https://doi.org/10.31449/inf.v49i30.8868 Informatica 49 (2025) 141–158 141

MSF-PSO: A Multi-Strategy Particle Swarm Optimization Framework for Dedicated Highway Traffic Control of Small Passenger Vehicles

Zichu Zhou[1*], Yaobei He[2], Yonghan Li[2]
[1]School of Traffic & Transportation Engineering, Central South University, Changsha 410075, China;
[2]Hunan Provincial Communications Planning, Survey & Design Institute CO, LTD, Changsha 410200, China
E-mail: zc101341@163.com
[*]Corresponding Author

Keywords: particle swarm optimization algorithm, highway traffic control, lane utilization, adaptive optimization, intelligent transportation

Received: April 11, 2025

This work aims to improve the passage efficiency of small passenger vehicles in highway environments. It proposes a Multi-Strategy Fusion Particle Swarm Optimization (MSF-PSO) algorithm to optimize travel time, lane utilization, lane-change frequency, and driving stability. The model architecture utilizes adaptive inertia weight adjustment to balance global and local search capabilities while implementing dynamic learning factor optimization to enhance individual and swarm learning capacities of particles. Also, it incorporates K-means clustering for population diversity maintenance and employs a Cauchy disturbance mechanism to facilitate particle escape from local optima. These components work synergistically to enable the algorithm to demonstrate faster convergence speed and superior global search capability in complex traffic environments. The algorithm undergoes validation on the SUMO traffic simulation platform using high-precision trajectory data from the HighD dataset. Experimental results demonstrate that compared to baseline algorithms, including Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Adaptive Particle Swarm Optimization (APSO), MSF-PSO significantly improves traffic efficiency and driving stability. Specifically, MSF-PSO reduces the average travel time to 243.7 seconds after 100 iterations, demonstrating remarkable improvements over baseline algorithms. This performance represents reductions of 8.6% compared to Haris & Nam (266.6 seconds), 11.4% versus APSO (275.4 seconds), 18.5% relative to PSO (299.2 seconds), and 19.8% compared to GA (303.8 seconds). Additionally, MSF-PSO achieves higher lane utilization (88.8%), lower lane-changing frequency (2.3 times/vehicle), and reduced speed fluctuation variance (7.9 km²/h²). Therefore, this work provides support for optimization in intelligent transportation.

Povzetek: Avtorji so razvili metodo MSF-PSO za namensko vodenje malih osebnih vozil po avtocestah, pri čemer upoštevajo: adaptivno težo, dinamična učenja, K-means diverziteto, Cauchy motnjo. V SUMO/HighD se tako zniža čas poti, menjave pasov in varianca hitrosti.

1 Introduction

In recent years, with the acceleration of global urbanization and the continuous increase in the number of motor vehicles, the traffic flow pressure on highways has been growing. This leads to problems such as decreased passage efficiency, frequent traffic accidents, and rising carbon emissions [1]. Traffic control systems have been evolving toward intelligent, refined, and dynamic optimization to optimize highway management and improve the passage efficiency of specific vehicle types. Small passenger vehicles (such as sedans and sports utility vehicles) dominate the highways, and their driving characteristics differ significantly from larger vehicles (such as trucks and buses) [2]. Therefore, a dedicated highway control system for small passenger vehicles helps optimize traffic flow, increase road capacity, and reduce safety risks associated with mixed traffic. However, traditional traffic control methods mainly rely on fixed rules and historical experience, which are difficult to adapt dynamically to changing traffic conditions. There is a pressing need to introduce advanced optimization algorithms to achieve smarter and more efficient traffic flow management.

Particle Swarm Optimization (PSO) is a typical swarm intelligence optimization algorithm. It simulates bird flock foraging behavior, where particles share information to look for the optimal solution and continuously adjust their movement trajectories to approach the global optimum [3, 4]. However, current research mainly focuses on signal control optimization and path planning, with limited involvement in dedicated traffic flow management for specific vehicle categories. Additionally, PSO faces challenges in convergence speed, global optimality, and real-time performance in complex traffic environments. This necessitates the combination of traffic flow modeling and simulation techniques to achieve more precise optimization control.

This work addresses the challenge of achieving coordinated optimization between passenger vehicle traffic efficiency and safety in dynamic highway environments through an improved PSO algorithm. The work specifically examines how the MSF-PSO algorithm outperforms conventional PSO and Adaptive Particle Swarm Optimization (APSO) across varying traffic densities (low, medium, high) and disruptive events (accidents, ramp merging). It demonstrates significant advancements in four key indicators: reduced lane-changing frequency, enhanced lane utilization efficiency, decreased average travel time, and minimized speed fluctuations. This work proposes a PSO-based dedicated highway control system for small passenger vehicles. It constructs a traffic flow optimization model and uses PSO for lane allocation, speed guidance, and traffic flow optimization. Specifically, the work first establishes a dedicated highway control model for small passenger vehicles. Then, it defines an optimization objective function that includes minimizing travel time, increasing lane utilization, reducing lane-change frequency, and improving passage safety. Moreover, an improved PSO algorithm is adopted to enhance convergence speed and computational efficiency through adaptive parameter adjustment. Finally, the system is tested in a simulation environment and compared with traditional rule-driven control methods to evaluate its optimization performance in different traffic flow scenarios. The work provides a new optimization approach for developing intelligent transportation system (ITS), offering theoretical support and practical reference for future autonomous driving environments and intelligent road infrastructure.

2 Related work

Recently, intelligent traffic control methods have been widely applied to highway management optimization, with numerous scholars conducting related research. Jin et al. [5] combined pattern recognition and deep learning methods to improve vehicle detection accuracy and real-time performance. Meanwhile, this method optimized traffic flow monitoring and accident early warning systems, thus providing data-driven support for highway intelligent control. Cui et al. [6] proposed a real-time monitoring and scheduling optimization framework based on the Internet of Things (IoT) and artificial intelligence, improving highway management efficiency. Agrahari et al. [7] focused on applying deep reinforcement learning (DRL), fuzzy control, and other methods in highway signal optimization. They indicated that improving signal system adaptability to dynamic traffic flow can effectively alleviate congestion. However, challenges in computational complexity and data quality should be solved. Wang & Geng (2024) [8] constructed a multi-node bilateral control model to deal with the traffic flow in blocked sections, which improved the vehicle traffic efficiency and the adaptive regulation ability of congestion. However, the delay and coordination of sensor data in actual deployment still posed challenges. Wang [9] explored the application of integrated intelligent signal control systems in highway traffic management. The study demonstrated that intelligent control systems could reduce average waiting time and energy consumption.

As a swarm intelligence-based optimization algorithm, PSO has gained wide application in intelligent traffic optimization in recent years due to its simple computation and fast convergence speed. It has been applied in tasks such as path optimization, signal control, and traffic scheduling. In path optimization, Zhi & Zuo [10] proposed a collaborative path-planning method for multiple autonomous underwater vehicles based on adaptive multi-population PSO. Although this method was primarily applied in underwater environments, its potential for intelligent traffic optimization improved PSO's global search capability and the efficiency of multi-vehicle cooperative path optimization. Haris & Nam [11] introduced a distance-dependent fast-converging PSO algorithm for optimizing the path planning of intelligent vehicles. The findings indicated that this method could effectively reduce computational complexity and improve path search speed, offering an efficient path-planning solution for intelligent traffic optimization. However, its adaptability in dynamic traffic environments should be further studied. Huang et al. [12] presented a collaborative path-planning method for multiple unmanned aerial vehicles based on column vector PSO combined with a genetic targeting mechanism. It provided new optimization ideas for path planning in autonomous vehicles within intelligent transportation, and enhanced path generation robustness and convergence speed.

In signal control, Zhang [13] combined gray system theory with intelligent signal optimization to achieve real-time traffic signal control optimization, reduce traffic delays, and improve the response capabilities of ITS. An & Bae [14] proposed a traffic signal timing optimization method based on PSO-Bacterial Foraging Optimization (BFO). The findings suggested that this method could reduce vehicle waiting time and increase intersection throughput. PSO optimized global search, and BFO fine-tuned signal timing, providing an efficient hybrid intelligent optimization strategy for intelligent traffic signal optimization. Qasim et al. [15] integrated the IoT, Arduino, and infrared sensors to achieve intelligent traffic signal control optimization. This boosted the dynamic adjustment ability of green light duration to adapt to complex traffic patterns, thus enhancing the adaptability of ITS.

Considering traffic scheduling, He (2024) [16] proposed an automated network traffic scheduling algorithm based on DRL, which effectively realized the intelligent allocation and load balancing of dynamic traffic data. However, the real-time performance and convergence speed in large-scale and complex road networks should be improved. Yang & Wang (2025) [17] combined DRL and convolutional neural networks (CNNs) for logistics path optimization, markedly improving scheduling efficiency and resource utilization rate. Nevertheless, the adaptability and robustness of the model to sudden traffic conditions must be enhanced.

The current research status of each scholar is further summarized in Table 1.

Table 1: Summary of the current research status of various scholars

Scholar	Application	Technology	Result/Performance	Disadvantage
Jin et al. [5]	Traffic flow monitoring and accident early warning	Pattern recognition + deep learning	Improved the detection accuracy and real-time performance	Relying on a large amount of data, the model training was complex
Cui et al. [6]	Improved the management efficiency of highways	IoT + artificial intelligence	Enhanced management efficiency	Insufficient adaptability to dynamic changes
Agrahari et al. [7]	Highway signal optimization	DRL and fuzzy control	Improved signal system adaptability, alleviated congestion	The computational complexity was high, and the requirements for data quality were strict
Wang & Geng [8]	Traffic flow control in blocked sections	A multi-node bilateral control model	Enhanced traffic efficiency and regulatory capacity	The problem of sensor data delay and coordination
Wang [9]	Highway traffic management	Intelligent signal control systems	Reduce waiting time and energy consumption	The adaptability to extreme traffic incidents was not taken into consideration
Zhi & Zuo [10]	A collaborative path-planning for multiple autonomous underwater vehicles	Adaptive multi-population PSO	Boosted global search ability	It was only applicable to specific environments and lacked universality
Haris & Nam [11]	The path planning of intelligent vehicles	Distance-dependent fast-converging PSO	Reduced computational complexity and improved search speed	Insufficient adaptability to dynamic traffic environments
Huang et al. [12]	A collaborative path-planning method for multiple unmanned aerial vehicles	Column vector PSO+ genetic targeting mechanism	Enhanced path generation robustness	The algorithm had a high complexity and limited real-time performance
Zhang [13]	Traffic signal optimization	Gray system theory + intelligent signal optimization	Realize real-time signal control optimization	Insufficient adaptability to sudden traffic incidents
An & Bae [14]	Traffic signal timing optimization	PSO-BFO	Reduced waiting time and increased intersection throughput	The convergence speed was slow, and it was prone to fall into a local optimum
Qasim et al. [15]	Intelligent traffic signal control	IoT + Arduino + infrared sensor	Improved the dynamic adjustment ability	Strong hardware dependence and poor scalability
He [16]	Automated network traffic scheduling	DRL	Improved the scheduling efficiency and real-time performance	Insufficient real-time performance in large-scale and complex road networks
Yang & Wang [17]	Route optimization	DRL+CNN	Enhanced scheduling efficiency and adaptability	Insufficient adaptability to sudden traffic conditions

In summary, existing research mainly focuses on overall traffic flow optimization, with limited attention given to refined control strategies for small passenger vehicles, making it difficult to meet the specific passage demands of certain vehicle categories. In addition, as a common optimization algorithm, PSO has made some progress in path planning and signal control. However, its real-time optimization capability in complex traffic environments still needs improvement. Existing methods are mostly used for static optimization, which struggles to adapt to sudden congestion and dynamic flow changes. Additionally, intelligent traffic management involves multiple optimization goals, such as lane allocation, speed guidance, and flow control. However, current research often focuses on single optimization tasks and fails to effectively balance multi-objective optimization (MOO) needs. To fill these research gaps, this work proposes a PSO-based optimization framework for dedicated highway control of small passenger vehicles, which expands the application of PSO in MOO, real-time flow regulation, and intelligent traffic decision-making. This provides new ideas for the development of future autonomous driving and intelligent road infrastructure.

3 Small passenger vehicle highway control method

3.1 Traffic flow modeling

To accurately assess the passage efficiency of small passenger vehicles, this work constructs a traffic flow model suitable for various traffic volume scenarios. It employs the SUMO traffic simulation platform [18] to model highway traffic flow and simulate the movement of small passenger vehicles on different lanes. The highway environment is set as a bidirectional six-lane road, with some lanes designated as priority lanes for small passenger vehicles to optimize passage strategies for specific vehicle categories. The experimental scenarios include three conventional traffic flow states: low, medium, and high, along with the inclusion of unexpected events (such as accidents or ramp merges) to simulate more complex traffic environments, as detailed in Table 2.

Table 2: Traffic flow modeling parameter settings

Scenario	Traffic density (vehicles/hour)	Speed range (km/h)	Lane occupancy (%)	Accident rate (%)
Low flow	1000 - 2500	90 - 130	30 - 50	1-3
Medium flow	3000 - 4500	70 - 110	50 - 70	3-7
High flow	5000 - 7000	50 - 90	70 - 85	5-12
Emergency event	4000 - 6000	30 - 80	75 - 90	10-20

The experimental implementation of accident scenarios involves configuring specific collision events within the SUMO traffic simulation platform. Based on accident characteristics documented in the HighD dataset, the simulation randomly selects a lane and generates a virtual accident obstruction at a predetermined location. The obstruction's dimensions correspond to average vehicle sizes and typical lane occupation ranges observed in actual traffic incidents. The simulation models vary accident severity levels (minor, moderate, severe) by adjusting obstruction duration parameters. Minor incidents persist for 5-10 minutes, moderate incidents for 10-20 minutes, and severe incidents for 20-35 minutes. During active accident periods, vehicles automatically adjust speed and routing according to SUMO's driver behavior models to circumvent the obstruction, realistically replicating accident-induced traffic flow disruptions.

In the traffic flow model for highway control of small passenger vehicles, the process specifically includes data collection, data preprocessing, SUMO traffic flow simulation, optimization computation using the improved PSO algorithm [19], optimization scheme output, and simulation verification. Figure 1 provides a detailed illustration.

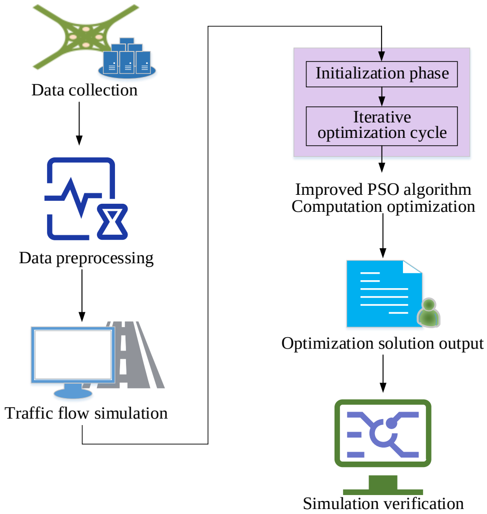

Figure 1: Flowchart of the traffic flow model for highway control of small passenger vehicles.

In Figure 1, the process begins with acquiring traffic flow data from the highway monitoring system, followed by data cleansing and normalization to ensure consistency and applicability. Next, a simulation scenario is constructed on the SUMO platform to simulate the operation of small passenger vehicles on highways, incorporating different traffic densities and traffic events. Subsequently, optimization computation is performed using the improved PSO algorithm to adjust lane selection, speed limit strategies, and traffic flow control, generating an optimized scheme. Finally, the optimized scheme is input into the simulation system for validation and compared with traditional rule-based methods to analyze improvements in small passenger vehicle traffic efficiency after optimization.

The traffic flow optimization objectives of this work include minimizing average travel time, improving lane utilization, reducing lane-changing frequency, and optimizing speed guidance strategies to achieve the optimal traffic scheme under different traffic flow conditions.

1) Minimizing Average Travel Time

This optimization objective aims to reduce the travel time of small passenger vehicles on highways, thereby improving traffic efficiency. It can be written as Equation (1):

$$\min T_{avg} = \frac{1}{N}\sum_{i=1}^{N} T_i \tag{1}$$

T_{avg} means the average travel time of all small passenger vehicles, in seconds (s); N denotes the total number of vehicles; T_i represents the travel time of the i-th vehicle from the entrance to the exit, in seconds (s). This optimization objective requires the improved PSO algorithm to adjust strategies such as lane allocation and speed limit guidance to reduce the travel time of small passenger vehicles and improve traffic efficiency.

2) Improving Lane Utilization

This optimization objective aims to improve the overall road efficiency by reasonably allocating lane resources and reducing congestion. It can be expressed as Equation (2):

$$\max U_{lane} = \frac{1}{M}\sum_{j=1}^{M} \frac{V_j}{C_j} \tag{2}$$

U_{lane} represents lane utilization, in percentage (%). M is the total number of lanes on the highway, V_j represents the current traffic flow on the j-th lane, in vehicles/hour; and C_j is the maximum capacity of the j-th lane, in vehicles/hour. This optimization objective requires balancing the load across different lanes to avoid severe congestion on some lanes while other lanes remain underutilized.

3) Reducing Lane-Changing Frequency

This optimization objective aims to reduce the number of lane changes, improve driving stability, and reduce accident risks. The objective function can be expressed as Equation (3):

$$\min LC_{avg} = \frac{1}{N}\sum_{i=1}^{N} LC_i \tag{3}$$

LC_{avg} represents the average number of lane changes, in occurrences; LC_i is the number of lane changes for the i-th vehicle, in occurrences; N refers to the total number of vehicles. This optimization objective encourages the improved PSO algorithm to adjust speed guidance and lane allocation strategies, thus minimizing lane changes as much as possible and improving traffic stability.

4) Optimizing Speed Guidance Strategy

This optimization objective aims to maintain a stable speed for small passenger vehicles through reasonable speed control, and reduce congestion and accidents caused by speed fluctuations. The objective function for optimizing the speed guidance strategy can be expressed as Equation (4):

$$\min \upsilon_{var} = \frac{1}{N}\sum_{i=1}^{N}(\upsilon_i - \upsilon_{opt})^2 \quad (4)$$

υ_{var} represents the variance of speed fluctuations, in km²/h²; υ_i stands for the actual speed of the i-th vehicle, in km/h; υ_{opt} denotes the optimal recommended speed for small passenger vehicles, in km/h. This optimization objective requires the PSO algorithm to dynamically adjust the recommended speed, reduce speed fluctuations, and enable vehicles to travel at a more stable speed, thereby improving the overall stability of the traffic flow.

3.2 Construction and analysis of the optimized control model based on the improved PSO algorithm

This work implements an intelligent traffic control model based on the improved PSO to address the optimization needs of small passenger vehicles on highways. The model enhances the traffic efficiency of small passenger vehicles by optimizing lane allocation, speed guidance, and traffic flow regulation. Due to the weak adaptability of traditional PSO in complex traffic environments, this work adopts an improved multi-strategy fusion PSO (MSF-PSO) algorithm. The algorithm incorporates optimization strategies such as adaptive inertia weight, dynamic learning factors, K-means clustering, and Cauchy disturbance to improve optimization performance. Figure 2 illustrates the optimized control model process based on the MSF-PSO algorithm.

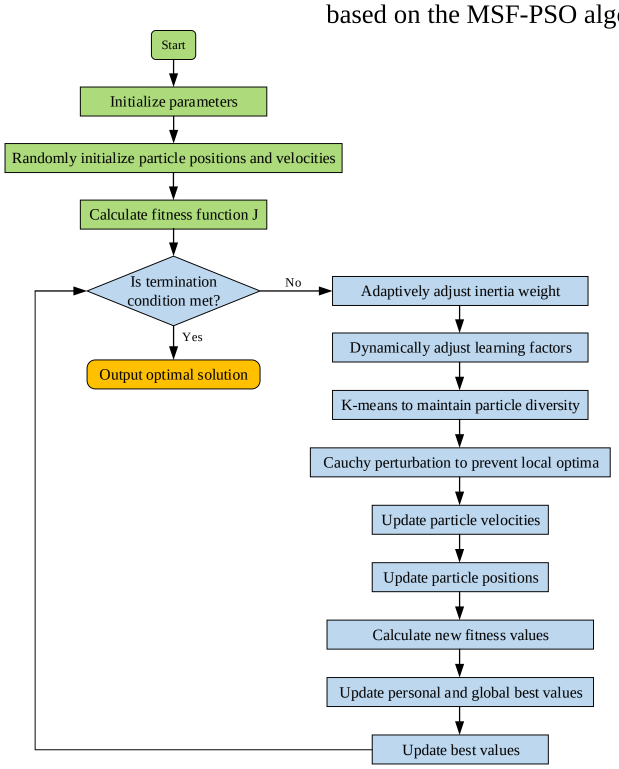

Figure 2: Flowchart of the MSF-PSO-based optimization control model.

In the proposed MSF-PSO-based optimization control model, a MOO function based on MSF-PSO is developed to optimize the highway traffic flow of small passenger vehicles. The function comprehensively considers minimizing average travel time, improving lane utilization, reducing lane-changing frequency, and optimizing speed guidance.

Taking into account the above optimization objectives, this work constructs the following weighted MOO function for the traffic flow model of small passenger vehicle highway control:

$$\min J = w_1 T_{avg} - w_2 U_{lane} + w_3 LC_{avg} + w_4 \upsilon_{var} \quad (5)$$

J represents the comprehensive optimization objective function; w_1, w_2, w_3, and w_4 are the weight coefficients for each objective (satisfying $w_1 + w_2 + w_3 + w_4$=1).

The selection of these weight coefficients is based on expert knowledge and practical traffic management requirements to balance performance and trade-offs among different objectives. Specifically, minimizing the average travel time (T_{avg}) receives the highest weight (w_1=0.4) as it represents the core indicator of traffic efficiency. Lane utilization (U_{lane}) is assigned a weight of w_2=0.3 to optimize road resource allocation. Lane-changing frequency (LC_{avg}) carries a weight of w_3=0.2 to reduce driving risks and improve traffic flow stability. Speed fluctuation variance (p) weights w_4=0.1 to decrease energy consumption and emissions. Experiments are conducted under different traffic flow conditions to verify the rationality of these weight coefficients, as detailed in Table 3. The result demonstrates that this weight combination achieves optimal performance across multiple scenarios, effectively balancing objectives while maintaining high adaptability and reliability.

Table 3: Optimization results under different traffic flow conditions.

Traffic flow conditions	T_{avg} (%)	U_{lane} (%)	LC_{avg} (%)	υ_{var} (%)
Low flow	12.5	8.8	33.9	43.7
Medium flow	11.8	9.2	32.5	41.3
High flow	10.5	10.1	30.2	38.5

The negative sign indicates the aim to maximize lane utilization, while the other optimization objectives are all minimized.

The optimization of this objective function needs to be solved under certain constraints. They mainly include traffic density, lane capacity, and safety distance constraints, ensuring the feasibility of the optimization results, as expressed in Equations (6) and (7):

$$\sum_{j=1}^{M}\rho_j \le \rho_{\max}, \sum_{j=1}^{M} V_j \le \sum_{j=1}^{M} C_j \tag{6}$$

$$D_{safe}(i) \ge \frac{\upsilon_i^2}{2a} + d_{\min}, \forall i \in N \tag{7}$$

ρ_j refers to the actual traffic density of the *j*-th lane; $\rho_{\max}$ is the maximum allowable density. $D_{safe}(i)$ represents the safety distance between the i-th vehicle and the preceding vehicle, in meters (m); *a* denotes the maximum deceleration of the vehicle, in meters per second squared (m/s²); and d_{min} stands for the minimum safe following distance, in meters (m). In this work, the safety distance calculation utilizes real-time vehicle trajectory data. For each vehicle, the HighD dataset provides current position coordinates (including lane position and longitudinal position within the lane) along with speed information. The longitudinal distance between consecutive vehicles in the same lane is computed using their positional coordinates. Considering each vehicle's braking performance and current speed, the minimum safe distance from the preceding vehicle is calculated according to Equation (7). The maximum deceleration rate (a) is predetermined based on vehicle type and road conditions, while the minimum safe following distance complies with traffic regulations and driving safety standards. This methodology enables real-time safety distance evaluation, providing critical safety constraints for the optimization control model.

In this work, the constraints of the optimization model (such as safety distance, lane capacity, traffic density, and others) are implemented through a penalty mechanism. Specifically, for infeasible solutions, that is, those that violate the constraints, they are not discarded directly. Instead, they are punished by introducing penalty terms into the objective function. The purpose of this penalty mechanism is to guide the optimization algorithm to gradually move away from the infeasible area during the search process, thus finding a feasible solution that satisfies all the constraint conditions.

For the safety distance constraint (Equation (7)), if the actual distance of a vehicle from the preceding vehicle is smaller than the minimum safety distance d_{min}, a penalty value is imposed on the vehicle's optimization objectives (such as average travel time or speed fluctuation variance) in the objective function. The magnitude of the penalty value is proportional to the degree of violation of the safety distance, meaning that a greater violation results in a higher penalty value. For example, if the actual distance d_{actual} is less than d_{min}, the penalty term can be expressed as Equation (8):

$$Penalty_{safe} = \alpha \times \left(\frac{d_{\min} - d_{actual}}{d_{\min}}\right)^2 \tag{8}$$

α is a penalty coefficient used to adjust the intensity of the penalty. In this way, the optimization algorithm prioritizes solutions that satisfy the safety distance constraint, ensuring the feasibility and safety of optimization results in practical applications.

For the lane capacity constraint (Equation (6)), if the actual traffic flow V_j on a lane exceeds its maximum capacity C_j, a penalty value is added to the lane's optimization objective (lane utilization) in the objective function. The penalty term can be written as Equation (9):

$$Penalty_{capacity} = \beta \times \left(\frac{V_j - C_j}{C_j}\right)^2 \tag{9}$$

β is a penalty coefficient used to adjust the intensity of the penalty. If the lane traffic flow $V_j \le C_j$ (the maximum capacity), the penalty term is zero, indicating that the lane's traffic flow is within the allowable range.

For the traffic density constraint, if the actual traffic density ρ_j on a lane exceeds the preset maximum allowable density $\rho_{\max}$, a penalty value is incorporated on the lane's optimization objective (lane utilization) in the objective function. The calculation of the penalty term reads:

$$Penalty_{density} = \delta \times \left(\frac{\rho_j - \rho_{\max}}{\rho_{\max}}\right)^2 \tag{10}$$

δ is a penalty coefficient employed to adjust the intensity of the penalty. The penalty term is zero if the traffic density ρ_j is less than or equal to the maximum allowable density $\rho_{\max}$. This illustrates that the density of this lane is within the safe range.

The incorporation of these penalty terms enables the optimization algorithm to simultaneously consider both objective optimization and constraint satisfaction when calculating the objective function value. Solutions violating constraints receive significantly increased objective function values through penalty terms, thereby reducing their fitness. This approach ensures gradual solution quality improvement during the search process. Also, it guarantees that final optimization results satisfy both performance objectives and practical traffic management constraints regarding safety and capacity.

Here, to enhance the efficiency of the optimization solution, the standard PSO is improved. First, adaptive inertia weight adjustment controls the particle's search ability, allowing the algorithm to perform a more extensive global search in the early stages and gradually converge to the optimal solution in the later stages. The dynamic adjustment of the inertia weight adopts a Sigmoid variation strategy, with its update equation given in Equation (11):

$$w(t) = w_{\min} + \frac{\left(w_{\max} - w_{\min}\right)}{1+e^{-\gamma(t-T/2)}} \tag{11}$$

$w(t)$ represents the inertia weight at the current iteration t; $w_{\max}$ and $w_{\min}$ are the maximum and minimum values of the inertia weight, typically set as $w_{\max}$ = 0.9 and $w_{\min}$ = 0.4; γ means the steepness parameter of the Sigmoid curve; T denotes the maximum number of iterations.

The parameter γ plays a critical role in governing the inertia weight decay process, where larger γ values accelerate decay while smaller values prolong it. This work systematically evaluates different γ values (ranging from 0.05 to 0.2) and observes their impact on optimization performance. Figure 3 visually demonstrates the effect of the adaptive inertia weight strategy by presenting decay curves under various γ values across iterations.

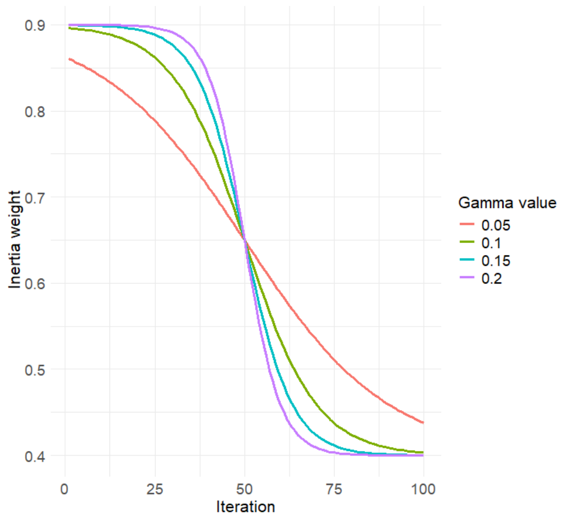

Figure 3: Convergence results under different γ conditions.

Figure 3 reveals that γ =0.1 produces an optimal inertia weight transition from 0.9 to 0.4, facilitating balanced search behavior. This smooth transition helps the algorithm conduct an extensive global search in the early stage and gradually focus on local search in the later stage, achieving a good balance between global exploration and local development. Consequently, γ =0.1 is identified as the most appropriate value for achieving robust optimization performance. Consequently, γ =0.1 is regarded as the most appropriate value. It can ensure that the algorithm smoothly transitions from global search to local search during iteration. It also prevents the algorithm from prematurely falling into the local optimal solution. Moreover, it does not affect the convergence speed of the algorithm due to the excessively long global search time.

Next, the adaptive adjustment of the learning factors can dynamically balance global and local search capability during the optimization process. Its update equations are given in Equations (12) and (13):

$$c_1(t) = c_1^{\max} - \frac{\left(c_1^{\max} - c_1^{\min}\right)\cdot t}{T} \tag{12}$$

$$c_2(t) = c_2^{\max} - \frac{\left(c_2^{\max} - c_2^{\min}\right)\cdot t}{T} \tag{13}$$

c_1 and c_2 represent the individual learning factor and the group learning factor, respectively; $c_1^{\max}$ and $c_1^{\min}$ are the maximum and minimum values for c_1, typically set as $c_1^{\max}$ = 2.5 and $c_1^{\min}$ = 1.5; $c_2^{\max}$ and $c_2^{\min}$ are the maximum and minimum values for c_2, typically set as $c_2^{\max}$ = 2.5 and $c_2^{\min}$ = 1.5.

To prevent premature convergence of PSO, this work uses K-means clustering after each iteration to classify the particle swarm and adjust the search direction of individual particles, maintaining the diversity of the population. In addition, a Cauchy disturbance strategy is introduced to enhance the ability of the particle swarm to escape from local optima in the later stages of iteration [20]. This strategy involves adding a Cauchy distribution random variable to the velocity update equation of some particles, as shown in Equation (14):

$$X_i^{new} = X_i + \tau\cdot Cauchy(0,1) \tag{14}$$

X_i^{new} represents the updated position of the particle; τ is the disturbance intensity factor; and $Cauchy(0,1)$ is the disturbance value randomly drawn from the Cauchy distribution.

Through the integration of the above optimization strategies, the improved PSO can achieve faster convergence speed, better global search capability, and more stable optimization results when solving the highway control problem for small passenger vehicles. Figure 4 depicts the pseudocode flow of the MSF-PSO-based optimization control model.

```
#Input: Control parameters (Lane_Selection, Speed_Limit, Traffic_Control)
#Output: fitness value J
# Initialize particle positions and velocities
# Particle position represents control parameters:
# [lane_speed_1, lane_speed_2, ..., speed_limit]
particles = np.random.uniform(low=30, high=120, size=(NUM_PARTICLES, 4))
velocities = np.random.uniform(-1, 1, (NUM_PARTICLES, 4))
# Evaluate initial personal best and global best
pbest_positions = particles.copy()
pbest_fitness = np.array([compute_fitness(pos) for pos in particles])
gbest_index = np.argmin(pbest_fitness)
gbest_position = pbest_positions[gbest_index]
# Main PSO loop
for t in range(MAX_ITER):
  # Update dynamic PSO parameters
  w = w_max - ((t / MAX_ITER) * (w_max - w_min))
  c1 = c1_min + ((t / MAX_ITER) * (c1_max - c1_min))
  c2 = c2_min + ((t / MAX_ITER) * (c2_max - c2_min))
  for i in range(NUM_PARTICLES):
    # Update velocity
    r1, r2 = np.random.rand(2)
    velocities[i] = (
      w * velocities[i]
      + c1 * r1 * (pbest_positions[i] - particles[i])
      + c2 * r2 * (gbest_position - particles[i])
    )
    # Update position
    particles[i] += velocities[i]
    # Run simulation and extract performance metrics
    sim_output = run_simulation(particles[i])
    metrics = extract_metrics(sim_output)
    # Check feasibility (e.g., safety distance constraint)
    if is_feasible(metrics['v_i'], metrics['a'], metrics['L_min']):
      # Compute fitness using multi-objective function
      current_fitness = compute_fitness(metrics)
      # Update personal best
      if current_fitness < pbest_fitness[i]:
        pbest_positions[i] = particles[i]
        pbest_fitness[i] = current_fitness
  # Update global best
  new_gbest_index = np.argmin(pbest_fitness)
  if pbest_fitness[new_gbest_index] <
compute_fitness(extract_metrics(run_simulation(gbest_position))):
    gbest_position = pbest_positions[new_gbest_index]
# Return final optimal solution
print("Optimal control parameters:", gbest_position)
```

Figure 4: Pseudocode flowchart of the MSF-PSO-based optimization control model.

Figure 4 presents the complete implementation process of the traffic control model based on the MSF-PSO algorithm, which focuses on optimizing traffic control parameters for passenger vehicles on highways through a simulation-driven approach. In this implementation, each particle represents a set of adjustable control parameters, including recommended speeds for different lanes and global speed limits. These parameters interact with the SUMO simulator to extract performance indicators such as average travel time, lane utilization, lane-changing frequency, and speed fluctuation variance, which are then used to calculate fitness values. The objective function employs a weighted multi-objective value J to guide particle updates, ensuring the optimization process aligns with real-world traffic behavior. Additionally, safety distance constraints are enforced by dynamically verifying whether inter-vehicle spacing meets the minimum safety distance requirement (d_{min}) during each simulation phase, thus guaranteeing traffic safety.

3.3 Experimental evaluation

To validate the effectiveness of the proposed MSF-PSO algorithm in optimizing highway control for small passenger vehicles, this work conducts experimental evaluations based on the SUMO traffic simulation platform. The experimental scenario is set as a six-lane highway, with a total length of 5 km, and the maximum vehicle speed is set to 130 km/h. To ensure the authenticity and comparability of the simulation, the experimental data comes from the HighD dataset (https://www.highd-dataset.com). It contains high-precision trajectory information of 11,500 vehicles on German highways, covering detailed traffic parameters such as speed, acceleration, lane position, and lane-changing behavior. Based on this dataset, different traffic flow scenarios are constructed, including low flow (1,000-2,500 vehicles/hour), medium flow (3,000-4,500 vehicles/hour), and high flow (5,000-7,000 vehicles/hour). Simulations for sudden events (such as accidents or merging disruptions) are performed to assess the optimization capability of MSF-PSO under sudden congestion conditions.

Emergency events are triggered by randomly establishing traffic flow interruption points during SUMO simulations, which may include temporary construction zones, sudden obstacles, or vehicle breakdowns. To maintain consistency of emergency scenarios across different algorithm tests, identical emergency scenarios are randomly generated at the start of each trial based on predefined parameters (event type, location, duration, etc.). These data are derived from statistical characteristics of similar emergencies recorded in the HighD dataset, ensuring test fairness and comparability. All algorithm evaluations use the same emergency scenario seed, enabling accurate performance comparisons under identical baseline conditions when responding to emergencies.

The experimental computing environment includes an Intel i7-12700H processor, 32GB RAM, and an NVIDIA RTX 3070 GPU to ensure the efficient operation of the optimization algorithm.

To evaluate the performance of the model, the MSF-PSO algorithm proposed is compared with Adaptive Particle Swarm Optimization (APSO) [21], PSO, Genetic Algorithm (GA) [22], and the model algorithm proposed by Haris & Nam (2024). Each algorithm is run 100 times, and the average value is taken for statistical analysis. To ensure the transparency and repeatability of the comparison, the configuration information of the benchmark algorithm used in the experiment is described in detail, including parameter settings and tuning processes, as listed in Table 4.

Table 4: Hyperparameter settings of each model algorithm.

The name of the algorithm	Range of inertia weight	Cognitive coefficient (c_1)	Social coefficient (c_2)	Cauchy distribution setting	Population size	Maximum number of iterations	Remark
MSF-PSO	0.9 - 0.4	Adaptive	Adaptive	τ=0.1, λ=1.5	30	100	-

PSO	0.7	2	2	-	30	100	-
APSO	0.9 - 0.4	2	2	-	30	100	-
GA	-	-	-	-	30	100	-
Haris & Nam	0.7	1.4	1.4	-	30	100	Distance-dependent fast convergence strategy

Notes: 1) GA does not utilize inertia weights or learning factors as it operates as a population-based evolutionary algorithm. The crossover probability is 0.8, the mutation probability is 0.01, and the tournament selection method is adopted, with a tournament size of 2. 2) The MSF-PSO's Cauchy distribution configuration includes perturbation intensity factor (τ) and shape parameter (λ). 3) The Haris & Nam model employs a distance-dependent rapid convergence strategy to enhance optimization speed.

The experiment mainly measures four key indicators: average travel time (T_{avg}), lane utilization (U_{lane}), lane-changing frequency (LC_{avg}), and speed stability (υ_{var}).

4 Results and discussion

4.1 Performance analysis under different traffic flows

The performance of the model algorithm proposed is analyzed across four key indicators under different traffic flow conditions. The specific results are suggested in Table 5.

Table 5: Performance results under different traffic flow conditions.

Traffic condition	T_{avg} (s)	U_{lane} (%)	LC_{avg} (Number/vehicle)	υ_{var} (km²/h²)
Low flow	210.4	78.2	1.75	6.2
Medium flow	243.7	88.8	2.3	7.9
High flow	315.2	92.4	3.89	15.1
Emergency event	398.5	70.3	5.62	25.6

In Table 5, a comparison of the results for each traffic flow condition reveals that under medium traffic flow (3,000-4,500 vehicles/hour), the traffic volume has noticeably increased but has not yet reached a state of severe congestion. The optimization algorithm shows remarkable effects and is highly representative. At this point, the average travel time is 243.7 seconds, which is an increase compared to the low-flow condition (210.4 seconds), but still much lower than the travel time under high-flow and sudden event conditions. This indicates that traffic under medium flow maintains relatively high passage efficiency. Meanwhile, lane utilization reaches 88.8%, substantially higher than the 78.2% under low flow. This indicates that the utilization of road resources is relatively balanced within this traffic range, allowing the optimization and control measures to be fully effective. Additionally, the lane-changing frequency is 2.3 times per vehicle, slightly higher than under low flow, but still significantly lower than the frequent lane changes observed under high flow and sudden event conditions. This suggests that, at this traffic level, driving behavior is relatively stable, reducing interference caused by lane changes. Finally, the speed fluctuation variance is 7.9 (km²/h²), slightly higher than under low flow but still much lower than under high flow and sudden event conditions. This indicates that, despite the increased traffic flow in medium flow, vehicle speeds remain relatively stable. Consequently, the indicators under medium flow reflect the effectiveness of traffic optimization while avoiding the severe congestion seen in high flow and the sharp deterioration caused by sudden events, making it a highly representative scenario.

The real-time performance of the proposed model algorithm under different traffic conditions is further analyzed, as revealed in Figure 5.

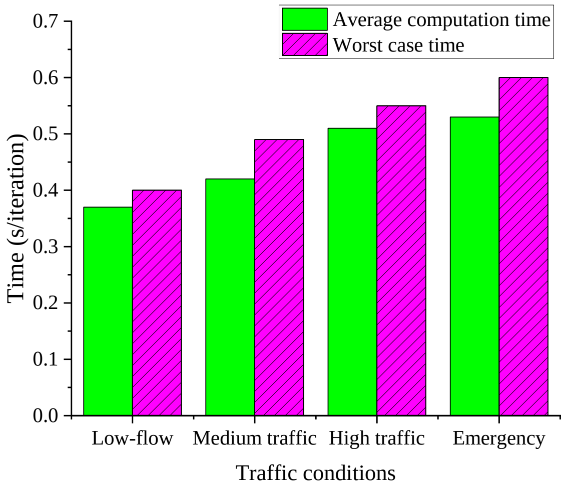

Figure 5: Real-time results under various traffic conditions

Figure 5 illustrates the computational performance of the MSF-PSO algorithm under different traffic conditions as follows. In low-traffic states, the average time per iteration is 0.37 seconds (with a worst-case scenario of 0.4 seconds), demonstrating efficient processing capabilities in light-load traffic. The average time increases to 0.42 seconds (the worst: 0.49 seconds) under medium traffic and reaches 0.51 seconds (the worst: 0.55 seconds) under high traffic, still maintaining sub-second response and meeting real-time control requirements. Notably, in emergency event scenarios, the algorithm's average time consumption stabilizes at 0.53 seconds (the worst: 0.6 seconds), consistently below the 1-second real-time threshold of ITS. This result indicates that the MSF-PSO algorithm maintains stable computational efficiency across all scenarios, from low-load to peak congestion and sudden incidents. This provides a reliable guarantee for real-time decision-making in actual road control systems.

4.2 Analysis of runtime complexity and convergence under diverse algorithms

The runtime complexity (including standard deviation) of each algorithm in 100 iterations is denoted in Table 6.

Table 6: The result of the runtime complexity of each algorithm.

Algorithm	Average computation time (seconds/iteration)	Standard deviation (seconds)	Total computation time (seconds /100 iterations)
MSF-PSO	0.4	0.05	40
APSO	0.43	0.06	43
PSO	0.38	0.04	38
GA	0.51	0.06	51
Haris & Nam	0.44	0.03	44

In Table 6, in terms of algorithm runtime complexity, MSF-PSO demonstrates excellent computational efficiency and stability. Regarding time per iteration, the standard PSO algorithm performs best (0.38 seconds), with MSF-PSO closely following at 0.4 seconds, incurring only a 5.3% increase in computational load compared to PSO and a 21.6% reduction compared to GA's 0.51 seconds. It is worth noting that although the iteration times of APSO and the Haris & Nam algorithm are similar to MSF-PSO (0.43-0.44 seconds), MSF-PSO has the smallest standard deviation (0.05 seconds). This indicates lower volatility in computation time and more stable operation. Considering total computational cost, MSF-PSO requires 40 seconds to complete 100 iterations, remarkably outperforming GA (51 seconds) and matching PSO's efficiency level of 38 seconds. This superiority in computational performance primarily stems from MSF-PSO's adoption of dynamic learning factors and K-means clustering strategies. This effectively controls computational complexity while ensuring optimization accuracy, giving it remarkable advantages in traffic control scenarios with high real-time requirements.

The convergence results of each algorithm are further explored as indicated in Figure 6.

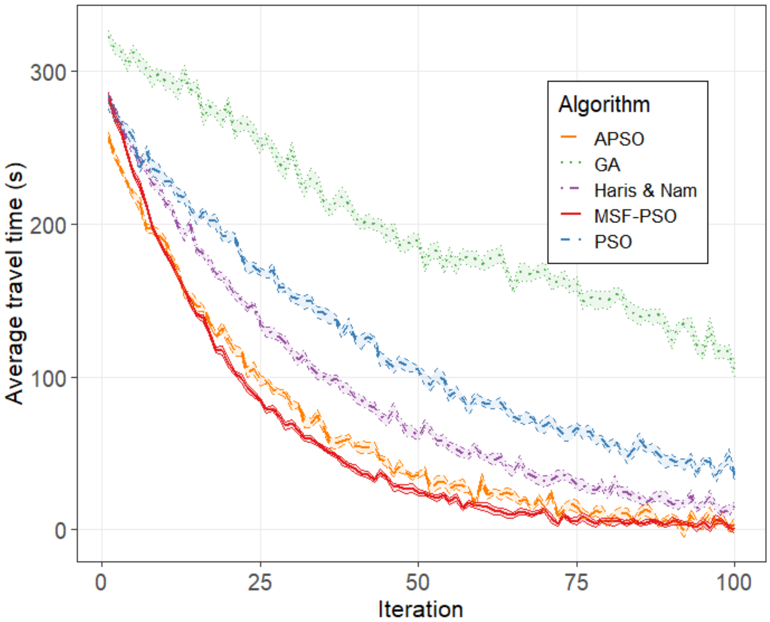

Figure 6: The convergence results of each algorithm.

The convergence analysis in Figure 6 reveals significant performance differences among the algorithms. MSF-PSO demonstrates superior convergence characteristics, rapidly stabilizing to the optimal solution within 30 iterations while maintaining the smallest standard deviation range (1-3). This verifies the effectiveness of its multi-strategy fusion approach in accelerating convergence. APSO and the Haris & Nam algorithm exhibit moderate convergence speeds, though the latter shows larger standard deviations (2-4), indicating weaker disturbance resistance. The standard PSO displays noticeably slower convergence due to its lack of adaptive mechanisms. GA presents characteristic oscillatory convergence patterns (with superimposed sinusoidal fluctuations), evidenced by its maximum standard deviation (3-6) and highest initial value (320). This reflects the inherent instability of traditional evolutionary algorithms in dynamic optimization scenarios. Notably, MSF-PSO's convergence curve consistently remains below those of other algorithms, and combined with its minimal standard deviation range, conclusively demonstrates dual advantages in both convergence speed and robustness.

4.3 Performance analysis under different algorithms

Further comparison of the results for each algorithm in terms of the four indicators under medium traffic flow is provided. Figures 7 to 10 present the specific comparison results.

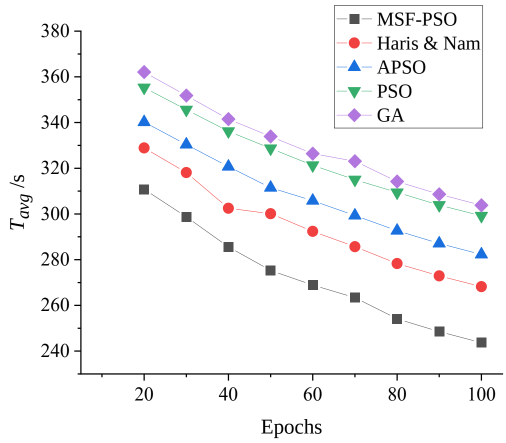

Figure 7: Comparison of average travel time under different algorithms.

Figure 7 illustrates that the proposed MSF-PSO algorithm can achieve a lower average travel time within all training epochs compared with APSO, PSO, GA, and the model proposed by Haris & Nam (2024). After 100 epochs of training, the average travel time of MSF-PSO is further optimized to 243.7 seconds, which is 18.5% lower than PSO (299.2 seconds) and 19.8% lower than GA (303.8 seconds). This demonstrates faster convergence speed and better optimization capability. The result indicates that MSF-PSO significantly enhances the traffic flow optimization efficiency by fusing multi-strategy techniques such as adaptive inertia weight adjustment, dynamic learning factors, and K-means clustering. It can reach a better convergence state with fewer training epochs, achieving a lower travel time.

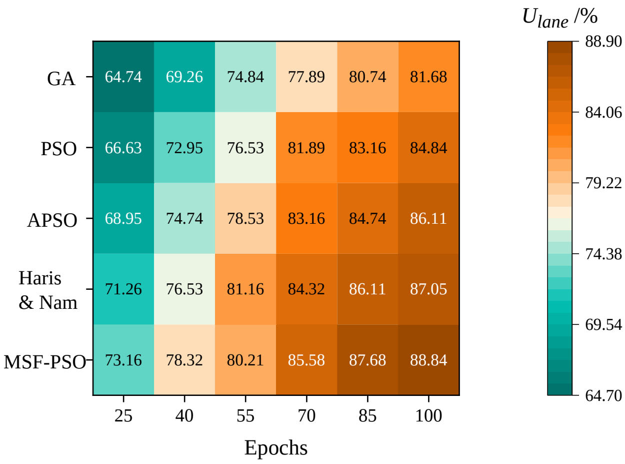

Figure 8: Comparison of lane utilization under different algorithms.

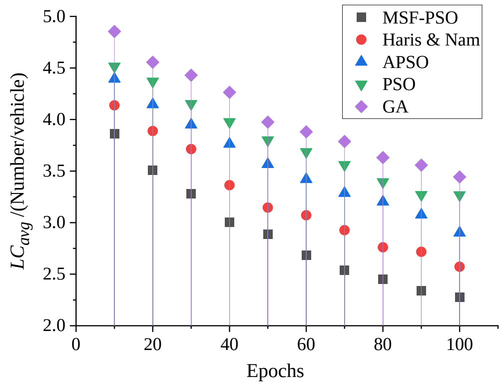

Figure 9: Comparison of lane-changing frequency under different algorithms.

In Figure 8, the experimental results indicate that the MSF-PSO algorithm proposed performs excellently in optimizing lane utilization. It consistently achieves higher lane utilization across all training epochs and has obvious advantages compared to the model proposed by Haris & Nam (2024), APSO, PSO, and GA. Moreover, lane utilization shows an upward trend as the iteration epochs increase. After 100 epochs, MSF-PSO further optimizes lane utilization to 88.84%, which is 1.8% higher than Haris & Nam (2024) (87.05%), 4.7% higher than PSO (84.84%), and 8.8% higher than GA (81.68%). This result demonstrates that MSF-PSO enhances search capability through adaptive inertia weight adjustment, dynamic learning factor adjustments, and K-means clustering, and achieves more efficient lane resource optimization. It reaches a better convergence state with fewer training epochs, improving lane utilization and enhancing the balance and efficiency of traffic flow.

Figure 9 suggests that the MSF-PSO algorithm proposed demonstrates outstanding performance in optimizing lane-changing frequency. It significantly reduces the average number of lane changes per vehicle across all training epochs compared to Haris & Nam (2024), APSO, PSO, and GA. After 10 epochs, MSF-PSO achieves a lane-changing frequency of 3.86 changes/vehicle, lower than GA (4.85 changes/vehicle) and PSO (4.52 changes/vehicle), showing initial optimization effects. After 100 epochs, MSF-PSO optimizes lane-changing frequency to 2.27 changes/vehicle. It is 11.5% lower than Haris & Nam (2024) (2.57 changes/vehicle), 21.6% lower than APSO (2.90 changes/vehicle), 30.5% lower than PSO (3.27 changes/vehicle), and 33.9% lower than GA (3.44 changes/vehicle). This result indicates that the proposed MSF-PSO model algorithm can enhance the search ability through diverse improvement measures, such as adaptive inertia weight adjustment, dynamic learning factor optimization, and K-means clustering. It enables vehicles to allocate lanes more rationally, reducing unnecessary lane changes, improving driving stability, reducing lane change interference, and optimizing traffic flow efficiency.

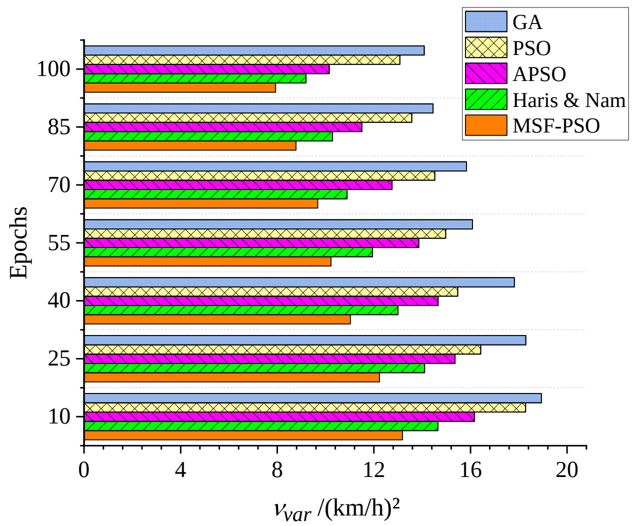

Figure 10: Comparison of speed fluctuation variance under different algorithms.

Figure 10 shows that the MSF-PSO algorithm proposed performs exceptionally well in reducing speed fluctuation variance. Across all training epochs, it outperforms Haris & Nam (2024), APSO, PSO, and GA, significantly improving driving stability. After 10 epochs, the speed fluctuation variance for MSF-PSO is 13.19 (km²/h²), about 30% lower than GA (18.94 (km²/h²)) and PSO (18.28 (km²/h²)), demonstrating initial optimization advantages. As training progresses, MSF-PSO further optimizes to 7.93 (km²/h²) after 100 epochs, which is 13.7% lower than the Haris & Nam (2024) model (9.19 (km²/h²)), 39.4% lower than PSO (13.08 (km²/h²)), and 43.7% lower than GA (14.09 (km²/h²)). It showcases superior speed and stability control. This result confirms that the MSF-PSO algorithm, through adaptive inertia weight adjustment, dynamic learning factor optimization, and Cauchy disturbance mechanisms, enhances the stability and safety of traffic flow, offering notable advantages in intelligent traffic optimization.

To verify the reliability of the results, analysis of variance (ANOVA) and independent sample t-tests (confidence level: 95%) are conducted on the performance indicators of each algorithm, as presented in Table 7.

Table 7: ANOVA and independent sample t-test results of various algorithms (mean ± standard deviation).

Indicator	MSF-PSO	Haris & Nam (2024)	APSO	PSO	GA	f-value	p-value
Average travel time (seconds)	243.7 ± 5.2	257.3 ± 6.1	268.4 ± 7.3	299.2 ± 8.7	303.8 ± 9.1	45.3	<0.001
Lane utilization (%)	88.8 ± 1.5	87.1 ± 1.8	86.2 ± 1.8	84.8 ± 2.1	81.7 ± 2.8	32.1	<0.001
Lane-changing frequency (changes/vehicle)	2.3 ± 0.4	2.57 ± 0.5	2.90 ± 0.6	3.27 ± 0.6	3.44 ± 0.7	28.7	<0.001
Speed fluctuation variance (km²/h²)	7.9 ± 0.9	9.19 ± 1.1	9.80 ± 1.2	13.08 ± 1.5	14.09 ± 1.8	37.6	<0.001

In Table 7, experimental results show MSF-PSO's statistically significant superiority ($p<0.001$) across all key performance indicators. Compared with other algorithms, MSF-PSO optimizes average travel time to 243.7±5.2 seconds, representing a 5.3% improvement over the second-best Haris & Nam model (257.3±6.1 seconds) and 18.5%-19.8% enhancements versus conventional PSO and GA approaches. The algorithm achieves 88.8±1.5% lane utilization, outperforming APSO (86.2±1.8%) by 3.0%. Simultaneously, this algorithm delivers the lowest lane-changing frequency (2.3±0.4 changes/vehicle) and speed fluctuation variance (7.9±0.9 km²/h²), representing 11.5% and 14.1% reductions, respectively, compared to the next-best performing algorithms. Crucially, MSF-PSO demonstrates the smallest standard deviations across all metrics, confirming the exceptional stability and reliability of its optimization results. These advantages primarily stem from the algorithm's integrated adaptive inertia weight, dynamic learning factors, and Cauchy disturbance mechanisms. These effectively address traditional algorithms' limitations in convergence speed and optimization accuracy while maintaining computational efficiency, thereby providing an advanced solution for real-time ITS optimization.

Table 8 shows the sensitivity analysis of multi-objective weights.

Table 8: Results of weight sensitivity analysis.

Weight combination (w_1, w_2, w_3, and w_4)	T_{avg} (s)	U_{lane} (%)	LC_{avg} (Number /vehicle)	υ_{var} (km/h)²
(0.4, 0.3, 0.2, 0.1)	243.7	88.8	2.3	7.9
(0.3, 0.3, 0.3, 0.1)	251.2	85.4	1.8	8.2
(0.5, 0.2, 0.2, 0.1)	238.5	82.1	2.6	9.1

Sensitivity analysis of multi-objective weight combinations reveals that different weight configurations significantly impact optimization results (as shown in Table 8). When using the baseline weight combination (0.4, 0.3, 0.2, 0.1), the algorithm achieves the best balance between travel time (243.7 seconds) and lane utilization (88.8%). Also, it maintains a low lane-changing frequency (2.3 times/vehicle) and speed fluctuation (7.9 km²/h²). Notably, increasing the weight of lane-changing frequency to 0.3 (combination 0.3, 0.3, 0.3, 0.1) markedly reduces lane-changing frequency to 1.8 times/vehicle (a 21.7% decrease). However, the cost is an increase of 7.5 seconds (3.1%) in travel time and a decrease of 3.4 percentage points in lane utilization. Conversely, prioritizing travel time (weight: 0.5) yields the shortest travel time (238.5 seconds) but reduces lane utilization to 82.1% and increases lane-changing frequency to 2.6 times/vehicle. These results demonstrate that weight settings for the MSF-PSO algorithm require trade-offs based on actual management needs. Increasing the lane-changing frequency weight is appropriate for safety-priority scenarios, while enhancing the travel time weight is suitable for efficiency-priority scenarios. Among all tested combinations, the baseline weights demonstrate optimal comprehensive performance, validating the rationality of the original weight design.

4.4 Discussion

Through analysis of the above model performance results, the proposed MSF-PSO algorithm can optimize the traffic flow of small passenger vehicles on highways. Particularly, it remarkably improves key indicators such as average travel time, lane utilization, lane-changing frequency, and speed fluctuation variance. Specifically, MSF-PSO optimizes the average travel time to 243.7±5.2 seconds, representing a 5.3% improvement over the model proposed by Haris & Nam and an 18.5%-19.8% improvement over traditional PSO and GA. Regarding lane utilization, it reaches 88.8±1.5%, a 3.0% improvement over APSO. Meanwhile, it achieves the lowest lane-changing frequency (2.3±0.4 times/vehicle) and speed fluctuation variance (7.9±0.9 km²/h²), which are 11.5% and 14.1% lower than those of the algorithm proposed by Haris & Nam, respectively. Therefore, compared with existing research, the proposed MSF-PSO algorithm outperforms traditional PSO, APSO, and GA in optimization performance. Also, it remarkably enhances the algorithm's global search capability and convergence speed through multi-strategy integration (such as adaptive inertia weight adjustment, dynamic learning factor optimization, K-means clustering, and Cauchy perturbation mechanism).

The MSF-PSO algorithm demonstrates remarkable advantages in optimizing traffic flow for small passenger vehicles on highways, particularly in convergence speed and global search capability. Compared with traditional GA, PSO, and APSO, MSF-PSO avoids local optima more effectively by introducing mechanisms such as Cauchy noise, thus achieving more efficient global search. Compared with the deep learning-based traffic flow monitoring method by Almukhalfi et al. [23], the MSF-PSO algorithm exhibits stronger adaptability in handling dynamic traffic flows, especially in optimizing lane utilization and reducing lane-changing frequency. Additionally, compared with Yazdani et al.'s [24] DRL-based signal optimization method, MSF-PSO demonstrates more balanced MOO performance, capable of optimizing multiple key indicators simultaneously rather than focusing solely on single signal control. Furthermore, compared with Haris et al.'s [25] fast-converging PSO algorithm, MSF-PSO possesses significant advantages in global search capability and optimization stability. When dealing with complex traffic flows, it effectively avoids local optimal solutions to achieve superior global optimization results.

However, these improvements are not without costs. The K-means clustering introduced in the proposed MSF-PSO algorithm, while helping maintain population diversity, increases computational complexity. In practical applications, this additional computational burden may affect the algorithm's real-time performance to some extent, particularly when processing large-scale traffic flow data. Nevertheless, MSF-PSO still maintains sub-second response times under different traffic flow conditions, meeting the real-time requirements of ITS.

Regarding generalizability and real-time implementation feasibility, the proposed MSF-PSO algorithm has demonstrated good performance in optimizing traffic flow for small passenger vehicles. However, further research is needed to determine whether it can be generalized to other traffic types (such as trucks or mixed traffic). Due to differences in driving characteristics and traffic demands among vehicle types, the algorithm may require corresponding adjustments and optimizations. For example, trucks have relatively poor acceleration and deceleration performance, resulting in different behavioral patterns in traffic flow compared to small passenger vehicles. Additionally, the real-time implementation feasibility of the MSF-PSO algorithm is a critical consideration. Experimental results show the algorithm maintains stable computational efficiency under various traffic flow conditions. However, real-time acquisition and processing of traffic flow data in actual highway environments may be affected by multiple factors, such as sensor accuracy and reliability, and communication network delays. These factors may interfere with the algorithm's real-time performance, thus impacting its optimization effects

Regarding practical significance, although experimental results indicate that the MSF-PSO algorithm achieves substantial improvements in optimizing traffic flow, its practical implications still need to be analyzed in combination with real-world conditions. In real highway environments, traffic flow is influenced by multiple factors, such as weather conditions, traffic accidents, and road construction. These factors may interfere with the optimization effects of traffic flow, thereby reducing the algorithm's practical application value. For example, under adverse weather conditions, even with the application of optimization algorithms, vehicle speeds may be restricted, leading to increased travel times. Additionally, the optimization of traffic flow needs to consider the traffic efficiency of vehicles and other factors (e.g., traffic safety and environmental impact). While the MSF-PSO algorithm performs excellently in reducing lane-changing frequency and speed fluctuations, its practical application necessitates further evaluation of how these optimization measures affect traffic safety and the environment. For instance, reducing lane-changing frequency may decrease interference between vehicles and thereby enhance traffic safety. However, it may also result in less flexible travel routes for some vehicles, affecting their traffic efficiency.

Regarding the algorithm's scalability and limitations, this work briefly identifies dynamic signal inputs and extreme events, further analyzing their impacts on actual traffic flow optimization. Dynamic signal inputs can real-time reflect changes in traffic flow, thereby providing more accurate input data for optimization algorithms. Extreme events (such as severe traffic accidents or adverse weather conditions) can cause sudden disruptions to traffic flow, affecting the performance of optimization algorithms. Meanwhile, the influence of mixed traffic (trucks, buses) is another critical factor to consider. Different vehicle types exhibit varying driving characteristics and traffic demands. Thus, in mixed traffic environments, further research is needed on optimizing traffic flows for diverse vehicle categories. For example, setting different lane priorities can provide more reasonable route and speed guidance for various vehicle types. Moreover, delays in the PSO update cycle represent an issue to be addressed. In practical applications, data processing, communication, and other factors may introduce delays in the PSO update cycle. Finally, robustness under changing weather/visibility conditions is also a key consideration. Adverse weather may impair the optimization effects of traffic flow, necessitating further research on enhancing the MSF-PSO algorithm's robustness across different weather conditions to ensure its effective operation in complex traffic environments.

Furthermore, the proposed highway lane allocation optimization model based on the MSF-PSO algorithm prompts further discussion on the broader impacts of such algorithms. For example, prioritizing certain vehicle categories over others may involve policy and equity considerations. In practical applications, multiple factors must be comprehensively evaluated to ensure the fairness and rationality of lane allocation algorithms. For instance, setting different priority levels can provide more reasonable route and speed guidance for various vehicle types. Meanwhile, policy factors such as environmental protection policies and traffic management regulations must be considered to ensure that the implementation of lane allocation algorithms complies with relevant policy requirements. In addition, the implementation of lane allocation algorithms may affect the overall performance of the traffic system. For example, prioritizing certain vehicle categories could reduce the travel efficiency of other categories, thereby impacting the system's overall performance.

5 Conclusion

This work proposes an MSF-PSO algorithm to optimize the traffic efficiency of small passenger vehicles in highway environments and conducts experimental validation on the SUMO traffic simulation platform. The results demonstrate that the proposed MSF-PSO exhibits exceptional optimization capabilities. It reduces average travel time (a 12.5% reduction), increases lane utilization (an 8.8% improvement), reduces lane change frequency (a 33.9% decrease), and enhances driving stability (a 43.7% reduction in speed fluctuation variance). However, this work has several limitations. For example, the current algorithm does not fully account for dynamic changes in real-time traffic signals, and its adaptability to extreme traffic events (such as severe accidents or inclement weather) requires further optimization. Additionally, the computational complexity of the algorithm when processing large-scale traffic flow data may compromise real-time performance. Future research focuses on integrating real-time data streams, such as those from connected vehicles or edge devices, to enable dynamic re-optimization. Meanwhile, exploring hybrid reinforcement learning frameworks (such as DDPG+PSO) for policy learning in dynamic environments enhances the algorithm's adaptability and optimization capabilities, offering more efficient and flexible solutions for intelligent traffic management systems.

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Data availability statement

The data used to support the findings of this study are all in the manuscript.

References

[1] Guo, H., Huang, R. and Xu, Z., 2024. The design of intelligent highway transportation system in smart city based on the internet of things. *Scientific Reports*, *14*(1), pp.28122. https://doi.org/10.1038/s41598-024-79903-0

[2] Mahini, H., Asadi, M., Mahini, A. and others, 2024. Lane and platoon assignment in intelligent transportation system: A novel heuristic approach. *International Journal of Intelligent Transportation Systems Research*, *22*(2), pp.298-315. https://doi.org/10.1007/s13177-024-00397-1

[3] Olayode, I.O., Du, B., Tartibu, L.K. and others, 2024. Traffic flow modelling of long and short trucks using a hybrid artificial neural network optimized by particle swarm optimization. *International Journal of Transportation Science and Technology*, *14*, pp.137-155. https://doi.org/10.1016/j.ijtst.2023.04.004

[4] Senthilkumar, A., Ramkumar, J., Lingaraj, M. and others, 2023. Minimizing energy consumption in vehicular sensor networks using relentless particle swarm optimization routing. *International Journal of Computer Networks and Applications*, *10*(2), pp.217-230. https://doi.org/10.22247/ijcna/2023/220737

[5] Jin, M., Sun, C. and Hu, Y., 2023. An intelligent traffic detection approach for vehicles on highway

using pattern recognition and deep learning. *Soft Computing*, *27*(8), pp.5041-5052. https://doi.org/10.1007/s00500-022-07375-3

[6] Cui, Y. and Lei, D., 2023. Design of highway intelligent transportation system based on the internet of things and artificial intelligence. *IEEE Access*, *11*, pp.46653-46664. https://doi.org/10.1109/ACCESS.2023.3275559

[7] Agrahari, A., Dhabu, M.M., Deshpande, P.S. and others, 2024. Artificial intelligence-based adaptive traffic signal control system: A comprehensive review. *Electronics*, *13*(19), pp.3875. https://doi.org/10.3390/electronics13193875

[8] Wang, X. and Geng, H., 2024. Intelligent traffic flow control based on multi-node bilateral control model in closed congested road scenarios. *Informatica*, *49*(17), pp.129-144. https://doi.org/10.31449/inf.v49i17.6703

[9] Wang, J., 2025. Analysis and research on the intelligent control system of comprehensive traffic signals. *Applied and Computational Engineering*, *129*, pp.46-50. https://doi.org/10.54254/2755-2721/129/2025.20251

[10] Zhi, L. and Zuo, Y., 2024. Collaborative path planning of multiple AUVs based on adaptive multi-population PSO. *Journal of Marine Science and Engineering*, *12*(2), pp.223. https://doi.org/10.3390/jmse12020223

[11] Haris, M. and Nam, H., 2024. Path planning optimization of smart vehicle with fast converging distance-dependent PSO algorithm. *IEEE Open Journal of Intelligent Transportation Systems*, *5*, pp.726-739. https://doi.org/10.1109/OJITS.2024.3486155

[12] Huang, C., Ma, H., Zhou, X. and others, 2025. Cooperative path planning of multiple unmanned aerial vehicles using cylinder vector particle swarm optimization with gene targeting. *IEEE Sensors Journal*, *25*(5), pp.8470-8480. https://doi.org/10.1109/JSEN.2024.3516124

[13] Zhang, S., 2024. Real-time application of grey system theory in intelligent traffic signal optimization. *Journal of Computational Methods in Science and Engineering*, *24*(4-5), pp.3137-3153. https://doi.org/10.3233/JCM-247560

[14] An, H.K. and Bae, G., 2023. A study of traffic signal timing optimization based on PSO-BFO algorithm. *The Journal of the Korea Institute of Intelligent Transport Systems*, *22*(6), pp.182-195. https://doi.org/10.12815/kits.2023.22.6.182

[15] Qasim, K.R., Naser, N.M. and Jabur, A.J., 2024. An IoT-enhanced traffic light control system with Arduino and IR sensors for optimized traffic patterns. *Future Internet*, *16*(10), pp.377. https://doi.org/10.3390/fi16100377

[16] He, H., 2024. Automatic network traffic scheduling algorithm based on deep reinforcement learning. *Informatica*, *48*(22), pp.163-178. https://doi.org/10.31449/inf.v48i22.6943

[17] Yang, Y. and Wang, K., 2025. Efficient logistics path optimization and scheduling using deep reinforcement learning and convolutional neural networks. *Informatica*, *49*(16), pp.151-170. https://doi.org/10.31449/inf.v49i16.7839

[18] Panigrahy, S.K. and Emany, H., 2023. A survey and tutorial on network optimization for intelligent transport system using the internet of vehicles. *Sensors*, *23*(1), pp.555. https://doi.org/10.3390/s23010555

[19] Li, Z., Li, Z., Pan, C. and others, 2025. Short-term traffic flow prediction based on spatial-temporal attention time gated convolutional network with particle swarm optimization. *Applied Intelligence*, *55*(2), pp.1-15. https://doi.org/10.1007/s10489-024-06117-2

[20] Wang, X., Liu, Q., & Zhang, L. (2023). An adaptive sand cat swarm algorithm based on cauchy mutation and optimal neighborhood disturbance strategy. *Biomimetics*, *8*(2), pp.191. https://doi.org/10.3390/biomimetics8020191

[21] Yang, H., Pu, C., Wu, J. and others, 2023. Enhancing OLSR protocol in VANETs with multi-objective particle swarm optimization. *Physica A: Statistical Mechanics and its Applications*, *614*, pp.128570. https://doi.org/10.1016/j.physa.2023.128570

[22] Zhang, Z., Zhu, H., Zhang, W. and others, 2023. Multi-objective optimization of traffic signal timing at typical junctions based on genetic algorithms. *Computer Systems Science and Engineering*, *47*(2), pp.1901-1917. https://doi.org/10.32604/csse.2023.039395 https://doi.org/10.32604/csse.2023.039395

[23] Almukhalfi, H., Noor, A., & Noor, T. H. 2024. Traffic management approaches using machine learning and deep learning techniques: A survey. *Engineering Applications of Artificial Intelligence*, *133*, pp. 108147. https://doi.org/10.1016/j.engappai.2024.108147

[24] Yazdani, M., Sarvi, M., Bagloee, S. A., Nassir, N., Price, J., & Parineh, H. 2023. Intelligent vehicle pedestrian light (IVPL): A deep reinforcement learning approach for traffic signal control. *Transportation research part C: emerging technologies*, *149*, pp. 103991. https://doi.org/10.1016/j.trc.2022.103991

[25] Haris, M., Bhatti, D. M. S., & Nam, H. (2024). A fast-convergent hyperbolic tangent PSO algorithm for UAVs path planning. *IEEE Open Journal of Vehicular Technology, 5,* pp. 681- 694. https://doi.org/10.1109/OJVT.2024.3391380