantity, and good condition [5]. Limited vehicle capacity and customer time windows are crucial issues and create challenges in this activity [6, 7]. One of the important decisions in distribution activities is the determination of delivery routes from depots to customers [8-10]. Optimizing distribution routes with limited vehicle capacity and customer time windows is called the Vehicle Routing Problem with Time Windows (VRPTW) [11].

VRPTW is a route determination problem with time window constraints in fulfilling each customer's demand [12, 13]. Solving this problem is to minimize the number of vehicles used. In addition, this problem aims to minimize the total mileage with capacity constraints and time windows. According to Truden, et al. [14], VRPTW produces a series of optimal routes carried out by a fleet of vehicles to serve a set of customers within set

https://doi.org/10.22219/JTIUMM.Vol23.No2.105-120 http://ejournal.umm.ac.id/index.php/industri ti.jurnal@umm.ac.id

Please cite this article as: Ibrahim, M. F., Mustofa, M. I., Meilanitasari, P., & Wijaya, S. U. (2022). An Improved Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows. Jurnal Teknik Industri, 23(2), 105–120. https://doi.org/10.22219/JTIUMM.Vol23.No2.105-120

time windows. In this problem, time windows are two-sided, meaning each customer must be served at the earliest and before the customer's latest time. If a vehicle arrives before the earliest time, it results in idle or waiting time. On the other hand, when the vehicle arrives after the latest time, the vehicle is tardy [11, 15]. There is also a service time required to serve each consumer.

In the last three years, there have been several studies on VRPTW problems. Most of them use metaheuristic methods. Wang, et al. [16] used the multi-Ant System algorithm in solving the VRPTW problem by considering service time. The algorithm is a development of the ant colony optimization algorithm. Hybrid Evolutionary Algorithm was developed to solve VRPTW by Zhang, et al. [17]. Numerical experiment results in a state that the algorithm provides effective and higher-quality results. Ant Colony Optimization algorithm was also developed by Suppan, et al. [18] with hard time windows. In the study, ACO was stated to perform better than genetic algorithms. Metaheuristics are widely recognized as efficient approaches to solving non-polynomial hard optimization problems. Meanwhile, according to Elshaer and Awad [19], metaheuristic algorithms can be classified into single solution-based and population-based. This method can also be called an approximate method because, in the search process, it cannot be guaranteed to find the best solution [20]. The population-based method is further divided into evolutionary computation, for example, Genetic Algorithm (GA). Meanwhile, one of the swarm intelligence procedures is Ant Colony Optimization (ACO) [19].

ACO takes inspiration from the foraging behavior of ants. These ants deposit pheromones on the ground to mark clear paths other colony members should follow. ACO exploits a similar mechanism to solve optimization problems [21]. According to Liu [22], this algorithm simulates an ant colony searching for paths during the foraging process. ACO is a type of paradigm development used in solving optimization problems with a collection of algorithms that use probabilistic methods. This algorithm replicates the behavior of a group of ant colonies in finding food. Along with the times, the ACO algorithm requires an improvement to get more optimal results. Many researchers are interested in developing the ACO algorithm, then called Improved Ant Colony Optimization (IACO). The improvement used does not combine the ACO algorithm with other algorithms. However, the improvement is in the form of changing the strategy in the ACO algorithm stages—for example, strategies in pheromone update and local search. Improvements aim to produce better results with minimal computing time [23].

This research proposes the Improved Ant Colony Optimization (IACO) algorithm for solving VRPTW. The contribution focuses on the developed algorithm's Mutation and Local Search processes. Improvements to the conventional ACO algorithm must be made to be sensitive to vehicle capacity constraints and time windows. IACO algorithm is proposed by adding mutation and a local search process. The mutation process considers the resulting distance, and local search considers time windows. The proposed IACO algorithm is implemented in a real case study. It is expected to overcome the problem of delayed product delivery.

2. Literature Review

This section presents the results of previous research literature studies that solve VRP problems with metaheuristic methods [24-26]. Li, et al. [27] developed an ACO algorithm to solve multi-depot VRP problems with an innovative approach to updating pheromones. The results show that improved ACO produces higher quality than conventional ACO. Huang, et al. [1], completed a case study of a Vehicle Routing Problem with a Drone (VRPD) utilizing the ACO algorithm with the 2-opt method for local search.

The results of numerical experiments state that the ACO algorithm performs well in solving classic VRP problems. ACO algorithm for the VRPTW case study was developed by adding a saving algorithm and λ-interchange mechanism by Ding, et al. [28]. The proposed algorithm successfully improved the search speed in large-scale problems.

Table 1. Literature Review

Authors	VRP				Method			Improvement	Fungsi Tujuan		
	C	MD	PD	TW	ACO	TABU	GA		Min-Total Cost	Max-Profit	Min-DIstance
Li, et al. [27]		√			√			Pheromone Updating		√	
Huang, et al. [1]	√				√			Local Search	√		
Ding, et al. [28]				√	√			Saving Algorithm & λ-Interchange mechanism	√		
Huang and Ding [29]					√			Path Weight Matrix & Save Matrix	√		
Jia, et al. [30]						√		Mutasi – Mixed Local Search			√
Fu, et al. [31]						√		Structure of The Tabu List	√		
Li and Li [32]				√		√		Adaptive Tabu Length And Neighborhood Structure	√		
Nazif and Lee [33]							√	Crossover Operators	√		
Zhong-yue, et al. [34]							√	Crossover Operators	√		
Huang and Zhang [6]			√	√			√	Local Search	√		
Ibrahim, et al.				√	√			Mutation & Local Search Operators			√

C: Capacitated; MD: Multi-Depot; PD: Pickup & Delivery; TW: Time Windows
ACO: Ant Colony Optimization; TABU: Tabu Search; GA: Genetic Algorithm

Meanwhile, Huang and Ding [29] developed the ACO algorithm by adding Path Weight Matrix and Save Matrix to solve the VRP problem. Computer simulation results prove that the resulting routes are better with other procedures. A different algorithm is offered by Jia, et al. [30]. They proposed a Tabu Search algorithm for VRP problems by adding Mutation and Mixed Local Search. The quality of the resulting solution is claimed to be better on large-scale problems. In addition, the Tabu Search algorithm for VRP problems with an improvised Structure of the tabu list was also conducted by Fu, et al. [31]. The computational results show that the algorithm has superior computation time advantages. Li and Li [32] developed a Tabu Search algorithm for the VRPTW problem by adding adaptive tabu length and neighborhood structure. The java programming language was used for solution computation. The results stated that the proposed algorithm proved to be better than the conventional tabu search algorithm.

In VRP solving, Genetic Algorithm with an improved crossover operator was proposed by Nazif and Lee [33] and Zhong-yue, et al. [34]. The results stated that the solution quality of the proposed procedure is competitive compared to other procedures. Another finding is that the optimal solution can be easily obtained. On the other hand, an improved genetic algorithm procedure with improvements in the local search was conducted by Huang and Zhang [6]. The research was applied to solve the VRPPDTW problem. The results concluded that improvement with local search has a faster global optimization capability.

Based on the previous research presented in Table 1, it can be concluded that many previous studies have utilized metaheuristic algorithms in solving various VRP problems. There have also been many improvements or developments of conventional metaheuristic algorithms to achieve better results with shorter computing times. It is also not uncommon to combine with other metaheuristic algorithms. However, especially in VRPTW, there has not been much development with mutation and local search. Hence, in this research, mutation and local search operators are developed in the IACO algorithm to solve VRPTW.

3. Methods

3.1 VRPTW Model

In this article, the problem of determining vehicle routes from a distributor to some destination nodes is designed as a Vehicle Routing Problem with Time Windows. In this problem, the optimal route is determined with the objective function of distance minimization. The assumptions include a symmetrical distance between nodes and do not consider congestion. Some constraints are required to ensure the boundaries that should not be violated, such as vehicle capacity and time windows of destination nodes. This VRPTW problem is used to solve the routing problem. The IACO is developed and used for optimization in distance minimization. The mathematical model for VRPTW in this paper is based on Toth and Vigo [35] and is formulated as follows:

$$Minimize\ Z = \sum_{k \epsilon K} \sum_{(i,j) \epsilon A} c_{ij} x_{ijk} \tag{1}$$

subject to:

$$\sum_{k \epsilon K} \sum_{j \epsilon \Delta+(i)} x_{ijk} = 1 \qquad ; \forall i \in N, \tag{2}$$

$$\sum_{j \epsilon \Delta-(0)} x_{0jk} = 1 \qquad ; \forall k \in K, \tag{3}$$

$$\sum_{i \epsilon \Delta-(j)} x_{ijk} - \sum_{i \epsilon \Delta+(j)} x_{jik} = 0 \qquad ; \forall k \in K,\ j \in N, \tag{4}$$

$$\sum_{i \epsilon \Delta-(n+1)} x_{i,n+1,k} = 1 \qquad ; \forall k \in K, \tag{5}$$

$$x_{ijk}(w_{ik} + s_i + t_{ij} - w_{jk}) \le 0 \qquad ; \forall k \in K,\ (i,j) \in A, \tag{6}$$

$$a_i \sum_{j \epsilon \Delta+(i)} x_{ijk} \le w_{ik} \le b_i \sum_{j \epsilon \Delta+(i)} x_{ijk} \qquad ; \forall k \in K,\ i \in N, \tag{7}$$

$$E \le w_{ik} \le L \qquad ; \forall k \in K,\ i \in \{0, n+1\}, \tag{8}$$

$$\sum_{i \epsilon N} d_i \sum_{j \epsilon \Delta+(i)} x_{ijk} \leq C \qquad ;\forall k \in K, \tag{9}$$

$$x_{ijk} \geq 0 \qquad ;\forall k \in K,\ (i,j) \in A, \tag{10}$$

$$x_{ijk} \in \{0,1\} \qquad ;\forall k \in K,\ (i,j) \in A, \tag{11}$$

Notations:

K : Set of vehicles with the same capacity
A : Node set= $\{0,1,\dots,n+1\}$
N : Customer set= $\{1,\dots,n\}$
c_{ij} : distance from vertex i and j
x_{ijk} : equal to 1 if art *(i,j)* is used by vehicle k and 0 otherwise.
w_{ik} : start of service at node i when serviced by vehicle k.
E : earliest possible departure from the depot.
L : latest possible arrival at the depot.
s_i : service time node i
t_{ij} : time travel from node i to node j
d_i : customer demand quantity i
$[a_i, b_i]$: time windows from node i

Equation (1) represents the objective function to minimize total distance, while Equation (2) is a constraint restricting the assignment of each customer to exactly one vehicle route. Equations (3)-(5) are constraints that characterize the flow on the path to be followed by vehicle k. Equations (6)-(9) are a constraint for guaranteeing schedule feasibility concerning time consideration and capacity. Equation (10) non-negativity constraint, and Equation (11) imposes a binary condition on the flow variable.

3.2 Improved Ant Colony Optimization

The method proposed in this research is Improved Ant Colony Optimization (IACO). The algorithm added a mutation process and local search before the pheromone update process. With the mutation and local search processes, the IACO algorithm is expected to have advantages, namely finding the optimal, effective, and efficient solution. The flowchart of the proposed method can be seen in Fig. 1. The seven main stages in the proposed IACO algorithm are explained as follows.

Step 1: Creating a Route Solution with State Transition

In IACO, an ant forms a route consisting of the customers visited by the ant. The customers visited by an ant are stored in a tabu list. In decision making, an ant k chooses customer (node) j after visiting customer (node) i using the probability formula Equation (12).

$$\boldsymbol{p_{ij}(k)} = \frac{(\tau_{ij})^{\alpha}.(\eta_{ij})^{\beta}}{\Sigma(\tau_{ij})^{\alpha}.(\eta_{ij})^{\beta}} \tag{12}$$

Please cite this article as: Ibrahim, M. F., Mustofa, M. I., Meilanitasari, P., & Wijaya, S. U. (2022). An Improved Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows. Jurnal Teknik Industri, 23(2), 105–120. https://doi.org/10.22219/JTIUMM.Vol23.No2.105-120

Where ($p_{ij}(k)$) is the probability of ant choosing customer (node) j after visiting customer (node) i on a k-th route. τ_{ij} is the pheromone of the distance of customer i to customer j, and η_{ij} is the visibility of the distance of customer i to customer j. α and β are, respectively the ant trail intensity controller and the visibility controller of the distance of customer i to customer j with values $\alpha \geq 0$ and $\beta \geq 0$, and Tabu is the set of customers that ant should not choose.

Step 2: Calculating value $Pm(t)$

Pm(t) is the probability of mutation at iteration t. The probability of mutation at iteration t (Pm(t)) is modeled in Equation (13).

$$p_m(t) = p_m^{min} + (p_m^{max} - p_m^{min})^{1-t/T} \quad (13)$$

p_m^{min} is the lowest mutation probability level defined by the Equation $p_m^{min} = 1/n$. n is the number of all customers (nodes), and p_m^{max} is the highest mutation probability level defined by the Equation $p_m^{max} = 1/n$, where n is the number of routes formed in the solution. t is the current iteration, and T is the maximum iteration in the solution search. The value of $Pm(t)$ is used for mutation.

Step 3: Mutations

At this stage, the mutation is performed if the value of the random value at each node <= $Pm(t)$. There is no mutation process if there is no random value <= $Pm(t)$. Therefore, a random value is required for each node. The mutation process is done by swapping a node's position with another node with a random value <= $Pm(t)$ on a route. An illustration of the random value at each node can be seen in Fig. 2. This random value is used for mutation on each route.

An illustration of mutation between routes is shown in Fig. 3. The results show that the position of node one on route two is swapped with node four on route one. The requirement in selecting nodes for mutation is that if, in one route, there is more than one node with a random value <= $Pm(t)$, then the node selected for mutation is the smallest random value.

Step 4: Local Search

Local search is performed by finding the route with the most time windows violations. Furthermore, the route is configured to minimize time windows violations. After configuring the route, whether the configuration violates, the time windows limitation is checked. An illustration of the Local Search process is shown in Fig. 4.

For example, based on the mutation process, there are most time windows violations on route 1. When visiting node 3, node three is already closed. Therefore, local search eliminates the delay by swapping with another node. It is shown in Fig. 4.

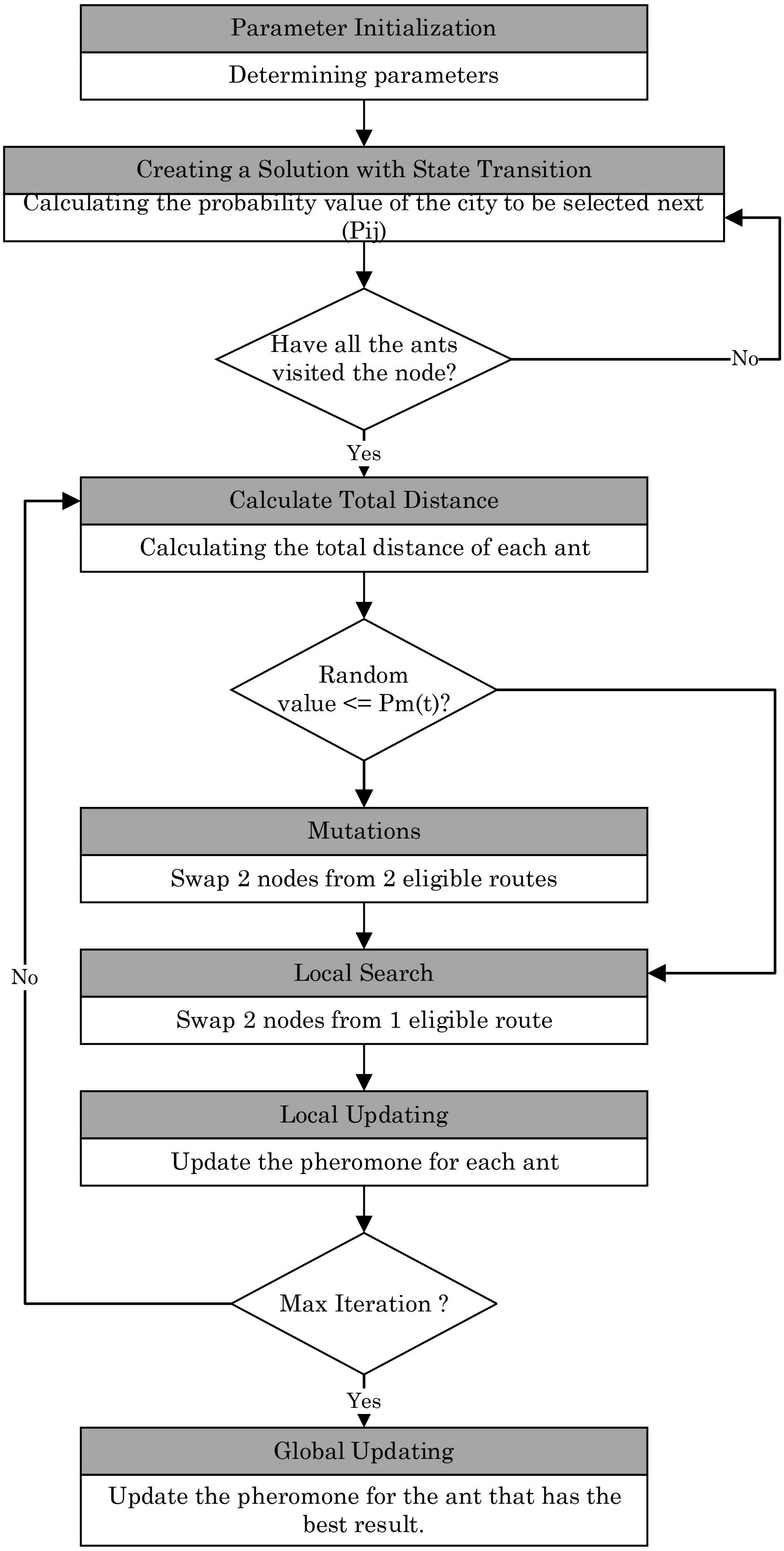


Fig. 1. Flow chart Propose IACO

Please cite this article as: Ibrahim, M. F., Mustofa, M. I., Meilanitasari, P., & Wijaya, S. U. (2022). An Improved Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows. Jurnal Teknik Industri, 23(2), 105–120. https://doi.org/10.22219/JTIUMM.Vol23.No2.105-120

Jurnal Teknik Industri
Vol. 23, No. 2, August 2022, pp. 105-120

Route 1	Depot	4	2	5	3	Depot
		0.099818701	0.902542	0.971266	0.738213	
Route 2	Depot	6	1	Depot		
		0.676614961	0.015803			

Fig. 2. Illustration of random numbers in the route mutation process

Route 1	Depot	1	2	5	3	Depot
Route 2	Depot	6	4	Depot		

Fig. 3. Illustration of route mutation

Route 1 Before Local Search :

Route 1	Depot	1	2	5	3	Depot

Route 1 After Local Search :

Route 1	Depot	1	5	3	2	Depot

Fig. 4. Illustration of Local Search process

Step 5: Local Updating

Ant colonies leave pheromones on the paths they travel. The strength of the pheromone changes because it evaporates, and the number of ants passing by changes. This phenomenon is modeled by Equation (14).

$$\tau_{ij} = (1-\rho).\tau_{ij} + \Delta\tau_{ij}^{k} \tag{14}$$

Where τ_{ij} is the change in the pheromone intensity value of each ant, $\Delta\tau_{ij}^{k}$ can be calculated based on Equation (15), where L_k is the total distance traveled by each ant.

$$\Delta\tau_{ij}^{k} = \frac{1}{L_k} \tag{15}$$

Step 6: Repeat Iterations

At this stage, iterations from state transition to local updating are repeated. The solution search is terminated if the maximum number of iterations has been reached. The search for a solution is repeated if the iteration condition has not been met.

Step 7: Global Updating

The pheromone value between customers on all paths may change due to evaporation and differences in the number of ants passing through. For global updating,

Please cite this article as: Ibrahim, M. F., Mustofa, M. I., Meilanitasari, P., & Wijaya, S. U. (2022). An Improved Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows. Jurnal Teknik Industri, 23(2), 105–120. https://doi.org/10.22219/JTIUMM.Vol23.No2.105-120

this pheromone is calculated after all iterations are complete and select the minimum distance to the ants in the last iteration. This behavior can be modeled in Equation (16).

$$\tau_{ij} = (1-\rho).\tau_{ij} + \frac{1}{Best\ distance} \tag{16}$$

Algorithm 1 Pseudocode Improvement Ant Colony Optimization

```
Input
distance, demand, time windows, speed, and vehicle capacity
Parameter initialization
alpha (α), beta (β), rho (ρ), number of ants (m), tho (τ), max iterations
Process of creating inverse distance matrix inverse distance calculates time matrix
for iteration = 1: iteration max
for i = 1:Ant
    for j = 1:number of node
            P (j) = Random probability of value r;;
            for b = 1:number of node s = cumulative P(i) ; If r ≤ s
                    Add the next route
                end
            end
     end
end
If If demand ≤ capacity && opening time ≤ arrival time < closing time(j) route(i,j)=b;;
    Calculate time and demand; break
else
   Return to Depot
end
   route starts from 1 route(i,j)=b;
   Set initial time;
   Set initial demand \; break
Calculate Pm(t)
Find <= 2 routes that have the largest distance;
Generate a random value for each node;
If random value <= Pm(t);
         swap route node 1 with route node 2 that has random value <= Pm(t);
Else Find the route that has the most time windows violations;
         Swap nodes that have time windows violations with other nodes until there are no time
         windows violations;
   Local updating
   min distance, selected ants, and selected route
Global updating
```

Algorithm 1 presents the detailed stages of Improvement Ant Colony Optimization (IACO). The first process begins with inputting data. Next, some parameter initialization in IACO is required, such as the value of alpha (α), beta (β), rho (ρ), the number of ants (m), initial tho ($\tau 0$), and maximum iteration. The next stage is to perform the inverse distance process and calculate the travel time from the distributor to the consumer and from node i to node j. This calculation is done because this data is needed in the state transition process or the formation of the next route visited by the vehicle concerning capacity constraints and time windows. After all, consumers are selected, the next process is to calculate the value of $Pm\ (t)$, which is used as the basis for mutation. The mutation process begins by calculating a random value for each destination node. The mutation is done by swapping a node's position with another node with a random value $<= Pm(t)$ on a route. The next process is local search if there is no random value <= $Pm(t)$. The local search process is done by looking for routes with the most time windows violations. Furthermore, the route with the most time windows violations is configured to eliminate time windows violations. After configuring the route, whether the configuration violates, the time windows limitation is checked. The next step is to update the local pheromone and the global pheromone. The calculation is finished by finding the best solution.

3.2 Case Study

The experiment in this research was conducted by implementing the developed IACO algorithm on a case study of shipping goods from one distribution center to forty-three customers. Demand and time windows for each customer are different. The lowest customer demand is five boxes; the most are 20 boxes. The total demand from customers is 229 boxes. Of all the customers who visited, the earliest opening time windows are at 7:00 and the latest at 9:00. Meanwhile, customers also have time windows for different closing hours. The earliest customer closing time is 12.30 to the latest at 19.00. Time windows in this problem are categorized as hard time windows problems. Vehicles wait if they arrive before the time windows of the customer's opening hours. However, vehicles that arrive beyond the customer's closing time are not served because the vehicle arrives after the time windows close. To compare the performance of the developed IACO algorithm, a comparison is also made with the ACO and Tabu search algorithms.

4. Results and Discussion

Data processing in this research uses the help of Matlab software. The experiment was conducted with 1000 iterations with a computer with Intel Core i5 2.4GHz Processor (8 CPUs), 8 Gb RAM, and 256 SSD. The output of this software model is the route and total mileage of each vehicle. Researchers tried to compare the routes generated by IACO with routes generated by other metaheuristic algorithms, namely ACO and Tabu Search. All routes generated by both IACO, conventional ACO and Tabu Search algorithms have met all VRPTW constraints. Each customer request can be sent, so delivery is on time.

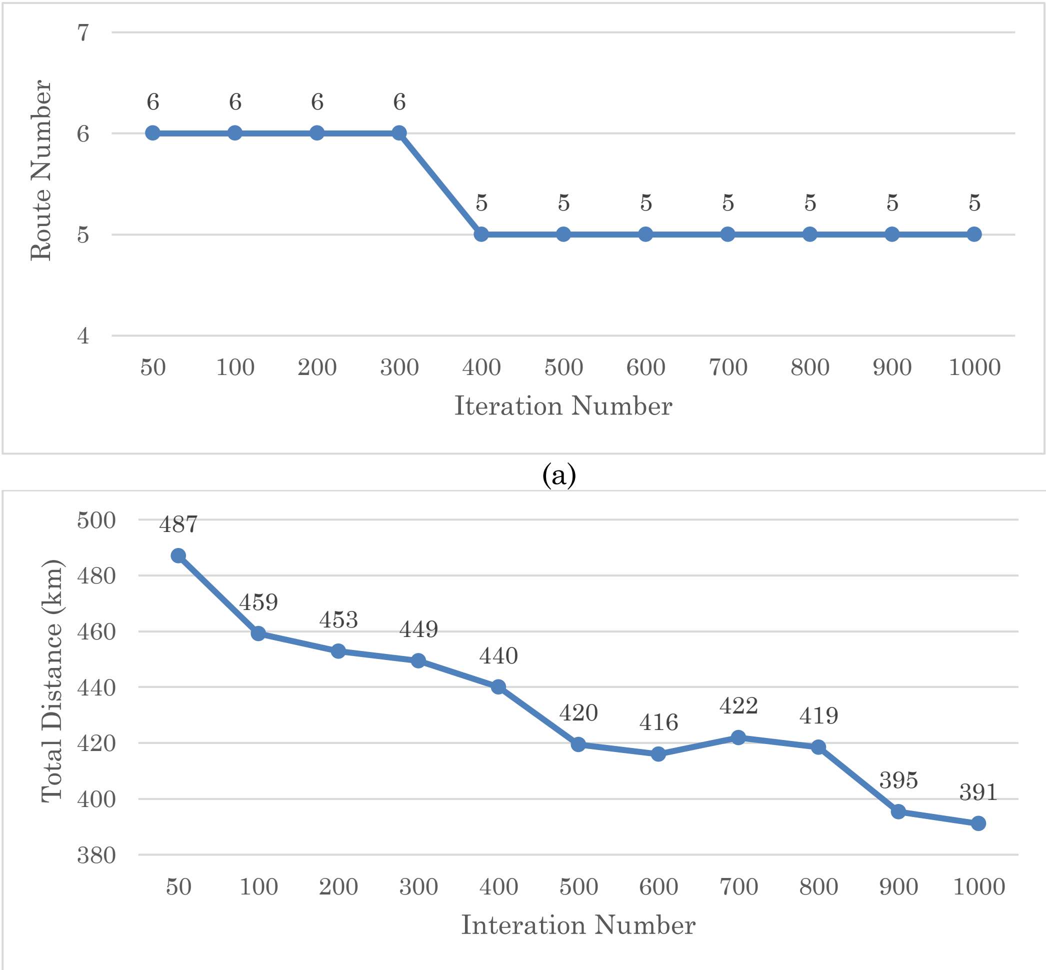
Table 2. Distance Traveled by ACO, Tabu Search, and IACO Routes

Vehicle	IACO (km)	ACO (km)	Tabu Search (km)
Vehicle 1	51	50.7	51
Vehicle 2	59.9	74.1	59.9
Vehicle 3	85.3	85	85.3
Vehicle 4	90.7	93.6	101.1
Vehicle 5	104.2	99.7	99.7
Total	391.1	403.1	397

A comparison of the distance traveled for each algorithm can be seen in Table 2. The results show that the route generated by the IACO algorithm has a minimum total distance of 391.1 km. While the route generated by the conventional ACO algorithm and Tabu Search has a distance of 403.1 km and 391 km, respectively. The comparison results convince us that the IACO algorithm produces better routes than the conventional ACO algorithm and Tabu Search. The conventional ACO algorithm route results are 12 km further than the IACO algorithm. So there are distance travel savings of 3%.

The IACO algorithm route is proven to deliver all requests with a minimum distance without violating the customer's time windows. These results show that the resulting route is more optimal than the route resulting from the conventional ACO algorithm. As presented in Table 2. there is a mileage savings of 2.9%. These savings certainly impact operational costs and generate savings for the company.

This study also carried out tests related to the number of iterations. It can be seen in Fig 5a. The number of routes formed decreases as the number of iterations increases. When iterations amounted to 50 to 300, the IACO algorithm gave the results of 6 routes. However, starting iterations amounted to 400, and the route decreased to 5. The decrease in the number of routes is directly proportional to the decrease in mileage. The higher the iterations attempted results in a set of routes with minimum distance traveled. The details of this finding are presented in Fig 5b. The trend is seen to decrease as the number of iterations increases. The graph shows that the number of iterations is very important for the set of routes generated by the IACO algorithm.



(b)

Fig 5. Experimental results (a) Number of iterations to number of routes and (b) Number of iterations to the total distance

We tried to analyze the effect of the number of iterations on computation time. The results of this experiment can be seen in Fig. 6. The computation time required is directly proportional to the number of iterations. The more the number of iterations, the longer the computation time. It is related to the more routes the ants take in each iteration. The computation time required by the IACO algorithm is considered quite short. It can be seen

Jurnal Teknik Industri
Vol. 23, No. 2, August 2022, pp. 105-120

that the computation time of 50 iterations takes 12.79 seconds while up to 1000 iterations only takes 17.18 seconds.

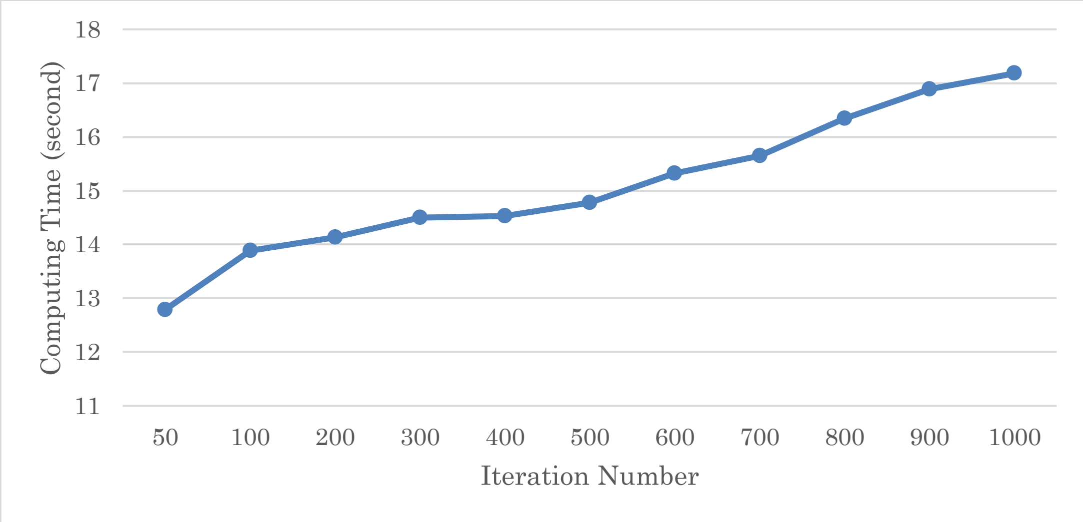


Fig. 6. Experimental results a number of iterations to the total distance

5. Conclusion

This research proposes an Improved Ant Colony Optimization (IACO) algorithm for solving Vehicle Routing Problems with Time Windows (VRPTW). The IACO algorithm is developed and applied to a VRPTW case study. The results of numerical experiments show that the developed IACO successfully provides a route with a minimum total distance compared to the Conventional ACO and Tabu Search algorithms. In addition, the IACO algorithm successfully provides a route with the minimum number of tours without violating the VRPTW constraints. The mutation and local search steps in the proposed IACO algorithm made the route minimal violation of time windows and a more optimal travel distance. This study also examined how iterations affect routes and distance. The results show that the trend of the number of routes and the total distance is also minimal with the increased iterations performed. This study concludes with various numerical experiments and analyses. The proposed IACO algorithm successfully provides better routes from all aspects with minimal computing time to solve VRPTW problems. Mutation operators and local search can be developed for further research to provide optimal results with a minimum number of iterations and computation time.

Declarations

Author contribution: All authors contributed to this paper.
Funding statement: This research is supported and financed by the Research and Community Service Institution of Universitas Internasional Semen Indonesia under the HRB scheme.
Conflict of interest: The authors declare no conflict of interest.
Additional information: No additional information is available for this paper.

Please cite this article as: Ibrahim, M. F., Mustofa, M. I., Meilanitasari, P., & Wijaya, S. U. (2022). An Improved Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows. Jurnal Teknik Industri, 23(2), 105–120. https://doi.org/10.22219/JTIUMM.Vol23.No2.105-120

References

[1] S.-H. Huang. Y.-H. Huang. C. A. Blazquez. and C.-Y. Chen. "Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm." *Advanced Engineering Informatics*. vol. 51. p. 101536. 2022. https://doi.org/10.1016/j.aei.2022.101536.

[2] C. Kunaka. "Logistics in the Developing World." in *International Encyclopedia of Transportation*. R. Vickerman. Ed. Oxford: Elsevier. 2021. pp. 150-156. doi: https://doi.org/10.1016/B978-0-08-102671-7.10235-0.

[3] S. K. Dewi and D. M. Utama. "A New Hybrid Whale Optimization Algorithm for Green Vehicle Routing Problem." *Systems Science & Control Engineering*. vol. 9. no. 1. pp. 61-72. 2021. https://doi.org/10.1080/21642583.2020.1863276.

[4] D. M. Utama. D. S. Widodo. M. F. Ibrahim. and S. K. Dewi. "A New Hybrid Butterfly Optimization Algorithm for Green Vehicle Routing Problem." *Journal of Advanced Transportation*. vol. 2020. p. 8834502. 2020. https://doi.org/10.1155/2020/8834502.

[5] I. Litomin. I. Tolmachov. and A. Galkin. "Use of the Distribution Center in the Ukrainian Distribution System." *Transportation Research Procedia*. vol. 16. pp. 313-322. 2016. https://doi.org/10.1016/j.trpro.2016.11.030.

[6] W. Huang and T. Zhang. "An improved genetic algorithm for Vehicle Routing Problem with Simultaneous Pickups and Deliveries and Time Windows." in *2016 35th Chinese Control Conference (CCC)*. 2016. pp. 9639-9643. https://doi.org/10.1109/ChiCC.2016.7554885.

[7] M. F. Ibrahim. M. M. Putri. D. Farista. and D. M. Utama. "An Improved Genetic Algorithm for Vehicle Routing Problem Pick-up and Delivery with Time Windows." *Jurnal Teknik Industri*. vol. 22. no. 1. pp. 1-17. 2021. https://doi.org/10.22219/JTIUMM.Vol22.No1.1-17.

[8] D. M. Utama. S. K. Dewi. A. Wahid. and I. Santoso. "The vehicle routing problem for perishable goods: A systematic review." *Cogent Engineering*. vol. 7. no. 1. p. 1816148. 2020. https://doi.org/10.1080/23311916.2020.1816148.

[9] D. M. Utama. B. N. I. Farida. U. Fitriani. M. F. Ibrahim. and D. S. Widodo. "Hybrid Henry Gas Solubility Optimization: An Effective Algorithm for Fuel Consumption Vehicle Routing Problem." *Jurnal Ilmiah Teknik Industri*. vol. 20. no. 2. pp. 141-152. 2021. https://doi.org/10.23917/jiti.v20i2.15640.

[10] D. M. Utama. T. A. Fitria. and A. K. Garside. "Artificial Bee Colony Algorithm for Solving Green Vehicle Routing Problems with Time Windows." *Journal of Physics: Conference Series*. vol. 1933. no. 1. p. 012043. 2021. https://doi.org/10.1088/1742-6596/1933/1/012043.

[11] P. Toth and D. Vigo. *The vehicle routing problem*. SIAM. 2002.

[12] R. Pérez-Rodríguez and A. Hernández-Aguirre. "A hybrid estimation of distribution algorithm for the vehicle routing problem with time windows." *Computers & Industrial Engineering*. vol. 130. pp. 75-96. 2019. https://doi.org/10.1016/j.cie.2019.02.017.

[13] M. F. Ibrahim. F. R. Nurhakiki. D. M. Utama. and A. A. Rizaki. "Optimised Genetic Algorithm Crossover and Mutation Stage for Vehicle Routing Problem Pick-Up and Delivery with Time Windows." *IOP Conference Series: Materials Science and Engineering*. vol. 1071. no. 1. p. 012025. 2021. https://doi.org/10.1088/1757-899X/1071/1/012025.

[14] C. Truden. K. Maier. and P. Armbrust. "Decomposition of the vehicle routing problem with time windows on the time dimension." *Transportation Research Procedia*. vol. 62. pp. 131-138. 2022. https://doi.org/10.1016/j.trpro.2022.02.017.

[15] K. Braekers. K. Ramaekers. and I. Van Nieuwenhuyse. "The vehicle routing problem: State of the art classification and review." *Computers & Industrial Engineering*. vol. 99. pp. 300-313. 2016. https://doi.org/10.1016/j.cie.2015.12.007.

[16] Y. Wang. L. Wang. Z. Peng. G. Chen. Z. Cai. and L. Xing. "A Multi Ant System based hybrid heuristic algorithm for Vehicle Routing Problem with Service Time Customization." *Swarm and Evolutionary Computation*. vol. 50. p. 100563. 2019. https://doi.org/10.1016/j.swevo.2019.100563.

[17] W. Zhang. D. Yang. G. Zhang. and M. Gen. "Hybrid multiobjective evolutionary algorithm with fast sampling strategy-based global search and route sequence difference-based local search for VRPTW." *Expert Systems with Applications*. vol. 145. p. 113151. 2020. https://doi.org/10.1016/j.eswa.2019.113151.

[18] M. Suppan. T. Hanne. and R. Dornberger. "Ant Colony Optimization to Solve the Rescue Problem as a Vehicle Routing Problem with Hard Time Windows." in *Proceedings of International Joint Conference on Advances in Computational Intelligence*. Singapore. 2022. pp. 53-65: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-0332-8_5.

[19] R. Elshaer and H. Awad. "A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants." *Computers & Industrial Engineering*. vol. 140. p. 106242. 2020. https://doi.org/10.1016/j.cie.2019.106242.

[20] M.-Y. Cheng and D. Prayogo. "Symbiotic Organisms Search: A new metaheuristic optimization algorithm." *Computers & Structures*. vol. 139. pp. 98-112. 2014. https://doi.org/10.1016/j.compstruc.2014.03.007.

[21] M. Dorigo. M. Birattari. and T. Stutzle. "Ant colony optimization." *IEEE Computational Intelligence Magazine*. vol. 1. no. 4. pp. 28-39. 2006. https://doi.org/10.1109/MCI.2006.329691.

[22] H. Liu. "Research on cloud computing adaptive task scheduling based on ant colony algorithm." *Optik*. vol. 258. p. 168677. 2022. https://doi.org/10.1016/j.ijleo.2022.168677.

[23] Y. Wang. L. Wang. G. Chen. Z. Cai. Y. Zhou. and L. Xing. "An Improved Ant Colony Optimization algorithm to the Periodic Vehicle Routing Problem with Time Window and Service Choice." *Swarm and Evolutionary Computation*. vol. 55. p. 100675. 2020. https://doi.org/10.1016/j.swevo.2020.100675.

[24] D. M. Utama. U. Fitriani. I. Amallynda. and R. D. Azmi. "A Novel Hybrid Yellow Saddle Goatfish Algorithm for Fuel Consumption Vehicle Routing Problem with Simultaneous Pick-up and Delivery Problem." *Jurnal Teknik Industri*. vol. 23. no. 1. pp. 43-66. 2022. https://doi.org/10.22219/JTIUMM.Vol23.No1.43-66.

[25] D. M. Utama. W. O. N. Safitri. and A. K. Garside. "A Modified Camel Algorithm for Optimizing Green Vehicle Routing Problem with Time Windows." *Jurnal Teknik Industri*. vol. 24. no. 1. pp. 23-36. 2022. https://doi.org/10.9744/jti.24.1.23-36.

[26] D. M. Utama. B. Widjonarko. and D. S. Widodo. "A novel hybrid jellyfish algorithm for minimizing fuel consumption capacitated vehicle routing problem." *Bulletin of Electrical Engineering and Informatics*. vol. 11. no. 3. pp. 1272-1279. 2022. https://doi.org/10.11591/eei.v11i3.3263.

[27] Y. Li. H. Soleimani. and M. Zohal. "An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives." *Journal of Cleaner Production*. vol. 227. pp. 1161-1172. 2019. https://doi.org/10.1016/j.jclepro.2019.03.185.

Please cite this article as: Ibrahim, M. F., Mustofa, M. I., Meilanitasari, P., & Wijaya, S. U. (2022). An Improved Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows. Jurnal Teknik Industri, 23(2), 105–120. https://doi.org/10.22219/JTIUMM.Vol23.No2.105-120

ISSN : 1978-1431 print | 2527-4112 online

[28] Q. Ding. X. Hu. L. Sun. and Y. Wang. "An improved ant colony optimization and its application to vehicle routing problem with time windows." *Neurocomputing*. vol. 98. pp. 101-107. 2012. https://doi.org/10.1016/j.neucom.2011.09.040.

[29] M. Huang and P. Ding. "An Improved Ant Colony Algorithm and Its Application in Vehicle Routing Problem." *Mathematical Problems in Engineering*. vol. 2013. p. 785383. 2013. https://doi.org/10.1155/2013/785383.

[30] H. Jia. Y. Li. B. Dong. and H. Ya. "An Improved Tabu Search Approach to Vehicle Routing Problem." *Procedia - Social and Behavioral Sciences*. vol. 96. pp. 1208-1217. 2013. https://doi.org/10.1016/j.sbspro.2013.08.138.

[31] Q. Fu. K. Zhou. H. Qi. and F. Jiang. "A modified tabu search algorithm to solve vehicle routing problem." *J. Comput*. vol. 29. pp. 197-209. 2018. https://doi.org/10.3966/199115992018062903019.

[32] G. Li and J. Li. "An Improved Tabu Search Algorithm for the Stochastic Vehicle Routing Problem With Soft Time Windows." *IEEE Access*. vol. 8. pp. 158115-158124. 2020. https://doi.org/10.1109/ACCESS.2020.3020093.

[33] H. Nazif and L. S. Lee. "Optimised crossover genetic algorithm for capacitated vehicle routing problem." *Applied Mathematical Modelling*. vol. 36. no. 5. pp. 2110-2117. 2012. https://doi.org/10.1016/j.apm.2011.08.010.

[34] S. Zhong-yue. G. Zhong-liang. and W. Qin. "An improved adaptive genetic algorithm for vehicle routing problem." in *2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM)*. 2010. vol. 1. pp. 116-120. https://doi.org/10.1109/ICLSIM.2010.5461457.

[35] P. Toth and D. Vigo. *Vehicle routing: problems. methods. and applications*. SIAM. 2014.

Jurnal Teknik Industri
Vol. 23, No. 2, August 2022, pp. 105-120

This page is blank

Please cite this article as: Ibrahim, M. F., Mustofa, M. I., Meilanitasari, P., & Wijaya, S. U. (2022). An Improved Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows. Jurnal Teknik Industri, 23(2), 105–120. https://doi.org/10.22219/JTIUMM.Vol23.No2.105-120