electronics

MDPI

Article

Hyperbolic-Embedding-Aided Geographic Routing in Intelligent Vehicular Networks

Ying Pan [1,2] and Na Lyu [2,*]

1 Graduate School, Air Force Engineering University, No.1(A) Changle East Road, Baqiao Distinct, Xi'an 710051, China; panda17691187697@outlook.com
2 Information and Navigation School, Air Force Engineering University, No.1 Fenghao East Road, Lianhu Distinct, Xi'an 710077, China
* Correspondence: lvnn2007@163.com

Abstract: Intelligent vehicular networks can not only connect various smart terminals to manned or unmanned vehicles but also to roads and people's hands. In order to support diverse vehicle-to-everything (V2X) applications in dynamic, intelligent vehicular networks, efficient and flexible routing is fundamental but challenging. Aimed to eliminate routing voids in traditional Euclidean geographic greedy routing strategies, we propose a hyperbolic-embedding-aided geographic routing strategy (HGR) in this paper. By embedding the network topology into a two-dimensional Poincaré hyperbolic disk, greedy forwarding is performed according to nodes' hyperbolic coordinates. Simulation results demonstrated that the proposed HGR strategy can greatly enhance the routing success rate through a smaller stretch of the routing paths, with little sacrifice of routing computation time.

Keywords: intelligent vehicular networks; geographic routing; routing void; hyperbolic space

check for updates

Citation: Pan, Y.; Lyu, N. Hyperbolic-Embedding-Aided Geographic Routing in Intelligent Vehicular Networks. *Electronics* **2024**, *13*, 661. https://doi.org/10.3390/electronics13030661

Academic Editor: Dongkyun Kim

Received: 18 December 2023
Revised: 31 January 2024
Accepted: 4 February 2024
Published: 5 February 2024

1. Introduction

With the rapid advancement of artificial intelligence, big data and 5G technologies, the concept of intelligent vehicular networks has become a reality in recent years [1]. As an increasing number of vehicles, especially autonomous self-driving cars, are becoming connected in the intelligent vehicular network [2], the performance requirement of disseminating transportation-critical information becomes much more stringent. In such a large-scale, heterogeneous, complex network, the design of a reliable and efficient routing strategy is of great necessity but quite challenging [3–5]. Therefore, the study of routing strategies has long been a hot topic in the research field of intelligent vehicular networks [6–8].

Unfortunately, traditional routing strategies [9–11] are often "topology-based" and lack full consideration of the dynamic topological characteristics in intelligent vehicular networks. Distinguished from general wired networks, the wireless connections in intelligent vehicular networks are impermanent and intermittent due to the mobility of vehicles [12]. Given the dynamic nature of network topology, traditional topology-based routing strategies are too sensitive to topology churns and are not well suited for intelligent vehicular networks [13]. To address this issue, geographic routing strategies [13–18] are introduced. Geographic routing strategies find routing paths based on nodal location, rather than the network topology. Utilizing the geographic location information offered by the global positioning system (GPS) on board, geographic routing strategies greedily forward packets hop by hop in a destination-distance-descending manner until the desired destination is reached. In this way, each node merely needs to know its location information, its neighbors, and the destination node, thus removing the precise requirement of global network topology information. This efficiency reduces the overhead of maintaining routing tables and removes the limitation of scaling in large-scale networks. Therefore, geographic routing strategies are more suitable for intelligent vehicular networks, and a plethora of geographic routing strategies have emerged in the context of intelligent vehicular networks.

However, existing geographic routing strategies suffer a lot from the problem of "routing void" [19,20]. That is, the current packet holder's distance to the destination is closer than any other neighbors, and the destination cannot be reached within one hop. When the routing void occurs, it can lead to serious packet delivery failures, which is unacceptable for the information transmission service, with demanding performance requirements in intelligent vehicular networks. Although efforts have been committed to developing countermeasures, such as the "right-hand rule" or face routing, the improved routing success rate is still lower than 100% [21]. Reliable and effective information dissemination in intelligent vehicular networks is calling for a brand-new design of routing strategies.

Hyperbolic geometric space brings us new ideas. In recent studies in network science, increasing evidence has been found that the hidden metric beneath many real-world networks is hyperbolic [22–24]. In contrast, existing geographic greedy routing in intelligent vehicular networks is typically in the Euclidean metric space, but the shortest path distance in the network topology graph can not be well-represented by the distance in the Euclidean metric space without distortion [25]. Experimental results also demonstrate that hyperbolic embeddings of complex networks enjoy the property of lower distortion compared to Euclidean network embeddings [26]. Most importantly, it is proved that every finite graph has a greedy embedding in the hyperbolic space that guarantees that greedy forwarding is always successful in finding a route to the destination, if such a route exists [27,28]. Inspired by this, we propose a hyperbolic-embedding-aided geographic routing strategy in intelligent vehicular networks, termed HGR. We will elaborate on HGR in the remainder of our paper, which is organized as follows. Section 2 discusses background and related works. Section 3 presents our detailed design of HGR. The performance evaluation results are reported in Section 4, and we draw conclusions in Section 5.

2. Background and Related Works

2.1. Background

The development of the intelligent vehicular network has gone through several stages. Originally known as the vehicular network, it was proposed to realize coordination among cars and roadside units. Thanks to rapid technological advances, vehicular networks are evolving into the Internet of Vehicles (IoV), which is extended from the concept of the Internet of Things (IoT). To avoid confusion with vehicle-mounted telematics, the term Vehicle-to-Everything (V2X) is more widely used. As its name suggests, V2X includes not only Vehicle-to-Vehicle (V2V) communications but also Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), and Vehicle-to-Network (V2N) communications. With the rapid development of artificial intelligence in recent years, the intelligent vehicular network has gained increasing attention due to the need for connected smart vehicles and driverless cars [1,2,29]. As a key enabler of intelligent transportation systems (ITS), intelligent vehicular networks will ultimately shift the role of cars into no longer merely a simple transportation tool, making our daily traveling safer, greener, and more convenient [30]. The networking scenario of intelligent vehicular networks is depicted in Figure 1.

Much effort has been made toward the standardization of vehicular networks across countries. In the U.S., vehicular networks are developed following the dedicated short-range communications (DSRC) technology [31]. DSRC is based on the IEEE 802.11p protocol and uses the 75 MHz frequency band from 5.850 GHz to 5.925 GHz. The purpose of the DSRC service is to improve traffic safety and reduce congestion. The European standard for V2X communication is cooperative intelligent transportation systems (C-ITS). C-ITS mainly focuses on one-to-one or one-to-many communication between cars, trucks, buses, trains and infrastructures. And C-ITS is compatible with cellular communication networks, Wi-Fi, and DSRC. In C-ITS, a frequency band of 70 MHz is reserved for DSRC, ranging from 5855 MHz to 5925 MHz. At present, IEEE 802.11p is the mainstream access protocol for DSRC. However, DSRC requires too much on building roadside infrastructures, resulting in a high cost of deployment. China promotes the Long Term Evolution V2X (LTE-V) technical route. Compared to DSRC, LTE-V utilizes the existing infrastructure and spectrum resources of cellular networks; hence, LTE-V is also often called cellular V2X (C-V2X). There are two types of air interfaces in C-V2X, i.e., Uu and PC5. The Uu interface uses the base station as the control center, providing wide-range, large-bandwidth communication services. The PC5 interface can support direct data transmission between vehicles. Recently, a new global organization, 5G Automotive Association (5GAA), has been formed to facilitate C-V2X standardization. C-V2X is becoming more and more popular, and it is a promising trend for future intelligent vehicular networks.

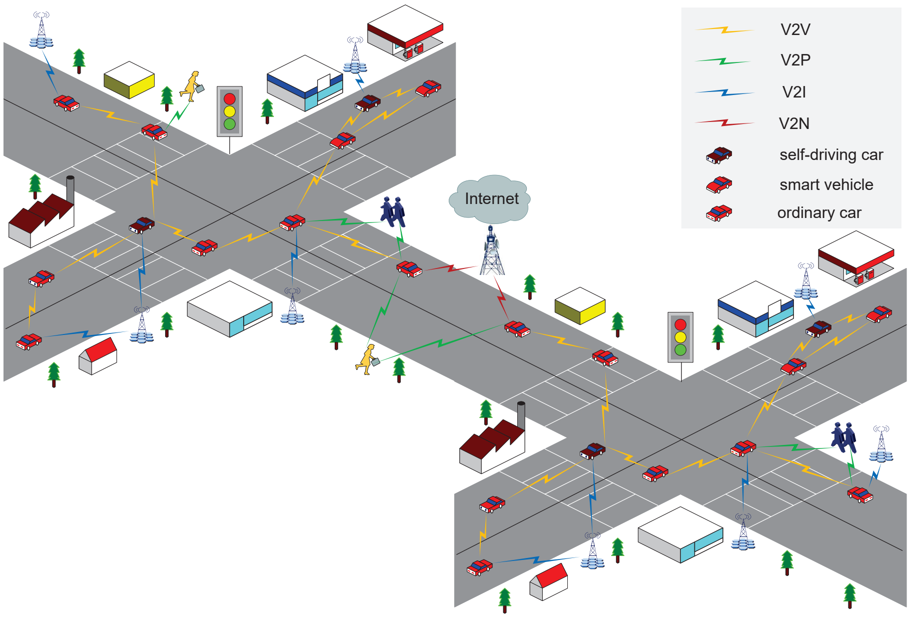


Figure 1. An illustration of intelligent vehicular networks.

In contrast to DSRC, which is more mature, the development of C-V2X started relatively late. With the proliferation of vehicular networks, multi-hop routing between vehicles in C-V2X is common, especially when using a PC5 interface. In this context, the design of a reliable and effective routing strategy is a fundamental issue that deserves more academic attention.

2.2. Related Works

Major challenges in designing routing strategies to provide reliable and efficient information services for intelligent vehicular networks result from the stringent requirements of delay-sensitive vehicular applications, as well as the inherent dynamics in vehicular network topology [12]. Taking both timeliness and reliability into consideration, routing strategies need to compute the shortest paths in dynamic vehicular networks. Existing routing strategies can be categorized into topology-based routing strategies and geographic routing strategies with respect to their shortest-path calculation methods [16]. Topology-based routing strategies [9–11], which need to collectively discover the current state of the network topology by exchanging information among nodes, have oblivious limitations when applied in intelligent vehicular networks. Alternatively, geographic routing strategies are much more suitable considering the time-varying topology characteristics of intelligent vehicular networks [21]. Geographic routing strategies transmit packets in a greedy manner by successively forwarding packets to the neighbor closer to the destination according to coordinates given by an embedding into some metric space, typically Euclidean. Geographic routing strategies are thus simple to implement but robust to topology churns, for only local information, rather than global topology, is required for routing decisions.

With the above advantages, as well as the proliferation of GPS-capable communication devices, geographic routing is a favorable choice for routing strategy design in intelligent vehicular networks [13–18]. In [13], the authors propose a connectivity-aware transmission quality guaranteed geographic routing in urban IoVs. In [14] transmission quality is guaranteed through an intersection-based geographic routing scheme. A geographic routing protocol based on trunk lines for vehicular networks is designed in [15]. And a delay-aware backbone-based geographic routing for IoV is proposed in [16]. There are also some intelligent routing methods using machine learning techniques, such as adaptive UAV-assisted geographic routing with Q-learning in [17] and reinforcement learning-based geographic routing in [18], to name a few.

Although geographic routing strategies are widely deployed in intelligent vehicular networks, they are based on the Euclidean distance metric and can hardly guarantee that greedy forwarding always succeeds in reaching the target destination. When there exists a node nearer to the destination than all of its neighbors, greedy forwarding is susceptible to failing into routing voids. Figure 2 offers an instance of a routing void, in which the current packet holder, the red-colored node v can not find a next-hop neighbor that is closer to destination node d (colored green) than itself according to the greedy forwarding rule.

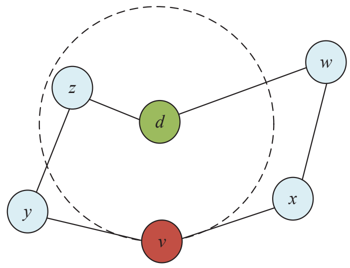


Figure 2. An instance of a routing void.

At first glance, routing voids are likely to occur when nodes are sparsely distributed. But after a deeper investigation, the core reason for routing voids lies in the fact that the Euclidean distance can not reflect the distance on the topology graph without distortion. Greedy routing can find the shortest paths only if the network topology is congruent with the metric space with respect to distances. Advances in network geometry have shown that structural properties observed in scale-free complex networks derived from real networks can emerge as a consequence of the geometrical properties of hyperbolic space [22–24]. Models in hyperbolic space can reproduce the basic topology properties of real-world complex networks, suggesting that the hidden metric space underlying practical networks is hyperbolic [32,33]. Hence, hyperbolic space can be an ideal metric space for real-world network embedding to solve network problems.

In recent years, there has been a plethora of hyperbolic embedding algorithms proposed. For instance, the paper [34] constructs an efficient quasi-linear time maximum likelihood estimation algorithm that embeds scale-free graphs in the hyperbolic space. In [35], the authors propose a method for embedding directed networks into hyperbolic space. In [36], a Laplacian-based hyperbolic embedding scheme is proposed. The paper [37] modified the Laplacian-based hyperbolic embedding scheme for temporal complex networks. Coalescent hyperbolic embedding was proposed in [38] based on manifold machine learning. A reliable embedding algorithm, Mercator, that maps real complex networks into their latent hyperbolic geometry, was introduced in [39]. And the D-Mercator method for the multidimensional hyperbolic embedding of real networks is designed in [40]. These hyperbolic embedding algorithms have been implemented in numerous kinds of downstream application tasks, such as community discovery [41,42], link prediction [43,44] and network routing [27,28,45–48], which proves their efficiency and effectiveness.

The combination of hyperbolic embedding and greedy routing turns out to be powerful. In 2007, Robert Kleinberg's groundbreaking work [27] proved the existence of greedy hyperbolic network embedding such that greedy forwarding can achieve 100% reachability. In [28], an online embedding and routing scheme for arbitrary connected network graphs is proposed. A novel hyperbolic embedding scheme for efficient computation of path centralities and adaptive routing in large-scale complex commodity networks is proposed in [46]. The paper [47] presented hyperbolic embedding algorithms that are well-suited for greedy routing and demonstrated that hyperbolic embedding can facilitate maximally efficient greedy routing in complex networks. In [48], the authors developed an optimization scheme for improving the greedy routing score in the hyperbolic space. Inspired by these works, in this paper, we present and implement a hyperbolic-embedding-aided greedy routing strategy for performance enhancement in intelligent vehicular networks.

3. Hyperbolic-Embedding-Aided Geographic Routing Strategy

In this section, we explain the proposed hyperbolic-embedding-aided geographic routing (HGR) strategy, including some preliminaries about the hyperbolic network model and the algorithm design.

3.1. Preliminaries

In this paper, we formally model intelligent vehicular networks as complex networks. We assume that each node in the network is equipped with a positioning system. By periodically sending location beacon messages to their neighbors, nodes become aware of the connectivity state around them. The connectivity state information is broadcast to all other nodes and finally converges into a network topology graph. If the connectivity state information changes, then a broadcast update is performed and the aggregated network graph will be updated accordingly. Let the undirected graph $G(V, E)$ be the topology of the intelligent vehicular network, where V and E represent the node sets and edge sets, respectively. The total number of nodes $|V|$ is denoted as n, and the total number of edges is denoted as $|E| = m$. Suppose that the node degree distribution of the intelligent vehicular network follows a power-law distribution with scaling exponent γ.

In our HGR, the network graph $G(V, E)$ is embedded into its latent hyperbolic space. Hyperbolic space is a non-Euclidean metric space that has constant negative curvature. Different from the Euclidean distance metric, the distance between nodes u and v in hyperbolic space is

$$hdis(u, v) = \operatorname{arccos}h(\cosh r_u \cosh r_v - \sinh r_u \sinh r_v \cos(\theta_u - \theta_v)), \tag{1}$$

where r_u and r_v denote the radial coordinates of nodes u and v in hyperbolic space, while θ_u and θ_v represent the angular coordinates of nodes u and v, respectively.

Hyperbolic space is even larger than the flat Euclidean space, for both the area and circumference of a disk in the hyperbolic space expand exponentially with the increase in the radius. This geometric property makes it hard to visualize on paper. In this work, we use the two-dimensional Poincaré hyperbolic disk to represent the embedding target space. The Poincaré disk is a hyperbolic model that represents the hyperbolic plane by mapping it to the interior of a Euclidean unit disk. Figure 3 shows a triangulation on a Poincaré disk, in which all triangles are equilateral, and all of them are of the same size.

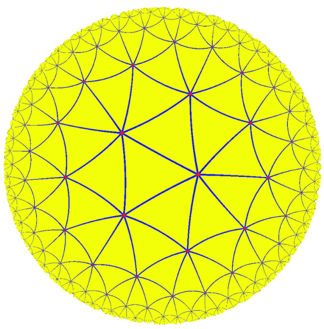


Figure 3. An instance of a triangulation on the Poincaré disk.

The core of our hyperbolic-embedding-aided geographic routing strategy is to assign an appropriate virtual hyperbolic coordinate for each node in the network. Before presenting our embedding algorithm, we would like to introduce the popularity-similarity optimization (PSO) model [33], which serves as a null model of our embedding algorithm.

The PSO model includes the following four parameters:

(1) $\mu > 0$, which is equal to half of the average node degree;
(2) $\beta \in (0, 1]$, defining the power-law degree distribution exponent $\gamma = 1 + 1/\beta$;
(3) $T \geq 0$, which controls the network clustering;
(4) $\zeta = \sqrt{-K} > 0$, where K is the curvature of the hyperbolic plane. Since changing ζ rescales the node radial coordinates and this does not influence the network topology structure, we set $K = -1$ without loss of generality.

To construct a network of n nodes on the Poincaré hyperbolic disk plane with curvature $K = -1$, the PSO model works as follows. Initially, the network is empty. At time $t \geq 1$, add a new node v_t at (r_t, θ_t), in which radial coordinate $r_t = 2\ln t$ and angular coordinate θ_t is uniformly sampled in $[0, 2\pi]$. At the same time, all existing nodes $v_s (s < t)$ added before time t increase their radial coordinates according to $r_s(t) = \beta r_s + (1 - \beta) r_t$. Then, link the newly-added node v_t with an existing node v_s that is not connected to it with the probability $p_s(t) = 1 \Big/ \left[1 + e^{(hdis(s,t) - R_t)/2T}\right]$, in which $hdis(s, t)$ is the hyperbolic distance between node v_t and node v_s. And $R_t = r_t - 2\ln\left[\frac{2T\left(1 - e^{-(1-\beta) r_t/2}\right)}{\mu(1-\beta)\sin(\pi T)}\right]$, which is the current radius of the hyperbolic circle containing the network. If $T = 0$, then $R_t = r_t - 2\ln\left[\frac{2T\left(1 - e^{-(1-\beta) r_t/2}\right)}{\mu\pi(1-\beta)}\right]$. Repeat this random connecting process until node v_t is connected to μ different nodes. Keep adding new nodes to the network as described above, and stop adding nodes when there are a total of n nodes in the network.

As its name suggests, the PSO model maintains a trade-off between node popularity, abstracted by the radial coordinate, and similarity, represented by the angular coordinate difference. It has been proven that the PSO model can generate random complex network topologies, reproducing common features of many realistic complex networks, including scale-free degree distributions [33]. Therefore, it is possible to implement the hyperbolic embedding method in the geographic greedy routing strategy in intelligent vehicular networks.

3.2. Algorithm Design

Our design for the proposed HGR is composed of two main steps. In the first step, we embed the network topology into the Poincaré hyperbolic disk based on the graph Laplacian. And in the second step, greedy forwarding is performed in the the Poincaré hyperbolic disk according to nodes' embedded hyperbolic coordinates. Figure 4 illustrates the design overview of the HGR strategy.

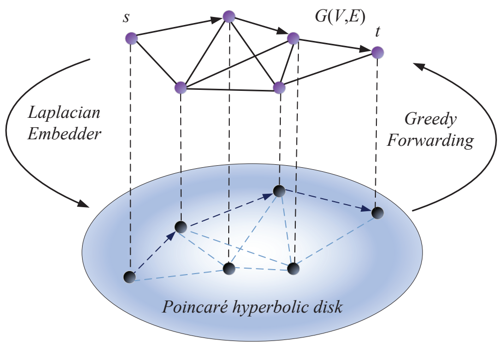


Figure 4. The design overview of HGR.

As is shown in Figure 4, the network graph $G(V, E)$ is embedded into the hyperbolic space via network embedding, and each node in the network is assigned to a virtual hyperbolic coordinate. When node s wants to deliver a data packet to node t, the packet is forwarded in a greedy manner based on the hyperbolic distance metric, as is shown in the arrow-annotated path.

3.2.1. Laplacian-Based Hyperbolic Embedding

Hyperbolic embedding refers to the process of mapping nodes back to their corresponding coordinates in the latent hyperbolic space. In fact, hyperbolic embedding is the inverse problem of generating complex network topologies by the PSO model.

For a network topology graph $G(V,E)$, hyperbolic embedding aims to assign nodes with coordinates $\boldsymbol{x} = (\boldsymbol{r}, \boldsymbol{\theta}) = \{x_i\} = \{(r_i, \theta_i)\}(i = 1, 2, \ldots)$ by maximizing the posterior probability $\mathcal{L}\{(x_i)|a_{ij}, \mathcal{P}\}$ under the premise that the network is generated as the PSO model. $\mathcal{P}$ denotes the set of parameters of the PSO, and $\mathcal{P} = \{\mu, \beta, T, \zeta\}$. According to the Bayes' rule, the probability is

$$\mathcal{L}\{(x_i)|a_{ij}, \mathcal{P}\} = \frac{\mathcal{L}\{a_{ij}|\{x_i\}, \mathcal{P}\} prob(x_i)}{\mathcal{L}\{a_{ij}|\mathcal{P}\}}, \tag{2}$$

where $\mathcal{L}\{a_{ij}|\{x_i\}, \mathcal{P}\}$ is the likelihood that a network with adjacency matrix $A = \{a_{ij}\}$ is generated as PSO, $prob(x_i)$ is the prior probability of node coordinates generated by PSO, and $\mathcal{L}\{a_{ij}|\mathcal{P}\}$ is the network that has been generated by the PSO model.

To solve this reverse problem, in the proposed HGR, we embed the observed network topology $G(V,E)$ of intelligent vehicular networks into a two-dimensional Poincaré disk via the Laplacian Embedder. The Laplacian Embedder was originally proposed in the paper [36] for dimension reduction in the context of manifold learning, and it is used here to recover the hyperbolic coordinates of nodes.

Given a network topology graph $G(V,E)$, the Laplacian matrix of it is

$$L = D - A, \tag{3}$$

where A is the adjacency matrix, and D is the degree matrix of $G(V,E)$.

In the proposed scheme, the radius R of the embedded Poincaré disk is calculated by

$$R = 2\log\left(\frac{n^2(\gamma-1)^2 T}{m\sin(\pi T)(\gamma-2)^2}\right). \tag{4}$$

in which n is the total number of nodes in $G(V,E)$, and m is the total number of edges in $G(V,E)$. The power law index of the complex network is denoted as γ. And T is the temperature parameter of the complex network.

For node v_i in $G(V,E)$, it is inferred that the radial coordinate r_i is

$$r_i = \min\left\{R, 2\log\left(\frac{nT(\gamma-1)}{\deg(v_i)\sin(\pi T)(0.5\gamma-1)}\right)\right\}, \tag{5}$$

in which $\deg(v_i)$ denotes the node degree of v_i.

Next, we will determine node angular coordinate θ_i. To do so, we use a $n \times 2$ matrix $Y = [y_1, y_2]$ in the interior of a Euclidean circle to represent the two-dimensional Poincaré disk. The ith row of Y denotes the embedding coordinates of node v_i. According to the PSO model, the connected probability of node pairs in the network is negatively correlated with the embedding coordinate distance. In this sense, the basic criterion for solving the embedding coordinates is to minimize the distance between the connected node pairs of the network in the embedding space. This is equivalent to minimizing the sum of the distances of all connected node pairs in the network. The objective function can be interpreted as

$$\frac{1}{2}\sum_{i,j} a_{ij}\left\|Y_i - Y_j\right\|^2 = tr\left(Y^T L Y\right), \tag{6}$$

in which $tr\left(Y^T LY\right)$ is the trace of the matrix after a Laplacian transformation. Considering the additional constraint $Y^T Y = I$, this optimization problem can be formulated as

$$\begin{cases} \min tr\left(Y^T LY\right) \\ s.t. \quad Y^T Y = I \end{cases} . \tag{7}$$

According to the Rayleigh–Ritz theorem, the solution to this is formed by the two eigenvectors corresponding to the two smallest non-zero eigenvalues of the generalized eigenvalue problem $LY = \lambda DY$. Since the smallest eigenvalue is zero, the generalized eigenvalue problem can be solved via the truncated spectral decomposition. And the solution is

$$Y = [y_1 = \mu_2, y_2 = \mu_3], \tag{8}$$

in which μ_2, μ_3 denote the two smallest non-zero eigenvalues, respectively.

Based on conformal properties, the node angular coordinates are approximated by

$$\boldsymbol{\theta} = \arctan\left(\frac{y_2}{y_1}\right), \tag{9}$$

in which $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_n]$ represents the angular coordinates of nodes sorted in a degree-descending order.

Up till now, each node's hyperbolic coordinate (r, θ) is determined.

3.2.2. Hyperbolic Greedy Forwarding

After embedding, each node in the network is assigned a virtual coordinate in the hyperbolic plane. To perform greedy forwarding, every node in the network not only needs to know its own hyperbolic coordinate but also the hyperbolic coordinates of its neighbors and the destination. In HGR, the data packet is encoded in IP format. And Figure 5 shows our design for the encapsulation format of data packets in intelligent vehicular networks.

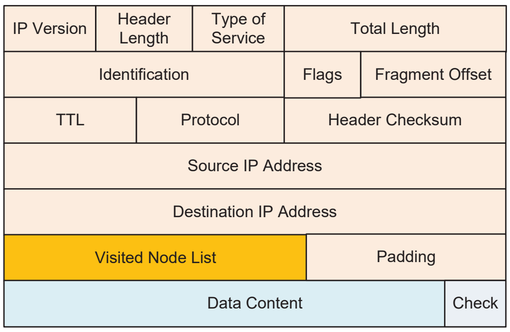


Figure 5. The packet encapsulation format of data in HGR.

In Figure 5, the orange field is the data packet header. The packet header stores the information about the packet's destination IP address, and it can be translated into the hyperbolic address by table lookup. In the packet header, the option field of the traditional IP packet is used to store the visited node list, which is colored a darker orange. At the source node, the packet's visited node list is empty. When a packet arrives at node v_i from v_j, the former forwarder, node v_j will be added to the visited node list. In addition to containing the packet's data content, the packet encapsulation format also contains a check field for calibration.

To avoid routing loops, we stipulate that the packet's next hop should be selected from the neighbors that haven't been visited. The hyperbolic distance between every unvisited neighbor of the current packet holder and the packet's destination is computed by the hyperbolic distance formula of Equation (1). Then, the neighbor with the closest distance to the destination is selected as the packet's next hop. Following this rule, packets are forwarded in a greedy manner until eventually reaching the desired destination. This scheme is referred to as hyperbolic greedy forwarding, and the operation flow chart is summarized in Figure 6.

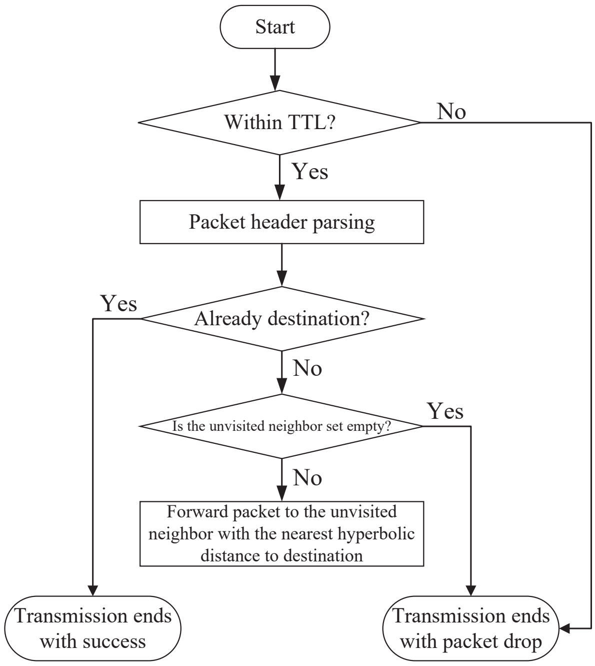


Figure 6. Flow chart of hyperbolic greedy forwarding.

4. Performance Evaluation

In order to verify the effectiveness of the proposed HGR strategy for intelligent vehicular networks under different network scenarios, a co-simulation was produced using EXata 5.3 and Mathematica 12.1 software. EXata is a popular simulation platform for emulating or simulating wireless networks, while Mathematica is an excellent software for scientific programming. EXata is used for generating vehicular networks, and Mathematica is used for routing calculation. The above two simulators are coupled with each other and work together as follows. Firstly, we randomly spread a certain number of nodes on the entire map canvas using the *Node Placement Tool* in EXata. Then, the location and connectivity information of these nodes are collected and abstracted as a network graph. Afterward, we input the network topology graph into Mathematica for network embedding and routing calculation, according to which the forwarding rules are configured in EXata. At the same time, the contrastive routing algorithm is calculated and configured under the same conditions. From the obtained experimental statistics, one can analyze routing performances and draw some conclusions.

In our work, we randomly sample 40 network topologies using the *Node Placement Tool* in EXata. These topologies of vehicular networks cover 10 different network scales, i.e., the total number of nodes ranges from 20 to 200, with a step length of 20. For each network size, four different topologies were randomly generated according to the random seeds of 1234, 2341, 3412, and 4123, respectively, to avoid the influence of specific network topology structures.

The node mobility model uses the Random Waypoint Model. And the packet size was set as 128 bytes. Other parameters were set as default in EXata 5.3. Under the same network settings, we compare the performance of our HGR strategy with the traditional Euclidean geographic greedy routing strategy (EGR). In this paper, we mainly use routing success rate and routing stretch as two indexes to evaluate the performance of the above routing strategies. And the simulation results are reported in the subsequent two subsections.

4.1. Routing Success Rate

Firstly, we analyzed the routing success rate of HGR and EGR. Routing success rate refers to the ratio of the number of packets that are successfully delivered to the destination to the total number of transmitted packets in the network. In order to quantitatively measure the effectiveness of our proposed routing strategy in mitigating the routing void, experiments were carried out in the 40 randomly sampled vehicular network topologies under the full communication demand matrix, where one unit of packet is assumed to be transmitted between each node pair in the network. We counted the routing success rate of each network scenario, and the results are plotted in Figure 7.

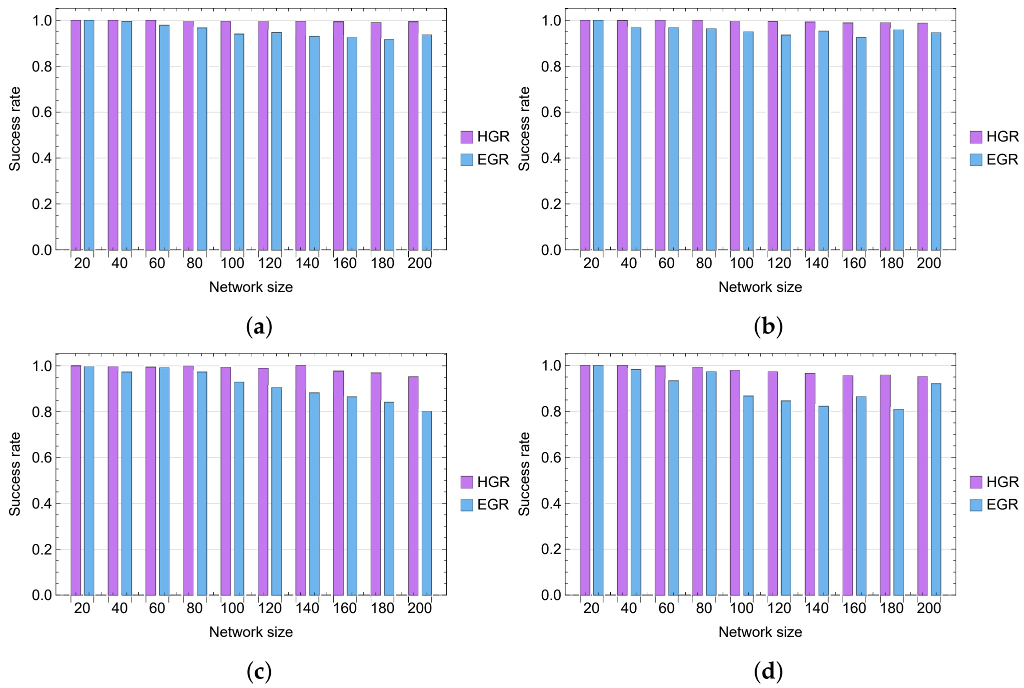


Figure 7. The bar charts of routing success rates comparing the above two routing strategies in different networks. (**a**) Random seed = 1234; (**b**) random seed = 2341; (**c**) random seed = 3412; (**d**) random seed = 4123 .

From Figure 7, we can see that no matter the network size or the topology structure, the routing success rate of the proposed HGR strategy is always higher than that of EGR. Moreover, the routing success rate of HGR is always close to 1. This indicates that our hyperbolic embedding is effective in alleviating routing voids.

As the network scale increases, the routing success rate of EGR presents a more significant descending trend than that of HGR. The performance of HGR will not significantly degrade when the network size increases. Therefore, HGR is scalable, and its application in large-scale networks, like future intelligent vehicular networks, is promising. All in all, our HGR outperforms EGR in terms of routing success rates.

4.2. Routing Stretch

Geographic routing strategies can find nearly optimal paths through greedy forwarding, but this does not guarantee the exact shortest routing paths between them. To quantitatively evaluate the approximation degree of routing path length versus the shortest path length, we define *Stretch* as

$$Stretch = \frac{l}{l_{spr}}, \tag{10}$$

in which the numerator represents the greedy forwarding path length between the source node and the destination using the geographic routing strategy, and the denominator represents the shortest path length on the topology graph between the source node and the destination. Hence, *Stretch* is equal to the ratio of output greedy routing path length to the shortest path length. For a given network topology and a certain greedy routing strategy, the value of *Stretch* between every node pair in the network is no less than 1. The closer the value of *Stretch* is to 1, the similar the output routing path length is to the optimal shortest path. A smaller value of *Stretch* indicates a lower greedy embedding distortion and a better routing performance.

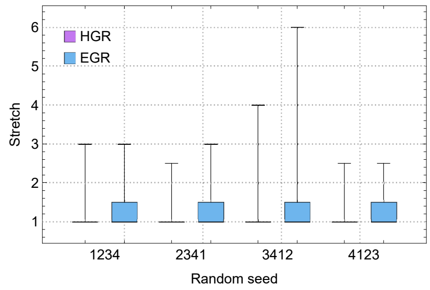
In this experiment, we compare the routing paths obtained from our HGR and the EGR strategy under the same network conditions. The value of *Stretch* between all node pairs in the network are calculated and analyzed, and the distribution charts are shown in Figure 8.

In Figure 8, statistics of *Stretch*s are depicted using box and whisker diagrams. The box and whisker diagrams can graphically depict groups of numerical data through their maximum, minimum, and median, as well as their 25% and 75% quartiles. Figure 8 contains 10 subplots, each of which corresponds to a certain network size. In Figure 8a, when the network size is 20, the minimum, median, and quartiles of *Stretch*s in HGR are all equal to 1 and coincide in the diagram. This demonstrates that HGR can forward packets along the shortest routing paths for most of the node pairs when $n = 20$. In contrast, the EGR strategy can not ensure that the routing paths between all node pairs in the network are strictly the shortest. Moreover, the maximum value of *Stretch* in the EGR strategy is higher than that in HGR, especially under the random seed of 3412, wherein the maximum value is 6. As the network size grows, from Figure 8a to Figure 8j, the maximum, minimum, median, and quartiles of all *Stretch* values remain relatively small. Hence, we can draw the conclusion that the output routing path lengths of HGR are generally very close to the optimal shortest paths. The routing stretch performance of HGR is also more stable than that of EGR, for the overall distribution of *Stretch* values of HGR is relatively concentrated, whereas, in EGR, the box and whisker diagrams have a higher length, suggesting that the distribution of the *Stretch* values are more scattered.

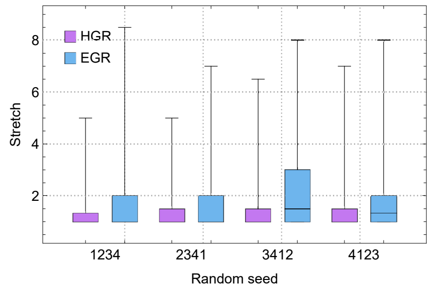
It can be seen from Figure 8 that regardless of the network size and the network topology, compared with the EGR routing strategy, the *Stretch* values of the HGR routing strategy's output greedy routing paths are closer to 1. This is due to the fact that hyperbolic embedding can better approximate the shortest paths on the graph. Therefore, the forwarding path length of the HGR routing strategy is closer to the ideal shortest routing paths and thus achieves better performance than the EGR routing strategy in terms of routing stretch.

To make the above statistics in Figure 8 more intuitive, we also record the experimental data in Table 1. For the decimal values of Stretch with infinite loops, we only keep the first two decimal places by rounding in Table 1.

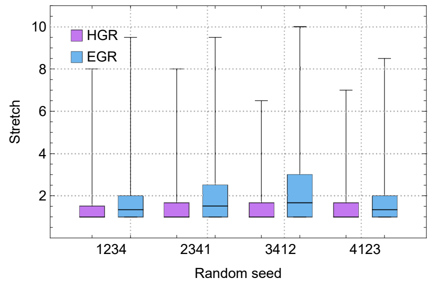
In Table 1, we can see that the minimum *Stretch* values of both HGR and EGR are all 1 under all network scenarios. This yields that geographic routing strategies have the ability to find the shortest routing paths via greedy forwarding, although they do not acquire global topology information. The 25% quartiles of the *Stretch* values in HGR are all 1. But in EGR, under the network size of 180 and 200, there are some cases where the 25% quartiles of the *Stretch* values are 1.33. This implies that HGR can always forward the packet along the shortest routing paths between more than 25% of the node pairs. The median *Stretch* values of HGR are all 1 when the network size is not larger than 60 nodes. However, the median *Stretch* values of EGR are generally more than 1, except in the case that the network size is 20. In addition, the median *Stretch* values of HGR are all smaller than 2, while there are a few cases where the median *Stretch* values of EGR are no less than 2. This indicates that HGR can guarantee that the routing path length between more than half of the node pairs in the network is no more than two times the shortest path. The 75% quartiles of the *Stretch* values in HGR are all below 3. However, in EGR, the 75% quartiles of the *Stretch* values are higher than 3 in the majority of cases. Nevertheless, in the network size of 200 with a random seed of 2341, the 75% quartile is even 5. The maximum of the *Stretch* values in HGR is no larger than that in EGR under the same conditions. Although, along with the increase in the network size, the maximum of the *Stretch* values in HGR shows an upward trend, both the increase rate and speed are much lower than that in EGR. To recap, our HGR strategy can achieve a lower routing stretch, regardless of the network size and network structure.



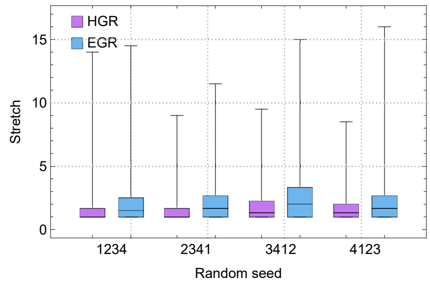
(**a**)



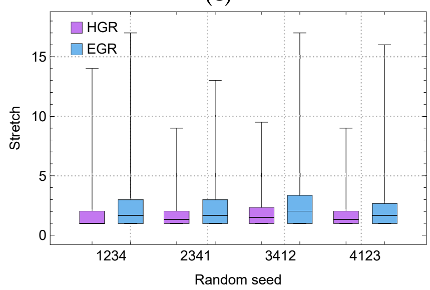
(**b**)



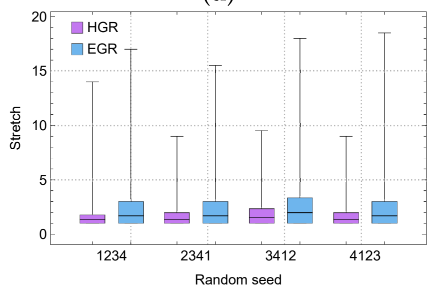
(**c**)



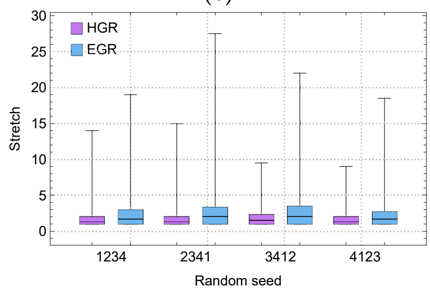
(**d**)



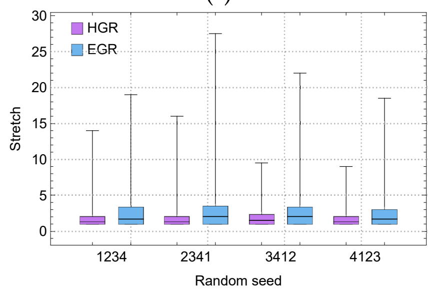
(**e**)



(**f**)

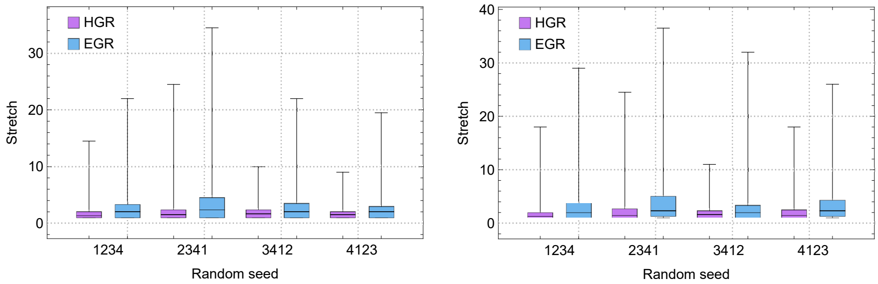


(**g**)



(**h**)

Figure 8. *Cont.*



(i) **(j)**

Figure 8. The box and whisker diagrams of Stretch comparing the above two routing strategies under different network scales. (**a**) n = 20; (**b**) n = 40; (**c**) n = 60; (**d**) n = 80; (**e**) n = 100; (**f**) n = 120; (**g**) n = 140; (**h**) n = 160; (**i**) n = 180; (**j**) n = 200.

Table 1. Routing stretch results.

Network Size	Random Seed	Max		75%		Median		25%		Min	
		HGR	EGR	HGR	EGR	HGR	EGR	HGR	EGR	HGR	EGR
$n = 20$	1234	3.00	3.00	1.00	1.50	1.00	1.00	1.00	1.00	1.00	1.00
	2341	2.50	3.00	1.00	1.50	1.00	1.00	1.00	1.00	1.00	1.00
	3412	4.00	6.00	1.00	1.50	1.00	1.00	1.00	1.00	1.00	1.00
	4123	2.50	2.50	1.00	1.50	1.00	1.00	1.00	1.00	1.00	1.00
$n = 40$	1234	5.00	8.50	1.33	2.00	1.00	1.00	1.00	1.00	1.00	1.00
	2341	5.00	7.00	1.50	2.00	1.00	1.00	1.00	1.00	1.00	1.00
	3412	6.50	8.00	1.50	3.00	1.00	1.50	1.00	1.00	1.00	1.00
	4123	7.00	8.00	1.50	2.00	1.00	1.33	1.00	1.00	1.00	1.00
$n = 60$	1234	8.00	9.50	1.50	2.00	1.00	1.33	1.00	1.00	1.00	1.00
	2341	8.00	9.50	1.67	2.50	1.00	1.50	1.00	1.00	1.00	1.00
	3412	6.50	10.00	1.67	3.00	1.00	1.67	1.00	1.00	1.00	1.00
	4123	7.00	8.50	1.67	2.00	1.00	1.33	1.00	1.00	1.00	1.00
$n = 80$	1234	14.00	14.50	1.67	2.50	1.00	1.50	1.00	1.00	1.00	1.00
	2341	9.00	11.50	1.67	2.67	1.00	1.67	1.00	1.00	1.00	1.00
	3412	9.50	15.00	2.25	3.33	1.33	2.00	1.00	1.00	1.00	1.00
	4123	8.50	16.00	2.00	2.67	1.33	1.67	1.00	1.00	1.00	1.00
$n = 100$	1234	14.00	17.00	2.00	3.00	1.00	1.67	1.00	1.00	1.00	1.00
	2341	9.00	13.00	2.00	3.00	1.33	1.67	1.00	1.00	1.00	1.00
	3412	9.50	17.00	2.33	3.33	1.50	2.00	1.00	1.00	1.00	1.00
	4123	9.00	16.00	2.00	2.67	1.33	1.67	1.00	1.00	1.00	1.00
$n = 120$	1234	14.00	17.00	1.75	3.00	1.33	1.67	1.00	1.00	1.00	1.00
	2341	9.00	15.50	2.00	3.00	1.33	1.67	1.00	1.00	1.00	1.00
	3412	9.50	18.00	2.33	3.33	1.50	2.00	1.00	1.00	1.00	1.00
	4123	9.00	18.50	2.00	3.00	1.33	1.67	1.00	1.00	1.00	1.00
$n = 140$	1234	14.00	19.00	2.00	3.00	1.33	1.67	1.00	1.00	1.00	1.00
	2341	15.00	27.50	2.00	3.33	1.33	2.00	1.00	1.00	1.00	1.00
	3412	9.50	22.00	2.33	3.50	1.50	2.00	1.00	1.00	1.00	1.00
	4123	9.00	18.50	2.00	2.75	1.33	1.67	1.00	1.00	1.00	1.00
$n = 160$	1234	14.00	19.00	2.00	3.33	1.33	1.67	1.00	1.00	1.00	1.00
	2341	16.00	27.50	2.00	3.50	1.50	2.00	1.00	1.00	1.00	1.00
	3412	9.50	22.00	2.33	3.33	1.50	2.00	1.00	1.00	1.00	1.00
	4123	9.00	18.50	2.00	3.00	1.33	1.67	1.00	1.00	1.00	1.00
$n = 180$	1234	14.50	22.00	2.00	3.33	1.33	2.00	1.00	1.00	1.00	1.00
	2341	24.50	34.50	2.33	4.50	1.50	2.33	1.00	1.33	1.00	1.00
	3412	10.00	22.00	2.33	3.50	1.67	2.00	1.00	1.00	1.00	1.00
	4123	9.00	19.50	2.00	3.00	1.50	2.00	1.00	1.00	1.00	1.00
$n = 200$	1234	18.00	29.00	2.00	3.75	1.33	2.00	1.00	1.00	1.00	1.00
	2341	24.50	36.50	2.67	5.00	1.50	2.33	1.00	1.33	1.00	1.00
	3412	11.00	32.00	2.33	3.33	1.67	2.00	1.00	1.00	1.00	1.00
	4123	18.00	26.00	2.50	4.33	1.50	2.33	1.00	1.33	1.00	1.00

4.3. Scalability

In order to further evaluate the scalability of the HGR routing strategy and test the actual effect of the HGR strategy in networks of different scales, simulation experiments were conducted to study the changing trend of routing computation time with the increase in network scale. Meanwhile, control experiments were conducted under the same conditions to compare the HGR routing strategy with the EGR routing strategy.

The experiments were carried out on a personal PC manufactured by HUAWEI of China with Intel Core i5 CPU , 2.40 GHz frequency, 16 GB memory, and a 64-bit Windows 10 operating system. The routing algorithms were compiled using Mathematica 12.1 software. Experiments were carried out in 40 different networks, covering 10 different sizes of networks, and the number of nodes is from 20 to 200 with a step size of 20. For each size of the network, four different random network topologies were generated. The average value of the routing computation time of the above two routing strategies in different scale networks was calculated, respectively. The curve of routing computation time with the increase in the total number of network nodes is plotted in Figure 9.

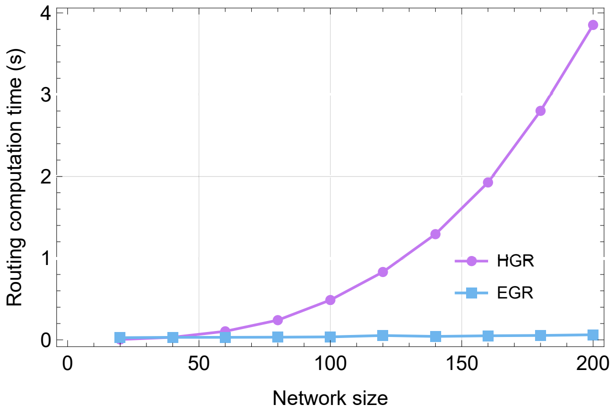


Figure 9. Routing computation time in networks of different scales.

From Figure 9, we can see that with the increase in the network size, the routing computation time of both HGR and EGR increases. However, the increased speed of EGR is slower than that of HGR. This is because EGR uses the original node position information in the Euclidean space, whereas additional embedding in hyperbolic space is required in HGR. And the time complexity of the embedding algorithm in HGR is $O(n^2)$. Considering the enhancement in the routing success rate and routing stretch, the time sacrifice is worthwhile, for even in a huge network with 200 nodes, the routing computation time is within 4 s. The computation time can be further reduced by using high-performance computers.

5. Conclusions

In this paper, a hyperbolic-embedding-aided geographic routing strategy (HGR) was proposed to tackle the problem of routing voids in intelligent vehicular networks. Unlike traditional geographic routing strategies, HGR transfers the metric space of greedy forwarding from the classical Euclidean space into a two-dimensional Poincare hyperbolic space by Laplacian embedding techniques. The effectiveness of the proposed HGR design was demonstrated through multiple sets of comparative simulation experiments. Simulation results have shown that the proposed HGR strategy can significantly improve the routing success rate with a much lower routing stretch compared to the Euclidean geographic greedy routing strategy.

Author Contributions: Conceptualization, Y.P. and N.L.; methodology, Y.P.; software, Y.P.; validation, Y.P. and N.L.; formal analysis, Y.P.; investigation, Y.P.; resources, N.L.; writing—original draft preparation, Y.P.; writing—review and editing, N.L.; visualization, Y.P. and N.L.; supervision, N.L.; project administration, N.L.; funding acquisition, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Outstanding Doctoral Dissertation Support Fund of Air Force Engineering University, grant number 081021011.

Data Availability Statement: All data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cheng, X.; Chen, C.; Zhang, W.; Yang, Y. 5G-Enabled Cooperative Intelligent Vehicular (5GenCIV) Framework: When Benz Meets Marconi. *IEEE Intell. Syst.* **2017**, *32*, 53–59. [CrossRef]
2. Fu, H.; Guan, J.; Jing, F.; Wang, C.; Ma, H. A real-time multi-vehicle tracking framework in intelligent vehicular networks. *China Commun.* **2021**, *18*, 89–99. [CrossRef]
3. Liu, L.; Chen, C.; Wang, B.; Zhou, Y.; Pei, Q. An Efficient and Reliable QoF Routing for Urban VANETs with Backbone Nodes. *IEEE Access* **2019**, *7*, 38273–38286. [CrossRef]
4. Zhao, L.; Bi, Z.; Lin, M.; Hawbani, A.; Shi, J.; Guan, Y. An intelligent fuzzy-based routing scheme for software-defined vehicular networks. *Comput. Netw.* **2021**, *187*, 107837. [CrossRef]
5. Fatemidokht, H.; Rafsanjani, M.K.; Gupta, B.B.; Hsu, C.H. Efficient and Secure Routing Protocol Based on Artificial Intelligence Algorithms with UAV-Assisted for Vehicular Ad Hoc Networks in Intelligent Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4757–4769. [CrossRef]
6. Ma, B.; Ren, Z.; Cheng, W. Traffic Routing-Based Computation Offloading in Cybertwin-Driven Internet of Vehicles for V2X Applications. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4551–4560. [CrossRef]
7. Chen, C.; Liu, L.; Qiu, T.; Jiang, J.; Pei, Q.; Song, H. Routing with Traffic Awareness and Link Preference in Internet of Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 200–214. [CrossRef]
8. Li, P.; Zeng, Y.; Li, C.; Chen, L.; Wang, H.; Chen, C. A Probabilistic Broadcasting Scheme for Emergent Message Dissemination in Urban Internet of Vehicles. *IEEE Access* **2021**, *9*, 113187–113198. [CrossRef]
9. Dijkstra, E.W. A Note on Two Problems in Connection with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
10. Bang-Jensen, J.; Gutin, G. Section 3.3.4: The Bellman-Ford-Moore algorithm. In *Digraphs: Theory, Algorithms and Applications*, 2nd ed.; Springer: London, UK, 2010; pp. 97–99.
11. Floyd, R.W. Algorithm 97: Shortest Path. *Commun. ACM* **1962**, *5*, 345. [CrossRef]
12. Wahid, I.; Tanvir, S.; Ahmad, M.; Ullah, F.; AlGhamdi, A.S.; Khan, M.; Alshamrani, S.S. Vehicular Ad Hoc Networks Routing Strategies for Intelligent Transportation System. *Electronics* **2022**, *11*, 2298. [CrossRef]
13. Liu, L.; Chen, C.; Ren, Z.; Qiu, T.; Song, H. A Connectivity Aware Transmission Quality Guaranteed Geographic Routing in Urban Internet of Vehicles. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6. [CrossRef]
14. Liu, L.; Chen, C.; Ren, Z.; Yu, F.R. An Intersection-Based Geographic Routing with Transmission Quality Guaranteed in Urban VANETs. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6. [CrossRef]
15. Chen, C.; Li, H.; Li, X.; Zhang, J.; Wei, H.; Wang, H. A geographic routing protocol based on trunk line in VANETs. *Digit. Commun. Netw.* **2021**, *7*, 479–491. [CrossRef]
16. Liu, L.; Chen, C.; Ren, Z.; Qiu, T.; Yang, K. A Delay-Aware and Backbone-Based Geographic Routing for Urban VANETs. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6. [CrossRef]
17. Jiang, S.; Huang, Z.; Ji, Y. Adaptive UAV-Assisted Geographic Routing with Q-Learning in VANET. *IEEE Commun. Lett.* **2021**, *25*, 1358–1362. [CrossRef]
18. Luo, L.; Sheng, L.; Yu, H.; Sun, G. Intersection-Based V2X Routing via Reinforcement Learning in Vehicular Ad Hoc Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 5446–5459. [CrossRef]
19. Al-Subhi, T.; Arafeh, B.; Alzeidi, N.; Day, K.; Touzene, A. A Void Avoidance Scheme for Grid-Based Multipath Routing in Underwater Wireless Sensor Networks. *Wirel. Sens. Netw.* **2018**, *10*, 131–156. [CrossRef]
20. Denardin, G.W.; Barriquello, C.H.; Campos, A.; Prado, R.N.D. A geographic routing hybrid approach for void resolution in wireless sensor networks. *J. Syst. Softw.* **2011**, *84*, 1577–1590. [CrossRef]
21. Ullah, A.; Yao, X.; Shaheen, S.; Ning, H. Advances in Position Based Routing Towards ITS Enabled FoG-Oriented VANET—A Survey. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 828–840. [CrossRef]
22. Boguñá, M.; Bonamassa, I.; Domenico, M.D.; Havlin, S.; Krioukov, D.V.; Serrano, M.Á. Network geometry. *Nat. Rev. Phys.* **2020**, *3*, 114–135. [CrossRef]
23. Krioukov, D.V.; Papadopoulos, F.; Kitsak, M.; Vahdat, A.; Boguñá, M. Hyperbolic Geometry of Complex Networks. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2010**, *82 Pt 2*, 036106. [CrossRef]
24. Boguñá, M.; Papadopoulos, F.; Krioukov, D.V. Sustaining the Internet with Hyperbolic Mapping. *Nat. Commun.* **2010**, *1*, 62. [CrossRef]
25. Sarkar, R. Low Distortion Delaunay Embedding of Trees in Hyperbolic Plane. In Proceedings of the International Symposium Graph Drawing and Network Visualization, Eindhoven, The Netherlands, 21–23 September 2011.

26. Almagro, P.; Boguñá, M.; Serrano, M.Á. Detecting the ultra low dimensionality of real networks. *Nat. Commun.* **2021**, *13*, 6096. [CrossRef]
27. Kleinberg, R.D. Geographic Routing Using Hyperbolic Space. In Proceedings of the IEEE INFOCOM 2007—26th IEEE International Conference on Computer Communications, Anchorage, AK, USA, 6–12 May 2007; pp. 1902–1909.
28. Cvetkovski, A.; Crovella, M. Hyperbolic Embedding and Routing for Dynamic Graphs. In Proceedings of the IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 19–25 April 2009; pp. 1647–1655.
29. Zheng, Z.; Bashir, A.K. Graph-Enabled Intelligent Vehicular Network Data Processing. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 4726–4735. [CrossRef]
30. Zhu, L.; Gu, W.; Yi, J.; Luan, T.H.; Li, C. On Mobility-Aware and Channel-Randomness-Adaptive Optimal Neighbor Discovery for Vehicular Networks. *IEEE Internet Things J.* **2021**, *8*, 6828–6839. [CrossRef]
31. Yin, X.; Ma, X.; Trivedi, K.S. An Interacting Stochastic Models Approach for the Performance Evaluation of DSRC Vehicular Safety Communication. *IEEE Trans. Comput.* **2013**, *62*, 873–885. [CrossRef]
32. Aldecoa, R.; Orsini, C.; Krioukov, D.V. Hyperbolic graph generator. *arXiv* **2015**, arXiv:1503.05180.
33. Papadopoulos, F.; Boguñá, M.; Krioukov, D.V. Popularity versus similarity in growing networks. *Nature* **2011**, *489*, 537–540. [CrossRef] [PubMed]
34. Bläsius, T.; Friedrich, T.; Krohmer, A.; Laue, S. Efficient Embedding of Scale-Free Graphs in the Hyperbolic Plane. *IEEE/ACM Trans. Netw.* **2018**, *26*, 920–933. [CrossRef]
35. Wu, Z.; Di, Z.; Fan, Y. An Asymmetric Popularity-Similarity Optimization Method for Embedding Directed Networks into Hyperbolic Space. *Complexity* **2020**, *2020*, 8372928. [CrossRef]
36. Alanis-Lobato, G.; Mier, P.; Andrade, M. Efficient embedding of complex networks to hyperbolic space via their Laplacian. *Sci. Rep.* **2016**, *6*, 30108. [CrossRef] [PubMed]
37. Jiang, H.; Li, L.; Zeng, Y.; Fan, J.; Shen, L. Low-Complexity Hyperbolic Embedding Schemes for Temporal Complex Networks. *Sensors* **2022**, *22*, 9306. [CrossRef]
38. Muscoloni, A.; Thomas, J.M.; Ciucci, S.; Bianconi, G.; Cannistraci, C.V. Machine learning meets complex networks via coalescent embedding in the hyperbolic space. *Nat. Commun.* **2017**, *8*, 1615. [CrossRef]
39. García-Pérez, G.; Allard, A.; Serrano, M.Á.; Boguñá, M. Mercator: Uncovering faithful hyperbolic embeddings of complex networks. *New J. Phys.* **2019**, *21*, 123033. [CrossRef]
40. Jankowski, R.; Allard, A.; Bogun'a, M.; Serrano, M.Á. The D-Mercator method for the multidimensional hyperbolic embedding of real networks. *Nat. Commun.* **2023**, *14*, 7585. [CrossRef] [PubMed]
41. Nie, Q.; Jiang, H.; Zhong, S.D.; Wang, Q.; Wang, J.J.; Wang, H.; Wu, L.H. Joint Detection of Community and Structural Hole Spanner of Networks in Hyperbolic Space. *Entropy* **2022**, *24*, 894. [CrossRef]
42. Sun, L.; Zhang, Z.; Zhang, J.; Wang, F.; Du, Y.; Su, S.; Yu, P.S. PERFECT: A Hyperbolic Embedding for Joint User and Community Alignment. In Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 17–20 November 2020; pp. 501–510. [CrossRef]
43. Kitsak, M.; Voitalov, I.; Krioukov, D. Link prediction with hyperbolic geometry. *Phys. Rev. Res.* **2020**, *2*, 043113. [CrossRef]
44. Moshiri, M.; Safaei, F. Application of hyperbolic geometry of multiplex networks under layer link-based attacks. *Chaos: Interdiscip. J. Nonlinear Sci.* **2022**, *32*, 021105 . [CrossRef] [PubMed]
45. Heszberger, Z. Hyperbolic trees for efficient routing computation. *J. Supercomput.* **2022**, *78*, 15250–15268. [CrossRef]
46. Stai, E.; Sotiropoulos, K.; Karyotis, V.; Papavassiliou, S. Hyperbolic Embedding for Efficient Computation of Path Centralities and Adaptive Routing in Large-Scale Complex Commodity Networks. *IEEE Trans. Netw. Sci. Eng.* **2017**, *4*, 140–153. [CrossRef]
47. Bläsius, T.; Friedrich, T.; Katzmann, M.; Krohmer, A. Hyperbolic Embeddings for Near-Optimal Greedy Routing. *J. Exp. Algorithmics (JEA)* **2020**, *25*, 1–18. [CrossRef]
48. Sulyok, B.; Palla, G. Greedy routing optimisation in hyperbolic networks. *Sci. Rep.* **2023**, *13*, 23026. [CrossRef]