Traffic & Transportation

Enhanced H-GASA Algorithm for Efficient Path Optimisation in Online Ride-Hailing Carpooling

Min GUO[1], Xinglin FENG[2], Haixiao WANG[3]

Original Scientific Paper
Submitted: 10 Sep 2024
Accepted: 24 Feb 2025

[1] guomin@imau.edu.cn, School of Energy and Transportation Engineering, Inner Mongolia Agricultural University, Hohhot, China
[2] 651132326@qq.com, School of Energy and Transportation Engineering, Inner Mongolia Agricultural University, Hohhot, China
[3] Corresponding author, wanghx@imau.edu.cn, School of Energy and Transportation Engineering, Inner Mongolia Agricultural University, Hohhot, China

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT

Online ride-hailing carpooling services often need help with bottlenecks, such as delayed response times and low computing efficiency, which negatively impact user experience and platform operation. Current path optimisation algorithms also need help managing real-time dynamic requests and large-scale computing challenges. In this respect, this paper proposes a bi-directional path-based online taxi carpooling optimisation model that considers road network conditions and time window constraints. It minimises operating and passenger travel costs under multiple constraints. The fitness assessment and acceptance criteria are optimised based on a genetic algorithm, combined with the temperature regulation mechanism of simulated annealing, and a hybrid genetic-simulated annealing algorithm (H-GASA) is proposed. In addition, this paper brings the parallel repair mechanism and accelerates the solution repair process using modern multi-core processors and parallel computing framework, significantly improving the solution efficiency. The experimental results show that the H-GASA algorithm substantially reduces the passenger travelling time and vehicle operating cost under multiple time windows, which is better than the existing algorithms and effectively solves the common premature convergence problem of genetic algorithms. The study verifies the efficiency and reliability of the algorithm in practical applications and provides strong technical support for optimising online carpooling services.

KEYWORDS

route planning; online ride-hailing carpooling service; H-GASA algorithm; time window thresholds.

1. INTRODUCTION

The rapid development of online ride-hailing services has become an essential part of modern urban travel, significantly improving the convenience of travelling and promoting the popularity of the sharing economy model. In particular, "one-to-many" carpooling services have become popular as they provide passengers with a more affordable travel option by sharing rides on the same trip. However, these services continue to encounter numerous challenges in practical applications, including improving the matching accuracy between passengers and drivers, ensuring the fairness of cost allocation, and avoiding service quality degradation when the expected travel time deviates from the actual service time. At the same time, it is critical to ensure the efficiency of the algorithms in complex and dynamic scenarios. Effectively addressing these challenges will optimise the overall efficiency of carpooling services and play an essential role in promoting sustainable transport development. Therefore, conducting an in-depth study on the optimisation strategies for these problems is of great practical significance.

Currently, many scholars have conducted in-depth research on related problems in online ride-hailing services, particularly with a particular focus on dynamic matching: for example, Bei Xiaohui et al. [1] proposed an approximation algorithm for solving the carpooling problem with the vehicle-trip assignment. Simonetto et al. [2] developed an efficient dynamic carpooling algorithm based on a linear allocation problem, which transforms the carpooling problem into a linear allocation problem between the fleet and the customer journey requests. Guo Yuhan et al. [3] studied the instantaneous vehicle sharing problem by developing a multi-strategy solution space graph search algorithm, which improves the efficiency of vehicle resource utilisation and reduces the environmental impact. Tafreshian et al. [4] developed graph partitioning methods to study the matching problem in dynamic carpooling systems to reduce the overall computational complexity of the problem and provide a near-optimal solution. Pang Mingbao et al. [5] developed a method for planning bus routes in large cities based on the demand density of complex networks and effectively improved the network's operational efficiency and service level through an ant colony optimisation algorithm. Li Xuefeng et al. [6] proposed a flexible order scheduling optimisation method with a one-to-three matching strategy considering a hybrid online car service model, which significantly improves service efficiency and customer satisfaction through multi-objective optimisation.

Dynamic matching provides technical support for carpooling services, while rational cost allocation encourages both passengers and drivers to participate. Current research on cost allocation includes the work of Ma Jie et al. [7], who used a variational inequality model to study origin-destination-based incentive pricing strategies in urban transportation networks, which effectively reduce travel costs and decrease intentional detours. Guo Junhua et al. [8] effectively solved the comprehensive optimisation problem of vehicle opening and cargo flow distribution using a collaborative optimisation model and numerical experiments and sensitivity analysis. Ning et al. [9] developed a joint bus scheduling and route planning framework that effectively reduces operational costs and improves passenger experience. Sun Jiahui et al. [10] developed a joint order scheduling and driver relocation approach, which optimises online taxi services' long-term efficiency and fairness while considering cost constraints. Feng Yiding et al. [11] developed a two-stage stochastic matching and pricing framework for online car rental platforms, which significantly improves operational efficiency and market responsiveness by batching requests and optimising supply efficiency and market efficiency using competitive algorithms. The constraints of time windows have a significant impact on travel costs. Guo et al. [12] addressed time window-based service constraints by simultaneously optimising vehicle routes and passenger assignments using a mixed-integer programming model and comparing different scenarios of partial and full-service coverage. Kashyap et al. [13] proposed the RWPSO algorithm to solve vehicle routing problems with time window constraints, achieving significant optimisation. Liu M et al. [14] introduced a hybrid brainstorming optimisation algorithm, effectively addressing dynamic vehicle routing problems under time window constraints.

Currently, many scholars have employed heuristic algorithms to address the multi-objective and multi-constraint path planning problem in carpooling services. For instance, Ruiz E et al. [15] proposed an improved ACO algorithm where the model-solving efficiency was optimised to a large extent. Li Nanman et al. [16] developed a hybrid multi-objective genetic algorithm for solving the bi-objective time-window assignment vehicle path problem, which effectively improves customer satisfaction and reduces the expected distribution cost; Jin Yuming et al. [17] proposed a prediction model based on stacked integrated learning to enhance the operational efficiency of the online taxi system. Feng Guiyun et al. [18] developed a heuristic algorithm to compare the efficiency of the matching mechanism with the traditional street-hailing mechanism under certain system parameter settings. Wang Qinyu et al. [19] used a two-stage framework to analyse and predict the spatial interaction gravity between cities. Gao Tianyang et al. [20] developed a hybrid variable neighbourhood search algorithm combining simulated annealing and variable neighbourhood search, which effectively reduces operational costs and improves computational efficiency by optimising electric flexible feeder bus services. These studies show that heuristic algorithms are particularly important in online ride-hailing carpooling services because they handle complex and dynamically changing optimisation problems.

Based on the existing research, this paper fully integrates the multiple constraints of the actual operation of the online carpooling problem and makes the following contributions.

— Comprehensive consideration of multiple constraints: This study incorporates constraints such as time windows and road network conditions into the model design, indirectly reflecting traffic conditions through average trip times. This approach ensures the minimisation of both operational costs and passenger travel costs under multiple constraints, thereby enhancing the practicality and accuracy of route planning.

— Enhanced computational efficiency of the algorithm: Existing large-scale path optimisation algorithms possess theoretical advantages but often fail to achieve desired results in practical applications due to high computational complexity and significant resource consumption. This research introduces a parallel repair mechanism that improves the algorithm's computational efficiency in large-scale environments.

— Dual-objective optimisation: Currently, there is substantial research focused on enhancing either the interests of one party within ride-hailing systems or the overall system efficiency, such as reducing operational costs, improving passenger experience, optimising matching efficiency and elevating service quality. However, systematic analyses that balance multiple objectives simultaneously remain insufficient. This study comprehensively considers the interests of both drivers and passengers, aiming to minimise operational costs and passenger travel costs through dual-objective optimisation. By achieving coordination and balance between these objectives, the study provides a more comprehensive and balanced solution.

With the emergence of online car rental platforms, researchers have focused on studying and identifying the spatiotemporal characteristics of online car rental trips in urban areas [21]. Similarly, the growing popularity of online ridesharing has prompted investigations into its structural and operational impacts on public transport systems [22]. This study builds on existing research and addresses multiple constraints, such as travel time windows, passenger satisfaction and costs associated with vehicle transfers. It develops a "one-to-many" online carpooling model (OMC-HCMC) that balances vehicle operating costs and passenger travel costs to ensure a fair distribution while minimising total travel costs.

This paper designed an enhanced genetic algorithm-simulated annealing hybrid algorithm (H-GASA) to handle the model, improving computational efficiency and providing real-time feedback for vehicle routing. The effectiveness of this optimisation algorithm is validated through a comparative analysis with various other algorithms and different time window settings. Ultimately, this study proposes a practical solution for real-world online carpooling that balances operational efficiency and passenger satisfaction.

2. PROBLEM DESCRIPTION AND MODELLING

As shown in *Figure 1*, the "one-to-many" ride-hailing carpooling route planning problem involves determining the optimal route for a ride-hailing car to pick up and drop off multiple passengers from their respective starting points to a designated endpoint [23]. In this process, a variety of factors must be considered, including the expected pick-up and drop-off time of each passenger, the capacity constraints of the vehicle and the operating costs associated with the car's operation. Developing a route planning model that incorporates these various factors can improve service efficiency, minimise diversion distances and increase passenger satisfaction while reducing energy consumption and operating costs.

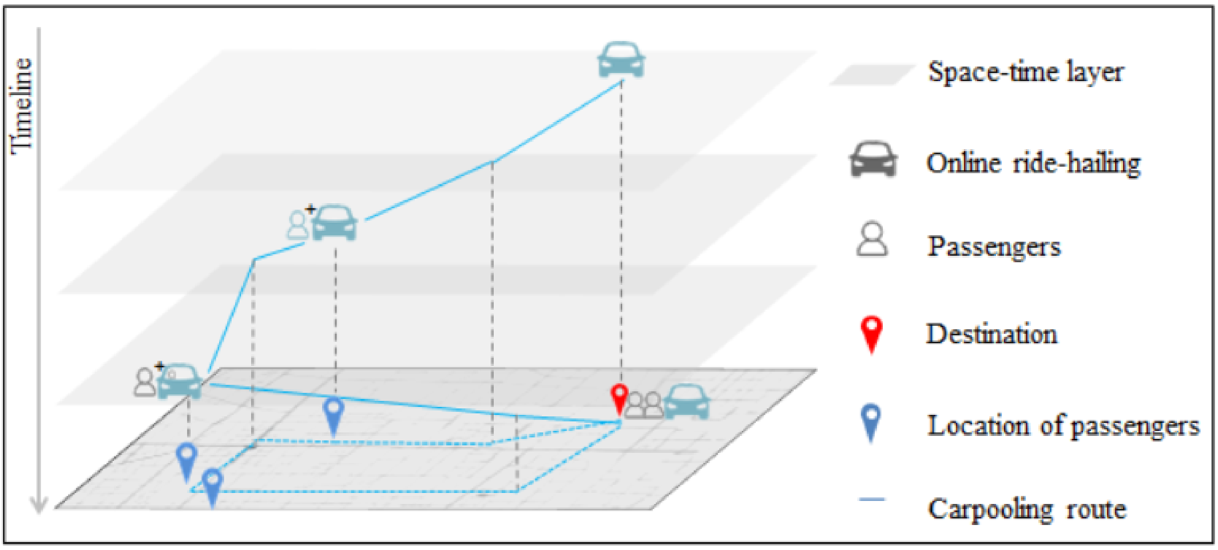


Figure 1 – Online ride-hailing carpooling route diagram

2.1 Model assumptions

— Passenger's real-time demand is randomly generated within the service area, and the location and travel time of each passenger's demand is known before the vehicle departs.
— All passengers who have booked a trip can arrive at the flexible pickup point within the vehicle's designated time window.
— Passengers at the same flex station may only be served by one vehicle once during that period.

2.2 Model parameter

In this paper, based on the above assumptions, the OMC-HCMC model is established. The model parameters and variables are shown in *Table 1* below:

Table 1 – Model symbol description

Variables	Meanings	Variables	Meanings
O	Passenger terminal stations.	$t_o(a)$	Departure time of vehicle a at passenger terminal o.
Z	Flexible site collection, $Z=(1,2,\dots,n-2,n-1,n)$	t_n^w	Represents the waiting time at station n, which can be divided into vehicle waiting time $t_n^w(a)$ and passenger waiting time $t_n^w(b)$. $in\ min$.
R	Requirements site collection, $R \in \{O \cup Z\}$	$t_n(a)$	Time vehicle a started service at site n, $t_n(a)=t_{n-1}(a)+t_{n-1}^n(a)+\ t_{n-1}^w(r)$.
M	Total number of vehicles available	$t_l(a)$	Latest time for vehicle a to arrive at a passenger hub site.
V	The set of vehicle kinds $V \in (V_1,V_2,V_3 \dots V_k)$	$b_n^{o,d}$	The number of passengers picks off b_n^o and pick up b_n^d at the n station. $in\ persons$.
N_k	The number of vehicles of a type V_k	$b_n(a)$	The number of passengers in vehicle a before picking up at demand point n, $in\ persons$.
a	Online ride-hailing vehicles involved in the service, numbered as a, $a \in (\sum_k N_k * V_k)$	T_n	The passenger time window in station n, $T_n=[E_n,L_n]$, where E_n is the start time and L_n is the end time. (soft time window).
C_k	Vehicle costs for model V_k, divided into origination costs $C_k(f)$ and fuel consumption costs $C_k(r)$, in ¥	f	The penalty factor $f \in (f_e,f_l)$, where f_e is the early arrival penalty factor, and f_l is the late arrival penalty factor.
C_b	The in-vehicle time cost per unit time for passengers	Q_k	Rated passenger capacity of vehicles V_k, $in\ persons$.
v_k	The average operating speed of online ride-hailing vehicles with numbered type V_k, $in\ km/h$	δ	Detour factor.
d_n^{n+1}	The shortest path distance based on the road network, $in\ km$	$x_n^{n+1}(a)$	Vehicle a connecting from station n to $n+1$ takes 1, otherwise 0.
$t_n^{n+1}(a)$	The average time for vehicle a to travel from station n to $n+1$ based on the road network, $in\ min$.	$y_n(a)$	Vehicle a connects to station n takes 1, otherwise 0.

2.3 Modelling

$$W=\min\,(W_1+W_2) \tag{1}$$

$$W_1=C_k(f)\sum_{n\in Z} x_o^n(a)+f\cdot C_k(r)\sum_{n\in Z} x_n^{n+1}(a)\cdot\left(t_n^{n+1}(a)+t_n^w(a)\right) \tag{2}$$

$$W_2 = C_b \cdot \delta \sum_{n \in Z} b_n(a) \cdot x_n^{n+1}(a) \cdot (t_n^{n+1}(a) + t_n^w(b)) \tag{3}$$

$$f = \begin{cases} f_e \cdot [E_n - t_n(a)], t_n(a) < E_n \\ 0, \qquad E_n \leq t_n(a) \leq L_n \\ f_l \cdot [t_n(a) - L_n], t_n(a) > L_n \end{cases} \quad (st. \forall n \in Z, E_u \& L_u \in T_n) \tag{4}$$

In this model, *Formula 1* is the objective function, which consists of two components, vehicle operating cost W_1 and passenger travel cost W_2. *Formula 2* and *3* are computational expressions for W_1 and W_2. Vehicle operating costs W_1 include fixed dispatch costs $C_k(f)$ and vehicle operating costs $C_k(r)$, vehicle operating costs $C_k(r)$ are related to the operating time of the vehicle. Additionally, if the flexible vehicle fails to accommodate the passenger's time-window needs during operation, a penalty factor f must be taken into account.

Passenger travel costs comprehensively reflect service quality and passenger satisfaction, incorporating essential factors such as service reliability, comfort and safety. These costs are quantifiable and easy to optimise, making passenger travel costs the optimisation objective. Consequently, transportation services are designed and evaluated to improve system efficiency, optimise resource allocation and enhance the overall travel experience for passengers. The passenger travel cost is calculated by multiplying the product of passengers' in-vehicle time $(t_n^{n+1}(a) + t_n^w(b))$ and the detour index δ caused by carpooling, with the number of passengers $b_n(a)$ at the corresponding station and the in-vehicle time cost per unit time C_b.

The objective function of the model is a multi-objective optimisation problem, involving both linear and nonlinear components. The linear components represent the operational costs, while the nonlinear components, such as passenger time windows and detours, reflect the dynamic nature of traffic conditions and the constraints imposed on vehicle routing. Among them, traffic conditions are indirectly reflected by the average travel time $t_n^{n+1}(a)$.

$$\sum_{n \in Z} x_n^{n+1}(a) = \sum_{n \in Z} x_{n+1}^n(a) = y_n(a) \tag{5}$$

$$\sum_{n \in Z} x_o^n(a) = 1 \tag{6}$$

$$\sum_{n \in Z} x_n^o(a) = 1 \tag{7}$$

Formula 5 is a flexible site constraint that ensures when a flexible site n is serviced, the drive into and out of each flexible site is the same vehicle a which ensures the continuity and orderliness of the vehicle in the route travel. *Formula 6* and *7* are origin-destination constraints that guarantee each route must start from and eventually return to the passenger hub station.

$$\sum_{i=1}^{k} N_i \leq M \tag{8}$$

$$\sum_{n \in Z} x_o^n(a) = \sum_{n \in Z} x_n^o(a) = \sum_{i=1}^{k} N_i \tag{9}$$

Formula 8 and *9* define constraints on fleet size. *Formula 8* limits the number of vehicles participating in the connection not to exceed the total available fleet size M. On the other hand, *Formula 9* enforces a conservation rule for the operating fleet, ensuring that the number of vehicles returning to the passenger hub $(\sum_{n \in Z} x_o^n(a))$, matches the number of vehicles departing from the passenger hub $(\sum_{n \in Z} x_n^o(a))$. Additionally, this value must correspond to the total number of vehicles for model$V$, denoted as $\left(\sum_{i=1}^{k} N_i\right)$.

$$\sum_{n \in R} b_n^o y_n(a) \leq Q_k, (st.\ k \text{ is the vehicle type corresponding to } a) \tag{10}$$

$$\sum_{n \in R} b_n^d y_n(a) \leq Q_k, (st.\ k \text{ is the vehicle type corresponding to } a) \tag{11}$$

$$b_n^o + b_n^d + b_n(a) \leq Q_k , (st.\ \forall n \in R) \tag{12}$$

$$x_n^{n+1}(a) \cdot (t_n^{n+1}(a) + t_n^w(a)) \leq \delta \cdot \frac{d_n^{n+1}}{v_k}, (st.\ \forall n \in R) \tag{13}$$

Formulas 10-12 are vehicle passenger capacity constraints that ensure the number of passengers on board a vehicle during operation does not exceed the vehicle's rated capacity. *Formula 13* is the maximum travel time constraint between stops, ensuring that each vehicle's actual travel time (both driving and waiting) does not exceed the allowable time for the shortest path as bounded by the detour factor δ.

$$\sum_{i \in a} y_n(a) = 1, (st.\ \forall n \in R) \tag{14}$$

$$\sum_{n \in Z} x_n^{n+1}(a) \cdot (t_n^{n+1}(a) + t_n^w(a)) \leq \delta \cdot \sum_{n \in Z} t_n^{n+1}(a) \tag{15}$$

$$t_o(a) + \sum_{n \in Z} x_n^{n+1}(a) \cdot (t_n^{n+1}(a) + t_n^w(a)) \leq t_l(a) \tag{16}$$

Formula 14 is a vehicle-order matching constraint that ensures passenger demand is satisfied at every flexible site; *Formula 15* and *16* ensure the rationality of the carpooling route. *Formula 15* represents a flexible route detour constraint, which prevents vehicles from excessive detours along the entire route within the service area. *Formula 16* is a time window constraint for the terminal station, ensuring that the connection vehicles can return to the passenger station on time or ahead of schedule. This guarantees that passengers arriving at the hub station are on time and can receive the services of hub station buses.

3. ALGORITHM DESIGN

Genetic algorithms are powerful heuristic search methods with strong scalability, and they are widely used in vehicle path planning. However, they usually encounter the problem of premature convergence, while simulated annealing algorithms have high computational efficiency and can effectively avoid local optimal solutions.

3.1 H-GASA algorithm

Given the advantages and disadvantages of the algorithms in existing studies, this paper proposes a hybrid genetic-simulated annealing algorithm (H-GASA). The algorithm combines the global search capability of the genetic algorithm and the local optimisation property of the simulated annealing algorithm to cope with the quality requirements of multi-objective, dynamics, and reconciliation in the "one-to-many" online ride-hailing carpooling path planning problem. Compared to traditional algorithms, the algorithm can balance solution quality and computational efficiency more effectively, providing superior solutions. Specific improvements include:

1) Improvement of the fitness function to reflect the actual quality of solutions more accurately. This optimisation improves the algorithm's ability to identify high-quality solutions during the search process, ensuring that potential solutions are evaluated and selected more effectively.
2) Combining the acceptance criteria of the simulated annealing algorithm with the genetic algorithm enhances the algorithm's ability to leapfrog local optima. This improvement allows the algorithm to be more flexible in the global search process and helps to avoid falling into local optima, thus improving the quality of the solution.
3) The parallel repair mechanism based on the road network combined with real-time road network information can quickly repair infeasible solutions and optimise path planning, thus improving the algorithm's performance under complex dynamic constraints.

3.2 H-GASA algorithm flow

The algorithm process is based on the basic process of a genetic algorithm with the addition of a simulated annealing mechanism. The specific process will be described in detail in the following section.

Coding operation

This paper proposes an algorithm solution set encoding method based on exact numbers for solving online carpooling path planning models. This method accurately describes the key elements in path planning (such as the order of passenger demand points and vehicle driving paths) by expressing the problem solution as a numerical sequence. Exact digital encoding can dynamically adjust the structural characteristics of the solution during the algorithm optimisation process, deeply explore the search space, avoid falling into local optimality, and thus improve the global search ability of the algorithm. With this encoding method, the model can optimise path planning more efficiently, balance passenger demand and vehicle scheduling costs and achieve a globally optimal solution in carpooling.

Population initialisation

The optimal population of the genetic algorithm will be affected by the quality of the initial population, a high-quality initial population can reduce the difficulty of the algorithm to search for the optimal solution, the steps to construct the initial solution are as follows:

— Step 1: Generate the initial chromosome. Assuming that the number of flexible stations is n and the total number of flexible feeder vehicles is a, a chromosome scheme r of length $n+a+1$ is randomly generated.
— Step 2: Constraint judgement. Vehicle passenger constraints (*Formula 10*), (*Formula 11*) and (*Formula 12*) and endpoint time window constraints (*Formula 16*) are judged for this scheme, and a new chromosome scheme r is regenerated if they are not satisfied until the constraints are satisfied.
— Step 3: Add to population. Add the program r^* that meets the constraints to the population q_0. If the initial population size is too large, it will reduce the efficiency of the algorithm; if it is too small, the algorithm is prone to "early convergence", which is usually taken as a value between 50-200.

Adaptation function

The fitness of the initial population was determined using the roulette wheel method. The fitness of subsequent chromosomal offspring, inherited from their parents, was assessed by calculating the improved fitness function.

Genetic design

1) Selection of individual parents

In the chromosome population pool, selection is based on chromosome fitness. The chromosome with high fitness is highly likely to be selected, and the chromosome with low fitness is unlikely to be selected. This principle also applies to the selection of the two parent chromosomes.

2) Crossover operation

To ensure a diverse combination of genes, three types of chromosome crossover are utilised: self-crossover, single-gene crossover between two chromosomes and genome crossover, as illustrated in *Figure 2a*.

3) Mutation operation

The crossover produces offspring that will mutate with a certain probability of mutation p_m, and the mutation process is shown in *Figure 2b*.

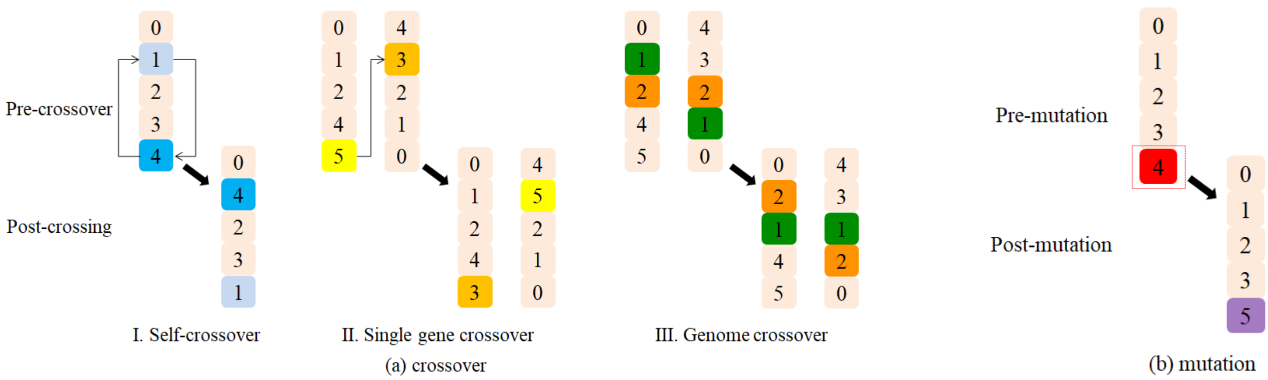


Figure 2 – Chromosome crossover and mutation schematic

Repair mechanism

This paper develops a parallel repair mechanism based on the actual road network to improve the algorithm's efficiency and solve complex problems. The mechanism integrates multi-core processors and parallel computing frameworks such as CUDA to distribute the repair tasks to multiple threads while considering time windows, capacity and path connectivity constraints. The repair mechanism uses the road network model to verify path connectivity, adjust node order, optimise task allocation and dynamically plan routes with real-time traffic data.

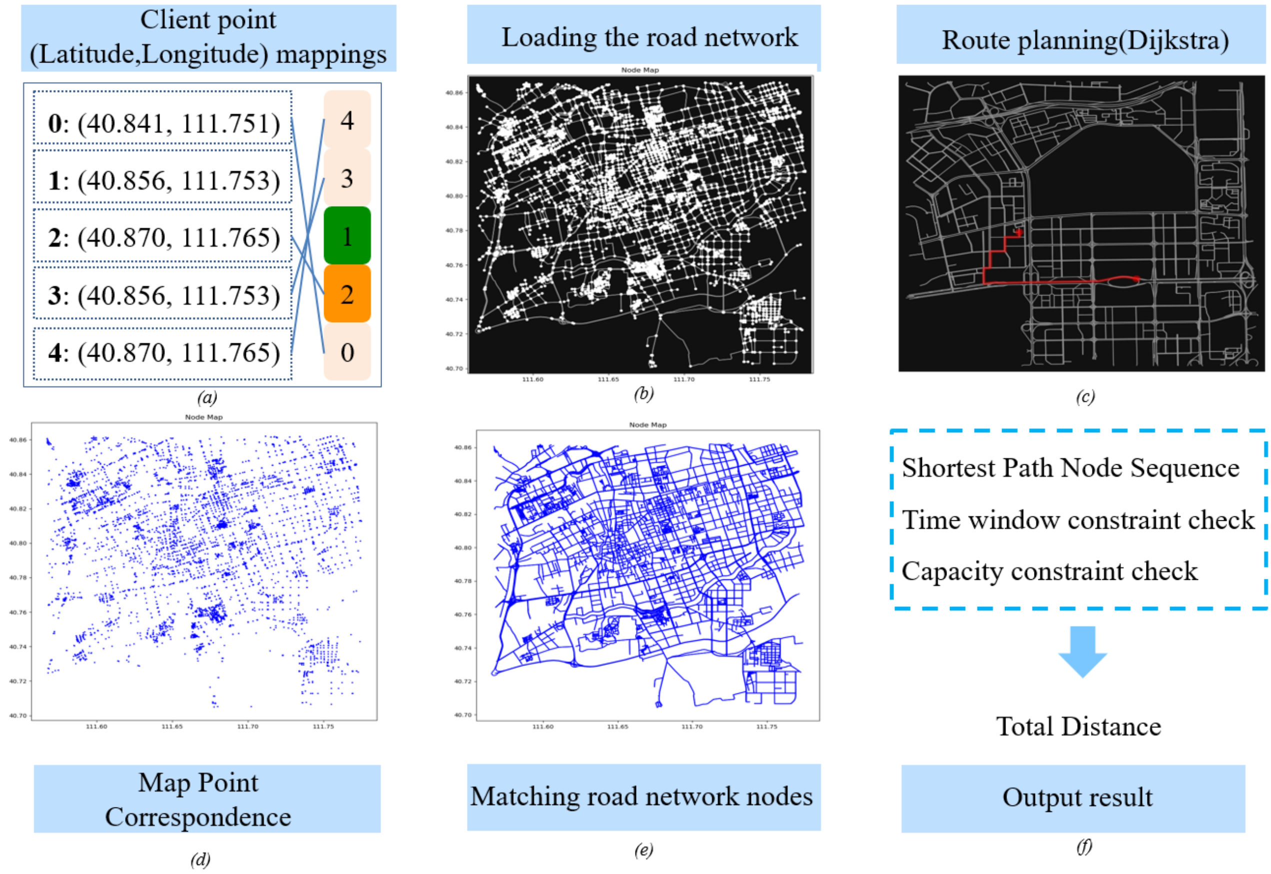


Figure 3 – Flowchart of the repair mechanism

In the implementation process, we use OSMnx and NetworkX tools to extract actual road network data and evaluate capacity. First, a chromosome scheme from the chromosome population is selected, and the actual position mapping of each gene is obtained (*Figure 3a*). Next, we extract the road network information of the target area (*Figure 3b*) and generate road nodes (*Figure 3c*) and actual roads (*Figure 3d*). Subsequently, we analyse the traffic attributes of the road, including one-way streets, speed limits and closures, to determine the path's capacity. Then, the shortest path algorithm plans the route according to weights such as road length and travel time (*Figure 3e*).

Simulated annealing design

The key to the effectiveness of the simulated annealing algorithm in jumping out of the local optimal solution for global search is the consideration of temperature t_z in the acceptance criterion. In this paper, simulated annealing has been performed to improve the fitness and acceptance criterion of individuals who produce offspring after genetic manipulation.

1) Initial temperature setting

   The initial temperature t_z should be set to a large value to ensure that there can be a large number of $P=1$ or $P\approx 1$ when the genetic operation can satisfy the requirement that the probability of acceptance of the offspring P meets the acceptance criterion, (i.e. $T_x \geq T_n$), thus, some offspring whose fitness function is not superior to the parent's is accepted, acting as a global search.

2) Acceptance probability
   The acceptance probability uses an improved acceptance criterion (*Formula 19*). This criterion takes into account the effect of temperature, allowing the algorithm to adapt to different temperature levels during the search process and enhancing the global search capability.
3) Cooling process
   As the number of iterations of the algorithm increases, the temperature will get lower and lower, and the cooling function uses isoperimetric cooling (*Formula 17*);

$$T_x = T_0 \cdot \alpha^x \tag{17}$$

   where x is the current number of iterations, T_x is the current simulated annealing temperature, T_0 is the initial simulated annealing temperature, α is the cooling coefficient of the simulated annealing algorithm and its value is set to 0.95 in this paper.

Improvement of the fitness function

In this paper, we improve the fitness function of the genetic algorithm (*Formula 18*) by using the calculation of acceleration and combining it with the core temperature t_z of the simulated annealing algorithm;

$$g_r(t_z) = \exp\left(\frac{g_b(t_z) - g_a(t_z)}{t_z}\right) \tag{18}$$

where t_z is the temperature at this point of simulated annealing, g_r represents the current fitness of the solution, g_b denotes the fitness of the solution before crossover and mutation and g_a indicates the fitness of the solution after crossover and mutation. The fitness function's improvement dynamically adjusts the fitness assessment's sensitivity by combining the fitness acceleration with the current simulated annealing temperature t_z. At the beginning of the algorithm, even higher temperatures are effective in limiting the range of variation of the fitness, thus preventing the algorithm from falling into a local optimum solution prematurely. As the temperature gradually decreases as the iteration progresses, the fitness acceleration will be more significantly affected by the difference between the fitness of the child and the parent. This adjustment mechanism enables the fitness function to improve the fitness value as the temperature decreases rapidly, thus improving the convergence speed and efficiency of the algorithm.

Improvement of the acceptance criterion

Based on the improvements mentioned above, the acceptance criterion has been optimised. Unlike the simulated annealing algorithm, which uses the temperature t_z directly to determine whether to accept a new individual, we have introduced (*Formula 19*) as the acceptance probability.

$$P = \min\{1, g_r(t_z)\} \tag{19}$$

P represents the acceptance probability, while $g_r(t_z)$ denotes the current fitness function value. According to this criterion, $g_r(t_z)$ is calculated using *Formula 19*, when the fitness of the offspring is better than that of the parent, $g_r(t_z)$ exceeds 1, resulting in an acceptance probability of 1. This guarantees that a high-quality solution will be selected, and when the fitness of the offspring is worse than that of the parent, $g_r(t_z)$ is less than 1, they will be selected. Conversely, if the fitness of the offspring is worse than that of the parent, $g_r(t_z)$ is less than 1, and the acceptance probability becomes equal to $g_r(t_z)$. In this case, the probability of accepting the offspring decreases, but there is still a chance for it to be selected.

Initially, the high temperature causes the acceptance probability to be close to 1, allowing many offspring to be accepted even if they are not optimal solutions. As iterations proceed and the temperature decreases, the fitness difference between the offspring and the parent significantly affects the acceptance probability, gradually tending to select higher-quality offspring. This process increases the possibility of finding the global optimal solution.

3.3 Algorithmic process

By improving the initial population construction, fitness function, simulated annealing combination and acceptance criterion, the H-GASA algorithm can more effectively avoid the local optimal solution when dealing with the complex "one-to-many" online carpooling path planning problem and significantly improve the computational efficiency, providing a better solution. The operation flow is shown in *Figure 4*.

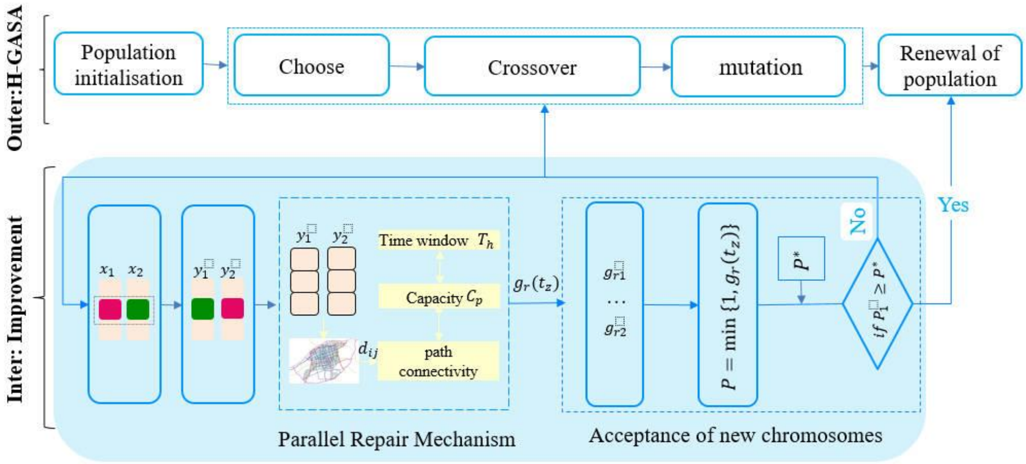


Figure 4 – H-GASA algorithm operation flowchart

4. CASE STUDY

To evaluate the effectiveness of the optimisation algorithm, we selected a ride demand scenario with a common destination in the road network. Hohhot East Station is the largest railway passenger station in the Inner Mongolia Autonomous Region, and many passengers need to take the same train. Shortly before the train departs, many trips to the East Station will occur in the surrounding areas. In this context, online carpooling services can effectively solve these concentrated travel needs and provide a convenient transportation solution.

Our research focuses on the roads surrounding Hohhot East Station. The designated range for the online taxi service extends from the coordinates (111.775°E, 40.862°N) to (111.687°E, 40.805°N). Based on our traffic survey, the distribution of various tourist attractions is illustrated in *Figure 5*. The algorithm was developed using Matlab R2022a and PyCharm 2024, and it runs on an Intel(R) Core(TM) i7-8750H CPU at 2.21 GHz with 16 GB of RAM.

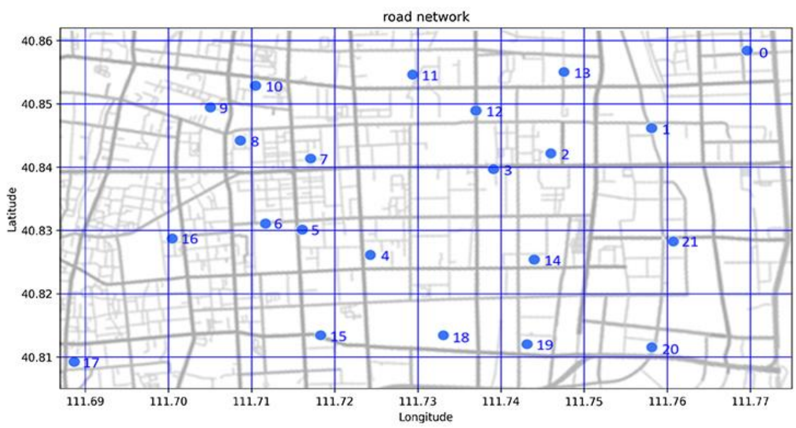


Figure 5 – Distribution map of the study area road network and stations

4.1 Case data

Hohhot East Station, marked as Station 0 in *Figure 5*, is both the end point of passenger travel and the starting point of online car-hailing service. The remaining 21 stations are flexible stops for passenger travel, including both scheduled stations and real-time demand stations. The service starts at 8 a.m., and vehicles travel in the service area according to the historical average speed of the road section. The distance between the stations is calculated using the road network from OpenStreetMap (https://www.openstreetmap.org/). We have established four different time window threshold intervals to enhance vehicle punctuality. The first time window threshold is set at 3.6 minutes, with each subsequent time window increasing by 1.2 minutes. As illustrated in *Table 2*, the starting value for the time window at each station is determined by the vehicle's actual arrival time and distance to other stations.

Table 2 – Starting values of time windows for each station

Station number	Starting moment	Station number	Starting moment	Station number	Starting moment
1	8:28	8	8:19	15	18:18
2	8:25	9	8:20	16	8:20
3	8:23	10	8:17	17	8:24
4	8:21	11	8:21	18	8:13
5	8:17	12	8:22	19	8:27
6	8:18	13	8:25	20	8:30
7	8:17	14	8:27	21	8:32

In this case, the main parameters are set as follows. Population size ($popsize$) was 200, the number of iterations (N) was 1,500, the probability of crossover (P_c) and the probability of variation (P_m) were 0.99 and 0.3. The initial temperature (T_0) for simulated annealing was $1{,}000{,}000°C$, the cooling factor (α) of 0.97; the tardiness penalty factor (m_1) was $¥33.15/min$, early arrival penalty factor (m_2) was $¥1{,}000/min$. The starting cost of a vehicle (C_1) is $¥2.28/veh$, the vehicle operating cost (C_2) is $¥46.8/km$ (data from local cab business operation data research).

4.2 Analysis of operational results

This study presents a comparative analysis of commonly employed heuristic algorithms, to evaluate their effectiveness. The selected comparative algorithms include the H-GASA algorithm, genetic simulated annealing algorithm (GSA), genetic algorithm (GA), ant colony algorithm (ACO) and particle swarm algorithm (PSO). By deeply analysing the performance of these algorithms in path planning, computational efficiency and resource utilisation, we proved the practical application value of the proposed algorithm.

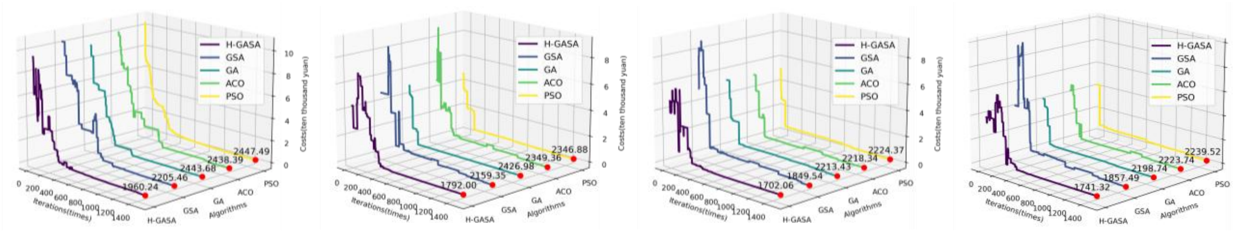
Results of the service reservation only passenger algorithm run

Table 3 gives the total cost and number of vehicles required for the optimal operating route in different time windows. The results show that the H-GASA algorithm outperforms other algorithms in all time windows, showing a clear advantage in operating costs. For example, in the first time window, the total cost of H-GASA, \$1,960.24, is significantly lower than that of GSA, \$2,205.46, and GA, \$2,443.68, and this trend remains consistent in the six-time windows, proving the stability and superiority of the algorithm. As the threshold interval of the time window expands, the superiority of H-GASA becomes more apparent. In the time window 8 (threshold interval is 7.2 minutes), the algorithm only needs six vehicles to complete all service tasks. In contrast, GA, PSO and ACO algorithms lag behind H-GASA in terms of vehicle demand and operating costs. In addition, the total cost of H-GASA in this time window is lower than that of the GSA algorithm.

Table 3 – Comparison of running results of various algorithms

Arithmetic	Time window 1		Time window 2		Time window 3		Time window 4		Time window 5		Time window 6	
	Cost (¥)	Veh	Cost (¥)	Veh	Cost (¥)	Veh	Cost (¥)	Veh	Cost (¥)	Veh	Cost (¥)	Veh
H-GASA	1 960.24	9	1 964.82	9	1 792.00	8	1 702.06	8	1 741.32	8	1 716.38	8
GSA	2 205.46	9	2 201.35	9	2 159.35	9	1 849.54	8	1 857.49	8	1 844.91	8
GA	2 443.68	9	2 431.86	9	2 426.98	9	2 213.43	9	2 198.74	9	2221.44	9
PSO	2 438.39	9	2 428.14	9	2 349.36	9	2 218.34	9	2 223.74	9	2 208.15	9
ACO	2 447.49	9	2 436.61	9	2 346.88	9	2 224.37	9	2 239.52	9	2 219.35	9

As shown in *Figure 6*, the iteration curves of various algorithms show significant differences in the optimisation performance in different time windows. In *Figure 6a* and *6b*, the curves of H-GASA and GSA show a rapidly rising trend in the first 200 iterations. The rapid rise is due to the high-temperature stage, which accepts many non-optimal solutions. This stage helps prevent the algorithm from prematurely converging on a local optimum by efficiently exploring the solution space. From the 200th to the 600th iteration, H-GASA obtains a better solution with a faster overall convergence speed, showing a good balance between global optimisation and local search. On the contrary, the convergence curve of GSA shows instability in *Figure 6b* and *6c*, affecting the stability and quality of the solution. At the same time, in *Figure 6a* and *6d*, the GA and ACO algorithms converge too early, resulting in poor solution quality. After the 600th iteration, H-GASA entered the slow convergence stage; the curve tended to be stable and gradually approached the global optimum, while the curves of the GA and ACO algorithms often fluctuated, indicating that the two algorithms had poor stability after convergence and were easily affected by local search disturbances and further verified that H-GASA had better convergence performance and global search capabilities.



(a) Time window 1 *(b) Time window 2* *(c) Time window 3* *(d) Time window 4*

Figure 6 – Algorithm iteration process diagram

The lowest travelling cost solution finally obtained is under time window 4, with a total cost of $1,702.06. The vehicle's diversion coefficient under this solution is 1.28. The vehicle route is shown in *Figure 7*. The H-GASA algorithm can rapidly repair infeasible solutions under complex constraints and optimise vehicle path planning by introducing a parallel repair mechanism based on the road network and matching it with real-time information. This mechanism ensures the paths' feasibility and efficiency and helps improve the scheduling system's real-time response capability.

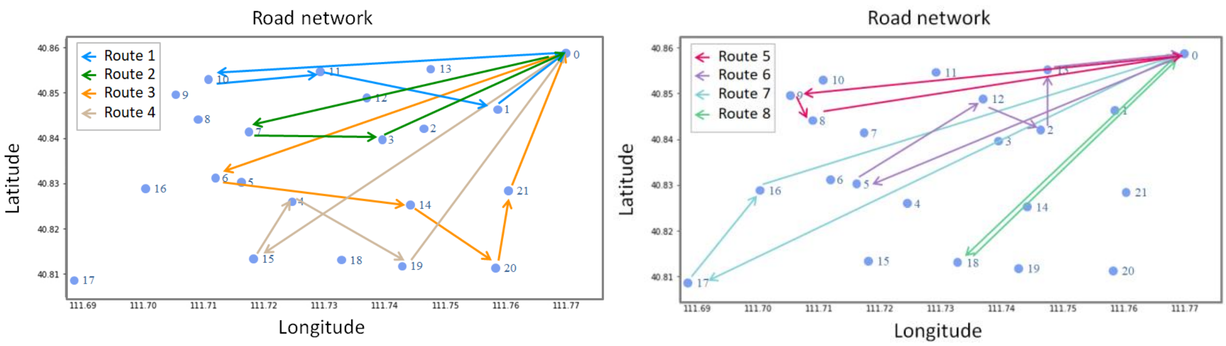
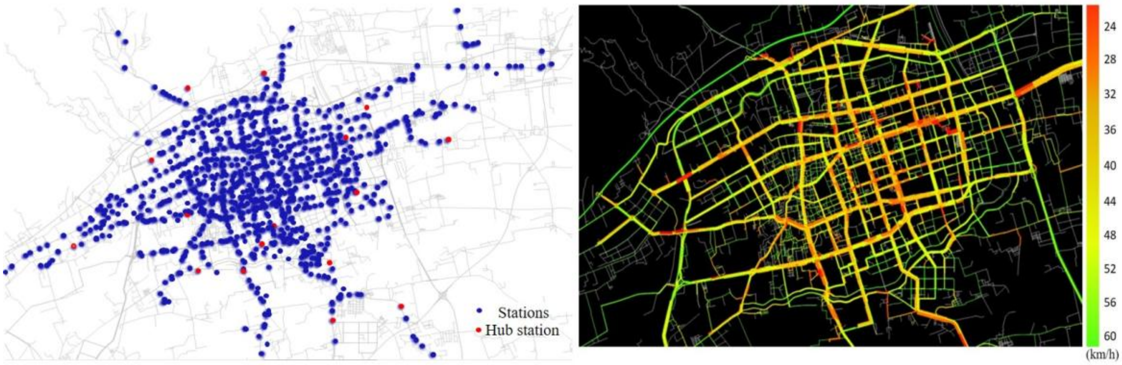


Figure 7 – Minimum cost operation roadmap (a total of 8 vehicles serve 8 routes)

Scalability experiments

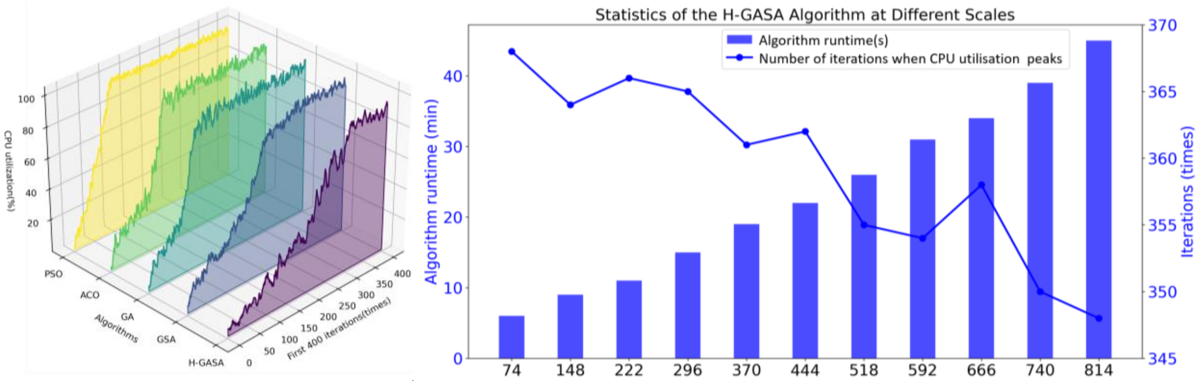
This study employs a series of progressively enlarged test cases to assess the algorithm's scalability. These cases evaluate the algorithm's runtime performance across the board. This study evaluates the algorithm's scalability using a series of progressively larger test cases. These cases systematically increase the input requirements to evaluate the runtime performance of the algorithm at different scene sizes – various scene sizes by systematically increasing the input dimensions. The data set is based on the public transportation data of Hohhot, and a road network model containing 16 public transportation hubs (shown in *Figure 8a*) and 814 stations is constructed. Assume that each station randomly generates 0 to 10 passengers to get on and off. *Figure 8b* shows traffic flow statistics collected in Hohhot City from 1 to 31 October 2021, showing the average speed distribution of road sections during off-peak hours (10:00-11:00). This case adopts a gradual and incremental approach to evaluating the algorithm's performance at different scales. This involves adding 74 demand points at a time, which helps avoid insufficient performance changes due to slow expansion or excessive performance differences due to rapid expansion.



(a) Site map *(b) Off-peak hour average speed*

Figure 8 – Expanded test case site map

In this case, the initial population size is set to 1,000, and the number of iterations is set to 1,500. Both are subsequently increased in the same proportion as the increase in the case size. The remaining parameters are kept unchanged.



(a) Runtime and CPU utilisation of different algorithms *(b) Statistics of the H-GASA algorithm at different scales*

Figure 9 – Algorithm performance metrics graph

Figure 9a shows that the peak CPU utilisation of the H-GASA algorithm occurs at the 348th iteration, which is significantly later than that of the benchmark algorithms such as GSA, PSO and ACO (220th, 120th and 175th iteration, respectively), reflecting its more substantial computational resource utilisation and global search capability. In contrast, PSO and ACO algorithms quickly reach 95% CPU utilisation in early iterations but then stagnate and suffer from premature convergence, which H-GASA effectively avoids by simulating annealing mechanisms and parallel repair strategies. As the data size increases, the CPU utilisation of all algorithms, including H-GASA, approaches 100% after the 400th iteration, showing the high load performance of the algorithms under complex constraints and large-scale scenarios. Meanwhile, H-GASA still shows apparent advantages in resource utilisation efficiency and robustness of optimisation results.

The results (*Figure 9b*) demonstrate that the H-GASA algorithm exhibits strong computational performance and resource utilisation efficiency across various test conditions. As the problem size extends from 74 to 814, the algorithm's running time increases from 6 to 45 minutes, showing an approximately linear growth trend. This indicates that the algorithm's computation time is proportional to the input size, further validating its efficient adaptation in large-scale datasets.

Meanwhile, as the problem size increases, the iteration position of the peak CPU utilisation gradually decreases from the 368th to the 348th, indicating that the algorithm can progressively optimise the allocation and utilisation efficiency of computing resources. Overall, these findings show that the H-GASA algorithm can balance computational performance and resource requirements in complex, large-scale scenarios.

5. CONCLUSION

This paper proposes an OMC-HCMC optimisation model that integrates a genetic algorithm and a simulated annealing algorithm (H-GASA). This model effectively optimises path planning and greatly reduces passenger travel and vehicle operating costs. The model achieves this goal by combining a parallel repair mechanism based on a road network and real-time traffic data. Experimental results show that the H-GASA algorithm performs well in multiple time windows and can quickly optimise the scheduling scheme. It overcomes the common premature convergence problem in genetic algorithms and improves the system's responsiveness and stability.

We apply the algorithm to real scenarios to test its scalability. The results show that the H-GASA algorithm can perform well even under large-scale datasets. Through gradually expanded test cases, the results highlight its potential for widespread application in real-world online carpooling services. In future research, the expression of traffic conditions will consider dynamically updating traffic fluctuation-related variables such as dynamic road congestion coefficients and average travel time in different periods to improve the model's mechanism for handling different traffic conditions. To improve computational efficiency, machine learning technology can be combined to improve the adaptability of the algorithm in dynamic and uncertain environments, consider environmental factors and verify the robustness of the algorithm through large-scale tests in different cities to explore improvement directions.

ACKNOWLEDGEMENTS

The authors thank the Transportation Engineering Laboratory of Inner Mongolia Agricultural University for providing data and assistance. This work was supported by the Inner Mongolia Natural Science Foundation (2024LHMS05024) and Inner Mongolia Higher Education Scientific Research Project (NJZY23112).

REFERENCES

[1] Xiaohui B, Shengyu Z. Algorithms for trip-vehicle assignment in ride-sharing. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2018;32(1). DOI: 10.1609/aaai.v32i1.11298.

[2] Andrea S, Julien M, Claudio G. Real-time city-scale ridesharing via linear assignment problems. *Transportation Research Part C: Emerging Technologies*.2019;101:208-232. DOI: 10.1016/j.trc.2019.01.019.

[3] Yuhan G, et al. A spatiotemporal thermo guidance based real-time online ride-hailing dispatch framework. *IEEE Access.* 2020;8:115063-115077. DOI: 10.1109/ACCESS.2020.3003942.

[4] Amirmahdi T, Neda M. Trip-based graph partitioning in dynamic ridesharing. *Transportation Research Part C: Emerging Technologies*. 2020;114:532-553. DOI: 10.1016/j.trc.2020.02.008.

[5] Mingbao P, Xing W, Lixia M. Transit route planning for megacities based on demand density of complex networks. *Promet-Traffic&Transportation*. 2022;34(1):13-23. DOI: 10.7307/ptt.v34i1. 3752.

[6] Xuefeng L, Mingyang D, Mei-Po K. Order dispatch optimization with considering flexible one-to-three matching strategy under hybrid ride-hailing service modes. *Transportation Letters.* 2023;15(10):1337-1353. DOI: 10.1080/19427867.2022.2150705.

[7] Jie M, et al. Ridesharing user equilibrium problem under OD-based surge pricing strategy. *Transportation Research Part B: Methodological.* 2020;134:1-24. DOI: 10.1016/j.trb.2020.02.001.

[8] Junhua G, Jiangyong W, Wuyang Y. Collaborative optimisation of high-speed rail express line planning and freight flow allocation considering multiple transportation modes and products. *Promet-Traffic& Transportation.* 2024;36(3):415-432. DOI: 10.7307/ptt.v36i3.531.

[9] Zhaolong N, Mengchu S, Xiping H. Online scheduling and route planning for shared buses in urban traffic networks. *IEEE Transactions on Intelligent Transportation Systems.* 2021;23(4):3430-3444. DOI: 10.1109/TITS. 2020.3036396.

[10] Sun J, et al. Optimizing long-term efficiency and fairness in ride-hailing under budget constraint via joint order dispatching and driver repositioning. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022,14 - 18 Aug.2022*, *New York, NY, USA,* 2022.p.3950–3960 DOI: 10.1145/3534678.3539060.

[11] Yiding F, Rad N, Amin S. Two-stage stochastic matching and pricing with applications to ride-hailing. *Operations Research*. 2024;72(4):1574-1594. DOI: 10.1287/opre.2022.2398.

[12] Rongge G, et al. Customized bus routing problem with time window restrictions: model and case study. *Transportmetrica A: Transport Science.* 2019;15(2):1804-1824. DOI: 10.1080/ 23249935. 2019.1644566.

[13] Kashyap GS, et al. Roulette-wheel selection-based pso algorithm for solving the vehicle routing problem with time windows. *ArXiv*. 2023; DOI: 10.48550/ arXiv.2306.02308.

[14] Mingde L, et al. A hybrid brain storm optimization algorithm for dynamic vehicle routing problem with time windows. *IEEE Access*.2023;11:121087-121095, DOI: 10.1109/ ACCESS. 2023. 3328404.

[15] Ruiz, E, et al. Solving the open vehicle routing problem with capacity and distance constraints with a biased random key genetic algorithm. *Computers Industrial Engineering*. 2019;133:207-219. DOI: 10.1016/ j.cie.2019.05.002.

[16] Manman L, Jian L, Wenxin M. A hybrid multi-objective genetic algorithm for bi-objective time window assignment vehicle routing problem. *Promet-Traffic &Transportation*. 2019;31(5):513-525. DOI: 10.7307/ ptt.v31i5.3057.

[17] Yuming J, et al. Demand forecasting of online car-hailing with stacking ensemble learning approach and large-scale datasets. *IEEE Access*. 2020;8:199513-199522. DOI: 10.1109/ ACCESS. 2020.3034355.

[18] Guiyun F, Guangwen K, Zizhuo W. We are on the way: Analysis of on-demand ride-hailing systems. *Manufacturing and Service Operations Management.* 2020;23(5):1237-1256. DOI:10.1287/msom.2020.0880.

[19] Qinyu W, Wejie Y, Wei W. Towards intercity mobility system–insights into the spatial interaction gravity model and determination approach. *Promet-Traffic&Transportation*. 2024;36(2):326-344. DOI: 10.7307/ ptt.v36i2.414.

[20] Tianyang G, et al. Optimizing electric flex-route feeder transit service with dynamic wireless charging technology. *Promet-Traffic&Transportation*. 2024;36(4):654-672. DOI:10.7307/ptt.v36i4.494.

[21] Yuhan G, Yu Z, Youssef B. Multi-dimensional spatiotemporal demand forecasting and service vehicle dispatching for online car-hailing platforms. *International Journal of Production Research.* 2022;60(6):1832-1853. DOI: 10.1080/00207543.2021.1871675.

[22] Stanko B, et al. Factors influencing driving time in public transport–a multiple regression analysis. *Promet-Traffic&Transportation*. 2023;35(1):37-54. DOI: 10.7307/ptt.v35i1.29.

[23] European Space Agency. Faster, greener and cheaper – your customised bus service. ESA Business Applications. https://business.esa.int/news/faster-greener-and-cheaper-your-customised-bus-service [Accessed 11th Dec. 2024].