

Research Article

Research on Optimization of Improved Gray Wolf Optimization-Extreme Learning Machine Algorithm in Vehicle Route Planning

Shijin Li¹ and Fucai Wang² 

¹Academic Affairs Office, Yunnan University of Finance and Economics, Kunming, Yunnan 650221, China

²Yunnan Business Information Engineering School, Kunming, Yunnan 650000, China

Correspondence should be addressed to Fucai Wang; wangfucai_km@sina.com

Received 2 August 2020; Revised 16 September 2020; Accepted 22 September 2020; Published 6 October 2020

Academic Editor: Jia-Bao Liu

Copyright © 2020 Shijin Li and Fucai Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of intelligent transportation, intelligent algorithms and path planning have become effective methods to relieve traffic pressure. Intelligent algorithm can realize the priority selection mode in realizing traffic optimization efficiency. However, there is local optimization in intelligence and it is difficult to realize global optimization. In this paper, the antilearning model is used to solve the problem that the gray wolf algorithm falls into local optimization. The positions of different wolves are updated. When falling into local optimization, the current position is optimized to realize global optimization. Extreme Learning Machine (ELM) algorithm model is introduced to accelerate Improved Gray Wolf Optimization (IGWO) optimization and improve convergence speed. Finally, the experiment proves that IGWO-ELM algorithm is compared in path planning, and the algorithm has an ideal effect and high efficiency.

1. Introduction

Traffic congestion has become a common problem faced by many cities and even most countries in the world. Traffic congestion will not only increase the waiting time people spend on the road and cause urban traffic paralysis but also aggravate environmental pollution. According to understanding, when vehicles are in traffic jams, gasoline is not fully burned in the engine due to the low speed of the vehicle, resulting in pollutants that are several times higher than that when driving at normal speed. Path planning plays an important role in the urban transportation system. A well-planned transportation system can not only reduce the pressure of urban traffic but also bring great convenience to people's travel, and it has a great role in improving the urban landscape, enhancing the image of the city, and providing people with a good working and learning environment [1]. The main purpose of path planning is to design a traffic route that best meets the actual needs of travelers, provide travelers with the most time-saving and least expensive route, reduce

traffic congestion, improve road traffic conditions, and realize urban traffic continued development [2]. As artificial intelligence plays an increasingly obvious role in traffic forecasting, it can plan the traffic path and greatly reduce the time of congestion on the road.

Path planning has been widely used in high technology, daily life, decision management, and other fields [3–6]. According to different classification types, the classification of path planning is also different. When classified according to the degree of mastery of the current path, it is mainly divided into two parts: global path planning and local path planning. As the core of autonomous planning and navigation technology for mobile intelligent vehicles, researchers at home and abroad have done a lot of research in recent years. Mrazovic et al. [7] used big data analysis technology to calculate the current vehicle circulation and parking utilization rate and developed a multivehicle routing algorithm to reduce road congestion caused by parking or specific areas. In this paper, a two-layer local search algorithm (greedy random adaptive) is used to train a large number of

historical data collected. Through algorithm optimization comparison and analysis, urban road optimization management and reasonable route planning are realized. Gündüz et al. [8] proposed a new path planning method based on an improved genetic algorithm. The evaluation index and optimization function are given through path length and path energy. The three-dimensional path layer structure is established based on fuzzy neural network, and then the genetic algorithm is used to solve the optimal path. Lin et al. [9] proposed a path optimization algorithm for dynamic programming based on PSO, which adopts multi-objective PSO method and can realize parallel processing of different optimization objectives. Ant colony algorithm adopts a positive feedback mechanism, and the search process converges continuously. It has parallelism and greatly improves the operation efficiency of the algorithm. Heuristic probabilistic search has better global optimization ability. However, the disadvantage of ant colony algorithm is that the computation is large [10]. Mauricio et al. [11] adopted a particle swarm optimization method to solve the nonuniform vehicle routing problem with terminal. Through adaptive inertia weight and local search strategy, a better vehicle route scheme is solved. Experimental results show that this method can effectively save the total delivery cost and the number of vehicles required for distribution. Vareias et al. [12] proposed a random time vehicle routing optimization method, which assigns corresponding time to vehicles to solve the optimization problem. The time-consuming and time-consuming of vehicles in the planned route are evaluated, a time window is set for each vehicle, and a penalty and restraint mechanism is set for vehicles late and early. A two-level solution method is implemented, and the cost of the path is optimized and the time window is decomposed into subwindows to evaluate the optimization effect. Hiraishi [13] proposed a route optimization scheme with passengers as the criteria for judging the vehicle status. The route is planned by judging the passenger status. For example, if the passenger is judged to be tired, a time-consuming route is planned. Saed et al. [14] used memory storage mode through the optimal route, adaptive decision-making cognitive framework; this method can memorize, store, and optimize the route of vehicle information and update the optimal route mode through continuous learning. These algorithms have a certain ability to intelligently process and optimize information, but they do not evaluate historical time and consumption. The restraint mechanism and real-time road conditions cannot be better evaluated, and the road conditions cannot be predicted in real time. The above-mentioned documents have certain advantages in solving real-time path optimization and have a good prediction effect in a short time.

2. Gray Wolf Optimization Algorithm Strategies

2.1. Standard Wolf Optimization Algorithm. Gray Wolf Optimization (GWO) algorithm is a population intelligence algorithm proposed by Mirjalili Seyedali inspired by gray wolf predation behavior in 2014. The gray wolf pack has a 4-layer hierarchical mechanism of α , β , δ , and ω . Among them, wolf α

is the leader with the best adaptability among the gray wolves. β and δ are the two individuals with the second best fitness. Their task is to assist wolf α in managing the wolf pack and making decisions in the hunting process. ω is the remaining common wolf pack. The predation process is described as follows: first, wolf α leads the gray wolves to search, track, and approach their prey; then, wolves β and δ besiege the prey under the command of wolf α and summon ordinary wolves to attack the prey until they capture the prey. GWO algorithm completes the predation behavior by imitating the predation tasks such as gray wolf surrounding, hunting, and attacking, thus realizing the process of global optimization.

2.2. Mathematical Model of GWO. In GWO's leadership system, wolf α has the highest level, and wolves β and δ led by wolf α are also better individuals. Wolf ω searches paths under wolf α , wolf β , and wolf δ . As ω continuously optimizes the target, while α , β , and δ wolves continuously update the location information, other ω also continuously optimize the target.

Assuming that in D -dimensional space, the gray wolf pack $\{X_i, i = 1, 2, \dots, N\}$ consists of N gray wolves. The GWO algorithm is described as follows:

Encirclement stage: after the wolves determine the position of their prey, they first encircle the prey. The mathematical description is as follows:

$$\begin{aligned} D &= |C \cdot X_p(t) - X(t)|, \\ X(t+1) &= X_p(t) - A \cdot D, \end{aligned} \quad (1)$$

where D is the distance between the gray wolf and its prey; $X_p(t)$ is the position of the prey after the t -th iteration (the current optimal solution); $X(t)$ is the position of the gray wolf (the position of the feasible solution) after the t -th iteration; A and C are coefficient factors, defined as follows:

$$\begin{aligned} A &= 2a \times r_1 - a, \\ C &= 2 \times r_2, \end{aligned} \quad (2)$$

where r_1 and r_2 are random numbers within $[0, 1]$; t is the number of iterations of the algorithm; and \max is the maximum number of iterations. $a = 2 \rightarrow 0$ with the increase of iteration times t :

$$a = 2 - 2\left(\frac{t}{\max}\right). \quad (3)$$

Hunting stage: wolf α can quickly find the position of the prey and search for the prey. When wolf α finds the position of the prey, wolves α , β , and δ have a certain understanding of the position of the prey and require ω

to approach the position of the prey. This is the prey process of wolves.

After the siege phase is completed, wolf α leads wolves β and δ to hunt down their prey. In the process of hunting, the position of individual wolves will move with the escape of prey:

$$\begin{aligned} X_1 &= X_\alpha - A_1 |C_1 X_\alpha(t) - X(t)|, \\ X_2 &= X_\beta - A_2 |C_2 X_\beta(t) - X(t)|, \\ X_3 &= X_\delta - A_3 |C_3 X_\delta(t) - X(t)|, \end{aligned} \quad (4)$$

where X_α , X_β , and X_δ represent the current positions of wolves α , β , and δ ; $X(t)$ indicates the current gray wolf position; C_1, C_2, C_3 are random vectors.

Update the location of wolf α :

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3}. \quad (5)$$

By continuously updating the location information, ω wolf continuously updates the location to obtain the optimal location information. The candidate solution in the gray wolf algorithm updates its own position by taking the position information of the leadership level. Candidate solutions are distributed in a random circle defined by the leadership level. From the above formula and schematic diagram, it can be summarized that the optimization process of gray wolf algorithm is to elect wolves α, β , and δ in turn and then use the elected elite individuals to evaluate the possible positions of the prey, and then other individuals in the group randomly update the positions around the prey with reference to the position information of wolves α, β , and δ .

3. Improved Gray Wolf Optimization Algorithm

Gray Wolf Optimization algorithm is a naturally inspired search algorithm, which has good performance and efficiency in solving various optimization problems. The ‘‘flame’’ variable in the process of lateral orientation is an important parameter used by GWO to update the new position of the population. However, the reduction of flame will slow down the convergence speed, thus affecting the quality of the final solution. In order to overcome this shortcoming, a new antilearning (OBL) moth flame algorithm is proposed. In addition, for cross-border moths, a mirror cross-border strategy is adopted to ensure that moths are within the dimension range and improve population diversity.

3.1. Antilearning Algorithm. OBL was first proposed by Tizhoosh in 2005. It is described as follows: suppose the point $\mathbf{X} = (x_1, x_2, \dots, x_i, \dots, x_d)$ in d -dimensional space, where $x_1, x_2, \dots, x_i, \dots, x_d \in R$ and $x_i \in [a_i, b_i]$. Its reverse point $\tilde{\mathbf{X}} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_i, \dots, \tilde{x}_d)$ is defined as follows:

$$\tilde{x}_i = a_i + b_i - x_i. \quad (6)$$

On the basis of GWO, a supplementary function H based on antilearning strategy is added to speed up the

convergence of GWO, and a high-precision solution is obtained. The H function is described as follows:

$$H = (A + B) - C \times \text{rand}() \quad (7)$$

$$M_i = H, \quad (8)$$

$$A = \begin{cases} \Gamma, & s > \text{rand}() \\ U, & \text{otherwise,} \end{cases} \quad (9)$$

$$B = \begin{cases} \Theta, & s > \text{rand}() \\ L, & \text{otherwise,} \end{cases} \quad (10)$$

where $s = (\text{Current number of iterations} / \text{Maximum number of iterations})$, Γ and Θ are the largest and smallest positions of gray wolf respectively, U and L are the upper and lower limits of the search space, C is that optimal flame position in any iteration process, and rand is a random number between (0, 1). Equation (7) is a new form of antilearning. H function generates a new reverse solution according to the reverse position of gray wolf.

3.2. Gray Wolf Cross-Border Mirror Strategy. In the GWO optimization process, the gray wolf updates its position according to formula (5), and its position coordinates will cross the border and be continuously placed on the boundary, which not only reduces the diversity of the gray wolf population but also easily makes the algorithm fall into local optimization, thus affecting the optimization performance of the algorithm. In view of this situation, use equation (11) to update the cross-border gray wolf:

$$X(t+1)_{i,j} = \begin{cases} 2L_j - \text{rand}() \times X(t+1)_{i,j}, & X(t+1)_{i,j} \leq L_j, \\ 2U_j - \text{rand}() \times X(t+1)_{i,j}, & X(t+1)_{i,j} > U_j, \end{cases} \quad (11)$$

where $X(t+1)_{i,j}$ is the j -dimensional cross-border position in the i -th gray wolf; L_j and U_j are the value range of the j -th dimension. Repeated operation formula (11) can map $X(t+1)_{i,j}$ to the range of dimensions, ensure population diversity, and effectively improve the optimization performance of GWO algorithm.

3.3. Steps to Improve Gray Wolf Optimization Algorithm. An Improved Gray Wolf Optimizer (IGWO) is proposed by improving the gray wolf algorithm. The implementation steps of IGWO are summarized as follows:

Step 1: set algorithm parameters, including population size N , maximum iteration times max , search space dimension dim , search space upper bound U , and lower bound L .

Step 2: initialize that population according to the method in Section 2.2, $p = 2 * \text{dim} + 3$.

Step 3: calculate and sort the fitness values of gray wolf individuals according to the objective function, find out the individuals with the highest fitness according to the

fitness values, and save their positions as X_α , X_β , and X_δ , respectively.

Step 4: update the position according to the step size segment step size update mechanism of formulae (4) and (5).

Step 5: update the gray wolf population position according to the algorithm designed by formula (11).

Step 6: update parameters a , A , and C , where the original linear convergence factor a is changed to exponential nonlinear convergence factor.

Step 7: determine whether the maximum number of iterations has been reached, and if so, stop the iteration and output.

4. Extreme Learning Machine (ELM)

Professor Huang et al. [15] proposed the Extreme Learning Machine in 2006. ELM algorithm is an improved new algorithm based on SLFN. The biggest advantage of ELM algorithm over traditional SLFN algorithm is that it does not need to update the parameters in training, such as the weight between the input layer and the hidden layer and the threshold value of hidden layer neurons. Once the number of hidden layer nodes is determined, ELM algorithm can obtain a unique global optimal solution with good generalization and general approximation capability.

Extreme Learning Machine is a kind of single-hidden layer feedforward neural network (SLFN), which has been widely concerned and applied because of its fast convergence and fast learning ability and good nonlinear classification ability.

Set n different training samples (x_i, y_i) , of which the k -th input $x = [x_1, x_2, \dots, x_N]^T \in R^N$ corresponds to the output $y = [y_1, y_2, \dots, y_m]^T \in R^m$, set the number of input layer nodes of the network as N , set the number of hidden layer nodes as M , and the activation function is expressed as $g(x)$; then the unified model of SLFN is

$$\sum_{j=1}^M \beta_j g(x_i) = \sum_{j=1}^M \beta_j g(w_j \cdot x_i + b_j) = y_i, \quad i = 1, \dots, n, \quad (12)$$

where w_j represents the j -th weight of the input layer and the hidden layer, b is the node bias of the hidden layer, $g(\cdot)$ is the excitation function of the hidden layer, the output of the output layer is $y = [y_1, y_2, \dots, y_m]^T \in R^m$, the expected output is $e = [e_1, e_2, \dots, e_m]^T \in R^m$, $w = \{w_{ij}\}$ represents the connection weight between neurons and neurons (i and j , resp., represent the serial numbers of neurons in the input layer and the hidden layer), and $\beta = \{\beta_{ij}\}$ represents the connection weight between neurons and neurons (i and j , resp., represent the serial numbers of neurons in the hidden layer and the output layer). Then the output of each sample is

$$y = [g(w_1 x + b_1)g(w_2 x + b_2), \dots, g(w_m x + b_m)]\beta_i + b_i. \quad (13)$$

The equation is expressed as

$$h(x) = g(w_1 x + b_1)g(w_2 x + b_2), \dots, g(w_m x + b_m). \quad (14)$$

$H(x)$ refers to the response of all neurons in the hidden layer to the input vector x and calculates the output:

$$y = h(x) \cdot \beta_i + b_i. \quad (15)$$

The system output for N samples is

$$Y = H \cdot \beta + b_i. \quad (16)$$

The error function is defined as

$$E = \|T - Y\| = \|T - (H \cdot \beta + b_i)\|, \quad (17)$$

where $\|\cdot\|$ represents norm. The above is the basic mathematical reasoning process of a single-hidden layer feedforward neural network. All its parameter values are continuously updated, and the thresholds and weights of both the hidden layer and the output layer will be optimized.

5. Experimental Scheme

In this paper, a directed graph $G = (V, E)$ is defined for urban intersections to represent the key intersection conditions of the city, as shown in Figure 1, where $V = (v_1, v_2, \dots, v_n)$ represents N key intersections in the city and $E = (e_1, e_2, \dots, e_m)$ represents all roads in the city, totaling m roads. The traffic flow between each key intersection is expressed by matrix $D = \{d_{ij}\}$, $1 \leq i, j \leq n$, where d_{ij} represents the number of vehicles from stations v_i to v_j . However, the cost matrix $C = \{c_{ij}\}$, $1 \leq i, j \leq n$, of vehicles passes between intersections. It shows that c_{ij} represents the cost of vehicles arriving at v_j from station v_i , which generally refers to time or distance. Because time is proportional to distance when the vehicle speed is the same, the cost is the same regardless of time or distance. However, if there is no road between station v_i and station v_j , c_{ij} is defined as a large number. For example, $c_{01} = 6, c_{10} = 4$ means that the two reintersections communicate at different times and the road conditions between the two intersections are different.

In Figure 1, the passing time before different intersections in the city at a certain moment shows different passing time at different times, different seasons, and different months, different time points in the day show different passing time, and the road conditions show different, specifically showing congestion, slow movement, and smooth flow. In Figure 1, the traffic conditions of different roads back and forth are shown, and the numbers represent specific locations. The number between location and location represents the cost that can be done between locations and generally refers to the time or cost that can be done.

Through the relevant statistical analysis of GPS data, the longitude and latitude of different sections at different times and the road conditions of the current map are analyzed. Relevant statistics and analysis are carried out on longitude, latitude, and geographic information in GPS data, vehicle information at different time points of road sections in GPS

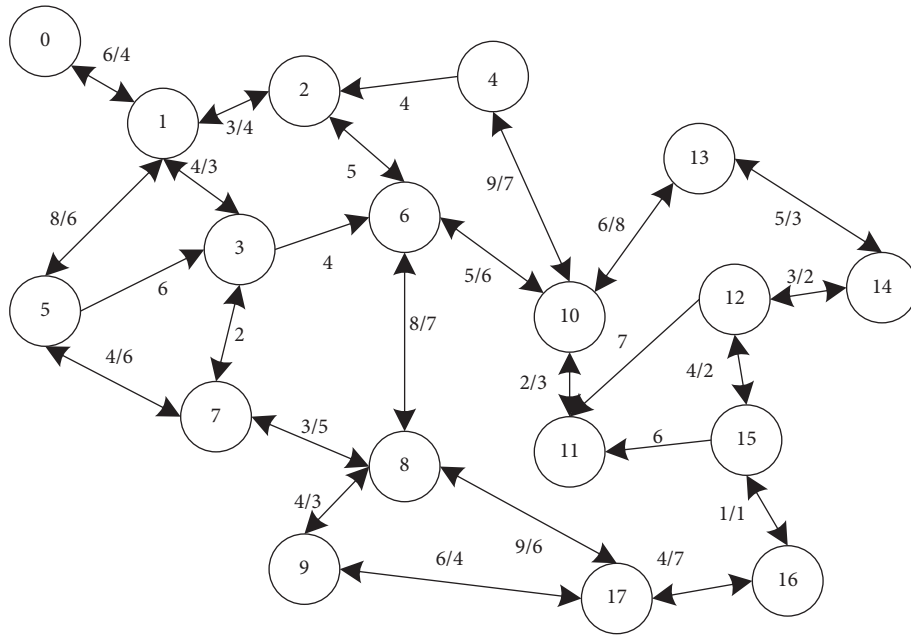


FIGURE 1: Abstract diagram of the route of the key intersection of urban traffic at a certain moment.

TABLE 1: Statistical table of road conditions of vehicles.

Road ID	Vehicles ID	Road length (m)	Entry time	Leave time	Speed (km/h)	Road condition
3	15	478.6	08:03:18	08:04:23	32.6	3
2	18	1242.8	12:52:00	12:54:10	23.2	2
1	121	1254.4	21:18:15	21:20:15	25.1	3
9	128	521.3	10:57:06	10:59:10	10.5	1
3	675	613.2	18:47:59	18:50:23	9.2	1
...						

data is counted, key intersections are numbered, and the sorted data are described in Table 1.

The above is the statistics of a certain point in time, and the road condition information is usually counted every 5 minutes. Generally, the statistical results of the current real-time road conditions can be divided into three states: blockage, slow travel, and comfort. 1, 2, and 3 are used, respectively. The threshold judgment criteria are as follows: blockage is less than 12 km/h, slow travel is 12~25 km/h, and unblocked is greater than 25 km/h. The data in Table 1 is obtained through raw data statistics. Road ID denotes the number of the road in the map; vehicle ID denotes the number of the vehicle; road section length denotes the actual length of a road section; entry time denotes the start time when the vehicle enters the road section; departure time denotes the end time when the vehicle leaves the road section; speed denotes general passing calculated by dividing the length of the road segment by the time; road conditions denotes classify road conditions by speed.

Through the gray wolf algorithm to predict the key intersections of urban traffic, predicting the key routes through

different intersections is the focus of this paper. In order to be able to predict the optimal route from the starting point to the ending point and to predict different intersections, Figure 2 compares the prediction effects of different routes.

The optimal route in Figure 2 is the route selection when the current key route is unblocked, which is referred to herein as the optimal route. The route selection recommended by IGWO-ELM algorithm is basically in high agreement with the optimal route in terms of route, but this is the actual route selection, considering the road conditions at different times. IGWO-ELM algorithm is ideal in execution effect, which shows that the algorithm has a good execution effect. In order to reflect the performance effect of the algorithm, the comparison effect between PSO and GWO algorithms is shown in Figure 3.

In Figure 3, compared with the time under different paths, the time of IGWO-ELM algorithm in practical application is close to the theoretical value. PSO and GWO algorithms may have differences in running time. In some paths, the running efficiency of the two algorithms is the same, sometimes PSO is better than GWO, and the time efficiency of the two algorithms alternates.

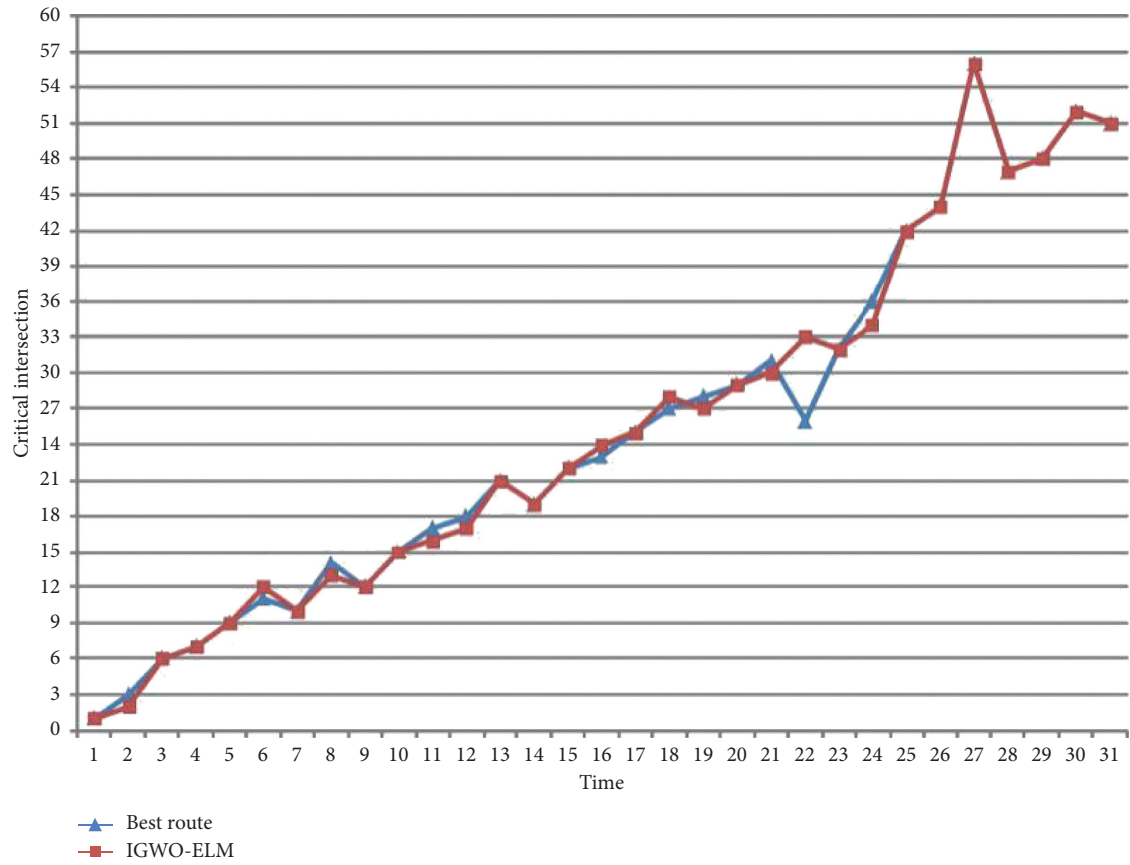


FIGURE 2: Comparison of IGWO-ELM route and optimal route in Figure 2.

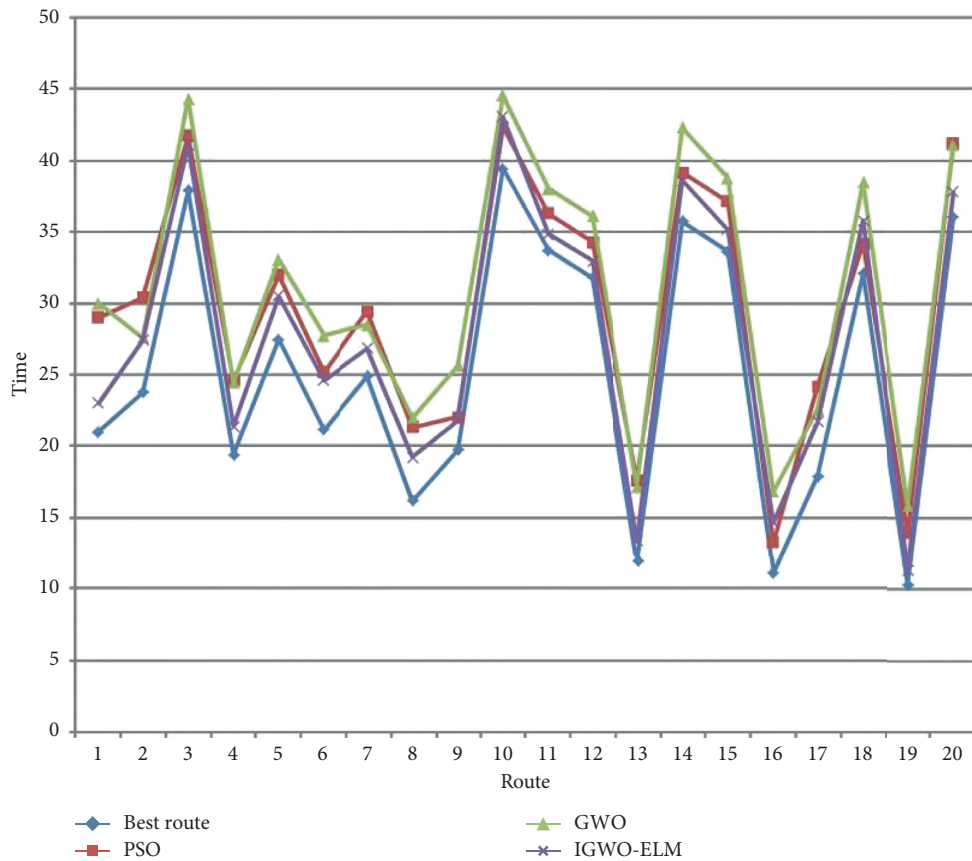


FIGURE 3: Comparisons of three algorithms under different route.

6. Conclusion

This paper proposes an improved gray wolf algorithm to be applied to urban traffic path planning. Reasonable traffic path planning can improve traffic operation efficiency and save a lot of time. The basic gray wolf algorithm is easy to fall into local optimization, and it is difficult to obtain global total optimization. The gray wolf algorithm is improved through the antilearning model; thus, the actual global optimization is achieved. Finally, the experimental comparison shows that IGWO-ELM algorithm has obvious advantages in execution efficiency and time. Considering the comparative advantage of IGWO-ELM algorithm in efficiency, different time points or seasonal points will be used as research objects in the later period. The historical data are trained to predict the traffic planning for a period of time in the future, and the vehicles are reasonably planned to alleviate the traffic congestion.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors contributed equally to the writing of this paper. All authors read and approved the final manuscript.

Acknowledgments

This work was supported by the Design and Application of Intelligent Teaching Model Based on Rain Classroom Grant no. 41620200001/002, Data Mining Research on MOOC Learning Behavior Based on Internet Environment under Grant no. 80059900227, and Research on the Model of Flip Classroom Teaching Based on Micro-Lectures under Grant no. 80059900253.

References

- [1] W.-H. Wang, Y. Wang, Z. Li, C.-Z. Wei, J.-C. Zhao, and L.-q. Sun, "Effect of a strengthened ecological floating bed on the purification of urban landscape water supplied with reclaimed water," *Science of the Total Environment*, vol. 622-623, pp. 1630-1639, 2018.
- [2] S. Zhao, Y. Liu, M. Sundhararajan et al., "Evaluation system for the sustainable development of urban transportation and ecological environment based on SVM," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 2, pp. 831-838, 2018.
- [3] J. B. Liu, J. Zhao, and Z. Cai, "On the generalized adjacency, Laplacian and signless Laplacian spectra of the weighted edge corona networks," *Physica A*, vol. 540, pp. 1-11, 2020.
- [4] W. Liu, J. Li, L. Ren et al., "Exploring livelihood resilience and its impact on livelihood strategy in rural China," *Social Indicators Research*, vol. 150, no. 3, pp. 977-998, 2020.
- [5] J.-B. Liu, J. Zhao, H. He, and Z. Shao, "Valency-based topological descriptors and structural property of the generalized sierpiński networks," *Journal of Statistical Physics*, vol. 177, no. 6, pp. 1131-1147, 2019.
- [6] W. Jiang, D. R. Carter, H. L. Fu et al., "The impact of the biomass crop assistance program on the United States forest products market: an application of the global forest products model," *Forests*, vol. 10, no. 3, pp. 1-12, 2019.
- [7] P. Mrazovic, E. Eser, H. Ferhatosmanoglu et al., "Multi-vehicle route planning for efficient urban freight transport," in *Proceedings of the 2018 international conference on intelligent systems (IS), IEEE, Funchal-Madeira, Portugal*, September 2019.
- [8] M. G. Gündüz, M. S. Kiran, and E. Özceylan, "A hierarchical approach based on swarm intelligence to solve the traveling salesman problem," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 23, pp. 103-117, 2015.
- [9] Y.-H. Lin, L.-C. Huang, S.-Y. Chen, and C.-M. Yu, "The optimal route planning for inspection task of autonomous underwater vehicle composed of MOPSO-based dynamic routing algorithm in currents," *Applied Ocean Research*, vol. 75, pp. 178-192, 2018.
- [10] W. Y. H. Adoni, T. Nahhal, B. Aghezzaf et al., "The Map-Reduce-based approach to improve the shortest path computation in large-scale road networks: the case of A* algorithm," *Journal of Big Data*, vol. 5, no. 1, p. 16, 2018.
- [11] G. E. Mauricio, R. I. Bolaños, and J. W. Escobar, "A meta-heuristic algorithm for the MultiDepot vehicle routing problem with heterogeneous fleet," *International Journal of Industrial Engineering Computations*, vol. 9, no. 4, pp. 461-478, 2018.
- [12] A. D. Vareias, P. P. Repoussis, and C. D. Tarantilis, "Assessing customer service reliability in route planning with self-imposed time windows and stochastic travel times," *Transportation Science*, vol. 53, no. 1, pp. 256-281, 2019.
- [13] H. Hiraishi, "Passenger condition based route-planning for cognitive vehicle system," *International Journal of Software Science and Computational Intelligence*, vol. 10, no. 2, pp. 25-35, 2018.
- [14] Y. Saeed, K. Ahmed, M. Zareei et al., "In-vehicle cognitive route decision using fuzzy modeling and artificial neural network," *IEEE Access*, vol. 7, pp. 20262-20272, 2019.
- [15] G. B. Huang, Q. Y. Zhu, C. K. Siew et al., "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489-501, 2006.