Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 2304097, 15 pages
https://doi.org/10.1155/2022/2304097

Research Article

Hyperpath Searching Algorithm considering Delay at Intersection and Its Application in CVIS for Vehicle Navigation

Ziyan Ju and Muqing Du

College of Civil and Transportation Engineering, Hohai University, Nanjing, Jiangsu 210098, China

Correspondence should be addressed to Muqing Du; dumqhhu@hhu.edu.cn

Received 23 August 2021; Revised 25 January 2022; Accepted 14 February 2022; Published 9 March 2022

Academic Editor: Peng Hang

Copyright © 2022 Ziyan Ju and Muqing Du. 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.

The periodic change of intersection signals in urban road systems is the leading cause of uncertain delays. Therefore, aiming to minimize the travel time on the road segments and the expected delay at the intersections, a hyperpath search method based on intersection signal timing is proposed. The expanded network representation is used to capture the different turning delays at intersections. According to the intersection signal timing, the maximum waiting time of each turn and the turning movement ratio is obtained. A "recommendation priority passable phase" guidance strategy is proposed for the cooperative vehicle-infrastructure system (CVIS) based on the optimal hyperpath. The simulation experiments show that the vehicles can shift the driving route in the hyperpath set according to the guidance strategy, which effectively reduces the actual delay at intersections, and further realize an optimized distribution of traffic flows in the road network.

1. Introduction

The intelligent transportation system, along with the driverless technology and the cooperative vehicle-infrastructure system (CVIS), has played a pivotal role in improving efficiency and reducing travel delays on the road network [1]. The route search algorithm is the key technology that makes for vehicle route guidance and traffic distribution optimization. Expressed more formally, in connection with the need to realize reliable route guidance, the intelligent transportation system mostly uses the Dijkstra algorithm and its improved algorithms (such as the A^* algorithm considered heuristic information) to solve the shortest path problem [2]. However, the simple-path algorithm cannot consider the several uncertain resources in the transportation networks such as traffic congestion, weather conditions, and vehicle accidents. In this way, traffic is easily guided to the same route, leading to overload along the way, then reducing the navigation system's practicability and credibility [3].

Lee [4] found that the test results indicated that the multiple path routing strategy performed better than the commonly used shortest path routing strategy using traffic simulation. Hence, shortest path methods have been extended to generate alternative routes, perhaps to avoid sites of congestion or specified areas.

Chen et al. [5] reviewed methods for generating multiple paths in the context of route guidance. Traditionally, alternati2ve paths could be calculated by two categories of the algorithm in graph theory, namely, (1) the k-shortest path (KSP) algorithms proposed by Eppstein [6], Martins [7], and Jimenez and Marzal [8] and (2) the totally disjoint path algorithms proposed by Dinic [9] and Torrieri [10].

However, these algorithms have some drawbacks for route guidance. For the k-shortest path algorithms, the searched k edges have considerable overlap, reducing choice diversity. For the disjoint path algorithms, the primary shortest path of the network may not be included, or the length of alternative paths may be unacceptable.

Intersections are the bottleneck in urban traffic networks, and the drivers' turning movement at signalized intersections can impact the traffic distribution on the road network. It may often be observed, apparently indeed in the great majority of experienced drivers, that they will change lanes early to cross the intersection instead of waiting. Based on the above, all paths, which can reduce the drivers' waiting time at signalized intersections, are defined as the hyperpath set on the road networks. Our numerical results show that the hyperpath algorithm performs better in reducing delays, rather than both the shortest path algorithms and the multiple path algorithms.

At last, this article applies the hyperpath algorithm to vehicle navigation with the CVIS. Under the "recommend priority passable phase" rule, the cooperative vehicle-infrastructure system is introduced to improve the vehicle guidance system based on the hyperpath algorithm. Under the information interaction, the driver can change the lane in advance by voice, text, picture, and video so that the driver can shift their routes in the hyperpath subset. In fact, the traffic distribution will be optimized, reducing the total time and improving the network performance.

The outline of this article is given as follows: Section 2 reviews the literature on multiple path algorithms and discusses the shortcomings of the existing path planning algorithms. Then, we describe the origin of the hyperpath from the transit assignment. Section 3 develops the hyperpath algorithm incorporated with the signal timing and then gives the constraints and the proofs. In Section 4, the hyperpath algorithm is tested on randomly generated road networks to demonstrate the advantages of the proposed algorithm. Section 5 introduces the CVIs to realize the vehicle route guidance. Section 6 presents the conclusions thus far and suggestions for future work.

2. Literature Review

The k-shortest path (KSP) problems involve finding the shortest path, the second shortest path, and so on to the kth shortest path between a given origin and a destination (O–D) pair [11]. KSPs are usually provided in route guidance systems to satisfy various preferences that different users have for path choices [12]. Chen et al. introduced the A^* technique to improve KSPs finding performance in stochastic networks under travel time uncertainty [13]. Some scholars proposed the improved KSPs that can find different shortest paths with a reasonable degree of similarity and close travel time, and the numerical result is satisfactory [14–16]. Recently, an efficient deviation path algorithm has been proposed for finding exactly the k-shortest simple paths without loops in road networks, which performs significantly better than the state-of-the-art algorithms [17]. A considerable amount of research has been devoted to developing KSP algorithms with better performance on the algorithm speed and ability to consider uncertainties. However, existing improved KSP algorithms do not change the nature, so that the k-shortest simple paths overlap each other and lack accessibility. The k-shortest simple paths have different optimal targets, which cannot be achieved when the drivers shift their routes.

The total disjoint path algorithm offers a major reduction in computation time for large networks. [10]. Here, "disjointness" can be considered in terms of either nodes or links [18]. A link-disjoint path-pair is a pair of paths between these nodes with no common links, but they still might share common nodes [19]. Accordingly, a node-disjoint path-pair is a pair of link-disjoint paths between these nodes with no common nodes [20]. Recently, scholars have taken into account the total disjoint path algorithm's vital feature to provide reliable service on the network, which guarantees to survive any single link failure [21]. As referred to earlier, the disjoint paths may not be shortest, but they are more suitable to uncertain traffic conditions.

Conventionally, path selection in the road network in the time-dependent vehicle routing problem is denoted as flexibility [22–24]. However, not all alternative paths have flexibility, but the ones with the same constraint conditions can have it. For instance, the k-shortest paths with different objective functions lack flexibility and are the same as the pair of the disjoint paths. The drivers cannot avoid the uncertain delay via shifting their routes among the results of multipaths. Consequently, there is an urgent need to realize path flexibility, which can reduce delays at intersections.

Indeed, empirical studies have shown that travel time variations significantly influence travelers' route-choice behavior [25]. Unlike previous algorithms, the driver can flexibly shift their routes to avoid delay in the hyperpath set. So the hyperpath algorithm has accessibility. Spiess and Florian [26] employed the hyperpath to solve passenger flow distribution on bus lines. In the public transportation model, the hyperpath that includes all the bus lines set between the origin and destination is interpreted as an optimal riding strategy. Bell [27] further considered the uncertainties of travel time on each link in the road network and applied Spiess and Florian's transit hyperpath to the road network. Bell pointed out that the hyperpath search algorithm was a special multipath algorithm that takes the minimum expected travel time as the goal. The hyperpath algorithm searches all paths that will be possible to become the shortest one between the origin and destination and incorporates them into the hyperpath subnet. Based on Bell's research, Ma et al. [28] proposed an improved method to speed up the hyperpath algorithm. However, the current hyperpath algorithms do not consider the impact of intersection delays on the expected travel time.

Moreover, we consider the delays at signalized intersections, which significantly contribute to the uncertainty of travel times, particularly in urban transportation networks [29–32]. In contrast, existing hyperpath studies about the road network have rarely addressed delays at signalized intersections due to the fact that signalized intersections can largely increase the complexity of the algorithm.

We aim to develop an efficient hyperpath searching algorithm typically applied to signalized road networks. The proposed algorithm conquers the inaccessibility of the most existing multiple path algorithms (e.g., KSP). Table 1 summarizes the differences of our proposed algorithm from the relevant studies. The contributions of this study are presented as follows:

(i) We propose a hyperpath-based vehicle routing method to realize the accessibility among the multiple paths, reducing the delay at intersections, which is more efficient and performs better than the existing multiple path algorithms

(ii) We propose a hyperpath algorithm that considers the uncertain delay at signalized intersections in the road networks

(iii) We introduce the CVIS to realize vehicle navigation under the optimal hyperpath

3. Models

3.1. Hyperpath-Based Choice Strategy. The concept of the hyperpath, i.e., a set of paths any one of which may be optimal, comes from the field of transit assignment and is associated with the common-line issue. We assume that the passenger arrives at bus stop A, whose destination is bus stop D. Assume that three attractive bus lines given the headways arrive randomly in Figure 1. The passenger taking the bus line depends on which one happens to arrive next. The lines that are attractive at a given stop hence constitute the common lines, namely, hyperpath. Under the "take whichever attractive line arrives next" rule, the hyperpath may be found by minimizing the expected travel time and that the resulting problem is a linear program.

(i) We assume that there is only bus line 1 from stop A to stop D, with a headway of 6 minutes. Hence, the expected waiting time of passengers is $T = 6\,\text{min}$.

(ii) We assume that three bus lines are attractive, which is referred to earlier. It will become evident that the expected waiting time of passengers is $T = 1/(1/6 + 1/3) = 2\,\text{min}$.

Since the applicability of the transit assignment model is satisfying, some researchers have introduced this concept into the road network.

Bell noted the parallel between link frequency and link delay and then extended the Spiess and Florian algorithm by adding node potentials into the link selection step, yielding an algorithm that resembles the Astar algorithm, but which generates a hyperpath.

Consider a road network with eight links and seven nodes, shown in Figure 2. In Bell's paper [27], each link delay is equal to a random number R. For simplicity, the delay at the intersection is not considered. In this case, it can be seen in Figure 2, as all the links are unreliable, the hyperpath contains two paths, and whichever path of the set of hyperpath will become the shortest one.

Since the delay at signalized intersections is the main component of travel time, we significantly emphasize its influence on the performance of the hyperpath algorithm.

This study improves the original hyperpath search algorithm by expanding the signalized intersection. Aiming to minimize the travel time on the road segments and the expected delay at the intersections, we get a hyperpath set between the origin and destination.

We discuss the relationship of the turning movement, signal timing, and delay, as shown in Figure 3. Take one entrance lane as an example, where it consists of a left-turn movement and a through movement, corresponding to different signal phases. Assuming that the delay at the intersection is entirely controlled by the signal timing, the turning delay varies with the turning movement. In this way, this article refers to the signal timing delay at intersections to the waiting time in the transit network and then proposes the equation of the delay at intersections in the road network.

The original hyperpath algorithm only considers uncertain delays on the road segment, which is the delays of leaving their upstream nodes. The turning delay will depend on the turning movement that vehicle passes through the intersection. As shown in Figure 4(a), the delays caused by signal timing vary in different entrances and exits. This article expands the intersection with signal control in the road network so that the hyperpath algorithm overcomes the original shortcomings that cannot capture the turning delay at intersections. Each turn is described as a virtual link. Link weight is used to represent the turning delay at intersections. Take Figure 4 as an example, and there are four types of turns and four virtual links at one-way intersections; for the two-way, there are twelve types of turns and twelve virtual links.

This approach is to expand each intersection in the network. Thus, taking the one-way intersection as an example, the label rule is shown in Figure 5. The node j is expanded to four nodes. The travel time between nodes i and j is c_{ij}, and the travel time on the virtual link is the turning delay. d_{ijk} denotes the turning delay from road segment (i, j) to road segment (j, k).

3.2. Hyperpath Model. In the extended road network, the optimal hyperpath problem is described as a mathematical optimization model that takes the minimum travel time on the road segment and the expected delay at intersections as the objective function.

Define the following sets and variables:

$G(V, A)$: A directed graph

V: A set of vertices

A: A set of edges

I: Node

J: Downstream node of the node i

K: Downstream node of the node j

L: Set of links

H: Set of links of the hyperpath

$\Gamma^{+}(i)$: The set of edges leaving the node i

$\Gamma^{-}(i)$: The set of edges entering the node i

Table 1: The most relevant existing studies on hyperpath.

Literature	Path flexibility	Travel time reliability	Delays at intersections
Chen et al., 2016	×	✓	✓
Shen et al., 2020	×	✓	✓
Bell, 2008	✓	✓	×
Ma et al., 2013	✓	✓	×
This article	✓	✓	✓

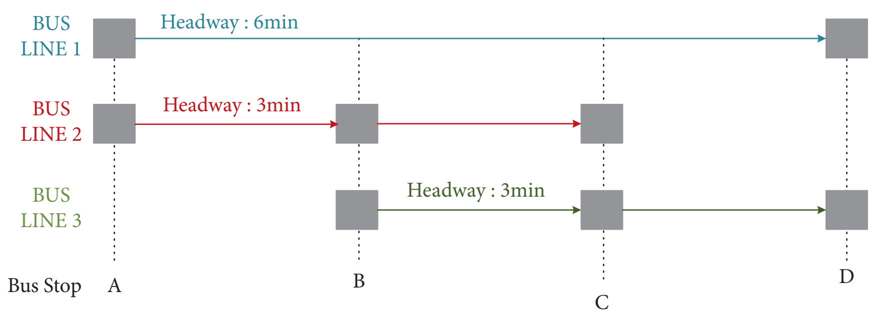

Figure 1: The hyperpath for the transit assignment model.

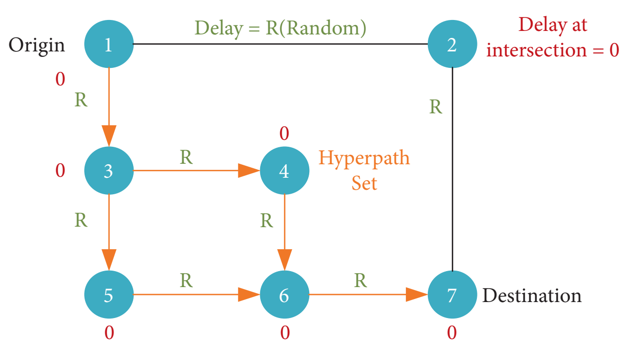

Figure 2: Hyperpath for unreliable links in the road network.

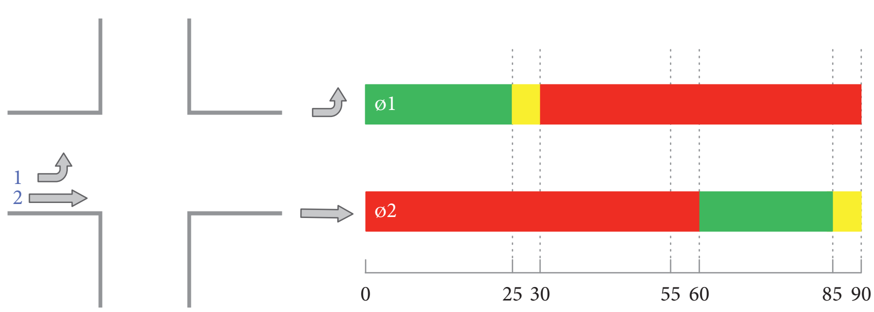

Figure 3: The turn delay varies with the turning movement.

u_i: Minimum travel time from node i to the destination

y_i: Probability of node i being selected

d_{ijk}: Turning delay from node i through node j to node k

c_{ij}: Travel time of on (i, j)

w_j: Expected delay time at node j

p_{ij}: Probability of (i, j) being selected

f_{ijk}: Service frequency of turning movement d_{ijk}, is $1/d_{ijk}$

f_j: Combined service frequency of node j

The hyperpath is identified by the following linear program:

$$\min_{p,w} \sum_{(i,j)\in A} c_{ij} p_{ij} + \sum_{i\in V} w_i, \tag{1}$$

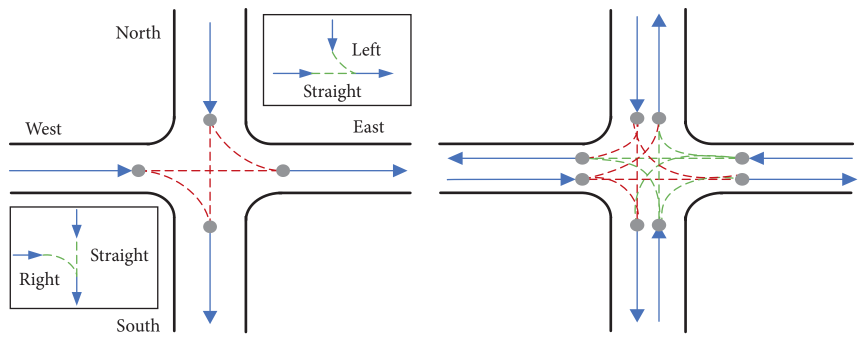

Figure 4: The expanded road at intersections. (a) The one-way intersections. (b) The two-way intersections.

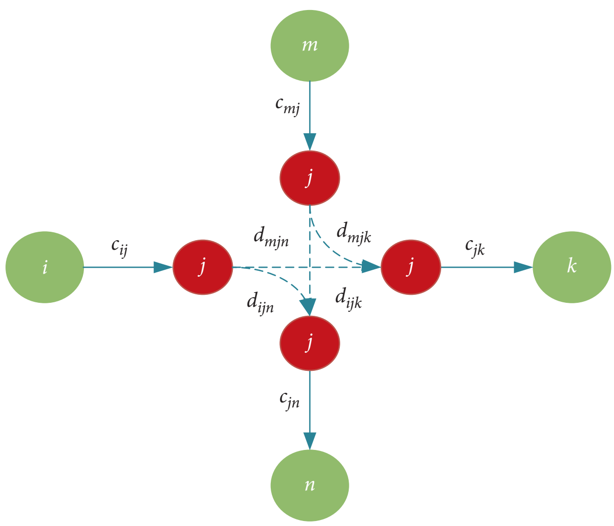

Figure 5: The label at the expanded network.

s.t.

$$\sum_{(j,k)\in A_j^+} p_{jk} - \sum_{(i,j)\in A_j^-} p_{ij} = g_j, \forall j \in V, \tag{2}$$

$$p_{ij} \in [0,1], \tag{3}$$

$$d_{ijk} \geq c_{jk} \cdot p_{jk}, \forall (j,k) \in A_j^+, i \in \Gamma^-(j). \tag{4}$$

In (1), the expected travel time is shown. Note that expected delay is interpreted as what would be expected by a pessimistic driver, namely, its exposure to maximum link delay. The travel time c_{ij} is a state of the free flow; i.e., the travel time does not change with increased traffic.

In (2), g_i is the origin and destination identification. When the node is the starting point r, $g_i = 1$. When the node is the endpoint s, $g_i = -1$; otherwise, $g_i = 0$. During the trip, there can be detours, but it is not allowed to arrive at a place twice. In (2), for the origin, its value is 1, which means the path starts from the origin. For the destination, the value is -1, which means that the trip stops at this point and ends the trip. Otherwise, its value is 0, which means that the path passes through the node with a certain probability. That is, it leaves after reaching the node.

In (3), the choice probability ranges from 0 to 1. The maximum probability of the link selected is 1. If not, the probability is 0.

Equation (4) indicates that the vehicle chooses to detour instead of waiting at the current intersection because the current turning delay is greater than the product of the travel time and the probability of being selected.

For the extended network, the virtual road segment corresponds to the turning movement at intersections, and the maximum delay d_{ijk} is caused by the periodic change of the signal light. This article assumes that the vehicle arrives randomly at the intersection and defines the intersection service frequency f_{r_j} ($f_{r_j} = 1/d_{r_j}$) of the turning r_j. In the hyperpath set, for node j, there may be multiple available turning r_j , and R_j (e.g., $R_j = \{d_{ijk}, d_{ijn}, d_{mjk}, d_{mjn}\}$) is the set of all available turns at this node. Further, we define the intersection service frequency $f_j = \sum_{r_j\in R_j} f_{r_j}$; then at the intersection, the expected waiting time w_j is as follows:

$$w_j = \alpha / \sum_{r_j \in R_j} f_{r_j}. \tag{5}$$

3.3. Optimal Hyperpath Algorithm

3.3.1. Initialization. Specify the origin r and the destination s; create a set L of links and add all links in the road network to the set L; create a set H of the hyperpath so that the set H is initially an empty set. Initialize the variables as follows:

$$\begin{aligned} u_s &= 0, u_i = \infty, \forall i \neq s, \\ y_r &= 1, y_i = 0, \forall i \neq r, \\ \forall i &\neq V, f_i = 0, d_{ijk} = 0, \\ \forall (i,j) &\in A, p_{ij} = 0, f_{ijk} = 0. \end{aligned} \tag{6}$$

3.3.2. Main Steps

Step 1: find the shortest link in the set L, take $(u_j + c_{ij})$ as the current link, and remove it from the set L.

Step 2: if the current link meets the conditions: $u_i \geq u_j + c_{ij}$, go to Step 3; otherwise, go back to Step 1.

Step 3: update the data.

If $i = r$, $f_{ijk} = 1$,

$$u_i = \frac{\beta + \left(u_j + c_{ij}\right) f_{ijk}}{f_i + f_{ijk}}, f_i = f_i + f_{ijk}. \quad (7)$$

If $u_i = \infty$ *or* $f_i = 0$,

$$\beta = 1; \quad (8)$$

otherwise,

$$\beta = u_i f_i. \quad (9)$$

Step 4: add the current link (i, j) to the set H. At this time, if the condition $L = \varnothing$ or $u_i + c_{ij} \geq u_r$ is satisfied, go to Step 5; otherwise, go back to Step 2.

Step 5: get the link and node choice probability: from largest to smallest, sort all links (i, j) in the network according to the value of $u_i + c_{ij}$ and traverse all the links according to the sorted order; if the link $(i, j) \in H$, access the database to obtain the probability of selected for each turn.

Starting from the origin r, trace back the hyperpath to the destination s and output it for route guidance strategy.

3.4. Algorithm Validation. The algorithm validation is as follows.

Proposition 1. *Assumption of maximum pessimistic expectation implies that* $c_{jk} \cdot p_{jk} = d_{ijk} > 0$ *if* $p_{jk} > 0$ *and* $c_{jk} > 0$ *for* $(j, k) \in A_j^+$ *and* $i \in \Gamma^-(j)$.

Proof 1. The Lagrangian function for equations (1)–(4)is as follows:

$$L_{p,w,\mu,\lambda} = \sum_{(i,j)\in A} c_{ij} p_{ij} + \sum_{i\in V} w_i + \sum_{i\in V} \mu_j \left(\sum_{(j,k)\in A_j^+} p_{jk} - \sum_{(i,j)\in A_j^-} p_{ij} - g_j \right) + \sum_{i\in\Gamma^-(j)} \sum_{(j,k)\in A_j^+} \lambda_i \left(c_{jk} \cdot p_{jk} - d_{ijk} \right), \quad (10)$$

where μ and λ are the Lagrange multipliers concerning $\mu \geq 0$ and $\lambda \geq 0$. The optimal solution of the Lagrange function is as follows:

$$\frac{\partial L_{p,w,\mu,\lambda}}{\partial w_i} = 1 - \sum_{i\in\Gamma^-(j)} \sum_{(j,k)\in A_j^+} \lambda_i \frac{\partial d_{ijk}}{\partial w_i} = 0. \quad (11)$$

According to (5), $\partial d_{ijk}/\partial w_i \leq 1$; then from (11), we can know that $1 - \sum_{i\in\Gamma^-(j)} \sum_{(j,k)\in A_j^+} \lambda_i \partial d_{ijk}/\partial w_i \leq 1 - \sum_{i\in\Gamma^-(j)} \lambda_i \Rightarrow \sum_{i\in\Gamma^-(j)} \lambda_i \leq 1$. At the solution, because w_i is equal to 0 or over. If λ_i equals 0, p_{jk} could be reduced to 0 at the solution, violating the assumption that $p_{jk} > 0$. So, by the complementary slackness conditions, $\lambda_i > 0$ implies that $c_{jk} \cdot p_{jk} = d_{ijk} > 0$ for $(j, k) \in A_j^+$ and $i \in \Gamma^-(j)$.

Proposition 2. *At the point at which (i, j) is selected,* u_j *has been reduced to its final value.*

Proof 2. See the Bell [27].

Proposition 3. *If, at the point at which* (i, j) *is selected,* $c_{ij} + u_j > u_r$, *the algorithm should terminate.*

Proof 3. See the Bell [27].

Proposition 4. *If* $d_{ijk} = 0$ *for all links* $(j, k) \in A_j^+$ *and all nodes* $i \in \Gamma^-(j)$ *then the Hyperstar algorithm finds only the path(s) with least undelayed travel time.*

Proof 4. See the Bell [27].

4. Illustrative Example

To demonstrate the Hyperstar algorithm, we use a grid-type transportation network, as shown in Figure 6, with node r as the origin and node s as the destination. All arcs can be traveled in both directions, leading to 62 directional links. Except for the nodes 5 and 16, the other nodes are controlled by signal timing. The network is expanded to virtual links referred to earlier coded in C# language. Figure 7 shows the signal timing at each intersection with signal control. Table 2 presents the travel time of links. Note that every link is two-way, but we just list one case when the travel time of two links, whose upstream and downstream nodes interchange, is the same. Table 3 lists the turning delays at intersections.

Based on the assumption of maximum pessimistic expectation, taking the T-shaped intersection two shown in Figure 6 as an example, the calculation of the turning delays will be described. When the forward intersection number of the path is origin r, and the backward intersection number is three, the vehicle needs to go through intersection two. According to the signal timing of Figure 7, going through the intersection, two corresponds to the first phase, and the green time of the first phase in one cycle time is 25 s, the yellow time is 5 s, and the red time is the 60 s. Since the vehicle stops during the yellow time and the red time, it is determined that the turning delay is 65 s. According to it, the turning delays at intersections in Figure 6 are derived in Table 3.

According to the algorithm flow, this algorithm is coded in C#. Each link is marked with its choice probability of the

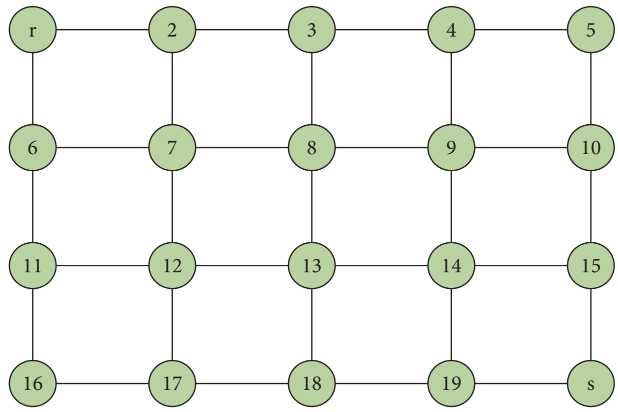

Figure 6: A grid-type network.

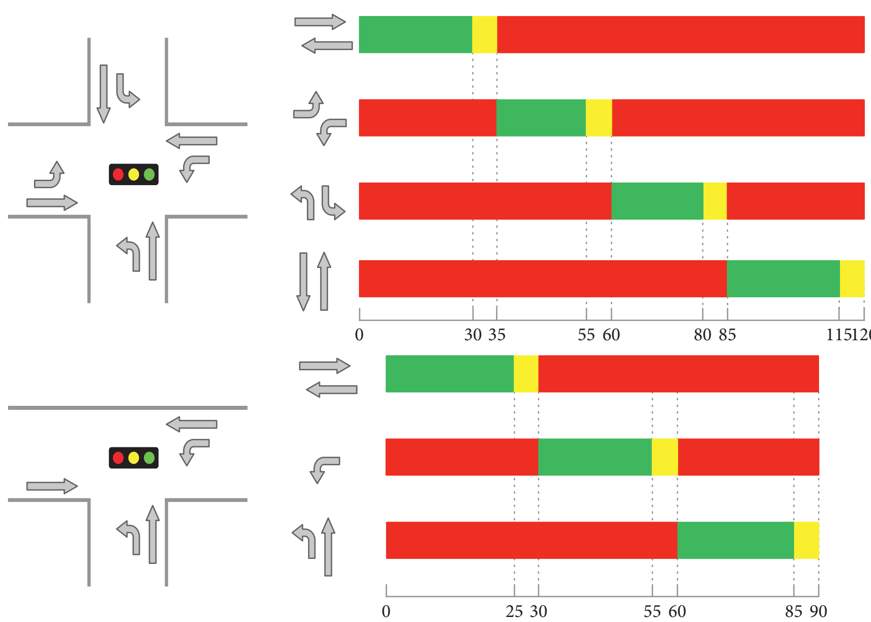

Figure 7: Intersection signal timing.

Table 2: Travel time of links.

Upstream node i	Downstream node j	Travel time (s)	Upstream node i	Downstream node j	Travel time (s)
R	2	60	19	S	40
2	3	60	r	6	30
3	4	40	6	11	20
4	5	30	11	16	40
6	7	6	2	7	60
7	8	70	7	12	30
8	9	30	12	17	60
9	10	20	3	8	20
11	12	50	8	13	30
12	13	70	4	9	20
13	14	60	9	14	40
13	18	30	14	19	30
14	15	30	5	10	50
16	17	70	10	15	60
17	18	60	15	S	70
18	19	30			

Table 3: Turning delays at intersections.

Forward intersection number	Intersection number	Backward intersection number	Turning delays (s)	Forward intersection number	Intersection number	Backward intersection number	Turning delays (s)
R	2	3	65	6	11	12	65
R	2	7	0	6	11	16	65
3	2	*r*	65	12	11	6	0
3	2	7	65	16	11	6	65
7	2	3	0	12	11	16	65
7	2	*r*	65	16	11	12	0
2	3	4	65	11	12	13	90
2	3	8	0	11	12	17	0
8	3	2	65	11	12	7	100
8	3	4	0	7	12	13	100
4	3	2	65	7	12	17	90
4	3	8	65	7	12	11	0
3	4	5	65	13	12	11	90
3	4	9	0	13	12	7	0
9	4	3	65	13	12	17	100
9	4	5	0	17	12	7	90
5	4	3	65	17	12	11	100
5	4	9	65	17	12	13	0
4	5	10	0	12	13	14	90
10	5	4	0	12	13	18	0
R	6	7	65	12	13	8	100
R	6	11	65	8	13	14	100
7	6	*r*	0	8	13	18	90
11	6	*r*	65	8	13	12	0
7	6	11	65	14	13	12	90
11	6	7	0	14	13	8	0
6	7	8	90	14	13	18	100
6	7	12	0	18	13	8	90
6	7	2	100	18	13	12	100
2	7	8	100	18	13	14	0
2	7	12	90	13	14	15	90
2	7	6	0	13	14	19	0
8	7	6	90	13	14	9	100
8	7	2	0	9	14	15	100
8	7	12	100	9	14	19	90
12	7	2	90	9	14	13	0
12	7	6	100	15	14	13	90
12	7	8	0	15	14	9	0
3	8	13	90	15	14	19	100
3	8	9	100	19	14	9	90
3	8	7	0	19	14	13	100
9	8	7	90	19	14	15	0
9	8	3	0	14	15	*s*	0
9	8	13	100	14	15	10	65
7	8	9	90	10	15	*s*	65
7	8	13	0	10	15	10	0
7	8	3	100	*s*	15	10	65
13	8	3	90	*s*	15	14	65
13	8	7	100	11	16	17	0
13	8	9	0	17	16	11	0
4	9	14	90	16	17	18	65
4	9	10	100	16	17	12	65
4	9	8	0	12	17	16	0
10	9	8	90	12	17	18	65
10	9	4	0	18	17	16	65
10	9	14	100	18	17	12	0
8	9	10	90	17	18	19	65
8	9	14	0	17	18	13	65

Table 3: Continued.

Forward intersection number	Intersection number	Backward intersection number	Turning delays (s)	Forward intersection number	Intersection number	Backward intersection number	Turning delays (s)
8	9	4	100	13	18	17	0
14	9	4	90	13	18	19	65
14	9	8	100	19	18	17	65
14	9	10	0	19	18	13	0
9	10	15	0	18	19	*s*	65
9	10	5	65	18	19	14	65
5	10	15	65	14	19	18	0
5	10	9	0	14	19	*s*	65
15	10	5	65	*s*	19	18	65
15	10	9	65	*s*	19	14	0

driver with the algorithm output. The probability is also the traffic distribution, as shown in Figure 8.

According to the link probability, there are nine possible shortest paths. The path choice probability is shown in Table 4.

5. The Cooperative Vehicle-Infrastructure System

5.1. Work Process of CVIs. Based on the hyperpath, this section introduces the CVIs to realize route guidance. The CVIs consist of four parts: intelligent traffic management system, intelligent communication system, intelligent vehicle system, and intelligent roadside system. The intelligent traffic management system processes the information obtained by both the intelligent vehicle system and the intelligent roadside system and then calculates the turning delay included in the hyperpath set, which is specifically divided into the following two situations:

Case 1. According to the transmitted information, when the turn encounters the green light, the intelligent traffic management system obtains the remaining green light time t_1 and the vehicle's distance l from the current position to the turn-stop line. After that, the intelligent traffic management system calculates the number of vehicles Q_2 that can pass during the remaining green time. If $Q_2 \geq Q_1$, the time when the vehicle arrives at the stop line is $t = l/v_1$, if $Q_2 < Q_1$. The signal cycle time is T, which is obtained by the database. The number of vehicles is Q_3 that can be passed during one signal cycle. The dissipation speed of the vehicles is v_2. The number of vehicles that need to wait is $n = \lfloor (Q_1 - Q_2)/Q_3 \rfloor$ during a signal cycle. The time to arrive at the stop line is $t = t_1 + nT + (Q_1 - Q_2 - nQ_3)/v_2$. The number of vehicles in front of the induced vehicle is Q_1, the symbol $\lfloor\rfloor$ indicates rounding down and $Q_2 = q \cdot t_1$.

Case 2. When the turn encounters the red light, the intelligent traffic management system obtains the remaining red time t_3 from the current phase. If the turn encounters the yellow light, the intelligent traffic management system should obtain the remaining yellow time plus red time. According to the database, the signal cycle time is T. The number of vehicles that can pass during one signal cycle is Q_3. The dissipation speed of the vehicles is v_2. The number of the vehicles that need to wait is $n = \lfloor Q_1/Q_3 \rfloor$. The time to arrive at the stop line is $t = t_3 + nT + (Q_1 - nQ_3)/v_2$.

The intelligent traffic management system determines the turning movement with the shortest time and recommends it to the driver. That ensures that vehicles always choose the passable priority phase at the intersection to

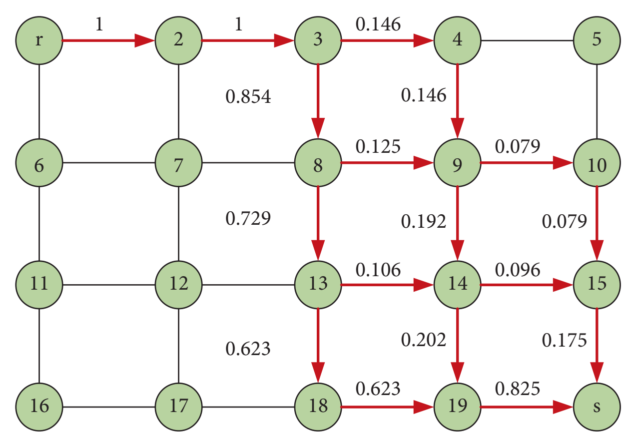

Figure 8: Link choice probability.

Table 4: Path choice probability.

Route	Choice probability
r-2-3-4-9-14-19-s	0.050
r-2-3-4-9-14-15-s	0.036
r-2-3-4-9-10-15-s	0.061
r-2-3-8-13-18-19-s	0.290
r-2-3-8-13-14-19-s	0.177
r-2-3-8-13-14-15-s	0.030
r-2-3-8-9-14-19-s	0.177
r-2-3-8-9-14-15-s	0.127
r-2-3-8-9-10-15-s	0.052

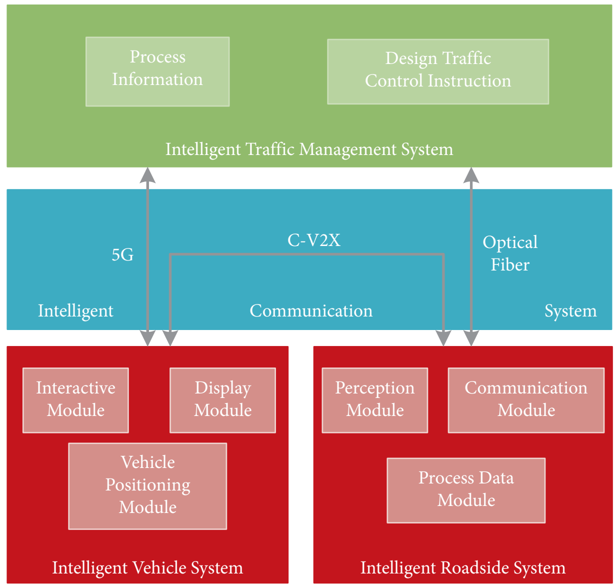

Figure 9: The cooperative vehicle-infrastructure system.

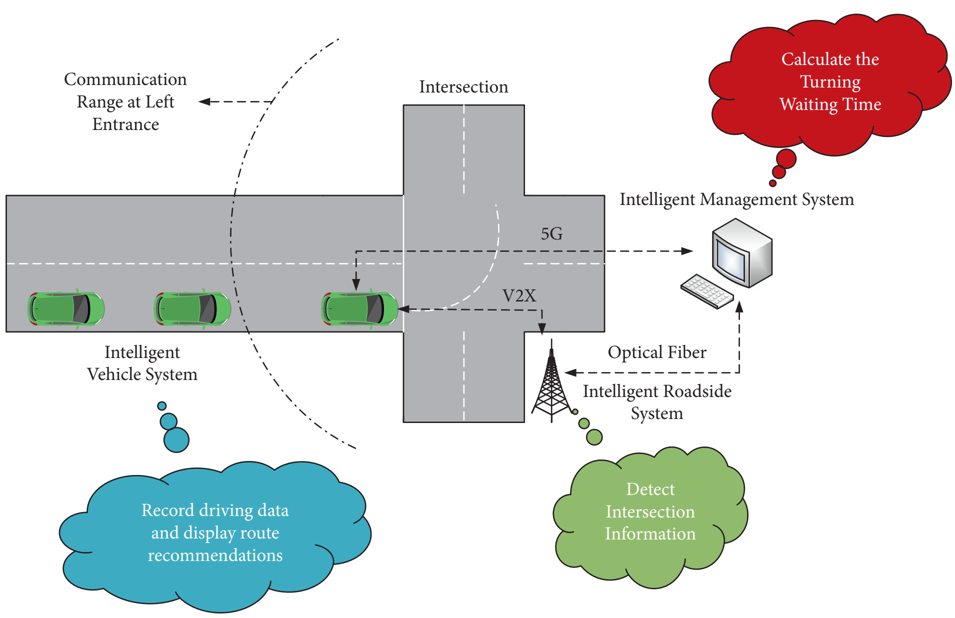

Figure 10: The information interaction on the cooperative vehicle-infrastructure system.

minimize waiting delays. Figure 9 shows the CVIs, and Figure 10 represents the information interaction principle of the vehicle route guidance system under the CVIs.

5.2. Numerical Experiment. Suppose a total of 1,000 drivers travel from the start point r to the end point s in a certain period. Travel along the nine paths was selected by the

Table 5: Hyperpath and single-path delay time comparison.

Hyperpath guidance under independence assumption							
Label	Route	Number of vehicles allocated	Expected vehicle delay time (s/pcu)	Total delay time (s/pcu)	Expected vehicle travel time (s/pcu)	Total travel time (s/pcu)
(1)	*r*-2-3-4-9-14-19-*s*	50				
(2)	*r*-2-3-4-9-14-15-*s*	36				
(3)	*r*-2-3-4-9-10-15-*s*	61				
(4)	*r*-2-3-8-13-18-19-*s*	290				
(5)	*r*-2-3-8-13-14-19-*s*	177	193.144	193,144	476.534	476,534
(6)	*r*-2-3-8-13-14-15-*s*	30				
(7)	*r*-2-3-8-9-14-19-*s*	177				
(8)	*r*-2-3-8-9-14-15-*s*	127				
(9)	*r*-2-3-8-9-10-15-*s*	52				
Shortest path guidance						
Label	Route	Number of vehicles allocated	Simple vehicle delay time (s/pcu)	Total delay time (s/pcu)	Simple vehicle travel time (s/pcu)	Total travel time (s/pcu)
(9)	*r*-2-3-8-9-10-15-*s*	1,000	205	205,000	525	525,000
Delays reduced by 5.8%						
Travel times reduced by 9.2%						

hyperpath algorithm (shown in Figure 9) and the only path with the shortest time. Calculate the delays of vehicles at the intersection, as shown in Table 5.

The results in Table 5 show that compared with the shortest path, applying the hyperpath guidance system in the road network can reduce the delays by 5.8% and travel times by 9.2%. It can be seen that the adoption of the hyperpath guidance strategy can realize the reasonable distribution of traffic in the network, reduce the delay at intersections, and achieve the shortest travel time in the network.

6. The Hyperpath in Nanjing Local Road Network

In order to validate our solution approach in a realistic context, we built our hyperpath network based on the local road network obtained from the center of Nanjing. By taking an investigation, there are 49 intersections and 37 out of them are controlled by traffic signals. Also, 29 road segments are involved, and some of them are one-way roads. Three road types (i.e., the arterial street, subarterial road, and branch road) are associated with different design speeds. The network is illustrated in Figure 11.

According to the output of the algorithm in Section 3.3, the optimal hyperpath is marked in Figure 12. Note that node 37 is the origin, and node 1 is the destination.

If we employ the Dijkstra algorithm on this network, the shortest path is 37-32-26-25-47-43-18-17-14-10-9-7-38-1, and the shortest travel time is 1,142 s. However, from the optimal hyperpath algorithm, the expected travel time is 1,093.88 s. Compared with the shortest path, the expected travel time of the optimal hyperpath is reduced by 4.2%. This result clearly showed the application of the hyperpath algorithm compared to the Dijkstra algorithm in terms of delays and travel time.

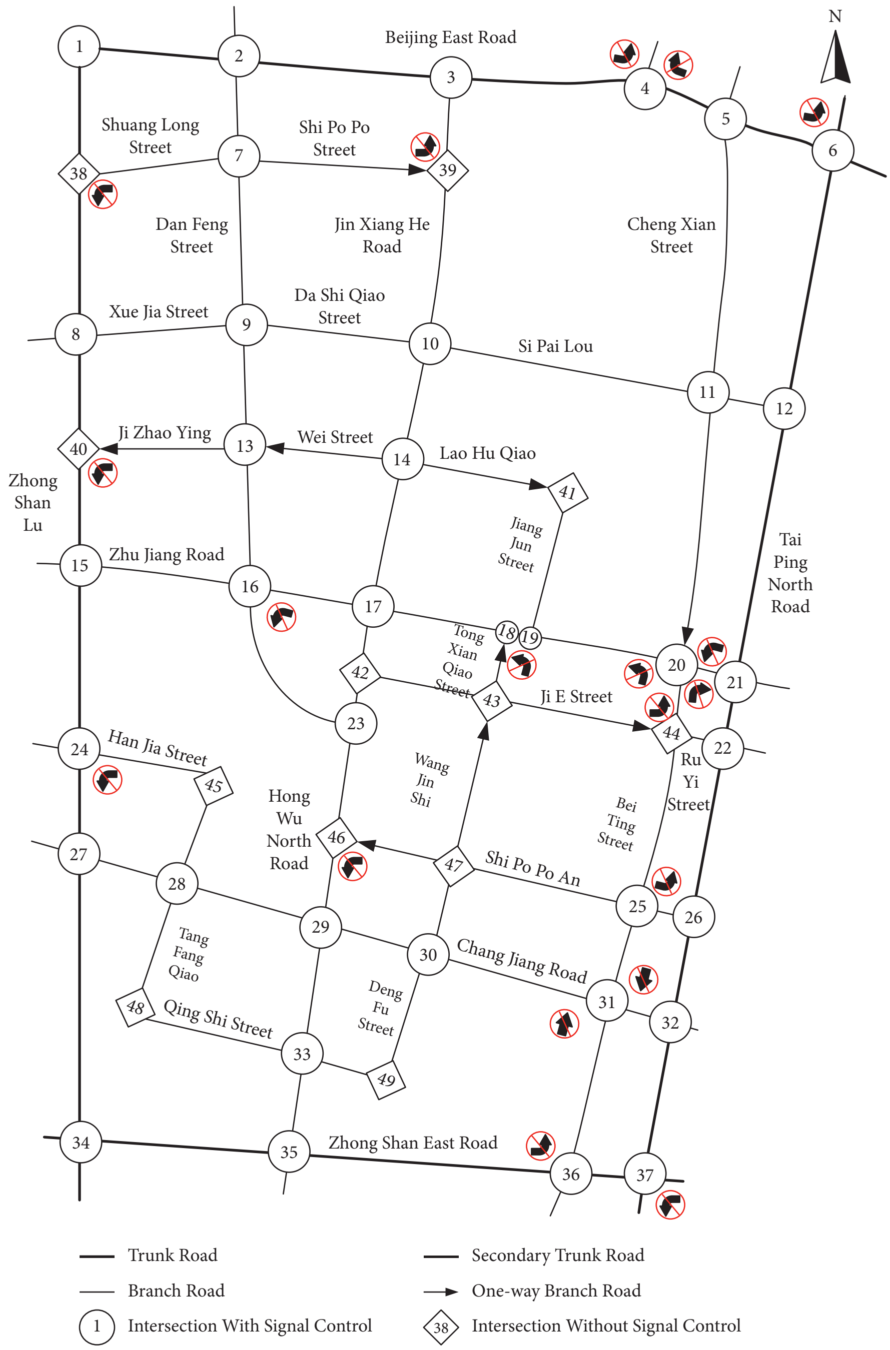

Figure 11: Local road network in Nanjing.

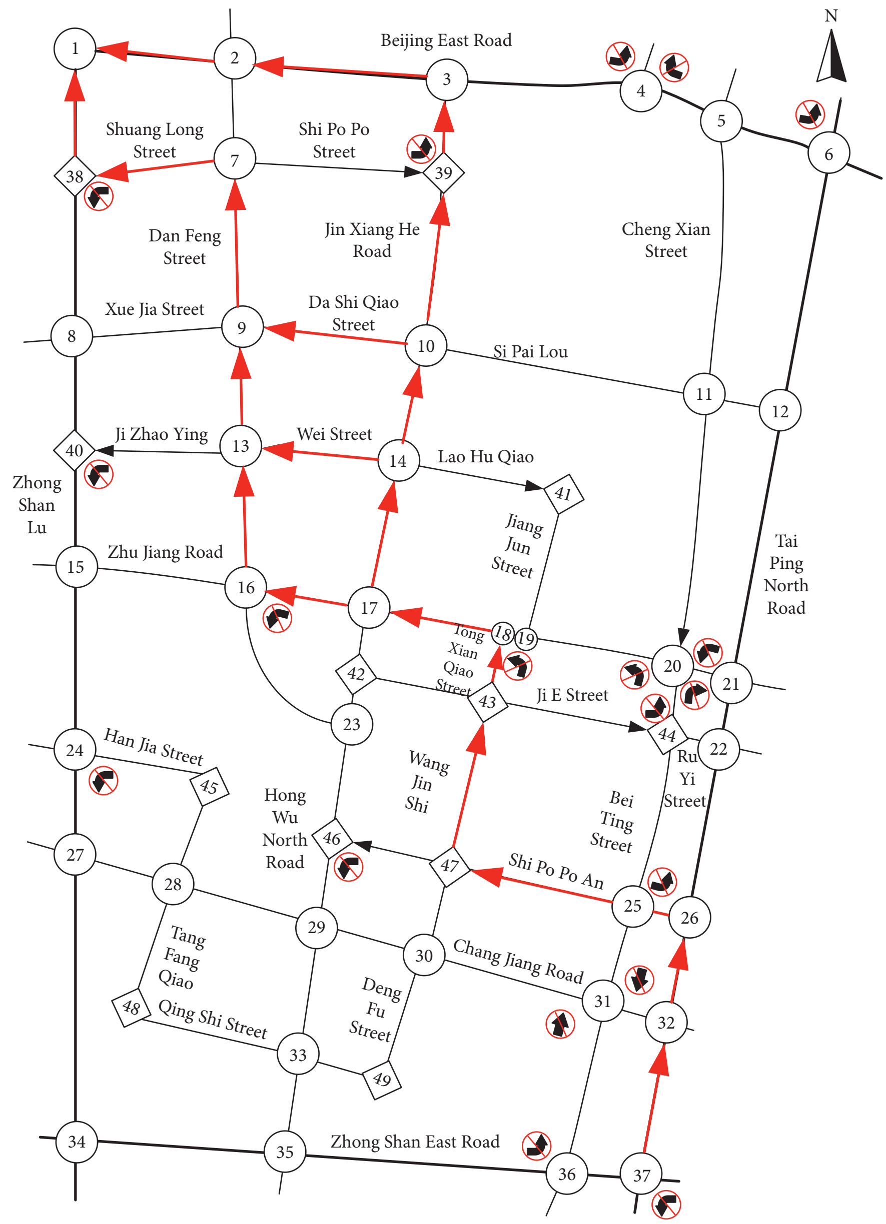

Figure 12: The optimal hyperpath from origin node 37 to destination node 1.

7. Conclusions

This article offers an efficient way to generate the optimal hyperpath set based on the signal timing and the CVIs for use in vehicle route guidance systems. Under the maximum pessimistic expectation assumption, this article generates the maximum waiting time for each turn at the intersection. Further, we find out the expected waiting time for linear combination, which avoids processing a large amount of data and accelerates the algorithm speed in the route guidance system. There is a "recommendation priority passable phase" strategy for the cooperative vehicle-infrastructure system, leading to a real-time and exact vehicle route guidance system.

For individuals, the driver can shift the driving route in the hyperpath subnet for the use of the route guidance system to avoid delays at intersections. As far as the overall network, this can be proven that using a hyperpath algorithm in the network will give a reasonable traffic distribution. Thus, the total travel time is reduced, and the transportation network's performance is improved, which will produce substantial economic benefits. In order to meet the increased demand for personalized travel, the hyperpath guidance system can introduce corresponding demand factors to change the objective function to improve network performance and further achieve personalized guidance to drivers. Meanwhile, it is vital to take into account the cooperative vehicle-infrastructure system when vehicles are guided. The tests concluded that compared with the shortest path, applying the hyperpath guidance system in the road network can reduce the delays by 5.8% and travel times by 9.2%. Such properties make the hyperpath algorithm a promising solution for use in intelligent transportation systems, which was confirmed by an experimental study comparing the travel time and delay time with the shortest path algorithms.

There are two assumptions about calculating the waiting time of vehicles at intersections based on signal timing. One is the assumption of maximum pessimistic expectations. The waiting time for each turn of the vehicle at the intersection is red time plus yellow time. The other is the independent hypothesis, assuming that the vehicle's phases are independent of each other when calculating the expected waiting time at the intersection. These two assumptions are ideal, leading to a specific deviation from the actual situation. Future research will further improve the mentioned assumptions to make the hyperpath algorithm more suitable for actual road network conditions.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was funded by the Natural Science Foundation of China, Grant no. 71801079, and the Fundamental Research Funds for the Central Universities, Grant no. B200202079.

References

[1] M. P. Dixon and L. R. Rilett, "Population origin-destination estimation using automatic vehicle identification and volume data," *Journal of Transportation Engineering*, vol. 131, no. 2, pp. 75–82, 2005.

[2] S. U. Shariff and M. Ganeshan, "A path finding visualization using A star algorithm and dijkstra's algorithm," *Journal of Trend in Scientific Research and Development*, vol. 5, no. 1, pp. 588-589, 2020.

[3] D. Mansoor and G. Mohsen, ""Shortest path problem on uncertain networks: an efficient two phases approach," *Computers & Industrial Engineering*, vol. 157, 2021.

[4] L. Chi-Kang, "A multiple-path routing strategy for vehicle route guidance systems," *Pergamon*, vol. 2, no. 3, pp. 185–195, 1994.

[5] Y. Chen, M. G. H. Bell, and K. Bogenberger, "Reliable pretrip multipath planning and dynamic adaptation for a centralized road navigation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 14–20, 2007.

[6] D. Eppstein, "Finding the k shortest paths," *SIAM Journal on Computing*, vol. 28, no. 2, pp. 652–673, 1998.

[7] E. De Queirós Vieira Martins, M. M. B. Pascoal, and J. L. E. D. Santos, "Deviation algorithms for ranking shortest paths," *International Journal of Foundations of Computer Science*, vol. 10, no. 03, pp. 247–261, 1999.

[8] V. M. Jiménez and A. Marzal, "Computing the K Shortest paths: a new algorithm and an experimental comparison," *Algorithm Engineering*, vol. 1688, pp. 15–29, 1999.

[9] E. A. Dinic, "Algorithm for solution of a problem of maximum flow in a network with power estimation," *Soviet mathematics - Doklady*, vol. 11, pp. 248–264, 1970.

[10] D. Torrieri, "Algorithms for finding an optimal set of short disjoint paths in a communication network," *IEEE Transactions on Communications*, vol. 40, no. 11, pp. 1698–1702, 1992.

[11] H. Yu, Z. Fang, F. Lu et al., "Impact of oil price fluctuations on tanker maritime network structure and traffic flow changes," *Applied Energy*, vol. 237, pp. 390–403, 2019.

[12] S. Vanhove and V. Fack, "An effective heuristic for computing many shortest path alternatives in road networks," *International Journal of Geographical Information Science*, vol. 26, no. 6, pp. 1031–1050, 2012.

[13] B. Y. Chen, Q. Li, and W. H. K. Lam, "Finding the k reliable shortest paths under travel time uncertainty," *Transportation Research Part B: Methodological*, vol. 94, pp. 189–203, 2016.

[14] X. Hu and Y.-C. Chiu, "A constrained time-dependent K shortest paths algorithm addressing overlap and travel time deviation," *International Journal of Transportation Science and Technology*, vol. 4, no. 4, pp. 371–394, 2015.

[15] W. Xu, S. He, R. Song, and S. S. Chaudhry, "Finding the K shortest paths in a schedule-based transit network," *Computers & Operations Research*, vol. 39, no. 8, pp. 1812–1826, 2012.

[16] M. Niksirat, M. Ghatee, and S. Mehdi Hashemi, "Multimodal K-shortest viable path problem in Tehran public transportation network and its solution applying ant colony and simulated annealing algorithms," *Applied Mathematical Modelling*, vol. 36, no. 11, pp. 5709–5726, 2012.

[17] B. Y. Chen, X.-W. Chen, H.-P. Chen, and W. H. K. Lam, "Efficient algorithm for finding k shortest paths based on re-optimization technique," *Transportation Research Part E: Logistics and Transportation Review*, vol. 133, Article ID 101819, 2020.

[18] J. Yallouz, O. Rottenstreich, P. Babarczi, A. Mendelson, and A. Orda, "Minimum-weight link-disjoint Node-"Somewhat disjoint" paths," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1110–1122, 2018.

[19] T. Gomes and J. Craveirinha, "Efficient calculation of the most reliable pair of link disjoint paths in telecommunication networks," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1055–1064, 2007.

[20] T. Gomes, L. Martins, S. Ferreira, M. Pascoal, and D. Tipper, "Algorithms for determining a node-disjoint path pair visiting specified nodes," *Optical Switching and Networking*, vol. 23, pp. 189–204, 2017.

[21] C.-N. Lai, "Optimal node-disjoint paths in folded hypercubes," *Journal of Parallel and Distributed Computing*, vol. 147, pp. 100–107, 2021.

[22] L. Wang, S. Gao, K. Wang, T. Li, L. Li, and Z. Chen, "Time-dependent electric vehicle routing problem with time windows and path flexibility," *Journal of Advanced Transportation*, vol. 2020, pp. 1–19, 2020.

[23] R. Guo, W. Zhang, W. Guan, and B. Ran, "Time-dependent urban customized bus routing with path flexibility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2381–2390, 2021.

[24] X. Liu, M. Qi, and C. Cheng, "Green vehicle routing problem with path flexibility," in *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, 10-13 Dec. 2017.

[25] L. Shen, H. Shao, T. Wu, E. Z. Fainman, and W. H. K. Lam, "Finding the reliable shortest path with correlated link travel times in signalized traffic networks under uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 144, Article ID 102159, 2020.

[26] H. Spiess and M. Florian, "Optimal strategies: a new assignment model for transit networks," *Transportation Research Part B: Methodological*, vol. 23, no. 2, pp. 83–102, 1989.

[27] M. G. H. Bell, "Hyperstar: a multi-path Astar algorithm for risk averse vehicle navigation," *Transportation Research Part B*, vol. 43, no. 1, pp. 97–107, 2008.

[28] J. Ma, D. Fukuda, and J.-D. Schmöcker, "Faster hyperpath generating algorithms for vehicle navigation," *Transportmetrica: Transportation Science*, vol. 9, no. 10, pp. 925–948, 2013.

[29] L. Hu, J. Ou, J. Huang et al., "Safety evaluation of pedestrian-vehicle interaction at signalized intersections in Changsha, China," *Journal of Transportation Safety & Security*, p. 26, 2021.

[30] L. Hu, X. Q. Zhou, X. Zhang, F. Wang, Q. Li, and W. Wu, "A review on key challenges in intelligent vehicles: safety and driver-oriented features," *IET Intelligent Transport Systems*, vol. 15, no. 9, pp. 1093–1105, 2021.

[31] L. Hu, X. Bao, M. Lin, C. Yu, and F. Wang, "Research on risky driving behavior evaluation model based on CIDAS real data," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 8, pp. 2176–2187, 2021.

[32] F. Zheng and H. Van Zuylen, "Uncertainty and predictability of urban link travel time," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2192, no. 1, pp. 136–146, 2010.