Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
 Volume 24, Issues 3, pp. 241–255, DOI 10.12694/scpe.v24i3.2372

TIME WINDOW ORIENTED IOT VEHICLE PATHWAY STUDY FOR THE DYNAMICALLY CHANGING NEEDS OF E-COMMERCE CUSTOMERS

XIN CHEN*

Abstract. The main dynamic truck routing problem also presents a significant difficulty in the logistics sector, which is an unavoidable development trend of the contemporary technological changing society. A dynamic vehicle routing problem with time window model is suggested by the study in order to establish an effective and low-energy dynamic response method. The fundamental concept is to disrupt the conventional strategy of static dynamic consumers responding in time slots by dividing the dynamic time window into a static time window with several time slice intervals. The study makes use of cutting-edge ideas including dynamic attitude, before-and-after time slicing, and continuous optimisation while proposing a new method for model solution to optimise dynamic vehicle route issues effectively and affordably. The study employs the Solomon optimisation dataset and runs simulation studies on the Java platform to confirm its efficacy. The experimental findings demonstrated that the optimisation technique employed in the study reduced the cost of travelling by 83.8 miles while also considerably increasing the average vehicle utilisation by 3.6%. Because driving distance cost and vehicle number cost are typically positively connected with dynamic attitude, the study employs solutions that can increase dynamic response efficiency and save money. As a result, their robustness is higher.

Key words: Internet of Things; path optimization; time windows; dynamic demand; cost

1. Introduction. All businesses are being moved towards intelligence by technological advancements, particularly by the emergence and growth of the Internet of Things (IoT) and intelligent algorithms, which are creating new prospects for the logistics sector. With the continuous development of science and technology, artificial intelligence has been elected as one of the hottest industries in recent years and is widely used in various industries. The field of intelligent vehicles has also introduced this advanced technology. Such transformation and upgrading plays an important role in national economic development and can greatly reduce the costs of vehicle driving. Many directions of its field have become research trends. In addition to automated driving, smart automobiles also effectively combine logistics and transport, which frequently leads to phenomena like bad customer experience because of fatal flaws like slow response in the traditional logistics industry, from the technical layer, industrial layer and application layer three aspects of the optimization industry, in the technical layer, will gradually change from manual to automation, to solve the problem of labor cost; At the industrial level, the transportation industry has become more network intelligent and performs better in route planning and other aspects. At the application level, transport vehicles are transformed into mobile intelligent terminals to provide convenience for people's life and work [2]. It is especially crucial to enhance the optimisation of dynamic vehicle path difficulties since task orders from IoT clients frequently alter in real time and with uncertainty in some medium and heavy transport operations. Vehicle scheduling is the process of determining the optimal route to take in order to satisfy various client needs and keep driving expenses, for example, within tolerable bounds [11]. Vehicle path optimisation (PO) systems with temporal windows (TW) are much more complicated. When it comes to the customer's dynamic demand (DD), the vehicle must first and foremost satisfy the customer's wants while attempting to design the optimum route to minimise driving expenses, etc., while still finishing the assignment on schedule [1]. Innovations in the logistics sector depend on research into putting forth a fresh set of optimisation models.

The first section of this research introduces the vehicle scheduling research application; the second section describes the construction process of this initial model and introduces the forbidden search algorithm as the model solving algorithm; the third section uses the Solomon dataset and conducts parameter tuning experiments

*Information Engineering College, Xinxiang Vocational and Technical College, Xinxiang, 453000, China (chenx963214780@163.com)

on the forbidden algorithm to visualise the customer distribution; and finally, the novel model is compared with the traditional model. Finally, Section 4 presents a thorough analysis of the experimental data from the study and draws the conclusion that the Dynamic Vehicle Routing Problem with Time Window (DVRPTW) model paired with the study's forbidding algorithm can be more effectively used to vehicle PO.

2. Related works. Adapting vehicle pathways to the DD of IoT clients is the key to attaining logistics vehicle scheduling in the logistics sector. To further improve the algorithm's learning capacity and combine user activity paths to produce optimal vehicle paths, Dai X et al. proposed a process model based on vehicle dynamic routes and user scenarios in order to realize the convenience of shared public transportation and reduce the burden on urban roads. According to the bus data set of multiple routes, the travel demand of the last kilometer is predicted, and the dynamic path is planned, and the shared public transportation is dynamically routing and scenario-based operation. Five prediction functions including time and location are used to further improve the learning ability of the algorithm, and the generation of the optimal vehicle path is realized by combining the user activity path [3]. According to Merka Z and colleagues, blockchain technology has a significant impact on the logistics sector, and the addition of IoT can make the entire logistics process more transparent, traceable, and productive. Supply chain refers to the whole process of producing parts, making intermediate products and final products, and finally sending them to consumers, involving suppliers, manufacturers, distributors and consumers through the connection of upstream and downstream members of the network chain structure. Supply chain management can minimize the entire system cost. They combined a number of cases to show the effectiveness of the current use of blockchain technology in a number of logistics industries, and they then proposed a method for building models to address the development limitations of the technology [8]. Lin R et al. proposed a logistics robot based on laser navigation system. The model consists of two sub-machines, which respectively contain two units driving the front and back, and both move in the same direction. The front-mounted laser is used for obstacle detection, and the on-board navigation system is used for positioning, while the rear laser is used for pallet picking [7]. Lasers at the front and rear ends are utilised for obstacle detection and logistics path planning, respectively. Wang X et al. proposed a vehicle PO model based on ant colony algorithm, introduced adaptive mechanism to tune the parameters, solved the shortcomings of the classical ant colony algorithm such as too fast convergence and too slow convergence, and achieved the effect of lowest cost as well as optimal path, and finally the study used data from two logistics enterprises in Huainan to verify the effectiveness of the heuristic intelligence algorithm.

The vehicle PO problem, which is one of the main challenges the system is currently facing, takes the maximisation of platform profit as the goal, while taking into account the driver's shift rest and other issues. Guo J et al. argued that the cross-city carpooling system needs to be further optimised, and the study proposes a VRP model that uses the sequential path construction method as the initial value solution path while introducing the variable neighbourhood search. The model was solved, and experiments were used to confirm its efficacy [4]. Himmich et al. investigated a novel primitive column construction framework applied to the vehicle scheduling problem and embedded it into a column generation scheme to solve the subproblem, and experimentally demonstrated that the method's solution falsity was reduced by a factor of seven in comparison to the conventional method by abandoning primitive algorithms like adjacency [5]. Vehicle scheduling was regarded by Li Y et al [6]. as an essential stage in managing public transit. In order to address the issue of randomness in vehicle assignment, the study applied the discrete artificial bee colony algorithm to the vehicle scheduling problem. It also used small differential coding and decoding to address the discretization defect and introduced three neighbourhood search schemes, such as initialization rules and heuristics, including discrete scheme, heuristic scheme, and learnable scheme to achieve the best vehicle scheduling optimisation results [5]. In order to find the best solution to the vehicle path problem, Raeesi R et al. proposed an optimisation of an electric vehicle path problem with TW and mobile battery swapping [12]. The study combined dynamic programming and integer programming algorithms.

Dynamic reaction models clearly play a crucial role in the logistics of vehicle transportation. This work proposes a dynamic vehicle PO model based on a taboo search algorithm that interprets the problem as a set of time-sliced intervals for a static vehicle problem and enhances performance, including cost.

3. One-stage DD-based Vehicle Path Study with TW. It is inevitable when customer demand changes dynamically in the process of logistics transportation. However, there are few researches on dynamic

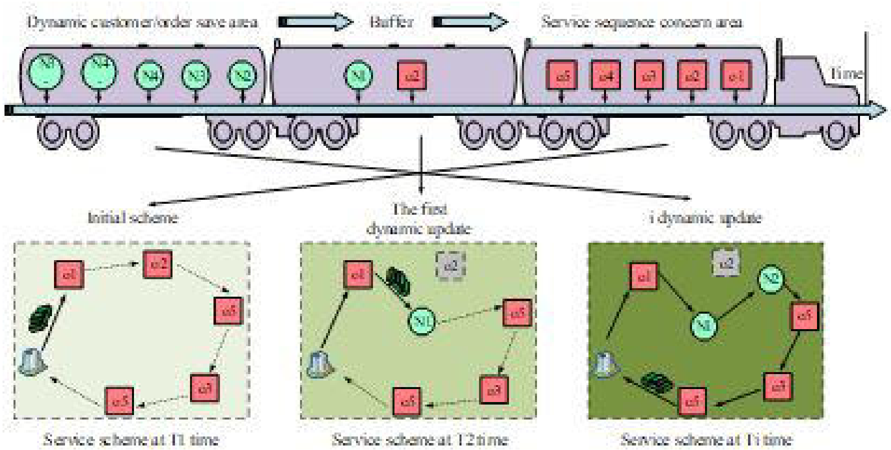

Fig. 3.1: Schematic diagram of DVRPTW

demand response at present, and most logistics still maintain the traditional fixed path planning scheme, which will greatly reduce the efficiency of vehicle transportation and increase its cost. Moreover, with the increasing number of logistics employees, it is more necessary for the model to realize the reasonable division of staff and work area. However, due to the randomness of dynamic requirements and the diversity of application scenarios, it is usually difficult to construct dynamic models, so a systematic construction system is needed.The PO of a vehicle is the key to achieving efficient work in logistics, which in practical application scenarios is usually reflected in a dynamic response process, requiring a dynamic vehicle path problem model with TW.

3.1. Vehicle Path Modeling for IoT Customer DD. In the static vehicle scheduling model, the time window includes hard time window and soft time window. The soft time window uses the penalty function for vehicle path planning. This method not only cannot obtain the best path, but also relies on the selection of penalty factors to a great extent, which is highly subjective. The hard time window provides the customer access of the specified time window, which can get the shortest path and the observability is stronger. Therefore, the model chooses a hard time window. In the dynamic vehicle scheduling model, the constraint on the time window is usually abandoned for the sake of timeliness, but this will also cause lateness and reduce customer satisfaction. Therefore, the concept of continuous time slice and equivalent time window transformation is introduced to optimize the time window. At the same time, dynamic attitude is introduced to simplify and improve the model. There is a certain relationship between dynamic attitude and time slice. The direction of commodity flow reflects the chain in which the logistics transport is located, including the three steps of the front and middle end. When it is at the front and middle end of the process, the vehicle needs to meet the requirements of the IoT customer, collect the goods in order and subsequently deliver them uniformly to the collection point. Online shopping is a common IoT model, individual users (IoT users) use smart devices to make online orders, followed by cooperation between merchants and logistics companies to achieve product pickup and transportation, in the process, the product transportation status will change in real time according to the DD of IoT users, which is the dynamic vehicle scheduling (Dynamic vehicle scheduling Problem (DVSP) process [14]. The DVRPTW is shown in Figure 3.1.

IoT users are volatile in terms of when and where they place their orders, and therefore need to meet higher standards in terms of specific services TW. The study establishes the DVRPTW model with the aim of accomplishing more efficient IoT platform services at a smaller cost. As can be seen from figure 3.1, vehicles are subject to the constraints of the undirected connectivity network graph, denoted by $G($Node, Edge$)$, which means that the correct edge $Edge$ of the full user $Node$ is selected. the dynamic vehicle path problem with TW is characterised by the inclusion of the length of the working day in the calculation process, where those who

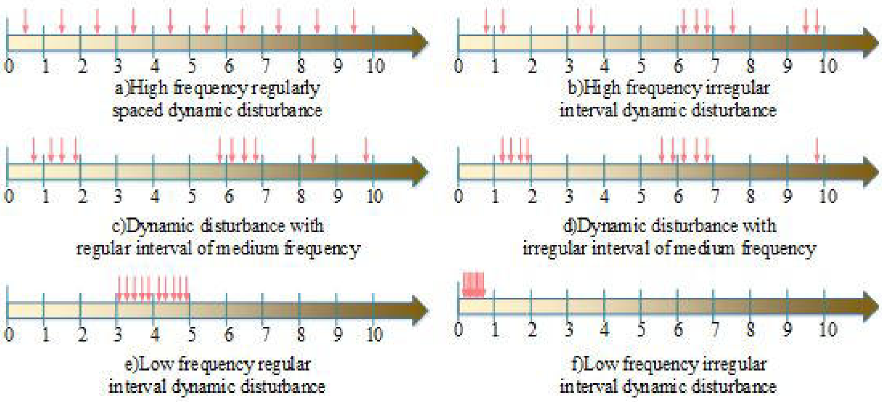

Fig. 3.2: Different dynamic event disturbance scenarios

place an order before 1/2 working day belong to dynamic users, and vice versa, they are classified as static users on the next working day, which working day with priority processing rights [13]. The upfront improvement of the DVRPTW is similar to the static problem, with the expression of the primary and secondary optimisation objective function shown in Equation 3.1.

$$\begin{cases} Primary.objective = \min \sum \sum \sum x_{[i],[j],[k]} \\ Secondary.objective = \min \sum \sum \sum d_{[i],[j]} x_{[i],[j],[k]} \end{cases} \tag{3.1}$$

In Equation 3.1, $d_{[i][j]}$ is the Euclidean distance between the random user i and j ; $x_{[i][j][k]}$ represents the (0, 1) decision variable of whether the vehicle passes through two customers. The two parameters are calculated as shown in Equation 3.2.

$$\begin{cases} d_{[i][j]} = \sqrt{(i_x - j_x)^2 + (i_x - j_x)^2} \\ x_{[i][j][k]} = \begin{cases} 1, \text{if } i - j \text{ are connected} \\ 0, \text{if } i - j \text{ aren't connected} \end{cases} \end{cases} \tag{3.2}$$

When an IoT user places an order before half a working day, it is already part of the pre-logistics phase, when a combination of dynamic and static customer status is required for real-time vehicle logistics updates. Assuming the vehicle state is on the way to the next user, the next logistics task will not be available until after that task state has been completed. And it needs to return to the initial point after all tasks have been completed, while the loading volume needs to meet the requirement of less than the maximum amount [16]. The difficulty of running the model is related to the value of its dynamic attitude, which refers to the ratio of dynamic users to the total number of users in a fixed time horizon, as shown in Equation 3.3.

$$Dyn = \frac{N_c^d}{N_c^d + N_c^s} \tag{3.3}$$

In the Equation 3.3, N_c^d is the total number of dynamic users; N_c^s is the total number of static quantities. The equation is simple and easy to calculate, but when the amount of time and the dynamic/static number of users are all 10, there are six different forms of its dynamic attitude, and the performance can produce limitations, as shown in Figure 3.2.

As can be seen from Figure 3.2, the neighbourhood search first generates an initial solution, followed by the selection of a neighbourhood range, or local search range, according to the definition. The neighbourhood solutions are compared one by one by evaluation, and the set of filtered neighbourhood solutions is shifted, and finally the local optimum is recorded, and by cycling through the above steps, the global optimum is eventually ranked. The amnesty rule can be used to solve the problem of falling into a local optimum, when a candidate value has violated the taboo rule, it needs to be released by this principle, and the release principle enables the restricted element to obtain a more optimal value. Among them, the indicators of the taboo table are mainly the taboo object and the taboo length, and the taboo length refers to the number of iterative failures. ts may obtain both correct and infeasible solutions throughout the search process, the infeasible solution has a positive effect on the global optimal search, so it can bring great convenience to the algorithm operation [10]. Therefore, the study introduces an adaptive penalty function and proposes an adaptive adjustment parameter as an optimisation. The adaptive penalty function is calculated as shown in Equation 3.4.

$$Dyn = \frac{N_c^d + DN(T/2n_{ts})}{N_c^d + N_c^s + N(T/2n_{ts})} \tag{3.4}$$

In Equation 3.4, $N(T/2n_{ts})$ is the amount of time slices in the full process; $DN(T/2n_{ts})$ is the amount of time slices in the request made by the dynamic user. This equation incorporates all the factors that will have an impact on the performance of the dynamic attitude. To improve the dynamic scheduling process of the vehicle path, multiple time slices of are set for the total duration of half a working day, and the initial operating moment is set to . The expression for half a working cycle is shown in Equation 3.5.

$$T_{\lim} = [(t_0, t_0 + n_{ts}), (t_0 + n_{ts}, t_0 + 2n_{ts}), ..., (t_0 + nn_{ts}, T/2)] \tag{3.5}$$

In Equation 3.5, $T_{\lim}$ denotes half a working day. It can be seen that the time slices are not discrete. According to the setting of the continuous interval, the study defaults that dynamic customers in each time slot are interpolated to the corresponding vehicle paths, and eventually real time updates of the paths are achieved. The model responds to the dynamic customer problem when it determines that the vehicle can achieve the user's time requirements, and the determination rules are shown in Equation 3.6.

$$f(t) = \begin{cases} 0, \ b_i < t_0 + in_{ts} \\ 1, \ b_i > t_0 + in_{ts} \end{cases} \tag{3.6}$$

The dynamic user demand TW is known to be denoted as $[a_i, b_i]$. In the equation 3.6, when b_i satisfies the condition greater than $t_0 + in_{ts}$, then the system is able to demand the corresponding user demand. Conversely, the vehicle system will reject the corresponding user demand. When the vehicle successfully corresponding demand, the fastest time for the dynamic user to complete the service is $t_0 + in_{ts}$. The updated TW expression is shown in Equation 3.7.

$$a_{i\_r} = \begin{cases} t_0 + in_{ts}, \ a_i < t_0 + in_{ts}, b_i > t_0 \\ a_i, \ a_i > t_0 + in_{ts}, b_i > t_0 \end{cases} \tag{3.7}$$

To ensure that the vehicle path can be as smooth as possible to facilitate vehicle travel and to further save user time, the study introduces an equivalent TW transformation concept that uses constraints to update the user task completion status as shown in Equation 3.8 [15].

$$[a_i, b_i] = \begin{cases} a_i = \mathrm{INT}(t_{ik\_a}) \\ b_i = \mathrm{INT}(t_{ik\_a}) + 1 \end{cases} \tag{3.8}$$

In the Equation 3.8, t_{ik_a} denotes the real moment when the vehicle numbered k completes the user i; INT denotes the rectification function. At any time on the node, when the first time need to serve the customer, will use the new vehicle, then the task completion time is the moment of arrival of the new vehicle. In the workflow, maintaining the customer's time rights comes first, and equation 3.8 allows for the optimal selection of paths. When the vehicle path is subject to this restriction, then the vehicle path is said to be unique. When the path to task completion is broken, then path planning violates the TW constraint, and conversely, the vehicle is able to complete the task successfully and on time. The study transforms the dynamic vehicle path problem into a static vehicle path problem for multiple short time intervals by means of dimensionality reduction at the time level. The basic idea is to refine the path at the next moment in the context of the previous time slice under the condition of minimum driving cost, and to keep cycling through the process until the end of the current working day, which is expressed as shown in Equation 3.9.

$$\begin{cases} \min F(y+[s+1]|y_{[s]}), s=1,2,....,t_{ns}+1 \\ y_{[s]} = \min \sum\sum\sum d_{[i\_s][j\_s]} x_{[i\_s][j\_s][k\_s]} \end{cases} \tag{3.9}$$

In equation 3.9, s denotes the time slice; $F(y+[s+1]|y_{[s]})$ denotes the driving cost. This equation is the objective function of the DVRPTW model. In the DVRPTW model, the driving path constraint of the vehicle is shown in equation 3.10.

$$\begin{cases} \sum_{k\in K}\sum_{j\in\Delta*(i)} x_{[i\_s][j\_s][k\_s]} = 1,\ \forall i\in N \\ \sum_{j\_s\in\Delta^+(1)} x_{[1][j\_s][k\_s]} = 1, \forall k\_s \in K \\ \sum_{i\_s\in\Delta^-(j-s)} x_{[i\_s][j\_s][k\_s]} - \sum_{i\_s\in\Delta^+(j-s)} x_{[i\_s][j\_s][k\_s]} = 0,\ \forall k\_s\in K \forall j\_s \in J \\ \sum_{i\_s\in\Delta^-(n+1} x_{[i\_s][n+1\_s][k\_s]} = 1, \forall k\_s\in K \end{cases} \tag{3.10}$$

In equation 3.10, this represents the constraint that the customer can only be served once; the vehicle departure point constraint; the vehicle departure place constraint; and the vehicle return constraint, i.e., the need to revert to the initial departure point with expression . The constraint on vehicle loading is calculated as shown in Equation 3.11.

$$\sum_{i_s\in N} d_{[i_s]} \sum_{j_s\in\Delta*(i)} x_{[i_s][j_s][k_s]} \le C, \quad \forall k_s \in K \tag{3.11}$$

As there is a rated load requirement for the vehicle, the DVRPTW model needs to constrain the maximum loading for each time slice, i.e., its load cannot be greater than the rated load size.

3.2. Improvement of Tabu Search Algorithm Based on Dynamic PO and Other Policies. The study introduces the Tabu Search (TS) algorithm, an algorithm that incorporates the concepts of taboo tables and domain search. It sets the storage rules as agile taboo criteria, and when the computation is constrained by the taboo length as well as the amnesty rules, the search process can be carried out more efficiently and accurately, resulting in the best value for the global search. TS is widely used in similar fields such as VRPTW, where neighbourhood search is an extremely important part of the algorithm. The underlying idea is that an iterative process searches the neighbourhood of the current value and eventually finds the optimal solution. When the neighbourhood range is too large, it will increase the burden of running the algorithm and cost more time, but at the same time, the local best value will be obtained [9]. The most common current strategies are single-path based internal search ($Or-opt, 2-opt$) and dual-path ($2-opt^*, swap/shift$) based search, respectively, for a total of four neighbourhood search strategies, expressed as shown in Equation 3.12 [18].

$$S_1 \to Selection\ function \begin{bmatrix} Or-opt \\ 2-opt \\ 2-opt^* \\ Swap/shift \end{bmatrix} \to S_2 \tag{3.12}$$

In the equation 3.12, S_1 denotes the initial solution of the vehicle route. denotes the final solution of the neighbourhood search. Where, AAA is constructed as shown in equation 3.13.

$$S_1 = \{0, c_1, c_2, c_3, c_4, ..., c_n, n+1\} \tag{3.13}$$

In the equation 3.13, c_n denotes the number of iterations for n time slices. Firstly the vehicle must meet the rated capacity as well as the TW condition, then nodes of completed users and newly added customers are

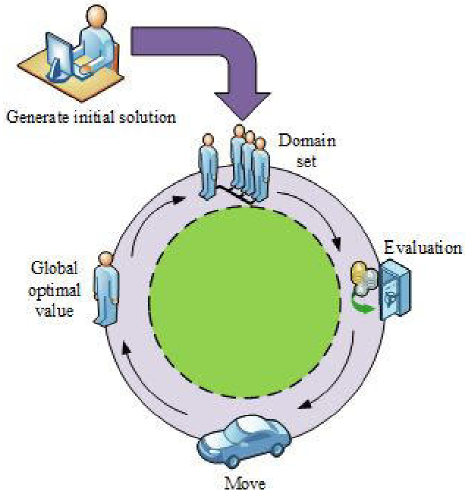

Fig. 3.3: Flow chart of neighborhood search strategy

arbitrarily selected and the former is placed to any node to achieve dynamic update. The neighbourhood search process is shown in Figure 3.3.

As can be seen from Figure 3.3, the neighbourhood search first generates an initial solution, followed by the selection of a neighbourhood range, or local search range, according to the definition. The neighbourhood solutions are compared one by one by evaluation, and the set of filtered neighbourhood solutions is shifted, and finally the local optimum is recorded, and by cycling through the above steps, the global optimum is eventually ranked. The amnesty rule can be used to solve the problem of falling into a local optimum, when a candidate value has violated the taboo rule, it needs to be released by this principle, and the release principle enables the restricted element to obtain a more optimal value. Among them, the indicators of the taboo table are mainly the taboo object and the taboo length, and the taboo length refers to the number of iterative failures. ts may obtain both correct and infeasible solutions throughout the search process, the infeasible solution has a positive effect on the global optimal search, so it can bring great convenience to the algorithm operation [10]. Therefore, the study introduces an adaptive penalty function and proposes an adaptive adjustment parameter as an optimisation. The adaptive penalty function is calculated as shown in Equation 3.14.

$$f(x) = y_{[s]} + \alpha Q_{over} + \beta T_{over} \tag{3.14}$$

In equation 3.14, the adaptive penalty function includes three cost indicators for distance, overload and violation time, with the latter two cost parameters, α and β , relating to the adaptive regulation parameters as shown in equation 3.15 [17].

$$\begin{cases} \alpha \begin{cases} \alpha/(1+\theta), \text{if } Q_{over} = 0 \& \alpha \geq \alpha_1 \\ \alpha \cdot (1+\theta), \text{if } Q_{over} = 0 \& \alpha \leq \alpha_1 \end{cases} \\ \beta \begin{cases} \beta/(1+\theta), \text{if } T_{over} = 0 \& \beta \geq \beta_1 \\ \beta \cdot (1+\theta), \text{if } T_{over} = 0 \& \beta \leq \beta_1 \end{cases} \end{cases} \tag{3.15}$$

In equation 3.15, α and β correspond to the vehicle load as well as the time cost indicators of the adaptive penalty function, respectively. It can be seen that α varies with the total user demand, and the process is controlled by the adaptive regulation parameter.When the total demand of the user is greater than the

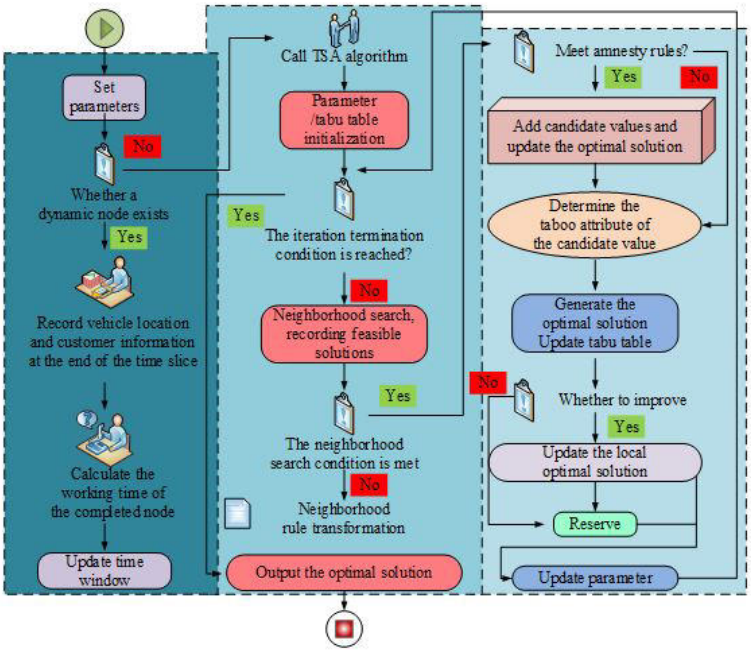

Fig. 3.4: Flowchart of Tabu algorithm implementation

maximum load of a vehicle, the adaptive parameters will be appropriately increased, so that the value of the fitness function can be increased; In the opposite case, an appropriately reduced value is needed to achieve a reduction in the fitness function.Moreover, if a vehicle completes its task overtime, β will also be adjusted accordingly by parameter θ, realising the automatic adjustment of the adaptation function.That is, when the vehicle does not arrive at the customer point on time, the adaptive parameter will increase it to achieve the improvement of the fitness function, otherwise the condition will automatically be lower[20]. In summary, the flow chart of the TS algorithm is shown in Figure 3.4.

The algorithm needs to ensure the minimum damage degree of the traversed path, which is realized by the adaptive penalty function. The concept of equivalent time conversion is used to achieve the purpose of partial path validity. For example, if a customer's time slice has been completed, a time window constraint is required to complete the correction. The generation of the initial solution is also assisted by the time window constraint. Customers meeting the constraint are randomly selected and inserted into the corresponding vehicle chain. The process is repeated continuously. When the number of customers reaches the capacity constraint, the next vehicle chain can be added. Among them, the adaptive penalty function optimizes the fitness value by adjusting its adaptive parameters, and then realizes the cost control. Initially, the vehicle traverses all customers at random time slices with the path planning direction, entering completed users at the end of the time interval; then, the static and dynamic users recorded by the system generate a new undirected connectivity network graph; finally, the system introduces TS to solve the VRPTW problem, and based on the optimal solution result, the path is updated in real time and provided to each vehicle.

4. Experiments on the Performance of a TS-based System for Optimizing Dynamic Vehicle Path Problems for IoT Users with TW. As the dynamic vehicle scheduling model adopted in this study has time window constraints, the performance simulation experiments of this model mostly adopt Solomon optimization data set, because the data set introduces time slice rules, and each time slice is equivalent to static vehicle path optimization, which is more suitable for the study of practical models. Therefore the Solomon optimisation dataset was selected as the basis for the study, and the optimisation scheme was specified as a random allocation of the next 50 users in the form of $max(0, e_i - \theta dis_{oi} - r)$ if the first 50 users all appeared at moment 0. The experimental environment parameters, i.e., the basic characteristic information of each data in the dataset, are shown in Table 4.1.

Table 4.1: Experimental environment parameters and data set characteristics

Content	Parameter
Computer model	MacBook Air 13.3 Core i5
CPU	1.8GHz
Memory	8G
SSD	512G
System	Windows 10 64bit
Programming environment	Java JDK-8u251
	R1 R2 C1 C2 RC1 RC2
Number	12 11 9 8 8 8
Working day length	230 1000 1235 3390 240 960
Customer service hours	10 10 90 90 10 10
Total demand	1457 1457 1810 1810 1725 1725
Vehicle capacity	200 1000 200 700 200 1000
Theoretical minimum vehicle	7.28 1.47 8.63 2.58 8.63 1.73
Mean time window	86.97 453.75 85.42 920.52 85.44 369.77
Mean EDD	0.13 0.13 0.17 0.18 0.17 0.13
Mean LoU	0.52 0.46 0.53 0.58 0.53 0.50

It is known that the study divides the DVRPTW model in the form of time slices and analogises each time slice to a refinement process for a static vehicle path problem. The study began by subjecting each time slice to performance validation experiments. In Table 4.1, information on the 56 data features with temporal constraints in the Solomon optimisation dataset is included.

4.1. Parameter Optimisation Experiments and Performance Testing of the TS. To investigate the relationship between parameter selection and algorithm optimisation results, a significance hypothesis testing method was introduced. The first group was set up as control group A/B with different parameters α/β and the second group was control group C/D with different adaptive parameters. after conducting several experiments on the data from the two groups respectively, a two-sample test was finally introduced for variance analysis and the results are shown in Figure 4.1.

The experiment was set up with values for the α/β parameters of (0.1/0.2) and (0.9/1.0) for groups A/B, respectively. From Figure 3.1(a), it can be seen that the experimental values obtained in Group A are significantly higher than those in Group B. The mean value of Group A is 1786.93, while that of Group B is only 1756.42, a difference of 30.51. The median values of the two groups are 1785.98 and 1736.27 respectively. The standard error and standard error of the mean for group A were 58.41 and 18.47 respectively, while those for group B were 74.91 and 23.70 respectively. it can be seen that the SEM value for group A was 5.23 smaller than that of group B, indicating that group B has better dispersion and stability of the mean. According to the significance hypothesis test, when the confidence level is set to 0.05, there is no significant difference between the resultant values under both conditions of assumed unity of variance and assumed unity of variance, so the algorithm is robust to the two parameters. In summary, the study selected data set B as the value for the α/β parameter. Figure 4.1(b) shows the control experiment set up to compare the values of the adaptive parameters, where the values for groups C/D are 0.5 and 0.8 respectively. it can be seen that the test means for groups C/D are 1785.91 and 1728.19 respectively; the SD and SEM values for the two data sets are (58.42/18.48) and (50.98/16.12) respectively. Again, the two sets of data were tested for significance at a confidence level of 0.05. The results showed that the test exhibited significant differences in both cases where the assumptions were

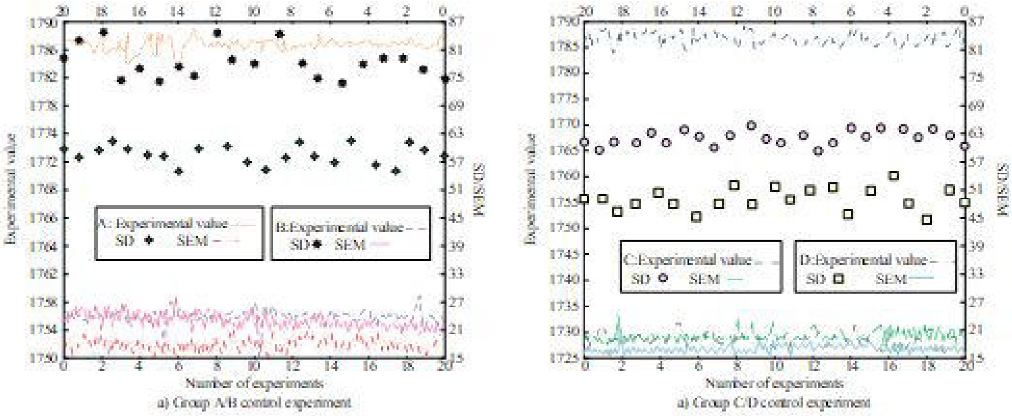

Fig. 4.1: Parameter tuning experiment

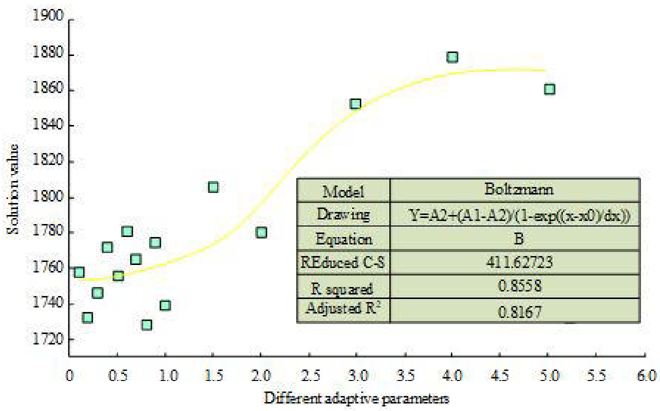

Fig. 4.2: Fitting results of adaptive parameters

flush and where the assumptions were not flush. Therefore, the value has a greater impact on the performance of the algorithm. The study carried out further tuning and optimisation of this parameter and set the value conditions to [0 0.11] and [1 1 5]. After 10 iterations, the experiments took the mean value of the data as the final selected value and used the Boltzmann function for scatter fit analysis. Figure 4.2 shows the fitting results and the final parameter selection values. As can be seen in Figure 4.2, the optimisation results of the algorithm show a tendency to become progressively weaker as the value of the adaptive parameter rises, with the algorithm optimising best when it takes values within 1, and the algorithm performance deteriorating when it is in the (1, 6) interval. Therefore, the study uses the average minimum value in the (0, 1) interval, i.e. $\theta = 0.8$. In general, the above experiments prove that the selection of the three parameters will change the total demand. Among them, the adaptive parameters are positively correlated with the optimization results, but the optimization effect will deteriorate when a certain equilibrium point is reached.

In Figure 4.3(b), the Z-axis represents the customer demand and the X/Y-axis represents the location of the customer. When the time slice interval is certain, if the dynamic customer's minimum order demand time

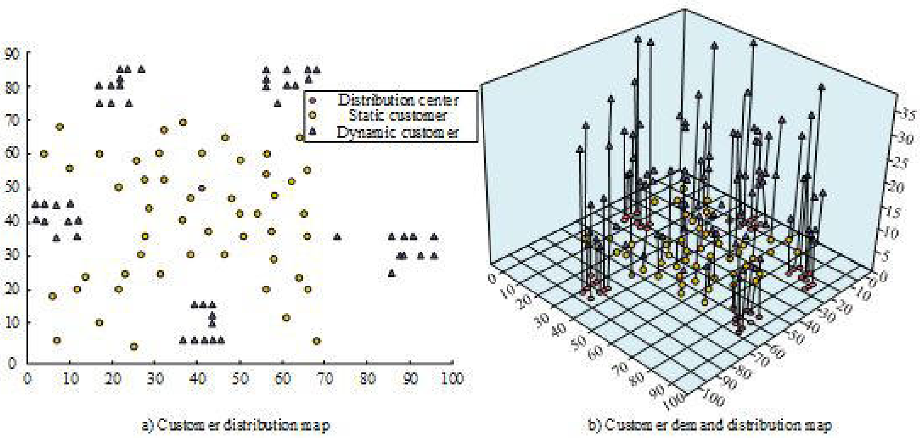

Fig. 4.3: Customer classification and customer demand analysis

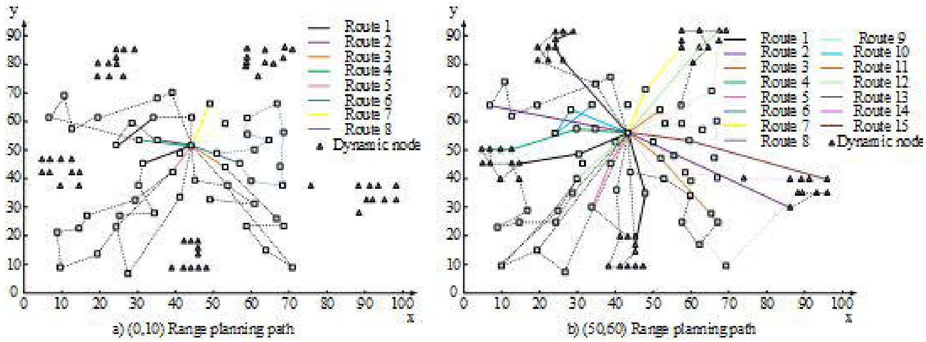

Fig. 4.4: Dynamic vehicle routing optimization simulation at different time slices

lies at the end of the area, the corresponding service time will not be less than the end of the interval. If the customer's demand time does not meet the requirement, the system will automatically reject the request. In the current prevalence of online shopping, customer points, points of sale and service points are basically interactive through intelligent means. Among them, the dynamic change of the relationship between the point of sale and the point of service requires the DVRPTW model to realize. After the vehicle gets the customer's demand, it should respond to it in a timely manner. The model can visualize static users and dynamic users, which can improve the operation efficiency of the model. First of all, static users should be satisfied first, because the dynamic requirements of users will affect the path generation, while the optimal path of static users is fixed, which greatly reduces the operation cost. Dynamic user paths follow static user paths until static user requirements are fully addressed. Therefore, the visualization of user distribution is very necessary.The study selects Instance RC102 as the test set and also sets the time slice parameter n_{ts} to 10, which yields 50 dynamic users in the first time slice interval. The study did dynamic vehicle PO simulations for the time slice intervals of [0, 10] and [50, 60] respectively, and the results are shown in Figure 4.4.

In figure 4.4, the dotted line represents path planning that does not incorporate the needs of dynamic

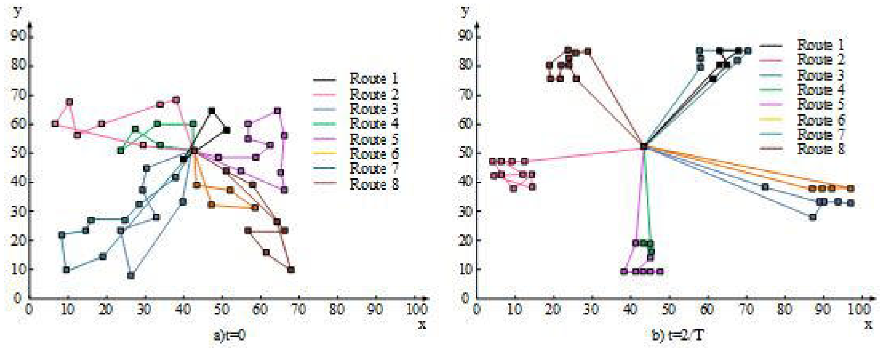

Fig. 4.5: Path analysis without considering dynamic customers

customers; realisations represent path planning that incorporates the needs of dynamic customers. The purple symbols then indicate dynamic customers. At the initial moment, the merchant assigns eight vehicles to serve the needs of 50 static customers from the previous working day, and the distance cost of driving is calculated as 743.81. Once the system has completed 50 dynamic customers, its distance cost rises to 1504.53 and the number of vehicles rises to 15. PO performance indicators include both vehicle load factor as well as time utilisation, with time and vehicle utilisation reaching 100% and 96.51% respectively, making the DVRPTW model extremely efficient for optimisation. However, the customer's TW constraint causes some waste of time, as the vehicle has already departed from the work centre at this point and its needs to complete its first order before the dynamic user's task can be considered. There are a large number of constraint algorithms in the model, which results in the vehicle having to complete the specified task before it can start the next order task. To further understand the impact of different conditions on the performance of the DVRPTW model, the study compares it to a traditional PO. The traditional approach does not consider requests from dynamic customers in the first half of the working day, serving only static users from the previous working day, and only processes dynamic customer requests from the previous half of the working day in the second half of the working day. The traditional optimisation path is shown in Figure 4.5.

As can be seen from Figure 4.5, the conventional PO divides it into two time slots, with the total number of vehicles in both time slots being 8. In the first time slice interval, the distance cost of the vehicles is 744.38. The distance cost of travelling in the second time slice interval is 843.95, and the total distance cost is 1588.33, which is more than 83.8 compared to the DVRPTW model. based on further calculations, the average vehicle utilisation for the two time slots reached 47.13% and 60.62% respectively, with a total average utilisation of 53.875%, which is significantly lower than the average load factor of 57.48% for the DVRPTW model. The experimental results show that the new DVRPTW optimisation model, which allows for better cost-efficiency, works better than traditional optimisation methods in terms of response speed and vehicle utilisation. Due to the excessive amount of data, T/2 time slice in the research model was selected for comparison. The performance of each model in terms of the number of customers, actual vehicle load, vehicle load utilization rate and route cost is shown in Table 4.2.

As can be seen from Table 4.2 above, compared with the traditional model, the average vehicle utilization rate of the optimized model has increased by 35.87%. This is because after route optimization, the number of customers that vehicles can satisfy at the same time has increased, which has greatly improved the vehicle loading rate and made it realize the maximum load on one route. The total customer task completion is twice that of the traditional model. Although the path cost of the optimization model is higher, it achieves a substantial improvement in completion efficiency at the lowest possible cost. This is because the path planning consistency of the optimized model is significantly better, and the vehicle utilization rate is naturally higher, which also leads to the rest of the performance improvement.The study repeated the above experiments several

Table 4.2: Performance comparison of each model

Model	Path number	Customer quantity	Actual load	Path cost	Loading rate/%
Optimization model	1	9	171	107.95	85.50
	2	6	129	110.05	64.50
	3	8	124	98.81	62.00
	4	9	149	115.02	74.50
	5	5	99	82.80	49.50
	6	6	69	76.15	34.50
	7	7	87	99.02	43.50
	8	8	147	113.07	73.50
	9	5	86	129.23	43.00
	10	4	61	60.96	30.50
	11	8	152	117.48	76.00
	12	6	146	93.23	73.00
	13	11	193	113.02	96.50
	14	6	95	155.14	47.50
	15	2	16	32.57	8.00
Sum total	100	1724	1504.52	96.50	
Traditional model	3	6	100	123.45	50.00
	4	3	60	78.73	30.00
	5	7	140	106.94	70.00
	6	4	80	114.45	40.00
	7	5	70	95.74	35.00
	8	10	190	105.96	95.00
Sum total	50	970	843.98	60.63	

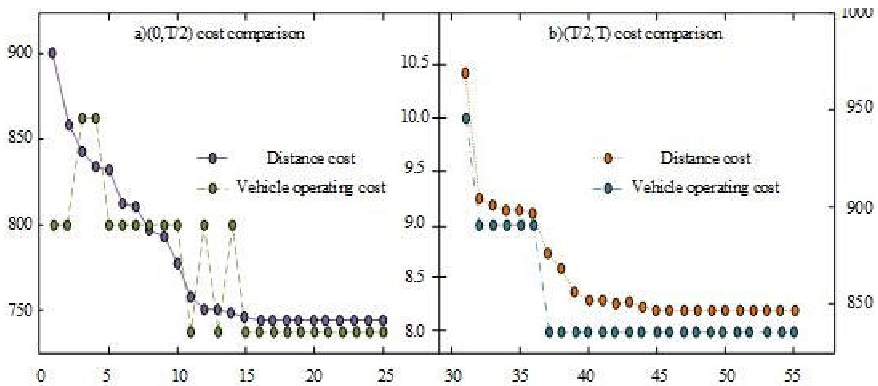

Fig. 4.6: Relationship between driving cost and vehicle use cost

times and recorded the distance cost and vehicle cost for the two time slice intervals respectively, distributed in a descending form, to obtain the relationship between driving cost and vehicle utilisation cost as shown in Figure 4.6. As can be seen from Figure 4.6, the trends of vehicle use cost and driving cost are more or less the same in the two time slice intervals, showing a positive correlation, especially in the (2/T, T) time slice interval, where both costs decrease almost simultaneously, while in the (0, 2/T) time slice interval, the vehicle use cost changes more curvilinearly and the driving distance cost is relatively moderate. As can be seen from

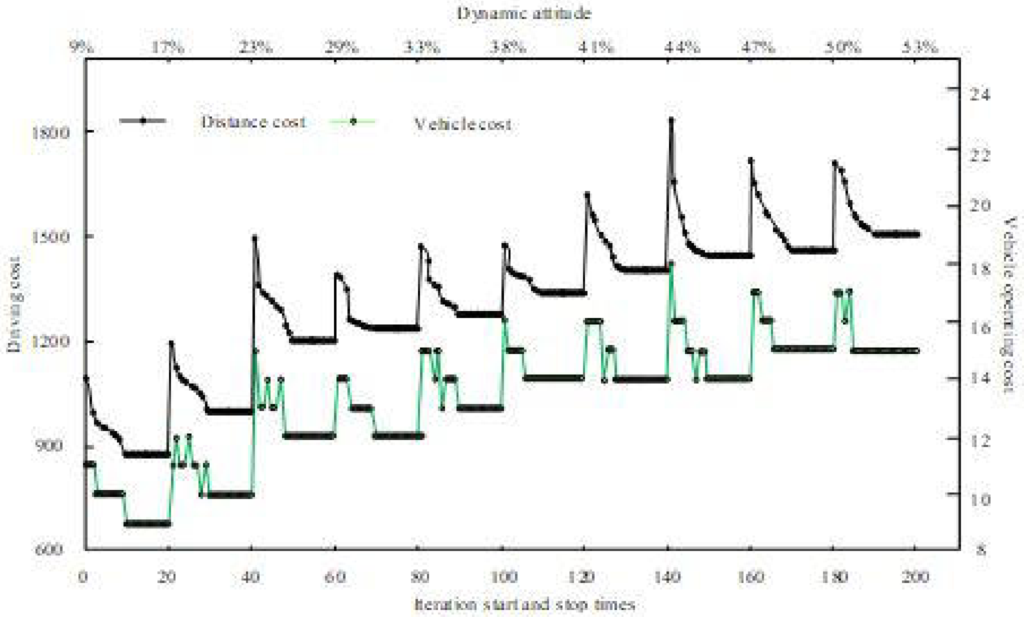

Fig. 4.7: Sensitivity of dynamic attitude to cost

the (10, 15) interval in Figure 4.6(a), an appropriate increase in the number of vehicles may lead to a reduction in driving costs when the vehicle path is in the pre-fluctuation phase. Experiments were introduced to the InstanceRC102-50-50 case to explore the sensitivity of dynamic attitude to cost, and the results are shown in Figure 4.7. When $t = 0$, the total number of static customers is 50, and as the time slice increases, the number of dynamic users increases by 5 in turn. It can be seen that cost and dynamic attitude are roughly positively correlated. In particular, the correlation of the cost of vehicle use shows interval intermittency, and within a certain range, its cost correlation will show a decreasing or stable constant trend. The experimental results show that the system is more robust to DD when the rated capacity of the vehicle is larger. Until an optimal solution is chosen, its optimisation scheme revolves around the minimisation of the number of vehicles, but may produce a phenomenon where the two costs appear in an opposing relationship.According to the relationship between vehicle routing cost and use cost, it can be seen that the optimal balance value can minimize the cost of both sides, which indicates that the relationship between the two is proportional. The dynamic attitude is the comprehensive consideration of the number of dynamic customers, the time point of occurrence and the frequency of occurrence, which are all factors that lead to cost changes. In the early stage, the change of the number of customers will only affect the distance cost, but with the increase of the vehicle capacity, the ability to meet the dynamic demand is greater. Therefore, the cost is positively correlated with the dynamic attitude.

5. Conclusion. The dynamic vehicle path problem must be optimised since inadequate merchant response frequently results in a bad IoT consumer experience. In the paper, a DVRPTW model is suggested, one is modelled, one is examined, and one is solved using a prohibited search method. Finally, the study utilised the Solomon optimisation dataset as the basis for the simulation experiments. Since the $\alpha/\beta/\theta$ parameters in TS have a large impact on the performance of the algorithm, the study first conducted parameter tuning experiments on them using significance hypothesis validation. Through two different sets of control experiments, the $\alpha/\beta/\theta$ parameters of 0.9/1.0/0.8 respectively were finally selected. The new optimisation approach and the conventional optimisation strategy were further compared in the experiments. The experimental results showed that the DVRPTW model achieved an average vehicle utilisation rate of 57.48% compared to only 53.875% in the traditional optimisation strategy, a relative decrease of 3.6%. This was in contrast to the traditional optimisation strategy, which only achieved an average vehicle utilisation rate of 53.875%. Additionally, the trials were run to visualise consumer distribution and examine how dynamic attitude and cost interacted, yielding a generally favourable association between the two. The DVRPTW optimisation model, which is a better option for dynamic vehicle routing issues since it allows for faster dynamic reaction and cost reduction than conventional PO approaches, is proven to be reliable by all of the aforementioned trials. Though there are still some problems to be solved, for example, is not high enough and needs to be further optimized. In addition, this study only designed the dynamic response of the vehicle to the customer, without considering the customer's response to the vehicle position. Only when there is two-way interaction between the two, can the model achieve the highest operating efficiency.

REFERENCES

[1] M. Abdirad, K. Krishnan, D. Gupta, and X. Li., *A two-stage metaheuristic algorithm for the dynamic vehicle routing problem in industry 4.0 approach*, Journal of Management Analysis, 8 (2021), pp. 69–83.

[2] D. Ambrosino and M. Paolucci, *Special issue of networks: New network models and approaches for logistics and transportation*, Networks, 78 (2021), pp. 227–228.

[3] X. Dai, M. Chen, and Y. Zhou, *Optimal logistics transportation and route planning based on fpga processor real-time system and machine learning*, Microprocessors and Microsystems, 80 (2021), p. 107.

[4] J. Guo, J. Long, X. Xu, and M. Y. Yuan, *The vehicle routing problem of intercity ride-sharing between two cities*, Transportation Research Part B: Methodological, 158 (2022), pp. 113–139.

[5] I. Himmich, I. E. Hallaoui, and F. Soumis, *Primal column generation framework for vehicle and crew scheduling problems*, Networks, 75 (2020), pp. 238–252.

[6] Y. Li, Y. Shen, and J. Li, *A discrete artificial bee colony algorithm for stochastic vehicle scheduling*, Complex System Modeling and Simulation, 2 (2022), pp. 238–252.

[7] R. Lin, H. Huang, and M. Li, *An automated guided logistics robot for pallet transportation*, Assembly Automation, 41 (2020), pp. 45–54.

[8] Z. Merka, D. Perkov, and V. Bonin, *The significance of blockchain technology in digital transformation of logistics and transportation*, International Journal of E-Services and Mobile Applications, 12 (2020), pp. 1–20.

[9] O. B. Olmez, C. Gultekin, B. Balcik, and A. E. Ozener, *A variable neighborhood search based matheuristic for a waste cooking oil collection network design problem*, European Journal of Operational Research, 302 (2022), pp. 187–202.

[10] A. Ponmalar and V. Dhanakoti, *Hybrid whale tabu algorithm optimized convolutional neural network architecture for intrusion detection in big data*, Concurrency and computation: practice and experience, 34 (2022), pp. 7038–7052.

[11] S. Pottel and A. Goel, *Scheduling activities with time-dependent durations and resource consumptions*, European Journal of Operational Research, 301 (2022), pp. 445–457.

[12] R. Raeesi and Z. KG., *The electric vehicle routing problem with time windows and synchronised mobile battery swapping*, Transportation Research Part B Methodological, 140 (2020), pp. 101–129.

[13] ———, *Coordinated routing of electric commercial vehicles with intra-route recharging and en-route battery swapping*, European Journal of Operational Research, 301 (2022), pp. 82–109.

[14] I. Sbai and S. Krichen, *A real-time decision support system for big data analytic: A case of dynamic vehicle routing problems*, Procedia Computer Science, 176 (2020), pp. 938–947.

[15] H. M. Stellingwerf, G. Lhc, G. Laporte, and A. K. B. Behdani, *The quality-driven vehicle routing problem: Model and application to a case of cooperative logistics*, International Journal of Production Economics, 231 (2021), p. 109.

[16] S. Yang, L. Ning, P. Shang, and L. C. Tong, *Augmented lagrangian relaxation approach for logistics vehicle routing problem with mixed backhauls and time windows*, Transportation Research Part E: Logistics and Transportation Review, 135 (2020), p. 101.

[17] X. Zhang, L. Chen, M. Gendreau, and A. Langevin, *A branch-and-cut algorithm for the vehicle routing problem with two-dimensional loading constraints*, European Journal of Operational Research, 302 (2022), pp. 259–269.

[18] Y. Zhao, B. Du, and S. Li, *Agile satellite mission planning via task clustering and double-layer tabu algorithm*, Computer Modeling in Engineering and Science, 122 (2020), pp. 235–257.

Edited by: Zhenling Liu
Special Issue on: Cloud Computing for Intelligent Traffic Management and Control
Received: Jun 26, 2023
Accepted: Aug 1, 2023