Optimization of Goods Distribution System Using Variable Neighborhood Search Method

Yunita*[1], Junia Kurniati[2], Desty Rodiah[3], Anggina Primanita[4], Munawaroh[5]
[1,2,3,4,5] Faculty of Computer Science, Sriwijaya University, Indonesia
E-mail: ***[1]yunita@ilkom.unsri.ac.id,** [2]niyaazulkarnain@ilkom.unsri.ac.id,
[3]destyrodiah@ilkom.unsri.ac.id, [4]anggina.primanita@gmail.com, [5]munawarohsyahfitri@gmail.com

Abstract

The problem of distribution of goods in the field of logistics is one of the problems that often occur today. In the distribution process, all companies expect to be able to optimize distribution costs to be more efficient. This makes the capacity of the transportation and distribution system must also be increased in order to create an efficient logistics system. To achieve this goal, producers must develop a proper distribution plan, because wrong distribution can lead to unsatisfactory distribution. Vehicle Routing Problem with Time Windows is a mathematical problem to determine the route for several identical vehicles in serving a number of customers with a certain stone limit. VRPTW is a Nondeterministic Polynomial Time problem, so the exact optimization method is difficult to solve the case. The purpose of this research is to solve the problem of VRPTW with the objective function that is the optimal distance by considering the time limit and vehicle capacity. This research uses the Variable Neighborhood Search algorithm in the solution search stage to solve the problem. Relying on the principle of selecting the nearest customer to be added to the vehicle route, the VNS method can utilize the concept of a varied solution environment, which enables the algorithm to explore the solution space in a more flexible way. The results of research on 100 customers obtained a total average mileage of 624.49. The total mileage without using the VNS method is 888.69, so the difference is 264.2 or there is a mileage efficiency of 29.73%.

Keywords *— Optimization, Distribution System, Vehicle Routing Problem with Time Windows, Variable Neighborhood Search*

1. INTRODUCTION

The problem of distribution of goods in the field of logistics is one of the problems that often occur today. In the distribution process, all companies expect to be able to optimize distribution costs to be more efficient. According to [1] transportation is about how to distribute an item from a number of sources to a number of destinations. This makes the capacity of the transportation and distribution system must also be increased in order to create an efficient logistics system. To achieve this goal, producers must develop a proper distribution plan, because wrong distribution can lead to unsatisfactory distribution. One of the transport decisions made in the field of distribution is the accuracy of determining the optimal vehicle visit route configuration. The problem in transport decision is to determine or find the optimal route. The optimal route can minimize the cost which is then called the Vehicle Routing Problem (VRP) [2]

Vehicle Routing Problem (VRP) is a mathematical problem to determine the route for several identical vehicles in serving a number of customers [3]. The problem of VRP in distribution is how to determine the optimal route from a group of vehicles in delivering orders from the depot to many customers. The route must be arranged so that each consumer can only be visited by one vehicle. Each route must start and end at the depot and the total demand from all consumers in one route must not exceed the capacity of the vehicle. In fact, there are various problems that cannot be solved with either the original VRP or the classic VRP. This led to the development of several variants of VRP. For example, when the route is found to have a one-way or two-way road, which will affect the selection of the VRP variant. Routes with one-way roads can use the VRP variant asymmetric model [4]. Research on VRP variants has continued to be done, both from the perspective of theory [5]and its application [6] dan [7].

In VRP there is one or more depots that will serve a number of customers with certain limitations. The most common limitation is the maximum capacity of the vehicles used in the logistics system. In fact, apart from the capacity limitations of the vehicles used, there are often limited or specific time limits (time windows) in the process of taking or distributing goods. This problem is called Vehicle Routing Problem with Time Windows (VRPTW). VRPTW is an NP-hard problem (Nondeterministic Polynomial Time) so the exact optimization method is difficult to solve the VRP case. According to [8] VRP is responsible for planning routes with a number of different vehicles, in one or more depots and serving geographically dispersed consumers.

A variant of VRP with additional time windows is known as VRPTW. Time windows can be interpreted as a time limit for a vehicle to be able to visit customers. This means that in addition to considering the selection of routes, there are limitations on the capacity and time period for vehicles in serving each customer [9]. Some research related to the algorithm used to solve the VRPTW problem is the exact method, the heuristic method and the metaheuristic method. The exact method that can be used to solve the VRPTW problem, for example, is Branch and Bound [10]. The weakness of this method is that it takes a long time to get a solution even for cases with a small number of consumers [9]. Therefore, the computation that is done is increasing with the number of consumers that must be served, so there is a need to use other methods to solve the VRPTW problem. Furthermore, the second method that can be used to solve VRPTW is the heuristic method. The heuristic method is a basic approach method that searches the solution space to get a good solution but it is not certain to get the most effective solution [11]. The heuristic method obtains an acceptable solution in a fast time for complex problems by doing trial and error [12]. The third method that can be used to solve VRPTW is the metaheuristic method. The metaheuristic method is a method that has a higher level than the usual heuristic method and generally has a better performance than simple heuristics [12].

Heuristic method reduces the number of steps or time required to solve the problem compared to the exact search method, especially when the problem is very complex and also produces a more optimal calculation than the metaheuristic method. This research proposes a Heuristic method for solving VRPTW, namely Variable Neighborhood Search (VNS). By using VNS for the scheme in the solution search stage (neighborhood search). Relying on the principle of choosing the closest customer to add to the vehicle route. By processing the available data using a time window and with the VNS method can take advantage of the concept of a varied solution environment, which allows the algorithm to explore the solution space in a more flexible way. In each solution environment, VNS can combine local search to improve the quality of the solution. The combination of local search with various solution environments helps increase the probability of finding an optimal or near-optimal solution.

2. RESEARCH METHOD

2.1. Vehicle Routing Problem With Time Windows (VRPTW)

VRPTW is a combinatorial optimization problem that often appears in the context of logistics and transportation. In this problem, a group of vehicles has the task of serving a number of customers scattered in a certain location, where each customer has a request for specific goods or services. As mentioned by [13] various local search approaches, such as Tabu Search, have been applied to VRPTW. The main objective of this problem is to determine the optimal route for each vehicle with the aim of minimizing the total cost, taking into account the limitations of vehicle capacity and the time window of customer visits.The following are the main components and limitations of VRPTW:

1) Customers, this problem involves a group of customers that need to be served by a fleet of vehicles, where each vehicle must start and end its route at a central depot.
2) Vehicle Capacity, each vehicle has a limited capacity, so it can only carry a certain amount of goods. The total demand from customers assigned to a vehicle must not exceed its capacity.
3) Time Window, each customer has a specific time window in which they can be served. The vehicle must visit each customer within a predetermined time window, which represents real-world conditions where the customer has a specific time limit for delivery or service.
4) Travel Time and Distance, travel time or distance between two locations (depot or customer) is known. The goal is to minimize the total travel time or distance.
5) Objective Function, the main objective is to minimize the total cost, which generally includes travel costs, vehicle operational costs, and possibly penalties for time window violations.

VRPTW is a known problem (NP-hard), which means that finding the optimal solution becomes computationally more difficult when the number of customers and constraints increases. Efficiently solving VRPTW is very important for logistics and transportation companies to optimize shipping routes, reduce operational costs, and meet customer requirements within set deadlines.

Minimize: $\sum_{k\in V} \quad \sum_{i\in N} \quad \sum_{j\in N} c_{ij}\, x_{ijk}$ (1)

Subject to:

$\sum_{k\in V} \quad \sum_{j\in N} x_{ijk} = 1\,, \forall_i \in C$ (2)

$\sum_{i\in C} d_i \sum_{j\in N} x_{ijk} \leq q, \forall_k \in V$ (3)

$\sum_{j\in N} x_{0\,jk} = 1, \forall_k \in V$ (4)

$$\sum_{i\in N} x_{ihk} - \sum_{j\in N} x_{hjk} = 0, \forall_h \in C, \forall_k \in V \quad (5)$$
$$\sum_{i\in N} x_{i,n+1,k} = 1, \forall_k \in V \quad (6)$$
$$s_{ik} + t_{ij} - K(1 - x_{ijk}) \leq s_{jk}, \forall_{i,j} \in N, \forall_k \in V \quad (7)$$
$$a_i \leq s_{ik} \leq b_i, \forall_i \in N, \forall_k \in V \quad (8)$$
$$x_{ijk} \in \{0,1\}, \forall_{i,j} \in N, \forall_k \in V \quad (9)$$

The VRPTW considered in this study is formulated as a mixed-integer optimization model. The objective of the model is to minimize the total transportation cost, which is represented by the total travel distance incurred by all vehicles. This objective is expressed in Equation 1, where the summation is taken over all vehicles $k \in V$, all origin nodes $i \in N$, and all destination nodes $j \in N$. The binary decision variable x_{ijk} indicates whether vehicle *k* travels directly from node *i* to node *j*, while c_{ij} denotes the travel cost or distance between nodes i and j.

Equation 2 ensures that each customer $i \in C$ is visited exactly once by exactly one vehicle. This is enforced by requiring that the sum of all incoming arcs to customer *i*, across all vehicles and predecessor nodes, equals one. This equation guarantees complete customer coverage and prevents multiple visits to the same customer.

Equation 3 represents the vehicle capacity limitation. It states that, for each vehicle $k \in V$, the total demand of the customers served by that vehicle cannot exceed its capacity *q*. The customer demand is denoted by d_i, and the summation over $i \in C$ and $j \in N$ accounts for all customers assigned to vehicle *k*.

Equation 4 ensures that each vehicle departs from the depot exactly once. The variable x_{0jk} represents the arc from the depot node *0* to the first customer *j*, and this equation guarantees that every vehicle starts its route from the depot.

Equation Eq. 5 enforces flow conservation at each customer node $h \in C$ for every vehicle $k \in V$. It requires that the number of arcs entering a customer node equals the number of arcs leaving that node for the same vehicle. This equation maintains route continuity and prevents disconnected or incomplete routes.

Equation 6 ensures that each vehicle returns to the depot at the end of its route. The node $n + 1$ represents the depot return node, and this equation enforces that exactly one arc enters the depot return node for each vehicle *k*.

Equation 7 incorporates the temporal feasibility of the routes by linking service start times with travel durations. The variable s_{ik} denotes the service start time of vehicle *k* at node *i*, while t_{ij} represents the travel time from node *i* to node *j*. The constant *K* is a sufficiently large positive number used in the Big-M formulation to deactivate the equation when $x_{ijk} = 0$. This equation ensures that service at node *j* cannot start before vehicle *k* arrives from node *i*, provided that arc *(i,j)* is used.

Equation 8 enforces the time window restrictions for each node $i \in N$ and each vehicle $k \in V$. The service start time s_{ik} must lie within the predefined time window $[a_i, b_i]$, where a_i and b_i represent the earliest and latest allowable service times at node i, respectively.

Finally, Equation 9 defines the domain of the decision variable x_{ijk}. It specifies that x_{ijk} is a binary variable, taking a value of 1 if vehicle k travels from node i to node j, and 0 otherwise. This equation formally establishes the combinatorial nature of the VRPTW.

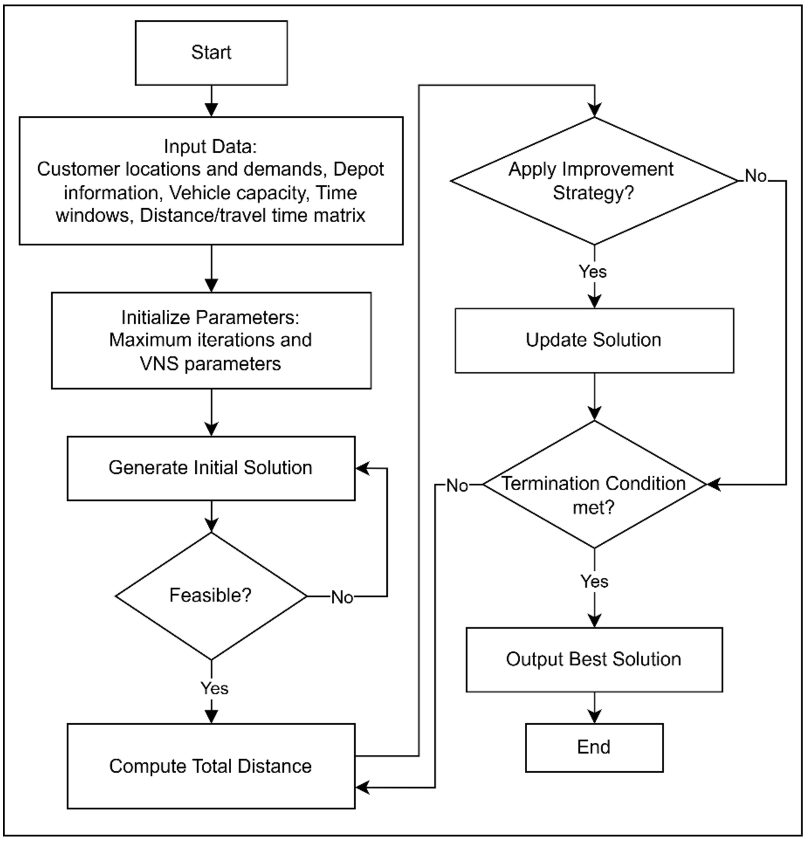

Figure 1. Flowchart of VRPTW

Figure 1 depicts the structured optimization procedure adopted to solve the VRPTW in this study. The process begins with the initialization stage, where all required problem data are provided. These inputs include customer locations and demands, depot information, vehicle capacity, predefined time windows for customer service, and the distance or travel time matrix. Together, these elements define the VRPTW instance and constitute the foundational parameters of the mathematical model and solution framework, as commonly established in previous study [14], [15].

Following data input, the algorithm proceeds to the parameter initialization stage. At this stage, the maximum number of iterations and the parameters specific to the Variable Neighborhood Search (VNS) framework are defined. These parameters regulate the search behavior and play a critical role in balancing solution exploration and exploitation throughout the optimization process. Proper initialization is essential to ensure convergence stability and prevent premature stagnation in metaheuristic-based VRPTW solvers [16].

Subsequently, an initial solution is generated by constructing vehicle routes that assign customers to vehicles while considering capacity and time window constraints. The generated solution is then subjected to a feasibility check. If the solution is infeasible, due to violations such as exceeding vehicle capacity, violating time window constraints, or improper route construction, it is rejected, and the algorithm returns to the initial solution generation stage. This iterative feasibility screening ensures that the search process remains confined to the admissible solution space, which is a standard practice in VRPTW optimization method [17].

Once a feasible solution is obtained, the total travel distance is computed. The total distance is calculated as the cumulative distance traveled by all vehicles across their respective routes. In this study, total travel distance serves as the primary objective value and performance indicator. The adoption of total distance as the evaluation metric is consistent with established VRPTW benchmarking practices, particularly when using Solomon benchmark instances, as it enables objective, transparent, and reproducible comparisons across different algorithms [18], [19].

After evaluating the solution, the algorithm determines whether an improvement strategy should be applied. If no improvement is applied, the algorithm directly proceeds to the termination condition check. Otherwise, an improvement strategy, typically involving neighborhood-based search operators, is executed to refine the current solution. The improved solution is then used to update the current solution state, allowing the algorithm to progressively enhance solution quality through iterative search. The integration of improvement strategies within an iterative framework is widely recognized as an effective approach for improving convergence speed and solution robustness in VRPTW research [20].

Following the solution update, a termination condition check is performed. Common termination criteria include reaching the maximum number of iterations, observing no further improvement in the objective value, or satisfying a predefined computational limit. If the termination condition is not met, the algorithm loops back to the evaluation and improvement stages, continuing the iterative optimization process. This loop ensures continuous refinement of feasible solutions until the stopping criterion is satisfied [21]. Finally, when the termination condition is met, the algorithm outputs the best solution obtained during the entire search process. This solution represents a set of vehicle routes that minimizes the total travel distance while fully satisfying all VRPTW constraints. The procedure then terminates, completing the optimization workflow.

2.2 Variable Neighborhood Search (VNS)

It is a heuristic method that belongs to the metaheuristic category. VNS is designed to handle combinatorial optimization problems, such as Vehicle Routing Problem (VRP), Traveling Salesman Problem (TSP), and the like. According to [22], "VNS is one metaheuristic method that aims to solve combinatorial optimization problems. The basic idea is a systematic neighborhood change combined with local search". VNS combines the idea of local search with the concept of various search environments. This approach tries to overcome the problem of getting stuck on the local solution which is generally faced by the usual local search algorithm. The stages of the VNS Algorithm are explained as follows:

1) Initialization
   a. Select Initial Solution: Start with an initial solution xxx, which could be generated randomly or through a heuristic.
   b. Set Parameters: Define the maximum number of iterations, stopping criteria, and a list of neighborhood structures Nk, where k = 1, 2, …, kmax.
2) Main Loop (while stopping condition not met)
   Set Neighborhood Index k=1: Start with the first neighborhood structure.While $k \leq k_{max}$ do:
   a. Shaking, generate a new solution x′: randomly generate a new solution x′ from the k-th neighborhood of the current solution x. This step is known as "shaking," where the algorithm perturbs the current solution to explore a different part of the search space.
   b. Local Search, apply a local search algorithm starting from the perturbed solution x′ to find a local optimum x″. This step refines the new solution to improve its quality within the given neighborhood.

Acceptance Criterion; if x″ is better than x: If the new solution x″ improves the objective function value compared to the current solution x, then accept x″ as the new current solution x, and reset k=1 to start with the first neighborhood structure again. Else, if no improvement is found, increment the neighborhood index k by 1 to explore the next neighborhood structure. End while, the loop continues until all neighborhood structures have been explored without finding an improved solution, or until a predefined stopping condition is met (e.g., maximum number of iterations).

3) Termination
   The algorithm terminates when the stopping criterion is met (e.g., no improvement is found after exploring all neighborhoods, or a maximum number of iterations is reached).
4) Output
   The best solution found during the search is returned as the final output.

3. RESEARCH RESULTS AND DISCUSSION

The data is presented using coordinate format (X, Y), demand (demand), ready time (Ready Time), due time (Due Time), service time (Service Time) and vehicle capacity: 200. The calculation is done using data from 100 customers.

Table 1. Customers Data

CUSTNO	XCOORD	YCOORD	DEMAND	READYTIME	DUETIME	SERVICETIME	CAPACITY
1	35	35	0	0	230	0	200
2	41	49	10	10	171	10	
3	35	17	7	50	60	10	
4	55	45	13	116	126	10	
5	55	20	19	149	159	10	
6	15	30	26	34	44	10	
7	25	30	3	99	109	10	
8	20	50	5	81	91	10	
9	10	43	9	95	105	10	

CUSTNO	XCOORD	YCOORD	DEMAND	READYTIME	DUETIME	SERVICETIME	CAPACITY
10	55	60	16	97	107	10	
…	…	…	…	…	…	…	
100	20	26	9	83	93	10	

The calculation steps are explained as follows:

1) Determine the Initial Solution
   Suppose we start with a simple initial route: Depot → Customer 1 → Customer 2 → ... → Customer 20 → Depot.
2) Calculate the distance between points
   Calculate the Euclidean distance between the points. and calculate the distance between all pairs of points in the same way.
3) Calculating Visit Time
   Depot to Customer 1:
   a. Arrival time = Distance (2.24)
   b. Start service = Arrival time
   c. Service completion = Arrival time + Service time (1) = 2.24 + 1 = 3.24
   d. Do similar calculations for all points
4) Route adjustment with VNS
   a. Shaking, for example, exchange the position of Customer 1 and Customer 2 in the route. New route: Depot → Customer 2 → Customer 1 → ... → Customer 20 → Depot
   b. New route evaluation, calculate the total distance for the new route and then calculate the total visit time to ensure all time windows are met and vehicle capacity is not exceeded.
   c. If the vehicle capacity is exceeded or the time window is not met, make adjustments such as changing the order of visits and dividing the route into several trips if necessary.

Table 2. Calculation results of VRPTW with VNS Integration

Number of Route	Total Distance	Total Completion Time	Sub Route	Number of Sub Route
Route 1	627.1327113946152	4.519683361053467	[[0, 9, 20, 1, 0], [0, 11, 19, 10, 0], [0, 14, 15, 13, 0], [0, 23, 22, 4, 25, 0], [0, 5, 16, 6, 0], [0, 2, 21, 0], [0, 7, 8, 17, 0], [0, 18, 0], [0, 12, 3, 24, 0]]	9
Route 2	627.1327113946152	4.318361759185791	[[0, 5, 16, 6, 0], [0, 11, 19, 10, 0], [0, 14, 15, 13, 0], [0, 23, 22, 4, 0], [0, 9, 20, 1, 0], [0, 2, 21, 0], [0, 7, 8, 17, 0], [0, 18, 0], [0, 12, 3, 24, 25, 0]]	9
Route 3	627.1327113946152	4.353008031845093	[[0, 5, 16, 6, 0], [0, 11, 19, 10, 0], [0, 14, 15, 13, 0], [0, 23, 22, 4, 0], [0, 9, 20, 1, 0], [0, 2, 21, 0], [0, 7, 8, 17, 0], [0, 18, 0], [0, 12, 3, 24, 25, 0]]	9
Route 4	627.1327113946152	4.796295642852783	[[0, 5, 16, 6, 0], [0, 11, 19, 10, 0], [0, 14, 15, 13, 0], [0, 23, 22, 4, 0], [0, 9, 20, 1, 0], [0, 2, 21, 0], [0, 7, 8, 17, 0], [0, 18, 0], [0, 12, 3, 24, 25, 0]]	9
Route 5	618.3299155462686	6.136072397232056	[[0, 5, 16, 6, 0], [0, 11, 19, 10, 0], [0, 14, 15, 13, 0], [0, 23, 22, 4, 0], [0, 2, 21, 3, 24, 25, 0], [0, 7, 8, 17, 0], [0, 18, 0], [0, 12, 9, 20, 1, 0]]	8

Number of Route	Total Distance	Total Completion Time	Sub Route	Number of Sub Route
Route 6	618.3299155462686	6.741713762283325	[[0, 11, 19, 10, 0], [0, 14, 15, 13, 0], [0, 23, 22, 4, 25, 0], [0, 5, 16, 6, 0], [0, 12, 9, 20, 1, 0], [0, 7, 8, 17, 0], [0, 18, 0], [0, 2, 21, 3, 24, 0]]	8
Route 7	618.3299155462686	5.972128629684448	[[0, 11, 19, 10, 0], [0, 14, 15, 13, 0], [0, 23, 22, 4, 25, 0], [0, 5, 16, 6, 0], [0, 2, 21, 3, 24, 0], [0, 7, 8, 17, 0], [0, 18, 0], [0, 12, 9, 20, 1, 0]]	8
Route 8	627.1327113946152	4.370475769042969	[[0, 9, 20, 1, 0], [0, 11, 19, 10, 0], [0, 14, 15, 13, 0], [0, 23, 22, 4, 25, 0], [0, 5, 16, 6, 0], [0, 2, 21, 0], [0, 7, 8, 17, 0], [0, 18, 0], [0, 12, 3, 24, 0]]	9
Route 9	627.1327113946152	4.132009506225586	[[0, 9, 20, 1, 0], [0, 11, 19, 10, 0], [0, 14, 15, 13, 0], [0, 23, 22, 4, 25, 0], [0, 5, 16, 6, 0], [0, 2, 21, 0], [0, 7, 8, 17, 0], [0, 18, 0], [0, 12, 3, 24, 0]]	9
Route 10	627.1327113946152	4.307651996612549	[[0, 5, 16, 6, 0], [0, 11, 19, 10, 0], [0, 14, 15, 13, 0], [0, 23, 22, 4, 0], [0, 9, 20, 1, 0], [0, 2, 21, 0], [0, 7, 8, 17, 0], [0, 18, 0], [0, 12, 3, 24, 25, 0]]	9

Table 2 presents the calculation results of the VRPTW using the Variable Neighborhood Search (VNS) method, revealing several key insights into the efficiency of the routing process. Notably, the total distances for the routes are relatively consistent, with values such as approximately 627.13 for both Routes 1 and 2, indicating the reliability of the VNS algorithm in producing stable and efficient routes across multiple evaluations. Additionally, the total completion times for these routes (e.g., 4.52 for Route 1 and 4.32 for Route 2) suggest that shorter distances correlated with reduced service times, reflecting an effective optimization strategy that balances time constraints with route efficiency.

The analysis of sub-routes is also critical, as each main tour comprises multiple sub-routes, with both Tours 1 and 2 consisting of nine sub-routes. This division indicates a well-structured approach to route management, facilitating balanced workloads among vehicles while adhering to customer demands and time windows. The consistency observed in the number of sub-routes emphasizes the algorithm’s capability to partition routes effectively.

Furthermore, performance metrics derived from the table indicate an overall improvement in route efficiency, with mean distances suggesting enhanced logistical performance. The standard deviation (Std) values, which are relatively low, point to the algorithm's ability to maintain a narrow variability in route outcomes, essential for logistics operations that require effective planning. Overall, the results from Table 2 illustrate the VNS method's strengths in optimizing vehicle routes, demonstrating its applicability in real-world logistics scenarios where timely delivery and efficiency are paramount. Parameters and Values of VRPTW Calculation Results from 100 customers are shown in Table 2.

Table 3. Parameters and Values of VRPTW Calculation Results

Parameter	Value
Length routes	10
Length Sub routes VNS	9
Min Sub route	8
Mean Sub route	8.7
Std Sub route	0.45825756949558394

Parameter	Value
Min Distance	618.3299155462686
Mean Distance	624.4918726401113
Std Distance	4.033947830229126
Mean Run Time	4.9647400856018065
Total Distance	888.6931100011278

Table 3 presents the parameters and values associated with the VRPTW calculations, providing critical insights into the performance of the Variable Neighborhood Search (VNS) methodology. The results indicate a total distance of approximately 888.69 for the optimization problem, which serves as a benchmark for comparison against the distances achieved using the VNS approach. The average distance traveled with the VNS method drops significantly to 624.49, highlighting an impressive mileage efficiency of 29.73% when utilizing the algorithm. This improvement underscores the effectiveness of VNS in optimizing routes, resulting in substantial cost savings for logistics operations.

Moreover, the analysis of sub-routes reveals a mean of 8.7 with a standard deviation of 0.46, indicating a well-distributed workload among vehicles with minor variations in the number of sub-routes utilized. The minimum number of sub-routes recorded at 8 suggests that the algorithm efficiently clusters deliveries, which is critical for ensuring timely service without overwhelming individual vehicles.

The findings also note a minimum distance of approximately 618.33 within the routes, suggesting that even under optimal conditions, vehicles can achieve shortened routes, thereby maximizing efficiency. The average run time of about 4.96 seconds illustrates the VNS method's practicality, indicating that it can deliver optimal or near-optimal solutions within a reasonable timeframe, making it suitable for real-time logistics applications.

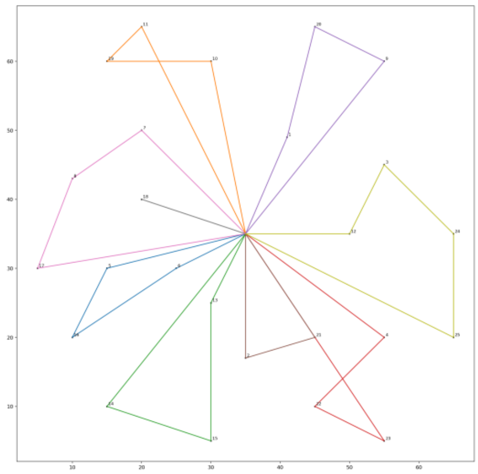

Figure 2. Optimal route results by using VNS

Figure 2 displays the optimal route results achieved through the application of the VNS algorithm for the VRPTW solution. The visual representation of the routes illustrates several key points regarding the optimization outcomes.

The configuration of the routes shows a systematic arrangement that efficiently connects the depot to various customer locations, ensuring that each customer is serviced while adhering to their specified time windows. This organization is crucial in a logistics context, where timely delivery is often a challenging aspect of operations. The VNS algorithm appears to have effectively minimized the total distance traveled while ensuring compliance with the constraints imposed by customer availability.

The routes depicted underscore the algorithm's ability to manage complex logistics scenarios. The distinct paths for multiple vehicles indicate that the VNS can effectively coordinate multiple deliveries without overlap, thereby maximizing vehicle utilization. Each route reflects a careful consideration of proximity to minimize travel distances and times, contributing to the overall mileage savings observed in the study.

Furthermore, the optimal route results imply that the VNS has successfully navigated the inherent challenges of the VRPTW, such as ensuring that no vehicle exceeds capacity while also respecting the time constraints for each customer's delivery. The visual feedback from Figure 2 likely helps logistics managers validate the selections made by the algorithm, offering a clear perspective on how the routes optimize logistical efficiency.

4. CONCLUSION

Based on the results of VRPTW calculations for 100 customers using the VNS method, the total average distance traveled was 624.49. The total distance traveled without using the VNS method is 888.69, so the difference is 264.2 or 29.73%. This shows that the mileage efficiency is 29.73% by using the VNS method. The results show that VNS can overcome the challenges in finding the optimal route for vehicles that serve customers by considering time constraints and vehicle capacity.

5. REFERENCES

[1] E. Nugraha and R. M. Sari, “Analisis Defect dengan Metode Fault Tree Analysis dan Failure Mode Effect Analysis,” *Organum J. Saintifik Manaj. dan Akunt.*, vol. 2, no. 2, 2019, doi: 10.35138/organum.v2i2.58.

[2] M. Schneider, A. Stenger, and D. Goeke, “The electric vehicle-routing problem with time windows and recharging stations,” *Transp. Sci.*, vol. 48, no. 4, 2014, doi: 10.1287/trsc.2013.0490.

[3] R. Baldacci, M. Battarra, and D. Vigo, “Routing a heterogeneous fleet of vehicles,” 2008. doi: 10.1007/978-0-387-77778-8_1.

[4] V. Leggieri and M. Haouari, “A matheuristic for the asymmetric capacitated vehicle routing problem,” *Discret. Appl. Math.*, vol. 234, 2018, doi: 10.1016/j.dam.2016.03.019.

[5] B. Afshar-Nadjafi and A. Afshar-Nadjafi, “A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet,” *J. King Saud Univ. - Eng. Sci.*, vol. 29, no. 1, 2017, doi: 10.1016/j.jksues.2014.04.007.

[6] L. Mingyong and C. Erbao, “An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows,” *Eng. Appl. Artif. Intell.*, vol. 23, no. 2, 2010, doi: 10.1016/j.engappai.2009.09.001.

[7] Z. Zhang, L. Wei, and A. Lim, "An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints," *Transp. Res. Part B Methodol.*, vol. 82, 2015, doi: 10.1016/j.trb.2015.10.001.

[8] A. K. Garside and D. N. Cahyanti, "Penyelesaian Vehicle Routing Problem with Simultaneous Pick Up and Delivery dengan Algoritma Tabu Search," *J. Ilm. Tek. Ind.*, vol. 17, no. 2, 2018, doi: 10.23917/jiti.v17i2.6703.

[9] N. A. El-Sherbeny, "Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods," *J. King Saud Univ. - Sci.*, vol. 22, no. 3, 2010, doi: 10.1016/j.jksus.2010.03.002.

[10] A. W. Kolen, A. H. G. Rinnooy Kan, and H. W. Trienekens, "Vehicle routing with time windows," *Oper. Res.*, vol. 35, no. 2, pp. 277–273, 1987.

[11] J. Pearl, *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc., 1984.

[12] X. S. Yang, *Nature-Inspired Optimization Algorithms*. 2014. doi: 10.1016/C2013-0-01368-0.

[13] J. Y. Potvin, T. Kervahut, B. L. Garcia, and J. M. Rousseau, "The vehicle routing problem with time windows part I: Tabu search," *INFORMS J. Comput.*, vol. 8, no. 2, 1996, doi: 10.1287/ijoc.8.2.158.

[14] T. Vidal, G. Laporte, and P. Matl, "A concise guide to existing and emerging vehicle routing problem variants," 2020. doi: 10.1016/j.ejor.2019.10.010.

[15] S. Tan and W. Yeh, "The Vehicle Routing Problem : State-of-the-Art Classification and Review," *Appl. Sci.*, vol. 11, 2021.

[16] A. G. Neogi, S. Mounika, S. Kalyani, and S. A. Yogananda Sai, "A comprehensive study of vehicle routing problem with time windows using ant colony optimization techniques," *Int. J. Eng. Technol.*, vol. 7, no. 2, 2018, doi: 10.14419/ijet.v7i2.32.13532.

[17] C. Chen, E. Demir, and Y. Huang, "An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots," *Eur. J. Oper. Res.*, vol. 294, no. 3, pp. 1164–1180, 2021, doi: 10.1016/j.ejor.2021.02.027.

[18] B. Boyacı, T. H. Dang, and A. N. Letchford, "Improving a constructive heuristic for the general routing problem," *Networks*, vol. 81, no. 1, 2023, doi: 10.1002/net.22119.

[19] M. J. C. S. Reis, "An Adaptive Hybrid Metaheuristic for Solving the Vehicle Routing Problem with Time Windows under Uncertainty," *Comput. Mater. Contin.*, vol. 85, no. 2, 2025, doi: 10.32604/cmc.2025.066390.

[20] G. Chen, J. Gao, and D. Chen, "Research on Vehicle Routing Problem with Time Windows Based on Improved Genetic Algorithm and Ant Colony Algorithm," *Electron.*, vol. 14, no. 4, 2025, doi: 10.3390/electronics14040647.

[21] X. Liu, Y. L. Chen, L. Y. Por, and C. S. Ku, "A Systematic Literature Review of Vehicle Routing Problems with Time Windows," *Sustain.*, vol. 15, no. 15, 2023, doi: 10.3390/su151512004.

[22] P. Hansen and N. Mladenović, "Variable neighborhood search," in *Handbook of Heuristics*, vol. 1–2, Springer International Publishing, 2018, pp. 759–787. doi: 10.1007/978-3-319-07124-4_19.