Application in Dynamic Path Selection for Emergency Vehicles Based on Hybrid Cuckoo Algorithm Optimizing Neural Network Model

Feng Yang[1, 2*], Chunming Ye[1], Jianfeng Ren[1]

ABSTRACT

If any emergency occurs in a city, emergency vehicle scheduling must be such as to shorten the path travel time. The biggest difficulty in scheduling is how to measure the real-time changes in the traffic conditions of urban road network. To study the dynamic path selection of emergency vehicles under city emergency, this paper abstracts the urban road network into a map composed of nodes and edges and takes the shortest path as the optimization objective. Firstly, the author takes the K-nearest sample set from the similar historical sample sets, predicts the real-time vehicle speed and establishes the path travel time function. Then, the author uses path reliability to measure the impacts of real-time traffic conditions on the overall travel time and constructs the two-stage objective optimization model for dynamic optimal path selection. Finally, based on this model, the author proposes a hybrid cuckoo search algorithm and uses it to optimize the weights and thresholds of neural network model to solve the K shortest-time paths in the dynamic road network, and take a partial road network in Yangpu District of Shanghai as an example for simulation test. The test results show that the proposed dynamic path selection model can reflect the actual scenario of emergency vehicle scheduling under emergency, and that the neural network model based on the hybrid cuckoo search algorithm is used to train weights and thresholds, so that the algorithm has a fast convergence speed and can solve the problem well. Compared with the classic cuckoo search algorithm and the particle swarm optimization algorithm, this algorithm has better performance.

Key Words: Emergency Vehicles, Dynamic Path, Hybrid Cuckoo Search Algorithm, Neural Network, FIFO Network

DOI Number: 10.14704/nq.2018.16.6.1578 **NeuroQuantology 2018; 16(6):625-637**

Introduction

City emergency refers to any natural disaster, accident and public event that suddenly occurs in a city, causes or may cause serious social harm and needs to be handled through emergency measures. Emergencies are generally sudden, highly uncertain and socially influential and re-quire non-procedural decisions (Xu *et al*., 2017). After an emergency occurs, it is necessary to respond to the emergency as soon as possible to shorten the time taken to rescue the injured persons. Therefore, emergency vehicles must be scheduled in such a way as to minimise the response time, of which the core is the travel time of emergency vehicles on the road, and what determines the travel time is the path choice for emergency vehicles. Due to the changes of traffic volume and the impacts of emergencies, the traffic conditions in various parts of the road network are dynamically changing. As a result, emergency vehicle scheduling must select the rational dynamic path based on the real-time dynamically changing conditions of the road network, with the shortest response time as the objective.

Corresponding author: Feng Yang
Address: [1]Business School, University of Shanghai for Science and Technology, Shanghai 200093, China; [2]College of Management, Henan University of Traditional Chinese Medicine, Zhengzhou 450046, China
e-mail ✉ yangfeng1126@126.com

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
Received: 2 March 2018; **Accepted:** 7 May 2018

Urban traffic conditions can change at any time. If the vehicle passing time is taken as the weight of the road section in the road network (Duan *et al*., 2015), the dynamic characteristics of the road net-work can be well expressed. In order to solve the event response and resources distribution problem in the traffic event management, reference (Ozbay *et al*., 2013) studied the scheduling decision-making time, proposed a integer programming model for scheduling decision-making time with probability constraints, and established a road network model used for the scheduling of emergency resources. By studying the differences between static network and time-varying network, reference (Ran *et al*., 2012) imposed the consistency constraint on the dynamic network, i.e. the first-in and first-out (FIFO) rule.

Due to the urgency of emergency rescue, emergency vehicle path selection is actually a matter of finding the shortest-time path, and its solution is based on the shortest-path algorithm. Generally, the algorithms for solving the shortest path are all based on static networks. If the network model is established using dynamic weights, the method for solving the shortest path problem is called the dynamic shortest-path algorithm. At present, there are only a few studies on such problem. Literature (Yang, 2013; Feng *et al*., 2014) proved that the shortest path problem in a time-dependent network is NP-Hard and proposed and proved the concept of the shortest path stability from three aspects - shortest path length, optimal solution stability and stable branch. In addition, a few scholars considered the reliability factor when choosing the driving paths of vehicles. Tomsen *et al*., (2014) used the reliability grade to characterize the dynamic features of the vehicle path arrangement, and by studying the typical velocity distribution in intercity highways, it turned real-time speed into reliability and solved the dynamic vehicle path planning problem. Reference (Fu *et al*., 2015) proposed a K shortest path (KSP) algorithm, which aims at searching multiple alternative optimized paths in the network. This is an important tool to solve the multi-objective, multi-constraint shortest path problem. The earliest KSP algorithm adopted the classic Dijkstra algorithm to find the shortest path. KSP is a typical NP-hard problem, for which, parallel computing can reduce the time complexity of the algorithm effectively, but currently there is only limited research based on parallel algorithm (Xu *et al*., 2013), so for such problems, the artificial intelligence optimization algorithm is usually used to perform approximate calculation (Zhang *et al*., 2016). Neural network is a complex network composed of a large number of simple neurons, simulating the structure and behaviour of human brain neural network (Zhao *et al*., 2008). Among them, BP (Back Propagation) neural network is the most widely used one. However, classical BP algorithm correct weights and threshold by the gradient descent method, which may cause learning speed slowdown, easily affected by the sample and falling into local optimum spot, thus limiting its practical application value. In order to overcome the shortcomings of BP algorithm, to overcome this drawback, many scholars have combined with swarm intelligent algorithms to put forward improved algorithms. Cuckoo Search (CS) is just one of the swarm intelligence optimization algorithms (Santillan *et al*., 2017), which mimics the behaviours of cuckoos like laying their eggs and fostering their young cubs. It was first proposed by Yang from University of Cambridge and Deb from Cambridge Institute of Technology in 2009.

Targeting the dynamic scheduling of emergency vehicles under city emergency, this paper abstracts the urban road network into nodes and sections, takes the travel speed function as the weight of the road section, proposes the dynamic path selection method for emergency vehicles, establishes a two-stage dynamic path selection model. Considering CS algorithm has the advantage of few parameters, simple and strong global search capability，this paper proposes a hybrid cuckoo algorithm to optimize the weight and threshold of BP neural network to improve the convergence speed of BP neural network, and then uses the random coding scheme to ensure the path connectivity and designs the optimization strategy based on the shortest path search principle for the FIFO network to obtain K shortest-time paths.

Dynamic path selection problem for emergency vehicles

Problem introduction

In the complex urban road network, the objective of path selection is to find the optimal one among the various paths, to make the total travel time shortest. The shortest-time path is a special shortest path problem, which takes the path travel time as the weight. Suppose the road

network is abstracted into a graph $G(R,E)$ consisting of nodes $r_i \in R$ and edges $(r_i,r_j) \in E$, where R is the node set, and E is the edge set. If the travel time of the road section (r_i,r_j) is T_{ij}, and the weight is ω_{ij}, then:

$$\omega_{ij}=\begin{cases} T_{ij} & (r_i,r_j)\in E \\ +\infty & others \end{cases}$$

Usually we suppose the travel time T_{ij} of the road section is constant, in which case, the urban road network is a static network. However, the vehicle speed and the road network traffic are dynamically changing and there are also impacts from emergencies, so the travel time of the road section is also dynamically changing. Specifically, the emergency vehicle enters the road section (r_i,r_j) at a different time t, and the road section travel time T_{ij} also changes accordingly. Therefore, the road section travel time is a time-dependent function.

For the time period U, if Δt is the minimum time interval, which divides U into H discrete time intervals, then $U=\{[(t_0+\gamma\Delta t),(t_0+(\gamma+1)\cdot\Delta t)]\}$, where, $\gamma=0,1,\ldots,H-1$, t_0 is the initial time, $U_H=t_0+r\cdot\Delta t$ is the last moment, and the emergency vehicle can reach the accident node within $[t_0,U_H]$. In fact, the real-time road section travel speed at the moment t_0 can be obtained, but the real-time road section travel speed at other time within the period U can only be predicted based on the historical speed data. Suppose the sample set is obtained from the historical vehicle speed data and that the travel speeds within the continuous acquisition cycles $[t_0-\varphi_1\Delta t,t_0+\varphi_2\Delta t]$ before and after the moment t_0 are selected as the sample vectors, then all subclasses of samples in the sample set are concentrated in the centre $x_c=(v_c(t_0-\varphi_1\Delta t),\ldots,v_c(t_0+\varphi_2\Delta t))$, $c=1,\ldots,d$, where $v_c(t)$ represents the road section travel speed at the moment t in the sample x_c and d is the dimension of the sample. For the sample to be tested $x=(v(t_0-\varphi_1\Delta t),\ldots,v(t_0+\varphi_2\Delta t))$, the Euclidean distance between the samples can be obtained using the Euclidean distance formula:

$$D(x,x_c)=\|x-x_c\|=\sqrt{\sum_{c=1}^{d}\left|v(t_0+\varphi_1\cdot\Delta t)-v_c(t_0+\varphi_1\Delta t)\right|^2}$$

A class of samples at the closest distance from the sample to be tested x is selected, denoted as:

$$x^k=(v^k(t_0-\varphi_1\cdot\Delta t),\ldots,v^k(t_0),\ldots,v^k(t_0+\varphi_2\cdot\Delta t)),\ k=1,2,\ldots,K$$

where, K is the number of samples contained in the nearest-neighbour sample set. The travel speed at the time φ_2 after the initial time t_0 can be predicted as:

$$v^2(t_0+\varphi_2\cdot\Delta t)=\frac{1}{K}\sum_{k=1}^{K}v^k(t_0+\varphi_2\cdot\Delta t)$$

In any road section (r_i,r_j), the vehicle will not always run at a constant speed; therefore, the predicted value $v_{ij}^2(t)$ of the travel speed must incorporate time-varying information. After the scheduling decision-making time t_0, if $v_{ij}^2(t)$ is used to fit the vehicle speed function, there must be some difference between the fitted result and the actual vehicle speed function, and as a result, there must be some difference between the road section travel time function $T_{ij}(t)$ and the actual function. Therefore, the shortest-time path obtained by the travel speed function may not be the optimal path. It is necessary to build a dynamic path selection model and use the heuristic algorithm to solve it.

Construction of the dynamic path selection model

(1) Analysis of road section travel time

Suppose the length of the road section (r_i,r_j) is L_{ij}, that the real-time travel speed at the time t_0 is $v_{ij}(t_0)$, and that $t_\gamma=t_0+\gamma\cdot\Delta t$, $\gamma=1,2,\ldots,H$, then the predicted travel speed at the time t_γ is $v_{ij}^{'}(t_\gamma)_{t_0}$. According to the broken-line vehicle speed function defined in Reference (Figliozzi, 2009), the vehicle speed function at the time t_0 in the road section (r_i,r_j) is shown in Figure 1:

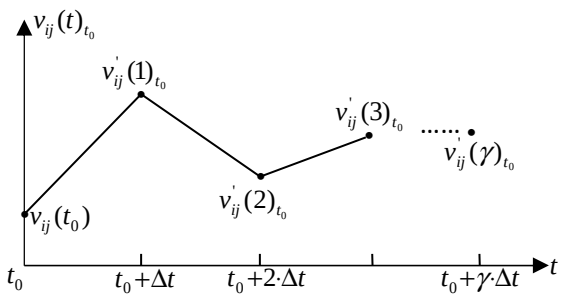

Figure 1. Travel speed function

The vehicle speed function is expressed as follows:

$$v_{ij}(t)_{t_0}=\begin{cases}\dfrac{v_{ij}^{'}(1)_{t_0}-v_{ij}(t_0)}{\Delta t}\cdot t+\dfrac{t_1\cdot v_{ij}(t_0)-t_0\cdot v_{ij}^{'}(1)_{t_0}}{\Delta t} \\ t_0\le t<t_1 \\ \dfrac{v_{ij}^{'}(\gamma+1)_{t_0}-v_{ij}^{'}(\gamma)_{t_0}}{\Delta t}\cdot t+\dfrac{t_{\gamma+1}\cdot v_{ij}^{'}(\gamma)-t_\gamma\cdot v_{ij}^{'}(\gamma+1)}{\Delta t} \\ t_\gamma\le t<t_{\gamma+1},\ \gamma\in[1,H-1]\end{cases} \quad (1)$$

$v_{ij}(t)_{t_0}$ is continuous within $[t_0,U_H]$, so it is integrable within $[t_0,U_H]$. If the emergency vehicle enters the road section (r_i,r_j) at the time $y(y\ge t_0)$, then the travel distance is the function of travel time $x(y\le x\le U_H)$:

$$\int_y^x v_{ij}(t)_{t_0}\,dt=\int_{t_0}^x v_{ij}(t)_{t_0}\,dt-\int_{t_0}^y v_{ij}(t)_{t_0}\,dt=\mu(x)-\eta(y) \quad (2)$$

Where,

$$\mu(x)=\int_{t_0}^x v_{ij}(t)_{t_0}\,dt=\int v_{ij}(x)_{t_0}\,dx+C=$$

$$\begin{cases}\dfrac{1}{2}\cdot\dfrac{v_{ij}^{'}(1)_{t_0}-v_{ij}(t_0)}{\Delta t}\cdot x^2+\dfrac{t_1\cdot v_{ij}(t_0)-t_0\cdot v_{ij}^{'}(1)_{t_0}}{\Delta t}\cdot x+C_1 \\ t_0\le x<t_1 \\ \dfrac{1}{2}\cdot\dfrac{v_{ij}^{'}(\gamma+1)_{t_0}-v_{ij}^{'}(\gamma)_{t_0}}{\Delta t}\cdot x^2+\dfrac{t_{\gamma+1}\cdot v_{ij}^{'}(\gamma)-t_\gamma\cdot v_{ij}^{'}(\gamma+1)}{\Delta t}\cdot x+C_i \\ t_\gamma\le x<t_{\gamma+1},\ \gamma\in[1,H-1]\end{cases} \quad (3)$$

$$\eta(y)=\int_{t_0}^y v_{ij}(t)_{t_0}\,dt=\int v_{ij}(y)_{t_0}\,dy+C=$$

$$\begin{cases}\dfrac{1}{2}\cdot\dfrac{v_{ij}^{'}(1)_{t_0}-v_{ij}(t_0)}{\Delta t}\cdot y^2+\dfrac{t_1\cdot v_{ij}(t_0)-t_0\cdot v_{ij}^{'}(1)_{t_0}}{\Delta t}\cdot y+C_1 \\ t_0\le y<t_1 \\ \dfrac{1}{2}\cdot\dfrac{v_{ij}^{'}(\gamma+1)_{t_0}-v_{ij}^{'}(\gamma)_{t_0}}{\Delta t}\cdot y^2+\dfrac{t_{\gamma+1}\cdot v_{ij}^{'}(\gamma)-t_\gamma\cdot v_{ij}^{'}(\gamma+1)}{\Delta t}\cdot y+C_i \\ t_\gamma\le y<t_{\gamma+1},\ \gamma\in[1,H-1]\end{cases} \quad (4)$$

where, C_1, C_i, $i=2,3,...,H-1$ is the integration constant, which is obtained based on the continuity of the primitive function of $v_{ij}(t)_{t_0}$. Let equation (2) be equal to the length of the road section L_{ij}, and then the time x at which the emergency vehicle leaves the road section x is:

$$x=\rho(L_{ij}+\eta(y)) \quad (5)$$

where, ρ is the inverse function of μ. Then the road section travel time function can be obtained:

$$T_{ij}(t)=\rho(L_{ij}+\eta(t))-t \quad (6)$$

Definition 1 in a time-dependent network, if the road section travel time $T_{ij}(t)$ is continuous and differential everywhere, and $\forall t,T_{ij}(t)\le\Delta t+T_{ij}(t+\Delta t)$, then the road section is called an FIFO arc. If each arc in the time-dependent network is an FIFO, the network is called an FIFO network, and a network that contains at least one non-FIFO arc is called a non-FIFO network.

Theorem 1: the urban road network in which the road section travel speed is a broken-line vehicle speed function is an FIFO network.

Proof: According to equation (6), the road section travel time $T_{ij}(t)$ continuous and differential everywhere. Below is the proof of $\forall t,T_{ij}(t)\le\Delta t+T_{ij}(t+\Delta t)$.

$$\begin{aligned}&T_{ij}(t+\Delta t)+\Delta t-T_{ij}(t)\\&=\rho(L_{ij}+\eta(t+\Delta t))-(t+\Delta t)+\Delta t-[\rho(L_{ij}+\eta(t))-t]\\&=\rho(L_{ij}+\eta(t+\Delta t))-\rho(L_{ij}+\eta(t))\end{aligned}$$

Where,

$$\begin{aligned}&\eta(t+\Delta t)-\eta(t)\\&=\int_{t_0}^{t+\Delta t}v_{ij}(t)_{t_0}\,dt-\int_{t_0}^{t}v_{ij}(t)_{t_0}\,dt\\&=\int_{t_0}^{t}v_{ij}(t)_{t_0}\,dt-\int_{t}^{t+\Delta t}v_{ij}(t)_{t_0}\,dt-\int_{t_0}^{t}v_{ij}(t)_{t_0}\,dt\\&=\int_{t}^{t+\Delta t}v_{ij}(t)_{t_0}\,dt\ge 0\end{aligned}$$

It can be seen that the function $\eta(y)$ increases monotonically, and similarly, $\mu(x)$ and its inverse function $\rho(x)$ also increase monotonically. Therefore, it can be concluded that $T_{ij}(t+\Delta t)+\Delta t-T_{ij}(t)\ge 0$, and $\forall t,T_{ij}(t)\le\Delta t+T_{ij}(t+\Delta t)$.

(2) Path reliability analysis

In case of a city emergency, the urban road network, especially the road traffic around the emergency area will inevitably be affected, so during the selection of emergency vehicle path, it is necessary to minimize the impact on the overall travel time when the estimated travel time of some road section along the path deviates from the actual time. This paper uses path reliability to measure this impact.

Under normal circumstances, the traffic flow density of the road section (r_i,r_j) is the theoretical normal value, and when the emergency vehicle is in a free travel state and the travel speed is close to the speed limit, the road

section travel time is T_{ij}. For a path $\prod_{i=s}^{d}(r_i,r_{i+1})$ from the origin r_s to the destination r_d, according to the predicted value of road section travel time, the emergency vehicle enters the road section (r_i,r_{i+1}) at the time t_i and exits after $T_{i,i+1}(t_i)$. Considering the difference between the actual time at which the emergency vehicle enters the node r_i and t_i, assuming that the time interval of the actual entry is $[Et_i,Lt_i]$, where, $Et_i=\sum_{k=s}^{i-1}T_{k,k+1}$ and $Lt_i\in[t_i,U_H]$, the maximum travel time it takes for the emergency vehicle to pass the road section after the vehicle enters it during this time interval is:

$$T_{i,i+1}=Max\{T_{i,i+1}(t),Et_i\le t\le U_H\} \tag{7}$$

Then the maximum delay of the road section (r_i,r_{i+1}) travel time is:

$$\Delta T_{i,i+1}=T_{i,i+1}-T_{i,i+1}(t_i) \tag{8}$$

Then the reciprocal of the total delay of the path $\prod_{i=s}^{d}(r_i,r_{i+1})$ is defined as path reliability:

$$W(\prod_{i=s}^{d}(r_i,r_{i+1}))=\frac{1}{\sum_{i=s+1}^{d-1}(T_{i,i+1}-T_{i,i+1}(t_i))} \tag{9}$$

(3) Two-stage model for dynamic optimal path selection

The urban road network is abstracted into a time-dependent network model $(R,E,T\times U)$, where, $R=\{r_1,r_2,...,r_M\}$ represents a set of nodes, and the connecting arc between adjacent nodes is the road section $(r_i,r_j)\in E$. U is time interval, and $T_{ij}(t)$ is the road section travel time function defined in the interval U, which indicates the time it takes for the emergency vehicle to travel from node r_i to node r_j after it enters the road section at the time U, $\forall t\in U$. For $t>t_0+H\cdot\Delta t$, let $T_{ij}(t)=\infty$. Assuming that the emergency vehicle is located at the node r_s, that the node where the accident is located is r_d, and that the emergency vehicle starts at the time t_0, then the optimal path for the emergency vehicle towards the accident node is $P_{sd}(t_0)$, and the shortest travel time is $T_{sd}(t_0)$. Based on the dynamic shortest path model, the path reliability is optimized and the two-stage objective optimization model is established as follows:

$$MaxW(\prod_{i=s}^{d}(r_i,r_{i+1})) \tag{10}$$

$$s.t.\begin{cases} T_{sd}(t_0)=Min\sum_{i=s}^{d}T_{i,i+1}(t_i) & (11)\\ s.t.\begin{cases} t_i=\begin{cases} t_{i-1}+T_{i-1,i}(t_{i-1}) & i=s+1,...,d-1\\ t_0 & i=s \end{cases} & (12)\\ (r_s,r_1),...,(r_i,r_{i+1}),...,(r_{d-1},r_d)\in E & (13)\\ r_s,r_1,...,r_i,r_{i+1},...,r_{d-1},r_d\in R & (14)\\ r_s\ne r_1\ne...\ne r_i\ne...\ne r_{d-1}\ne r_d & (15) \end{cases} \end{cases}$$

Equation (10) is the objective of the path reliability, which is to maximise the reliability of the selected path. Equation (11) is the objective of path travel time. Due to the time urgency of emergency rescue, the optimization objective is defined as minimising the emergency vehicle travel time, which is, minimising the sum of the travel time that the emergency vehicle spends in passing each road section. Equation (12) calculates the time t_i at which the emergency vehicle enters each road section (r_i,r_j), $i\in[s,d-1]$ from the starting point r_s. Equation (13) (14) and (15) are constraints, to ensure that the path sequence is a no-return path.

Neural network model of hybrid cuckoo search algorithm optimization for solving the dynamic path selection problem

Hybrid cuckoo search algorithm

The Cuckoo Search Algorithm (CSA) solves the optimization problem based on the Lévy flight (Sharma *et al.*, 2016) mechanism of animals and the cuckoo's nesting and spawning behaviours. CSA has such advantages as strong global search ability, few input parameters, excellent search path and strong problem solving ability (Nanda *et al.*, 2014). In many aspects, CSA outperforms particle swarm optimization (PSO) algorithm and genetic algorithm and ant colony algorithm (Guerrero *et al.*, 2015) and (Kanagaraj *et al.*, 2014) and (Liu *et al.*, 2017). However, as CSA controls the step size through Lévy Flights, it would slow down the convergence and lead to poor convergence accuracy and prematurity though it

makes it easy to skip local optimal solutions. This paper introduces leapfrog local search and chaos theory into CSA, and proposes a hybrid cuckoo search algorithm (HCSA). Below is a description of the basic CSA.

According to cuckoo's host hatching behaviour, the cuckoo search algorithm needs to assume some basic rules: each cuckoo produces only one egg at a time and is randomly placed in a nest; in a randomly selected nest, the best nest will be retained for the next generation; there is a probability P_α that each nest will be abandoned and a new nest will be re-established, where, $P_\alpha \in [0,1]$, which is usually set as $P_\alpha = 0.25$.

According to the above basic rules, the cuckoo selects the nest and lays eggs based on the Lévy Flights pattern. The cuckoo's nest selection route and location update formula is expressed as follows:

$$x_i^{t+1} = x_i^t + \alpha \oplus L(\lambda),\ i = 1,2,\ldots,n \tag{16}$$

where, x_i^{t+1} stands for the location of the i-th nest in the generation $t+1$; $\oplus$ refers to point-to-point multiplication; x_i^t represents the location of the i-th nest in the generation t; $L(\lambda)$ stands for the random search route of Lévy Flights; α is the controlled quantity of constant step size, which is usually set as $\alpha = 0.01$. As $L(\lambda)$ is a Lévy distribution function, for the convenience of application, Yang simplies $L(\lambda)$ and obtains a probability density function (Nanda *et al*., 2014) through Fourier transform:

$$Levy\ u = t^{-\lambda} (1 < \lambda < 3) \tag{17}$$

where, λ is exponential coefficient. In algorithmic programming, this paper uses the hop route formula that simulates Lévy Flights proposed by Mantegna (Nasa *et al*., 2013):

$$s = \frac{\mu}{|v|^{1/\beta}} \tag{18}$$

Reference (Li *et al*., 2015) proved that the Mantegna algorithm can be used to achieve equivalent calculation. In formula (18), s is the Lévy Flight hop route $L(\lambda)$, the relation between the parameter β and λ in Formula (17) is $\lambda = \beta + 1$, $0 < \beta < 2$, and usually in CSA, $\beta = 1.5$, μ and v are random numbers obeying the following normal distribution:

$$\mu \sim N(0, \sigma_\mu^2) \qquad v \sim N(0, \sigma_v^2) \tag{19}$$

where, the standard deviations σ_μ and σ_μ corresponding to μ and v are:

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta)\cdot \sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\cdot \beta \cdot 2^{(\beta-1)/2}} \right\}^{1/\beta} \qquad \sigma_v = 1 \tag{20}$$

Γ is the standard Gamma function.

According to Reference (Pinto *et al*., 2015), a good initial population can greatly improve the global optimization effect of the algorithm. The initial population in the classical CSA is randomly generated, resulting in poor population diversity and affecting the efficiency of the algorithm iteration. Chaos is a kind of aperiodic motion, which can traverse all conditions without repetition within a certain range. Therefore, it is better to use chaotic search than random search. This paper uses Logistic chaotic mapping to initialize the solution of CSA. Logistic mapping is derived from the demographic statistics in the dynamical system. The system equation given in (Yang *et al*., 2015) is as follows:

$$x(t+1) = \varpi x(t)(1 - x(t)) \tag{21}$$

where, t is the number of iterations, $x(t) \in [0,1]$ and ϖ is the control parameter. The linear transformation formula $u^{(k)} = a + (b-a)x^{(t)}$ is used to map the values of chaotic variables to the domain of definition of the optimization problem.

The shuffled frog leaping algorithm has strong local search ability and thus can well make up for CSA's shortcoming. Its basic idea is that n frogs are divided into m groups, with p ones in each group and that different groups represent frog sets with different information. Frogs in a group conduct deep local search and internal information exchanges within the solution space in accordance with the following meta-evolution strategy.

$$Step = rand * (P_{bi} - P_{wi}) \tag{22}$$

$$x_w = P_{wi} + Step \qquad -S_{\max} < Step < S_{\max} \tag{23}$$

where, *rand* indicates a randomly generated number within (0,1) ; $S_{\max}$ represents the maximum step size the frogs are allowed to hop at; P_{wi} represents the worst frog in the i-th group and P_{bi} the best frog in the i-th group.

In order for CSA to have a good global search ability in the early iteration and a good expansion effect in the late iteration, so as to speed up the convergence of the algorithm, this paper introduces inertial weight into the location update formula (16) for cuckoo's nest selection and egg laying route:

$$x_i^{t+1}=\omega x_i^{(t)}+\alpha\oplus L(\lambda),\ i=1,2,...,n \tag{24}$$

Regarding the inertia weight, this paper uses a non-linear decreasing strategy provided in (Zhang *et al*., 2015), as shown in the following equation, where t indicates the current iteration number.

$$\omega=(2/t)^{0.3} \tag{25}$$

If the urban road network satisfies the FIFO condition, according to the two-stage model for dynamic path selection given in Sections 1, 2 and 3, based on the hybrid cuckoo search algorithm, this paper designs a solving algorithm for the two-stage objective optimization model. The two-stage objective optimization is a process of optimization from bottom to top. The core is the hybrid cuckoo search algorithm (HCSA-KSP) that solves K shortest paths, as shown in Figure 2:

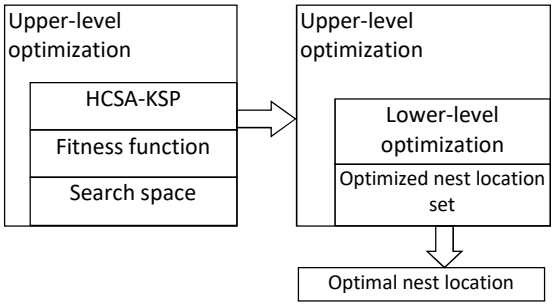

Figure 2. Principle of the two-stage HCSA

The idea of solving K shortest-time paths in a dynamic road network is to initialize the cuckoo population randomly in a chaotic way, with the location of each cuckoo corresponding to path information in the dynamic road network, and obtain the worst and best nest locations under the fitness function, then complete local iterations, remix all populations and build new cuckoo populations according to the principle of the shuffled frog leaping algorithm and let them evolve according to the "survival of the fittest" rule. As it is difficult for the algorithm to express the connectivity constraints on the sequence of nodes in the road network during the initial coding and evolution, it is necessary to redesign a coding scheme and local optimization strategy for solving the dynamic shortest path.

BP neural network model based on hybrid cuckoo search algorithm

BP network is a kind of multilayer feed forward neural network. The traditional BP neural network adopts the error back-propagation algorithm, that is to say, positive signal propagation and error back-propagation to adjust the weights and thresholds of each layer, so that the actual output of the network is closer to the expected output. However, the actual problem is often complicated multidimensional surface, which may lead to the algorithm fall into the local extreme point. Furthermore, the choice of initial threshold and weight values seriously affects the convergence speed.

The improved hybrid cucurbit search algorithm has a strong global optimization capability, which is combined with BP neural network, as the weight and threshold of the learning algorithm training network, which can overcome the shortcomings of the algorithm.

Determine the structure of the neural network and the dimensions of the search space, combine the backlog of weights and threshold as a cuckoo search algorithm to find the bird's nest coordinates, thus established the bird's nest coordinates, and the weights of neural network mapping relationship between the thresholds. Algorithm of the search target is to find the coordinates of one of the most suitable for bird's nest (i.e., a set of weights and thresholds) makes the selection of fitness function to get the minimum value, and then gives the coordinates of the bird's nest component to the BP neural network as the initial weights and thresholds of the network.

The purpose of network training is to adjust weights to minimize network and error, so choose the following fitness function:

$$S_{SSE}=\sum_{i=1}^{n}(\gamma'(i)-\gamma(i))^2 \tag{26}$$

Where n is the total number of samples, $\gamma'(i)$ is the actual output value of the i-th sample, and $\gamma(i)$ is the expected output value of the i-th sample.

Coding scheme and fitness function design

In the hybrid cuckoo search algorithm, each individual cuckoo is a carrier of the road network node information, represented by vector. Each vector corresponds to a feasible solution to the optimization problem. Coding is a mapping from the decision space of the optimization problem to the search space of the hybrid cuckoo search algorithm. The search space consists of combinations of all cuckoo location vectors and is a subset of the integer space. Therefore, it needs to be mapped to the integer space according to the characteristics of the decision variables. The decision vector of the dynamic shortest-path model is the path selection scheme $(r_s,...,r_i,r_{i+1},..,r_d)$ from the starting node of the vehicle r_s to the accident node r_d, where, $(r_i,r_{i+1})\in E$, r_i, $r_{i+1}\in R\subset R_+$, and R_+ is a positive integer set. With the help of the integer coding scheme, the path is coded into $X=(r_s,...,r_i,r_{i+1},..,r_d)$, and then for an urban road network containing M nodes, there are M^d coding schemes. In this case, the search space of the algorithm is much larger than the constraint space of the dynamic path problem and the algorithm is easy to fall into the local optimum, making it hard to obtain the optimal path. Therefore, it is necessary to add path connectivity constraints into the integer coding scheme and design a new coding scheme to reduce the search space of the algorithm. In the urban road network with 17 nodes shown in Figure 3, a feasible path for the emergency vehicle from the node r_{14} where it is located to the accident node is $r_{14}\to r_{15}-r_{11}\to r_9\to r_5\to r_6\to r_7$, and the corresponding cuckoo location code is Cuckoo: 14 15 11 9 5 6 7.

In order to ensure that the path sequence corresponding to the cuckoo location meets the connectivity requirement, a random integer coding scheme is proposed:

Let the set of adjacent nodes be $Nx_i\subset R_+$ and the current path sequence be $CP_i=\{r_s,r_1,...,r_i\}$. If there is a set $Er_i=Nx_i\cap\overline{CP_i}$, and a node is randomly selected as the effective node r_{i+1}, then there is

$$r_{i+1}=Er_i[INT(|Er_i|\cdot\varepsilon)] \tag{27}$$

where, ε is a random number and $0<\varepsilon<1$. The current path sequence is updated to $CP_{i+1}=\{r_s,r_1,...,r_i,r_{i+1}\}$, and the search is conducted step by step towards the accident node r_d, forming the cuckoo location code $X=(r_s,...,r_i,r_{i+1},..,r_d)$.

Fitness is a measure of the pros and cons of individuals, and the driving force behind the evolution of cuckoo populations. According to the stage 1 objective function of the dynamical optimal path model, the fitness function of the HCSA-KSP algorithm is defined as:

$$f(X)=-\sum_{i=s}^{d-1}T_{i,i+1}(t_i) \tag{28}$$

$$\text{where, } t_i=\begin{cases}t_{i-1}+T_{i-1,i}(t_{i-1}) & i=s+1,...d-1\\ t_0 & i=s\end{cases} \tag{29}$$

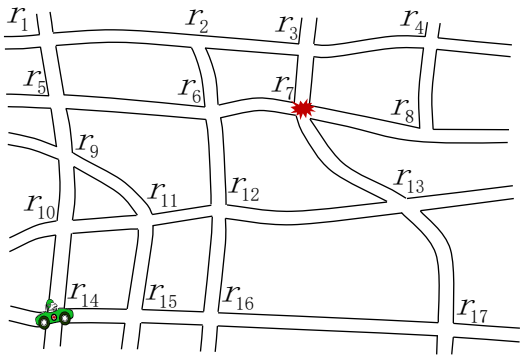

Figure 3. Urban road network model

Local optimization strategy

For the urban road network that satisfies the FIFO conditions, it is necessary to ensure that every road section that the vehicle passes through satisfies the shortest-time path requirement, that is, the local optimization process must be reasonable. In other words, if a shortest-time path $\prod_{j(t_s)}$ passes r_q, it must be the shortest-time path from the source node r_s to the node r_q.

In fact, the sub-path of each shortest-time path in the FIFO network is also the shortest-time path. Hence, the local optimization strategy of the cuckoo search algorithm is as follows: first mark the common node $SN=\{r_1',...,r_k',...,r_y'\}\subset P_{best}\cap P_{worst}$ in the optimal cuckoo location

$P_{best}=(r_s,...,r_i^b,r_{i+1}^b,...,r_d)$ and the worst cuckoo location $P_{worst}=(r_s,...,r_i^w,r_{i+1}^w,...,r_d)$. Then from the first coincidence node $r_1^{'}$, compare the travel time T_{sk}^w and T_{sk}^b of the paths $(r_s,...,r_i^w,...,r_k^{'})$ and $(r_s,...,r_i^b,...,r_k^{'})$. If $T_{sk}^w>T_{sk}^b$, then for P_{worst}, there is a shorter-time path from the source node r_s to $r_k^{'}$, and mark the node $r_k^{'}$ as the bottleneck node; otherwise, let $k=k+1$, and compare T_{sk}^w and T_{sk}^b, until the bottleneck node is marked. Finally, a random code scheme is used to generate a path from the node $r_{k-1}^{'}$ (flag $Wn_k^{'}=1$) to the bottleneck node to replace the corresponding path in the original P_{worst}, thus completing the update of the worst cuckoo location.

Algorithm design

With K optimal paths as the basis and the path reliability as the optimization objective, the two-stage hybrid cuckoo search algorithm is designed to finally obtain the optimal path. The steps of the HCSA-KSP algorithm are as follows and the process flow of the algorithm is shown in Figure 4.

Step 1: the initial population size is n, the search dimension d, the maximum number of iterations T, the discovering probability P_a, the range of search space $[Lb,Ub]$, the number of updates in frog leap search group Ne, the number of groups divided m, the number of birds in each group p, and the iteration counter $t=1$;

Step 2: perform chaos initialization of population individuals: according to the random coding scheme, generate F populations n consisting of cuckoo individuals in the decision space, generate n-dimensional vectors, and ensure that each cuckoo code $X_f=(r_s,...,r_i^f,r_{i+1}^f,...,r_d)$ is a path from the starting node r_s of the emergency vehicle to the accident node r_d, where, $(r_i,r_{i+1})\in E$, r_i, $r_{i+1}\in R\subset R_+$, and R_+ is a collection of positive integers. Update each dimension of X_f using (21) to generate chaos variables, and then use the carrier formula $X^{(n)}=Lb+(Ub-Lb)x^{(n)}$ to map them to the solution space;

Step3: according to (11) and (12), the fitness value $f(X_f)$ corresponding to the locations of the F nests, and sort all individual nests in descending order according to their fitness values to determine the current optimal nest location and its fitness value;

Step 4: divide the n nests into m groups, with p nests in each group. First, determine the best solution $P_{bi}=(r_s,...,r_i^b,r_{i+1}^b,...,r_d)$ and the worst solution $P_{wi}=(r_s,...,r_i^w,r_{i+1}^w,...,r_d)$ in the group, and update them according to (22) and (23). If the worst nest location is improved, that is, $P_{wi}^{'}>P_{wi}$, then reserve $P_{wi}^{'}$; otherwise randomly generate a nest location in the group to replace the worst location. Then remix the groups and sort them in descending order to construct a new cuckoo population, and calculate the fitness value corresponding to the new nest location, and retain the optimal location until the number of local searches is reached;

Step 5: update the new nest locations according to (16), (17), (18), (19), (20), (24) and (25) to obtain a new group of nest locations;

Step 6: randomly generate a uniformly distributed number r $(0\le r\le 1)$, and if $r>P_a$, randomly change the nest location with a high discovering probability and keep the location with a low discovering probability so as to obtain a new group of nest locations;

Step 7: according to (9), calculate the reliability of the shortest path $\prod_{i=s}^{d}(r_i^k,r_{i+1}^k)$, and choose the one with the highest reliability as the optimal solution to the model;

Step 8: evaluate the fitness of each nest location and then update the historical optimal location;

Step 9: if the algorithm meets the termination conditions, then the coordinate component of the best nest is used as the initial weight and threshold of BP neural network, and is trained to obtain the best model, and then output the result; otherwise, return to Step 3 to proceed.

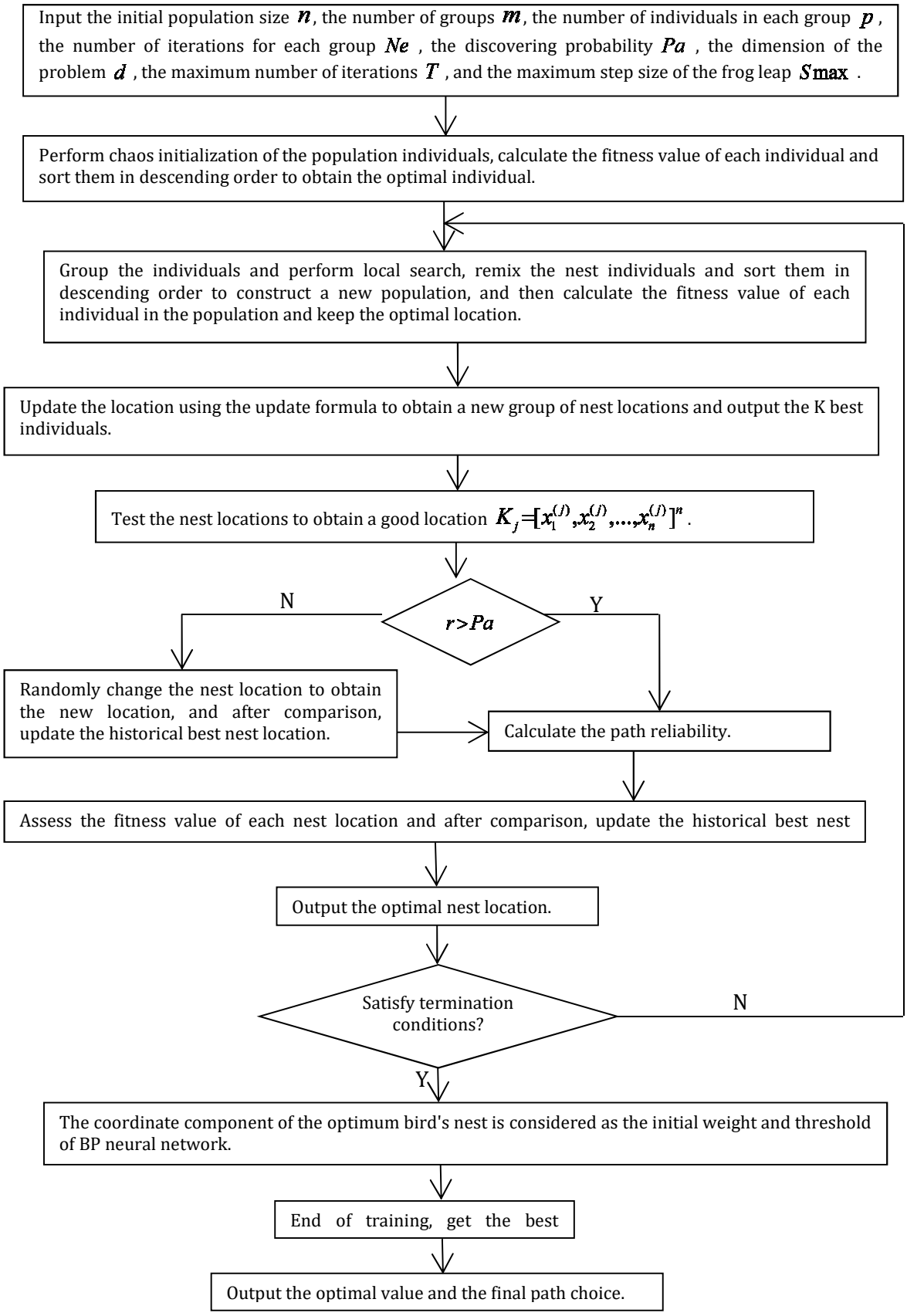

Figure 4. Process of HCSA-KSP

Example verification

By taking the urban road network in some sections of Yangpu District, Shanghai (Jungong Road - Songhuajiang Road - Elevated Inner Ring Road - Zhoujiazui Road) as an example, selecting 8:00-9:00 on December 29, 2017 as the time period and setting the shortest time interval at Δt=5min , this paper constructs a real-time road section travel speed function, and obtains the road section travel time function. The example is shown in Figure 5. 10 pairs of nodes are randomly selected. The start point is the area of departure and the end point is the accident scene. In order to simulate the situation of a city emergency, the morning rush hour during a normal working day is selected as the scheduling period after the accident occurs. Considering the complex dynamic road network changes, the HCSA-KSP algorithm is used to solve the K shortest paths for the emergency vehicle from the start point to the end point and the travel time. The results are compared with those of the PSO algorithm and the classic CSA to verify the performance of the HCSA-KSP algorithm. Parameters of the HCSA-KSP algorithm are shown in Table 1.

Table 1. Parameter selection of HCSA-KSP

n	*Pa*	*T*	α	*m*	*Ne*	*S*max	*P*
100	0.25	100	0.01	10	10	50	10

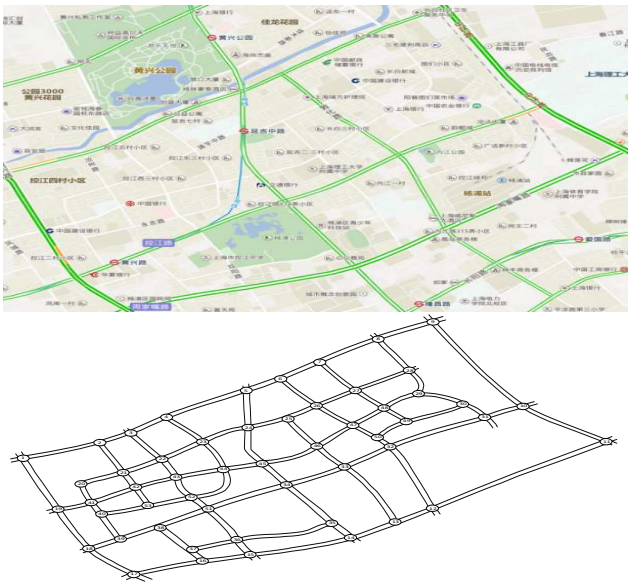

Figure 5. Map and road network map of some areas in Yangpu District, Shanghai

Where, α is the constant step control quantity. In order to obtain the K shortest paths and travel time between 10 nodes, the HCSA-KSP algorithm, PSO algorithm and classic CSA are operated and the calculated results obtained are listed in Table 2.

Through analysis of the shortest path selection results in Table 2, it can be seen that HCSA-KSP, PSO-KSP and CSA-KSP algorithms can all obtain the K shortest paths between nodes and the corresponding travel time. From node r_1 to node r_{43}, the K shortest paths obtained by HCSA-KSP and CSA-KSP are 1, 2, 21, 42 and 43, with a length of 1.32km, and the shortest travel time is 4.036, the K shortest paths obtained by PSO-KSP are 1, 2, 3, 22, and 43, with a length of 1.43 km, and the shortest travel time is 4.331. From this, we can see that the K shortest path scheme obtained by HCSA-KSP and CSA-KSP is better. From node r_{13} to node r_{24}, HCSA-KSP, PSO-KSP and CSA-KSP all obtain the same K shortest paths (13, 33, 46, 25, 24) and the same shortest travel time, indicating that the three schemes have the same performance. However, from the path choices of other node pairs, it can be seen that the K shortest paths obtained by the HCSA-KSP algorithm have the shortest travel time, which is the best result.

Take the path optimization from node r_1 to r_{43} and node r_{13} to r_{24} for example. The convergence processes of HCSA-KSP, PSO-KSP and CSA-KSP are shown in Figure 6.

As can be seen from Figure6, the travel time from node r_1 to r_{43} obtain by the HCSA-KSP algorithm reaches the global convergence after about 20 iterations while PSO-KSP converges after about 50 iterations and CSA -KSP converges after around 70 iterations. HCSA-KSP and CSA-KSP need shorter travel time than PSO-KSP algorithm. From node r_{13} to r_{24}, HCSA-KSP also converges faster than PSO-KSP and CSA-KSP, but all three obtain the same travel time.

Generally speaking, between the same start point and end point, as different algorithms obtain different optimal paths, the shortest path travel time is different and the computation time is also different even at the same average vehicle speed. Comparatively, the HCSA-KSP algorithm obtains better shortest path travel time, takes less computation time and converges faster to the optimal solution, indicating that the HCSA-KSP algorithm is a feasible method to solve the dynamic path selection problem of emergency vehicles.

Table 2. Results of the K shortest path selection

Start point	End point	Algorithm	K shortest paths	Path length (km)	Average Speed (km/h)	Shortest travel time(min)	Computation time (s)
	r_{43}	HCSA-KSP	1,2,21,42,43	1.32	19.62	4.036	0.4101
		PSO-KSP	1,2,3,22,43	1.43	19.81	4.331	1.3322
		CSA-KSP	1,2,21,42,43	1.32	19.62	4.036	1.0601
r_{13}	r_{24}	HCSA-KSP	13,33,46,25,24	2.11	20.02	6.324	0.4852
		PSO-KSP	13,33,46,25,24	2.11	20.02	6.324	1.5101
		CSA-KSP	13,33,46,25,24	2.11	20.02	6.324	1.1273
r_{11}	r_{48}	HCSA-KSP	11,10,31,30,29,48	2.05	19.98	6.156	0.5101
		PSO-KSP	11,12,32,50,49,48	2.53	19.81	7.663	1.9854
		CSA-KSP	11,10,31,30,49,48	2.07	20.11	6.176	1.3233
r_{9}	r_{26}	HCSA-KSP	9,8,7,27,26	1.42	19.73	4.318	0.4122
		PSO-KSP	9,8,28,27,26	1.43	19.72	4.351	0.9881
		CSA-KSP	9,8,28,27,26	1.43	19.72	4.351	0.8917
r_{12}	r_{25}	HCSA-KSP	12,13,33,46,25	2.08	19.73	6.325	0.4799
		PSO-KSP	12,32,33,46,25	2.12	19.74	6.444	1.5301
		CSA-KSP	12,13,33,46,25	2.08	19.73	6.325	1.1198
r_{5}	r_{43}	HCSA-KSP	5,24,23,22,43	1.34	19.82	4.057	0.4271
		PSO-KSP	5,24,23,44,43	1.37	19.84	4.143	1.3505
		CSA-KSP	5,4,23,22,43	1.36	19.87	4.107	1.3101
r_{3}	r_{34}	HCSA-KSP	3,22,43,52,51,34	1.57	20.05	4.698	0.5012
		PSO-KSP	3,22,23,24,45,34	1.61	20.01	4.828	1.2347
		CSA-KSP	3,22,43,44,45,34	1.58	20.02	4.735	1.2255
r_{39}	r_{22}	HCSA-KSP	39,38,51,52,43,22	1.34	19.65	4.092	0.4202
		PSO-KSP	39,40,41,20,21,22	1.37	19.62	4.190	1.3441
		CSA-KSP	39,40,41,42,43,22	1.35	19.62	4.128	1.2554
r_{36}	r_{23}	HCSA-KSP	36,51,52,43,22,23	1.47	19.73	4.470	0.4240
		PSO-KSP	36,51,52,43,44,23	1.52	19.73	4.622	1.4122
		CSA-KSP	36,51,52,44,23	1.49	19.72	4.533	1.0577
r_{25}	r_{29}	HCSA-KSP	25,26,47,48,29	1.28	18.95	4.053	0.4201
		PSO-KSP	25,46,47,48,29	1.34	18.93	4.247	1.3521
		CSA-KSP	25,26,27,48,29	1.30	18.94	4.118	1.0922

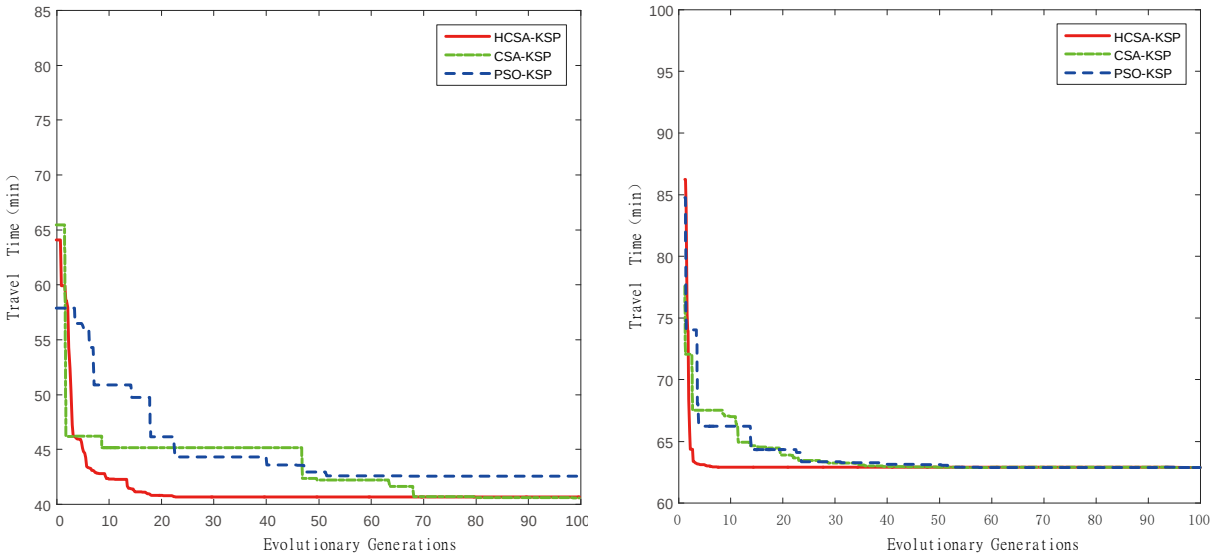

Figure 6. Evolutionary processes of the three algorithms in path optimization of r_1 to r_{43} and r_{13} to r_{24}

Conclusions

Whenever an emergency occurs in a city, in order to get the injured treated as soon as possible, the emergency vehicles must be scheduled in such a way as to minimise the rescue time. However, the predetermined rescue paths often cannot adapt to the complex and changeable traffic conditions of the urban road network; therefore, a dynamic path selection scheme is needed to schedule these emergency vehicles. This paper takes the vehicle passing time as the weight of the road section to solve the shortest-time path selection

problem. It abstracts the road network into nodes and edges, builds the road section travel time function, uses the path reliability to measure the negative impacts of dynamic traffic conditions on the whole travel time, design is a two-stage objective optimization model for dynamic optimal path selection and at last presents a BP neural network model optimized by hybrid cuckoo algorithm to solve this problem. The simulation test proves that this algorithm is a feasible method to solve the dynamic path selection problem of emergency vehicles, and through comparison test, it can be found that this algorithm has better performance than the classic cuckoo search algorithm and the particle swarm optimization algorithm.

References

Duan X, Song S, Zhao J. Emergency vehicle dispatching and redistribution in highway network based on bilevel programming. Mathematical Problems in Engineering 2015; 2015(1923): 1-12.

Feng LY, Yuan LW, Luo W. Geometric algebra-based algorithm for solving nodes constrained shortest path. Acta Electronica Sinica 2014; 42(5): 846-51.

Figliozzi MA. A route improvement algorithm for the vehicle routing problem with time dependent travel times//Proceeding of the 88th Transportation Research Board Annual Meeting CD ROM, 2009.

Fu Y, Zhu LJ, Han HG. Analysis of K Shortest paths algorithms and applications. Technology Intelligence Engineering 2015; 1(1): 112-19.

Guerrero M, Castillo O, García M. Cuckoo search via lévy flights and a comparison with genetic algorithms//Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics. Springer, Cham 2015: 91-103.

Kanagaraj G, Ponnambalam SG, Jawahar N. An effective hybrid cuckoo search and genetic algorithm for constrained engineering design optimization. Engineering Optimization2014; 46(10): 1331-51.

Li X, Yin M. Modified cuckoo search algorithm with self adaptive parameter method. Information Sciences 2015; 298: 80-97.

Nanda SJ, Panda G. A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm and Evolutionary Computation 2014; 16: 1-18.

Nasa-ngium P, Sunat K, Chiewchanwattana S. Enhancing modified cuckoo search by using Mantegna Lévy flights and chaotic sequences//Computer Science and Software Engineering (JCSSE), 2013 10th International Joint Conference on. IEEE 2013: 53-57.

Ozbay K, Iyigun C, Baykal-Gursoy M. Probabilistic programming models for traffic incident management operations planning. Annals of Operations Research 2013; 203(1): 389-406.

Pinto ARF, Crepaldi AF, Nagano MS. A Genetic Algorithm applied to pick sequencing for billing. Journal of Intelligent Manufacturing 2015: 1-18.

Ran B, Boyce D. Dynamic urban transportation network models: theory and implications for intelligent vehicle-highway systems. Springer Science & Business Media 2012.

Santillan JH, Tapucar S, Manliguez C. Cuckoo search via Lévy flights for the capacitated vehicle routing problem. Journal of Industrial Engineering International 2017:1-12.

Sharma H, Bansal J C, Arya KV. Lévy flight artificial bee colony algorithm. International Journal of Systems Science2016; 47(11): 2652-70.

Xu SH, Han CF, Meng LP. Research on adaptive of emergency management organization system based on NK model. Systems Engineering-Theory & Practice 2017;37(6): 1619-29.

Xu T, Ding XL, Li JF. Review on K shortest paths algorithms. Computer engineering and design 2013; 34(11): 3900-06.

Yang JM, Ma MM, Che HJ. Multi-objective adaptive chaotic particle swarm optimization algorithm. Control and Decision 2015; 30(12): 2168-74.

Yang XF Stochastic scenario-based two-stage model and algorithm for the least expected time shortest path. Beijing Jiaotong University, 2013.

Zhang J, Zheng J. Capacitated vehicle routing problem using a novel hybrid swarm optimization approach. Academic Journal of Manufacturing Engineering 2016;14(1):90-95.

Zhang L, Tang Y, Hua C. A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques. Applied Soft Computing 2015;28: 138-49.

Zhao HY, Wang JD, Liu SL. Compound Fault Diagnosis Technique Based on Artificial Neural Network and Support Vector Machine. Fluid Machinery 2008;36(1): 39-63.