Journal of Database Management
Volume 34 • Issue 3

Modeling and Optimization of Multi-Model Waste Vehicle Routing Problem Based on the Time Window

Hongjie Wan, Nanjing University of Posts and Telecommunications, China
Junchen Ma, Nanjing University of Posts and Telecommunications, China
Qiumei Yu, Key Laboratory of Urban Land Resources Monitoring and Simulation MNR, China
Guozi Sun, Nanjing University of Posts and Telecommunications, China*
Hansen He, Nanjing Jiangshipin Data Technology Co. Ltd., China
Huakang Li, XJTLU Entrepreneur College, China

ABSTRACT

With the development of China's economy, the urban floating population is also increasing, resulting in a sharp increase in the amount of urban waste. How to recycle and dispose of municipal waste more efficiently has become the top concern of municipalities and other relevant departments. In this article, the above problem is transformed into the municipal waste collection vehicle routing problem (MWCVRP) to solve the problem with the minimum total waste transportation cost. Because the carrying capacity of different models is different, this article introduces a cost calculation criterion that combines the total mileage of different models of transport vehicles and the number of station services. A multi-model garbage truck path optimization model is established, and then a heuristic-based task dynamic assignment algorithm is designed to solve the problem. The Solomon dataset is used to verify the feasibility and effectiveness of the model and algorithm through experiments.

KEYWORDS

Cost Calculation, Heuristic, Task Dynamic Assignment Algorithm, Urban Waste, Vehicle Routing Problem

INTRODUCTION

With the continuous development of China's economy and the explosive increase in the population, the production of municipal waste has also entered an exponential growth stage. Lie (Yang and Chen et al., 2013) found that in the past fifteen years, the production of municipal waste in China has increased significantly compared with that in the United States. In the face of massive urban waste production, how to recycle and dispose of waste has become the most concerning issue of municipal and other related departments. This paper converts the waste collection problem into a classic Vehicle

DOI: 10.4018/JDM.321543 *Corresponding Author

Routing Problem (VRP) and further expands it to the Municipal Waste Collection Vehicle Routing Problem (Han and Ponce Cueto, 2015) (MWCVRP), which aims to study how to minimize the total cost of transporting waste.

This problem has been identified as an NP-hard problem since it was proposed (Kim and Ong et al., 2015) (Yousefikhoshbakht and Didehvar et al., 2014); that is, it is difficult to solve with an exact algorithm, so people have proposed two less accurate but very effective solutions to this problem (Beliën and De Boeck et al., 2014), namely, the classical heuristics algorithm and the meta-heuristics algorithm. Gevorg (Guloyan and Aydin, 2020) used the genetic greedy hybrid algorithm to find a route with the lowest total transportation cost in the case of municipal solid waste recycling in Yerevan. Sobhan (Darmian and Moazzeni, et al., 2020) used a heuristic-based multi-objective local search algorithm to determine the collection center and the route with the lowest total cost as the recycling route. Emre (Eren and Tuzkaya, 2021) converted the vehicle routing problem into the traveling salesman problem (TSP) and solved a safe recycling route when recycling COVID-19 medical waste. Meiling (He and Wei et al., 2021) adopted an adaptive variable neighborhood. The search ant colony algorithm solves the vehicle routing problem with time windows. Luis (Flores-Luyo and Agra et al., 2020) brought the traditional vehicle routing problem into the wireless network environment and used greedy combined with a heuristic algorithm to improve the loading rate of vehicles. They (Zhang and Yang et al., 2020) used a multi-objective evolutionary algorithm to reduce the vehicle waiting time in the path planning problem with time windows. Combining (Dong and Zhou et al., 2018) the discrete firefly evolution mechanism with the variable neighborhood evolution mechanism, and its design scheme was efficient in dealing with multi-objective path planning problems. Based on the Solomon dataset, they (Nasri and Hafidi et al., 2020) adopted a parallel adaptive large neighborhood search metaheuristic algorithm to obtain the best robust solution under different scenarios in the dataset.

The above studies have good results in their respective application scenarios, but there are few studies on multi-model routing problems with time windows and capacity constraints (Mojtahedi and Fathollahi-Fard et al., 2021). For this reason, this paper is closer to reality (Ticha and Absi et al., 2017)(Fallah and Tavakkoli-Moghaddam et al., 2019), studies the routing problem based on multi-model waste transport vehicles, considers the difference in the mileage fee and station service fee of different types of transport vehicles, introduces a cost calculation criterion that combines the total mileage and the number of service stations, establishes a multi-model waste transport vehicle path optimization model, and then like Chenxin (Sun and Huang et al., 2022) designs a simple and effective heuristic-based task dynamic assignment algorithm to solve the problem. Different from the genetic algorithm (Ombuki and Ross et al., 2006) (Yusuf and Baba et al.,2014) that requires multiple iterations, the algorithm used in this paper focuses more on a timely assignment strategy, and we believe that this can better combine the real-life urban waste transportation problem with the theoretical multi-model routing problem with time windows. At the same time, the Solomon dataset (Solomon, 1987) is used to verify the feasibility and effectiveness of the proposed model and algorithm through experiments, which provides methodological guidance for decision support for urban waste transportation.

The remaining chapters of the article are arranged as follows: Section 2 introduces the multi-model route optimization model with a time window; Section 3 proposes an algorithm to solve the optimization model, that is, the heuristic-based task dynamic assignment algorithm; and Section 4 proves the proposed algorithm through experiments. The algorithm can solve the above practical problems very well. Section 5 gives a summary.

Model of the Municipal Waste Collection Vehicle Routing Problem

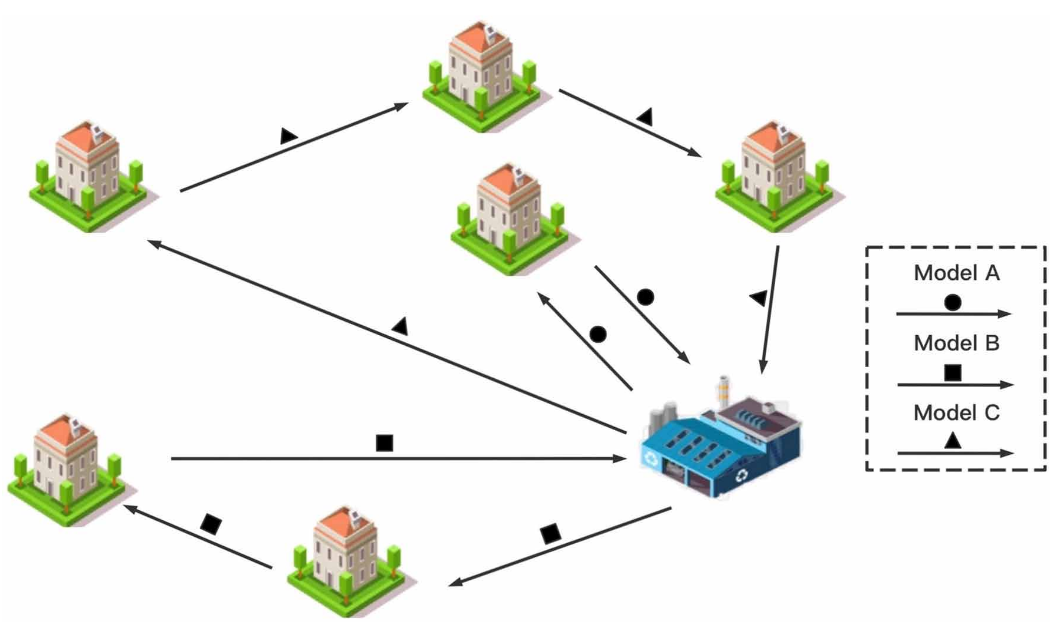
A typical MWCVRP model consists of a garage, a waste disposal station, and multiple waste transfer stations, in which there are multiple waste trucks in the garage. Each waste truck must start from the garage, visit and collect part of the waste at the waste transfer station, then transport it to the waste disposal station for unloading treatment, and finally return to the garage to complete a transportation process (Golden and Assad et al., 2002).

The model in this paper is extended on the basis of the MWCVRP model according to the waste transportation requirements in the real world to form the model of the mixed capacitated vehicle routing problem with a time window. That is, there can be various types of waste trucks, and an accessible time interval for each waste transfer station is set. The model concept diagram is shown below (Fig. 1).

The points involved in the model are represented by the set $V = N \cup \{0, n+1\}$, where N is the set of waste transfer stations, node represents the garage, and $node_{n+1}$ represents the waste disposal station. The distance between $node_i$ and $node_j$ is denoted by D_{ij}, where $1 \leq \mathrm{i}, \mathrm{j} \leq \mathrm{n}$. The amount of waste in each waste transfer station is represented by q_i, where $q_0 = q_{n+1} = 0$. There are m vehicles in the garage, which is represented by the set $P = \left\{v_1, \ldots, v_k, \ldots, v_m\right\}$, where $1 \leq k \leq m$, and Q_k is used to represent the capacity of vehicle v_k. The types of waste trucks are represented by a set T. This paper assumes that there are three types of vehicles, namely, $T = \{a, b, c\}$. C_t^k represents the service fee of the t-type transport vehicle v_k, and K_t^k represents the kilometer fee of the t-type transport vehicle v_k, where $t \in T$. The accessible time interval of the waste transfer station is represented by $[\pm_i, {}^2{}_i]$, where $\pm_i$ indicates the start time when the transfer station is accessible, and ${}^2{}_i$ indicates the end time when the transfer station is accessible.

The goal of building this model is to minimize the total cost of waste transportation, which includes not only the shortest total transportation distance but also the smallest possible station service fee received per vehicle. Although the total station service fee is a fixed value, the station service fee of m vehicles can be kept within a normal range, that is, an average value, through an assignment algorithm. This can ensure that the total cost of each vehicle is roughly the same, and the cost of each vehicle in the real world is roughly the same, which is conducive to the coordinated assignment of resources.

Figure 1.
Model of the municipal waste collection vehicle routing problem



Then, x_{ij}^{k} is used to indicate whether $node_i$ accesses $node_j$ after transport vehicle v_k visits, where $x_{ij}^{k} = 1$ is Access, and $x_{ij}^{k} = 0$ is no access (Kumar and Panneerselvam, 2012). s_i^k represents the start time when waste transfer station N_i is visited, and d_i represents the remaining amount of waste in $node_i$. Combining the above parameters and decision variables, the mathematical model expression of the MWCCVRPTW is:

$$min\sum_{k=1}^{m} C_t^k \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}^{k} + \sum_{k=1}^{m} K_t^k \sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij} x_{ij}^{k} \tag{1}$$

subject to:

$$\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}^{k} = n \tag{2}$$

$$\sum_{i=1,i\neq j}^{n}\sum_{k=1}^{m} x_{ij}^{k} = 1, j = 1,\ldots,n \tag{3}$$

$$\sum_{j=1,i\neq j}^{n}\sum_{k=1}^{m} x_{ij}^{k} = 1, i = 1,\ldots,n \tag{4}$$

$$\sum_{i=0,i\neq p}^{n} x_{ip}^{k} = \sum_{j=1,j\neq p}^{n+1} x_{pj}^{k}, p = 1,\ldots,nk = 1,\ldots,m \tag{5}$$

$$\sum_{j=1}^{n} x_{0j}^{k} = \sum_{i=1}^{n} x_{jn+1}^{k} = 1, k = 1,\ldots,m \tag{6}$$

$$\pm_i \leq s_i^k \leq {}^{2}{}_i, i = 1,\ldots,nk = 1,\ldots,m \tag{7}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} q_j x_{ij}^{k} \leq Q_k, k = 1,\ldots,m \tag{8}$$

$$\sum_{d=1}^{n} d_i = 0 \tag{9}$$

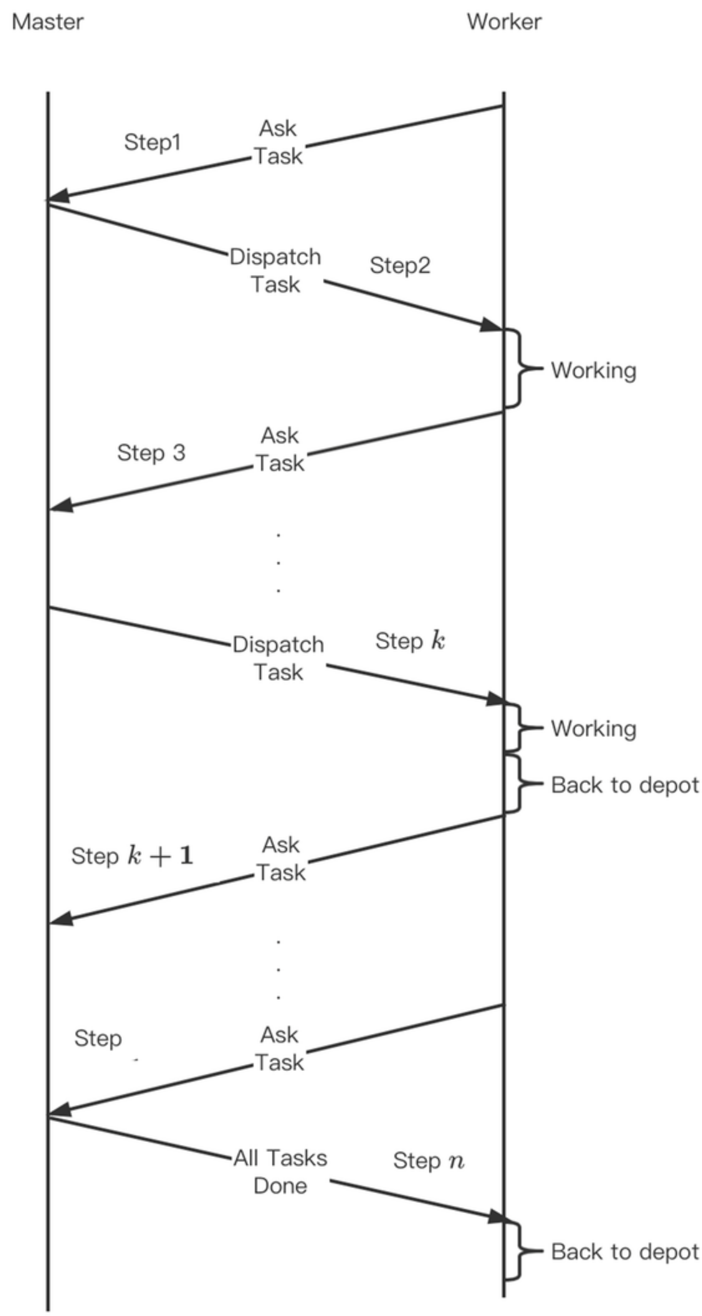
Constraint (1) is the optimization objective function, that is, the total cost of waste trucks; Constraint (2) ensures that all waste transfer stations can be accessed; Constraints (3) and (4) ensure that each waste transfer station can only be accessed once; Constraint (5) ensures the correct direction of the vehicle, that is, the vehicle should leave immediately after visiting $node_i$; Constraint (6) emphasizes that the vehicle must start from $node_0$, then visit other nodes, finally visit $node_{n+1}$ and return to $node_0$; Constraint (7) is the time window constraint to ensure that the vehicle must reach within the specified time of $node_i$; Constraint (8) is a capacity constraint, limiting the amount of waste to not exceed the capacity of the car; Constraint (9) means that after all nodes have been visited, each node has 0 waste remaining.

A Heuristic-Based Dynamic Task Assignment Algorithm

The Basic Thinking

The multi-model waste transportation vehicle routing problem studied in this paper is a typical discrete optimization problem. To solve the optimal solution agreed upon in the model, this paper adopts a heuristic-based task dynamic assignment algorithm. The general idea of the algorithm is to regard each waste truck as a Worker and regard all waste transfer station processing requests as tasks and push them into the task queue of Master so that Master can dynamically assign the optimal task for each Worker requesting task so that the dynamic assignment algorithm can solve the discrete optimization problem of the vehicle path. The flow chart of the assignment algorithm is as follows (Fig. 2).

Figure 2.
A Heuristic-based dynamic task assignment algorithm



Init Master and Worker

Suppose depot has *m* vehicles, that is, there are *m* Worker, and only Ask Task, Working and Back to depot three behaviors. Because this paper studies the multi-model vehicle routing problem, it is assumed that there are three types of vehicles, namely, A, B, and C, and the corresponding rated capacity is x, y and z; each transporter, namely, $Worker_i$, must determine its own model t_i during initialization, where $1 \leq i \leq m$. The processing request task of each transfer station mainly includes Cust No., Xcoord., Ycoord., Demand, Ready Time, Due Date, Service Time properties, push all Tasks into Master TaskQueue T_v.

The Process of Algorithm

When $Worker_i$ sends an Ask Task request to Master, Master selects from T_v according to a heuristic-based assignment strategy. The most suitable task is assigned to $Worker_i$, that is, the master sends the dispatch task signal to $Worker_i$. After $Worker_i$ receives the signal, execute the working operation, that is, go to the transfer station marked in Task for service.

When there is no Task available for assignment in the Master TaskQueue T_v, Master will send the signal of The All Tasks Done to all Workers that all transfer stations have been serviced. The worker automatically ends the worker process after receiving the signal.

1. **The Classification of Task Type:** Calculate $delay_i$ based on Ready Time, Due Date and Service Time of $Task_i$.

$$delay_i = DueDate - ServiceTime \tag{10}$$

Perform NeedFix Classification on each Task in Master TaskQueue T_v.

$$NeedFix = \begin{cases} false, delay_i \leq ReadyTime \\ true, delay_i > ReadyTime \end{cases} \tag{11}$$

where $delay_i$ represents the latest time that the task should theoretically be served. Such a classification operation enables Master to identify urgent tasks and delay tasks, thereby improving the accuracy of assignment.

For example, four tasks have been reached now, as shown in the following table (Table 1).

Table 1.
Classification example table

Cust No.	Ready Time	Due Date	Service Time
1	0	1127	90
2	727	782	90
3	621	702	90
4	0	1125	90

Perform type classification on four tasks, according to the above algorithm, to identify that $Task_1$ and $Task_4$ are delay tasks, which can be assigned slowly, and the other tasks are all urgent tasks, which need to be assigned to the free worker immediately.

However, it is not enough to have such a Task type classification because some Task Ready Time is too early or too late, which will lead to many Tasks being aggregated for a certain period of time. When the number of idle workers is less than the number of tasks, it will affect the master assignment strategy and task execution process, so the next step of data correction is needed.

The data correction is mainly to reset the Reay Time so that there is a certain time interval between Task*i* and Task *j*, which can greatly reduce the degree of the assignment difficulty of Master and improve the efficiency of scheduling idle Worker. The idea of the data correction algorithm is simple, and it has achieved good results on the Solomon public dataset.

2. **Date Correction:** First, initialize two empty sets $S_{needFix}$ and S_{nonFix} and a hash table map with all zero status bits. Traverse T_v of Master, for each $Task_i$, if its Need Fix status bit is true, then $Task_i$ is pushed into $S_{needFix}$, and map is subscripted to the state position of Ready Time; otherwise, it is pushed into S_{nonFix}.

After the push-to-set operation is completed, the data correction process begins. Traverse the set $S_{needFix}$, and perform a ready time reset operation for each $Task_i$. The random time method is used here, that is, the random number time is generated by calling the rand function of the system to ensure that time is not greater than $delay_i$, not less than Ready Time. If the status bit of the map subscripted by time is zero, set the value of ready time of this $Task_i$ to time, set the corresponding state position of the map to 1, and add it to S_{nonFix}; otherwise, regenerate the random number time until time meets the requirements.

After the above set division operation, there are $Task_1$ and $Task_4$ in $S_{needFix}$ and S_{nonFix} in $Task_2$ and $Task_3$. Assuming that the random generation time of $Task_1$ is 50, and the status bit of the 50th position in the map is zero, then the ready time is modified to 50, and the map corresponds to the position state turn 1. The random time generated by $Task_4$ is also 50, but the 50th position in the map. The status bit is already 1, so it is necessary to regenerate time, and it is 60. The status bit of the 60th position in the map is zero, and the data are corrected in the same way.

Among them, $1 \leq \mathrm{j} \leq \mathrm{length}\left(T_v\right)$, map status bit is 1 means that there is Task at this point, and zero means no Task at this point. The rand function is used to reset Ready Time instead of the traditional iterative reset method because the data processed by the rand function combined with the status bit reset method are more accurate and conform to the law of uniform distribution. For example, $Task_1$ and $Task_4$ are classified as delay tasks. If the traditional iterative reset algorithm is used, that is, traversing map, if Ready Time is reset, then there will be a high probability that the Ready Time of $Task_1$ will be lower than $Task_4$. This is something we don't want to see. Although both $Task_1$ and $Task_4$ are delay tasks, there are also task priorities. According to the design idea of the master assignment strategy and under the nature of delay tasks, the task with the lower Due Date should also be assigned first. Therefore, the Ready Time of $Task_4$ should be as low as possible as $Task_1$. Using the rand function can greatly increase the probability of this happening, but the traditional iterative reset algorithm cannot do so.

3. **Init TaskQueue:** Push Task in S_{nonFix} into T_v in Master, according to the corrected Ready Time Sort in ascending order. If Ready Time is equal, then sort in ascending order according to Due Date of the task to complete the initialization of the task queue.
4. **Dispatch Algorithm of Master:** When Master receives Ask Task request from $Worker_i$, first record the arrival time of $Worker_i$ request TimeStamp, current location Location, rated capacity VehcCap, and remaining capacity RestCap. Then, Master will find the Task available for assignment at this TimeStamp moment in the task queue T_v, that is, Ready Time ≤ TimeStamp ≤ Due Date, and Demand ≤ VehcCap, then push these tasks into the candidate queue C_v.

This paper assumes that there is no penalty function, that is, it is not allowed that TimeStamp exceeds the Due date of the Task; otherwise, the dispatch scheme fails. The Elitist Genetic Algorithm (EGA) proposed by YanFei (Zhu and Lee et al., 2021) is assisted by a penalty function; even if TimeStamp exceeds the Due date of the Task, it will not lead to the failure of the dispatch scheme, at most increasing the cost of its scheme. Because this paper studies the Mixed Capacitated Vehicle Routing Problem with a Time Window, we feel that the method without a penalty function is more suitable for the scene where the Task needs to be serviced in time in this paper.

Traverse the candidate queue C_v, according to the idea of heuristic search (Yelmewad and Talawar, 2021), calculate the evaluation function $f\left(k\right)$, the minimum cost function $g\left(k\right)$ and the heuristic function $h\left(k\right)$. Where $Task_k \in C_v$, $f\left(k\right) = g\left(k\right) + h\left(k\right)$, $g\left(k\right)$ is $Worker_i$ the actual cost, that is, from the current position to the location of $Task_k$; $h\left(k\right)$ is $Worker_i$ the estimated cost, that is, from the location of $Task_k$ to Depot.

When calculating the heuristic function $h\left(k\right)$, because the next route of $Worker_i$ cannot be determined, this algorithm uses the Manhattan method to roughly calculate the distance between $Task_k$ and Depot, which is taken as $h\left(k\right)$.

Sort C_v in ascending order according to the *f* value of Task, select the Task with the smaller first two *f* values, and record them as $Task_{\pm}$ and $Task_2$, where the *f* value of $Task_{\pm}$ is $f_{\pm}$, and the demand is $demand_{\pm}$. The *f* value of $Task_2$ is f_2, and the demand is $demand_2$. If $f_{\pm} < f_2$, directly assign $Task_{\pm}$ to $Worker_i$; in the case of $f_{\pm} = f_2$, then compare the demand of the two $demand_{\pm}$ and $demand_2$, and select the Task with the larger demand for assignment, for example (Table 2).

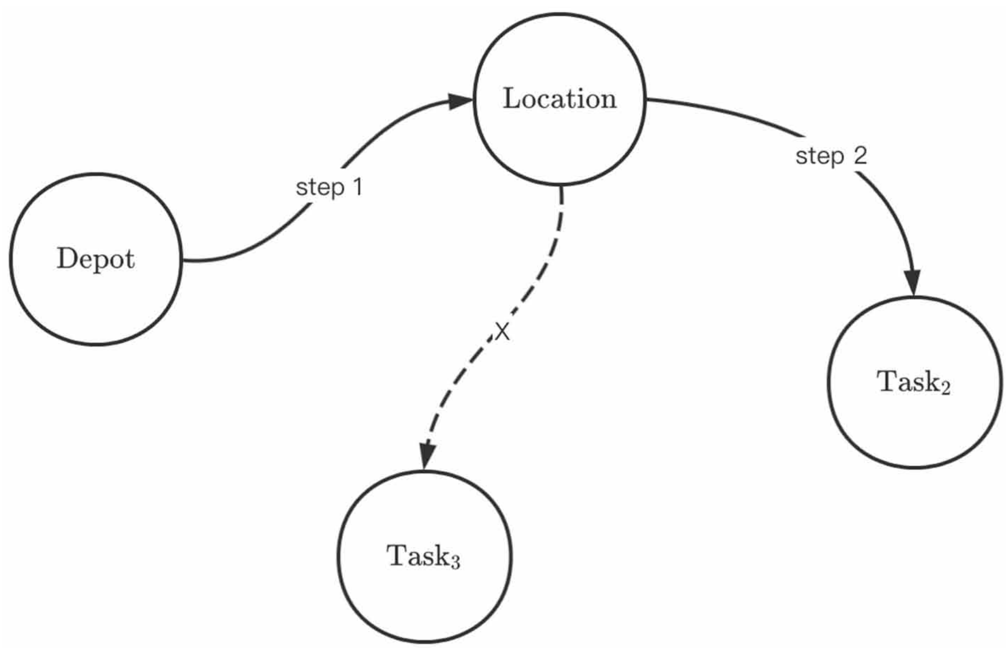
The arrival time TimeStamp of the request of $Worker_i$ is 728, the current location Location is (30, 70), the rated capacity VehcCap is 36, and the remaining capacity RestCap is 25. The selection scheme is shown below (Fig. 3).

The master selects two tasks with the smallest *f* value from the task queue T_v, that is, the *f* value of $Task_3$ is 50, where g_3 =25, h_3 =25; the *f* value of $Task_2$ is 50, where g_2 =20, h_2 = 30. At

Table 2.
Classification example table

Cust No.	XCoord.	YCoord.	Ready Time	Due Date	Demand	Service Time
3	50	65	621	702	10	90
2	45	75	727	782	20	90

Figure 3.
Selection between two tasks



this time, $f_2 = f_3$, and it is necessary to compare the demand $demand_2$ and $demand_3$. It can be seen from the above table that $demand_2$ is larger, so Master assigns $Task_2$ as the optimal task to $Worker_i$.

5. **The Working Process of Worker:** When the $Worker_i$ process starts, record the current location, that is, the location of the garage depot, determine the car capacity VehcCap according to your own model, and initialize the remaining capacity RestCap = VehcCap.

After making an Ask Task request to Master, wait for the Dispatch Task signal from Master. Perform Working or Back to depot operations according to the type of signal; if the signal of Master is TaskEnd, it means that at this time TaskQueue T_v of Master has no tasks, and $Worker_i$ will end the work process; if the signal of Master is TaskWait, it means that there is no suitable task in TaskQueue T_v of Master for $Worker_i$, notify $Worker_i$ to rest for 1 time unit and then request Master; other in the case of $Worker_i$, the Working operation should be performed, but it needs to be checked before execution. If the signal of Master is TaskNonHandle, it means There is a suitable task for $Worker_i$, but the demand of this task is greater than the remaining capacity of $Worker_i$, and it needs $Worker_i$ to execute Back to depot Back to warehouse unloading operation, and then go to the task marked location to serve.

The working operation is realized by letting the worker process of $Worker_i$ sleep for service time unit time, that is, simulating the operation of going to the target transfer station for service.

Table 3.
The main parameters of the three models

Vehicle type	Displacement/ml	Rated load	Fuel cost(¥/km)	Service fee(¥/per)
A	2982	36	0.2982	5.7
B	400	52	0.4	4.2
C	7500	100	0.67	2.7

EXPERIMENT

Dataset and Experimental Environment

The prototype of the dataset used in this experiment is the Solomon dataset. According to the category of waste trucks provided by Chengli Special Automobile Co., Ltd., this paper selects three types of vehicles with different loads, and the parameters of the three vehicles are also proportional to each other's zoom. The vehicle parameters are shown in Table 3, in which the transfer station service cost includes loss cost and loading cost (mechanical cost or labor cost). All the experimental environments tested in this article are Ubuntu 20.04LTS.

Vehicle Type's Influence

To illustrate the difference between a single model and a hybrid model, we designed the following experiments. In the experiment, according to the algorithm mentioned in this paper, the scheme using only three single models is tested, and the test dataset is the dataset C101. Compared with other datasets with scattered data points, the station distribution of the C1 series is more in line with the urban structure and is more scientific and realistic. The test results are shown in Table 4.

The travel cost refers to the sum of the costs of each type of vehicle travel distance in each scheme, and the total cost is equal to the sum of the travel cost and the station service fee. Compared with S_2, the total distance of S_3 is greatly shortened due to the large capacity of the C-type models, but the total cost is 21.33% higher due to the higher fuel cost per mileage of the C-type models. Table 5 shows the access path and loading rate of the C-type vehicles in solution S_3, where $node_1$ is the waste disposal station, and the S_1 solution that uses all the A-type vehicles. Due to the small capacity of the A-type vehicles, some large-capacity sites may be unable to be served, so it has no practical significance.

From the results in Table 5, among the 11 C-type waste trucks, only 2 waste trucks have a loading rate of 90%, the No. 3 vehicle's loading rate is only 60.00%, the average loading rate of the 11 vehicles is 79.09%, and the loading rate is too low, resulting in an extremely high waste of resources. From the above test results for a single model, it can be seen that a single model will bring problems such

Table 4.
Single vehicle type test comparison

Solution	Vehicle type	Number of vehicles	Total distance	Travel cost	Service cost	Total cost
S_1	A	--	--	--	--	--
S_2	B	11	5574	2229.60	420	2649.60
S_3	C	11	4624	3098.08	270	3368.08

Table 5.
Access paths and loading rates for C-type waste trucks

C-type vehicle No.	Access paths	Loading rates
1	1-58-79-9-75-1-31-46-49-69-1	85.00
2	1-68-43-8-20-16-1-17-40-52-67-1	90.00
3	1-44-56-55-54-1-57-39-78-80-1-51-50-1	60.00
4	1-14-18-19-1-72-47-10-65-5-70-1	80.00
5	1-25-26-63-45-1-85-86-101-35-1	85.00
6	1-6-4-87-84-93-94-1-15-37-81-2-48-1	75.00
7	1-99-97-77-11-12-29-89-1-100-22-1	70.00
8	1-82-42-28-30-73-98-1-13-23-1	70.00
9	1-21-66-64-41-83-1-59-61-60-3-76-1	100.00
10	1-33-34-32-1-36-71-62-27-24-53-1	85.00
11	1-91-88-96-95-38-74-1-7-90-92-1	70.00

as high total cost or low efficiency in the process of vehicle transportation. Therefore, this paper proposes a transportation solution based on a multi-model model to solve the unequal distribution and high total costs of the resources brought by the single vehicle type model.

Two approximate optimal solutions S_4 and S_5 are obtained through the calculation of the transportation plan based on the multi-model model, where S_4 is the approximate solution of S_1, that is, while ensuring that the transportation task can be completed on time, use as many vehicles of A-type as possible; S_5 is an approximate solution of S_2, that is, use as many vehicles of B-type as possible while ensuring that the transportation task can be completed on time. As N is the number of vehicles, TD is the travel distance, and SC is the service cost.

As seen from Table 6, after using multiple models and reducing the use of large capacity C-type vehicles, not only can the transportation task be completed on time, but the total transportation cost has been greatly reduced, and the reduction in the use of large capacity C-type vehicles has also solved the problem. For the problem of uneven resource allocation (too low loading rate), Table 7 records the access path and loading rate of the only C-type vehicle in the optimal solution S_5.

Figure 4 shows the path on a time slice in the optimal solution S_5. It can be seen from the figure that the B-type vehicle (dotted line) has a wider service range than the A-type vehicle (solid line), and the A-type vehicle is used for those service points with farther distances and smaller clusters. The model reduces the travel cost of long-distance station clusters in this way and further improves the efficiency of resource allocation and vehicle loading rate.

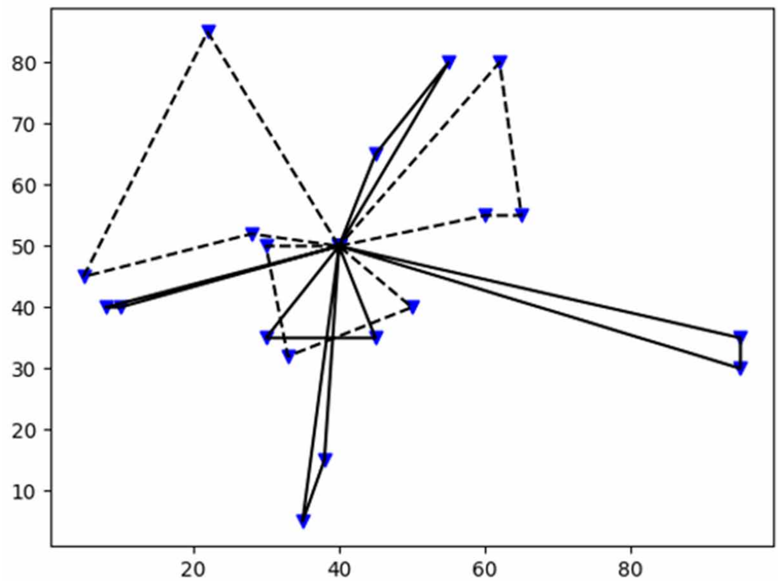
Table 6.
Approximate solution S4 and optimal solution S5 of S1

Solution	vehicle-A(N/TD/SC)	vehicle-B(N/TD/SC)	vehicle-C(N/TD/SC)	Total distance	Travel cost	Service cost	Total cost
S_4	7/3694/359.1	2/946/79.8	2/968/48.6	5608	2128.51	487.5	2616.01
S_5	5/2890/262.19	5/2410/189.0	1/346/24.30	5646	2057.61	475.49	2533.11

Table 7.
Access path and loading rate of C-type waste trucks in optimal solution S5

C-type vehicle No.	Access paths	Loading rates
1	1-91-88-64-54-1-81-17-15-24-28-1	100.00

Figure 4.
Partial path of optimal solution



The above tests are the approximate solution S_1 that uses as many A-type vehicles as possible and the optimal solution S_5 that uses as many B-type vehicles as possible. It can be seen from the comparison that under the same conditions, the B-type vehicles are preferred, the transportation scheme assisted by other models has the best effect, and the total transportation cost is saved by approximately 3.27%. Since the B-type vehicles inherit the advantages of A-type and C-type vehicles (low cost and high load), the optimal solution can be proposed based on the approximate solution S_4 and the optimal solution S_5. Then, for the datasets C101, R101 and RC101, this idea is proven by experimental comparison with a single vehicle model. The experimental results are shown in Table 8 below.where Δ_1 represents the percentage of the travel cost of a single B-type vehicle higher than the optimal solution, and Δ_2 refers to the percentage of the total cost of a single B-type vehicle higher than the optimal solution. From the comparison of the three datasets, it can be seen that the optimal plan is to give priority to the use of B-type vehicles, and the proportion of investment exceeds 45.45%. In summary, solving the problem of mixed vehicles can reduce the travel cost and the total cost to a certain extent.

Comparisons with GA and p-SA

Finally, to verify the efficiency of the heuristic-based task dynamic assignment algorithm proposed in this paper in solving the time window-based multi-model urban waste transport vehicle routing problem, the algorithm in this paper is compared with two traditional algorithms for solving vehicle

Table 8.
Comparison between the solution of a single B-type vehicle and the optimal solution

Dataset	Number of A-type vehicles	Number of B-type vehicles	Number of C-type vehicles	Total distance	Travel cost	Δ_1	Total cost	Δ_2
C101	5	5	1	5646	2057.61	8.35	2533.11	4.59
C101	0	11	0	5574	2229.60		2649.60	
R101	3	7	1	4976	1730.76	18.79	2230.26	11.01
R101	0	11	0	5140	2056.00		2475.99	
RC101	4	6	0	6478	2340.97	10.68	2817.97	6.85
RC101	0	10	0	6478	2591.20		3011.2	

routing problems, GA (Wang and Chen, 2012) and p-SA (Wang and Mu et al., 2015). Under the condition of ensuring the use of the same vehicle type and the same number of vehicles, the optimal solutions of the three algorithms in the C101, R101 and RC101 datasets are solved. The evaluation indicators mainly include the total distance, travel cost and total cost. The experimental comparison results are shown in Table 9.

It can be seen from Table 9 that on three typical datasets, the algorithm proposed in this paper is better than GA and p-SA in solving R101 because the M2W algorithm gives full play to the characteristics of dynamic task assignment, which can be more reasonable and accurate. When solving RC101, better than p-SA but slightly inferior to GA, it is because most of the points in the dataset RC101 are aggregated and mixed with some randomly distributed points, so M2W cannot give full play to the heuristic-based characteristics of the task dynamic assignment strategy. In summary, the M2W proposed in this paper outperforms traditional pathfinding algorithms on relatively discrete and randomly distributed datasets.

CONCLUSION

This paper takes the vehicle routing problem with a time window as the basic model, introduces multi-model and capacity constraints and cost calculation criteria, establishes a multi-model waste transporter route optimization model, and then designs a heuristic-based task dynamic assignment algorithm to solve the problem.

Table 9.
C101 experimental comparison results

Algorithm	Dataset	Number of A-type vehicles	Number of B-type vehicles	Number of C-type vehicles	Total distance	Travel cost	Service cost	Total cost
GA	C101	5	5	1	5279	1954.02	462.00	2416.02
p-SA	C101	5	5	1	5402	1991.20	460.49	2451.70
M2W	C101	5	5	1	5646	2057.61	475.49	2533.11
GA	R101	3	7	1	5366	1684.91	530.10	2233.91
p-SA	R101	3	7	1	5512	1683.17	555.00	2238.17
M2W	R101	3	7	1	4976	1730.76	499.49	2230.26
GA	RC101	4	6	0	6698	2308.85	490.50	2799.35
p-SA	RC101	4	6	0	6980	2310.48	525.00	2835.48
M2W	RC101	4	6	0	6478	2340.97	477.00	2817.97

The research in this paper provides a model and method for municipal and other related departments to solve the problem of urban waste recycling, especially in the case of multi-model waste trucks. The method in this paper will greatly improve the utilization of resources and save the cost of manpower and material resources. The innovation in the algorithm also includes considering that the vehicle has a capacity limit, so there is a high possibility that the vehicle will be fully loaded during transportation. If this happens, it needs to return to the processing station for the unloading operation and then go to the next station for service.

In developing countries, research on the problem of urban waste vehicle routing is still in its infancy. This paper only considers the economic factors of the recycling process, ignoring the impact on the surrounding environment, so it studies how to minimize the environmental impact while ensuring the total transportation cost. The minimum will be the next research content.

ACKNOWLEDGMENT

Conflict of Interest

The authors of this publication declare there is no conflict of interest.

Funding Agency

This research was supported by the National Natural Science Foundation of China [grant numbers 61906099]; the Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources [grant numbers KF-2019- 04-065].

REFERENCES

Beliën, J., De Boeck, L., & Van Ackere, J. (2014). Municipal solid waste collection and management problems: A literature review. *Transportation Science*, *48*(1), 78–102. doi:10.1287/trsc.1120.0448

Darmian, S. M., Moazzeni, S., & Hvattum, L. M. (2020). Multi-objective sustainable location-districting for the collection of municipal solid waste: Two case studies. *Computers & Industrial Engineering*, *150*, 106965. doi:10.1016/j.cie.2020.106965

Dong, W., Zhou, K., Qi, H., He, C., & Zhang, J. (2018). A tissue P system based evolutionary algorithm for multi-objective VRPTW. *Swarm and Evolutionary Computation*, *39*, 310–322. doi:10.1016/j.swevo.2017.11.001

Eren, E., & Tuzkaya, U. R. (2021). Safe distance-based vehicle routing problem: Medical waste collection case study in COVID-19 pandemic. *Computers & Industrial Engineering*, *157*, 107328. doi:10.1016/j.cie.2021.107328 PMID:33879956

Fallah, M., Tavakkoli-Moghaddam, R., Alinaghian, M., & Salamatbakhsh-Varjovi, A. (2019). A robust approach for a green periodic competitive VRP under uncertainty: DE and PSO algorithms. *Journal of Intelligent & Fuzzy Systems*, *36*(6), 5213–5225. doi:10.3233/JIFS-179323

Flores-Luyo, L., Agra, A., Figueiredo, R., & Ocaña, E. (2020). Heuristics for a vehicle routing problem with information collection in wireless networks. *Journal of Heuristics*, *26*(2), 187–217. doi:10.1007/s10732-019-09429-6

Golden, B. L., Assad, A. A., & Wasil, E. A. (2002). Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairy, and newspaper industries. In *The vehicle routing problem* (pp. 245–286). Society for Industrial and Applied Mathematics. doi:10.1137/1.9780898718515.ch10

Guloyan, G., & Aydin, R. (2020, December). Optimization of capacitated vehicle routing problem for recyclable solid waste collection using genetic and seed genetic algorithms hybridized with greedy algorithm. In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 595-599). IEEE. doi:10.1109/IEEM45057.2020.9309901

Han, H., & Ponce Cueto, E. (2015). Waste collection vehicle routing problem: Literature review. *PROMET-Traffic&Transportation*, *27*(4), 345–358. doi:10.7307/ptt.v27i4.1616

He, M., Wei, Z., Wu, X., & Peng, Y. (2021). An adaptive variable neighborhood search ant colony algorithm for vehicle routing problem with soft time windows. *IEEE Access : Practical Innovations, Open Solutions*, *9*, 21258–21266. doi:10.1109/ACCESS.2021.3056067

Kim, G., Ong, Y. S., Heng, C. K., Tan, P. S., & Zhang, N. A. (2015). City vehicle routing problem (city VRP): A review. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4), 1654–1666. doi:10.1109/TITS.2015.2395536

Kumar, S. N., & Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants.

Mojtahedi, M., Fathollahi-Fard, A. M., Tavakkoli-Moghaddam, R., & Newton, S. (2021). Sustainable vehicle routing problem for coordinated solid waste management. *Journal of Industrial Information Integration*, *23*, 100220. doi:10.1016/j.jii.2021.100220

Nasri, M., Hafidi, I., & Metrane, A. (2020). Multithreading parallel robust approach for the VRPTW with uncertain service and travel times. *Symmetry*, *13*(1), 36. doi:10.3390/sym13010036

Ombuki, B., Ross, B. J., & Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, *24*(1), 17–30. doi:10.1007/s10489-006-6926-z

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*(2), 254–265. doi:10.1287/opre.35.2.254

Sun, C., Huang, C., Zhang, H., Chen, B., An, F., Wang, L., & Yun, T. (2022). Individual tree crown segmentation and crown width extraction from a heightmap derived from aerial laser scanning data using a deep learning framework. *Frontiers in Plant Science*, *13*, 914974. doi:10.3389/fpls.2022.914974 PMID:35774816

Ticha, H. B., Absi, N., Feillet, D., & Quilliot, A. (2017). Empirical analysis for the VRPTW with a multigraph representation for the road network. *Computers & Operations Research*, *88*, 103–116. doi:10.1016/j.cor.2017.06.024

Wang, C., Mu, D., Zhao, F., & Sutherland, J. W. (2015). A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Computers & Industrial Engineering*, *83*, 111–122. doi:10.1016/j.cie.2015.02.005

Wang, H. F., & Chen, Y. Y. (2012). A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Computers & Industrial Engineering*, *62*(1), 84–95. doi:10.1016/j.cie.2011.08.018

Yang, L., Chen, Z., Liu, T., Wan, R., Wang, J., & Xie, W. (2013). Research output analysis of municipal solid waste: A case study of China. *Scientometrics*, *96*(2), 641–650. doi:10.1007/s11192-013-0982-z

Yelmewad, P., & Talawar, B. (2021). Parallel version of local search heuristic algorithm to solve capacitated vehicle routing problem. *Cluster Computing*, *24*(4), 3671–3692. doi:10.1007/s10586-021-03354-9

Yousefikhoshbakht, M., Didehvar, F., & Rahmati, F. (2014). An efficient solution for the VRP by using a hybrid elite ant system. *International Journal of Computers, Communications & Control*, *9*(3), 340–347. doi:10.15837/ijccc.2014.3.161

Yusuf, I., Baba, M. S., & Iksan, N. (2014). Applied genetic algorithm for solving rich VRP. *Applied Artificial Intelligence*, *28*(10), 957–991. doi:10.1080/08839514.2014.927680

Zhang, W., Yang, D., Zhang, G., & Gen, M. (2020). Hybrid multi-objective evolutionary algorithm with fast sampling strategy-based global search and route sequence difference-based local search for VRPTW. *Expert Systems with Applications*, *145*, 113151. doi:10.1016/j.eswa.2019.113151

Zhu, Y., Lee, K. Y., & Wang, Y. (2021). Adaptive elitist genetic algorithm with improved neighbor routing initialization for electric vehicle routing problems. *IEEE Access : Practical Innovations, Open Solutions*, *9*, 16661–16671. doi:10.1109/ACCESS.2021.3053285

Hongjie Wan received his B.S degree from Nanjing Normal University taizhou College in 2019. He is currently a graduate student at Nanjing University of Posts and telecommunications. His research interests include software engineering, distributed system, database and operating system.

Junchen Ma received his B.S degree from Nanjing University of Posts and telecommunications in 2021. He is currently a graduate student at Nanjing University of Posts and telecommunications. His research interests include Vehicle Routing Problem and Natural Language Processing.

Qiumei Yu received her B.S. degree in computer science and technology from Xi'an Jiaotong University City College in 2019. She is currently a graduate student at Nanjing University of Posts and telecommunications. Her main research interests include Artificial Intelligence and Natural Language Processing.

Guozi Sun is a profess or in the School of Computer Science and Technology at Nanjing University of Posts and Telecommunications, China. His research interests include blockchain forensics, digital forensics, and digital investi- gation. Sun received his Ph.D. in mechanical engineering and automation from Nanjing University of Aeronautics and Astronautics, China. He is a member of the IEEE Computer Society, ACM, China Computer Federation (CCF), Chinese Institute of Electronics (CIE), and Information Security and Forensics Society (ISFS), China. Contact him at sun@njupt. edu.cn.

Hansen received his M.S.degree from Nanjing University of Posts and Telecommunications. His research interests include machine learning natural language processing, fake news detection and digital data forensics.

Huakang Li received his PhD degree in Computer Science from School of Computer Science and Engineering, University of Aizu in 2011. Currently, Huakang Li works as an associate professor at the School of Artificial Intelligence and Advanced Computing, XJTLU Entrepreneur College (Taicang). From May 2011 to Sep. 2013, Huakang Li worked as a postdoctoral research fellow in the Department of Computer Science and Engineering, Shanghai Jiaotong University. During that period, Huakang Li went to Ali Cloud Computing Ltd. as a visiting researcher to develop the National 863 Project. From Sep. 2013 to July 2020, Huakang Li was a lecturer at School of Computer Science, Nanjing University of Posts and Telecommunications. His research interests are focused on artificial intelligence and big data mining, specifically natural language processing, social computing, knowledge engineering, knowledge graphs and the related domain applications.