IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 3, 2020, accepted January 12, 2020, date of publication January 20, 2020, date of current version January 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967898

Vehicular Delay Tolerant Network Routing Algorithm Based on Bayesian Network

JIAGAO WU, (Member, IEEE), YAHANG GUO, HONGYU ZHOU, LU SHEN, AND LINFENG LIU

School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing 210023, China

Corresponding author: Jiagao Wu (jgwu@njupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872191 and Grant 41571389, in part by the Open Research Fund from Key Laboratory of Computer Network and Information Integration (SEU), Ministry of Education, China, under Grant K93-9-2014-05B, and in part by the Scientific Research Foundation of NUPT under Grant NY214063.

ABSTRACT Delay Tolerant Networks (DTNs) are novel wireless mobile networks, which suffer from frequent disruption, high latency, and the lack of a complete path from source to destination. Vehicular Delay Tolerant Network (VDTN) is a special type of DTNs with vehicles as nodes. In VDTN, most nodes have specific movement patterns, however, traditional routing algorithms in DTNs do not take this characteristic into considerations very well. In this paper, a new routing algorithm based on Bayesian Network (BN) is proposed to construct the prediction model, which intends to predict the movement patterns of nodes in the real VDTN scenarios. Firstly, a comprehensive BN model is established, where more attributes of nodes are selected to improve the accuracy of the model prediction. Then, considering the complexity of the structure learning problem of BN, a novel structure learning algorithm, K2 algorithm based on Genetic Algorithm (K2-GA), is proposed to search the optimal BN structure efficiently. At last, Junction Tree Algorithm (JTA) is adopted in the inference of BN, which can accelerate the inference process through variable elimination and calculation sharing for large scale BN. The simulation results show that the proposed VDTN routing algorithm based on the BN model can improve the delivery ratio with a minor forwarding overhead.

INDEX TERMS Vehicular delay tolerant network, routing algorithm, Bayesian network, genetic algorithm, optimization.

I. INTRODUCTION

Traditional networks rely on end-to-end paths from source to destination and special network protocols (such as TCP/IP or UDP) to transmit data in minimal delay and high reliability. However, this scheme does not work once the end-to-end paths are broken by sudden and tragic disasters including earthquakes, tsunamis, etc. Hence, Delay Tolerant Networks (DTNs) are proposed to cope with the lack of connectivity between source and destination [1]. Such networks have been applied in various fields, including the interplanetary internets [2], underwater sensor networks [3], mobile ad-hoc networks [4] and wireless sensor networks [5], etc. In DTNs, the store-carry-forward strategy is often adopted to deliver messages [6]. In this routing scheme, messages are carried by the source or intermediate nodes, which will keep moving until they meet the destination node or other nodes having the capability to deliver the messages as relays, and then transmit the messages to them. Vehicular Delay Tolerant Network (VDTN) is a new type of wireless network derived from the combination of DTNs and vehicular network [7]. It takes advantage of the opportunities that vehicles meet each other to transmit messages. Due to the high-speed movement of vehicles, the contact time of vehicles is short and limited. Therefore, a novel routing protocol with optimal performance is required to solve the problem.

The well-known routing algorithms in DTNs include Epidemic [8], Spray and Wait [9], Prophet [10], etc. Epidemic algorithm is based on flooding strategy, which is guaranteed to get the optimal path for messages delivery with the shortest delay, because it utilizes all available communication paths to deliver messages. However, flooding in the network may result in tremendous congestion. Although Spray and Wait improves the Epidemic algorithm through limiting the

The associate editor coordinating the review of this manuscript and approving it for publication was Tariq Umer.

number of replicas of a given message, the delivery ratio is also limited and the delay is increased significantly. Prophet algorithm takes the historical information of nodes movement into account to determine the likelihood that a node will encounter other nodes. Nevertheless, in VDTN, vehicles usually have specific movement patterns, (e.g., public transport buses are governed by route and time schedules, personal cars tend to follow regular trajectories, and taxis reflect the hot spots of individual mobility, etc.), but Prophet does not take the vehicle movement patterns into consideration very well, and Community-based Bus System (CBS) is only for public buses [11]. With the popularity of machine learning in recent years, many machine learning algorithms have been applied to DTNs such as decision tree [12], reinforcement learning [13], [14], Artificial Neural Network (ANN) [15] and Naive Bayesian (NB) classification model [16], [17], etc. However, decision tree and ANN are prone to the over-fit problem, and reinforcement learning may result in a large number of copies in DTNs. NB classification model is a simple and effective classification model, but its conditional independence assumption makes it unable to express the dependency among attributes and reduces the classification accuracy. On the contrary, the Bayesian Network (BN) [18] represents a set of attributes and their conditional dependencies via a probabilistic graphical model, thus can improve the accuracy of inference significantly. According to our previous work [19], a routing algorithm based on BN was proposed for VDTN, which utilizes the BN model to preserve the dependency of the attributes of node (e.g., location, time, contact, etc.) and obtain the movement patterns. However, BN structure learning and inference algorithm proposed in [19] are preliminary yet. In order to obtain the optimal structure of BN, more advanced structure learning algorithms need to be adopted. The K2 algorithm [20] has been proven to be efficient and accurate for BN structure learning. But, for searching the global optimal BN structure, K2 algorithm need to iterate all the attribute variables ordering, which is a NP-hard problem. Therefore, in this paper, we propose a new structure learning algorithm called K2-GA that combines K2 algorithm with Genetic Algorithm (GA) to search the optimal BN structure efficiently. Besides, there are many classical exact inference algorithms for BN, among which Junction Tree Algorithm (JTA) is the fastest and most widely used for BN [21]. We choose the JTA to calculate the probability according to the BN model. Furthermore, the reward mechanism based on the ant colony algorithm in [19] may cause the reward value of the source node to be much higher than that of all relay nodes, which causes the source node will no longer transmit any message to the relay nodes. Therefore, we also improve the reward mechanism in this paper.

The main contributions of this paper are as follows:

1) A new BN model is proposed, where more attributes of nodes are introduced to predict the delivery ability of nodes accurately. These attributes are supposed to be able to reflect the movement patterns, which will reduce the number of copies of messages in VDTN. In addition, the reward mechanism based on the ant colony algorithm is improved.
2) A new structure learning algorithm, named K2-GA, is proposed, which combines K2 algorithm with genetic algorithm to search the optimal BN structure efficiently.
3) The JTA algorithm is adopted, which can accelerate the inference process through variable elimination and calculation sharing, and is more feasible for inference based on the joint distribution in large scale BN.

The rest of this paper is organized as follows: Section II presents a brief overview of related work. Section III provides a detailed description of our BN model in VDTN. Section IV presents a comprehensive illustration of K2-GA algorithm. The JTA algorithm is introduced in Section V. In Section VI, the simulation results are discussed, and finally, the paper is concluded in Section VII.

II. RELATED WORK

Delay tolerant networks are wireless mobile networks, which lack of continuous connections, but only exist opportunistically connections among nodes. DTNs are intended to deal with scenarios involving intermittent connectivity between adjacent nodes, lack of contemporaneous end-to-end links, exceptionally high delays and error-rates. To solve these problems, many DTNs routing algorithms have been proposed. These routing protocols are generally divided into two categories: forwarding-based and replication-based. Forwarding-based routings can conserve resources, because only one copy of a message exists in the network. However, the performance of the routing scheme is poor in latency and delivery ratio, because it does not fully utilize the message buffer of nodes. First Contact is a forwarding-based routing protocol where nodes only forward the message copies to the first encounter node [22]. Direct Delivery routing protocol forwards messages only when the destination node is encountered [23]. Although reducing the excrescent message copies and saving the buffer, the forwarding-based routing scheme limits the message delivery.

Replication-based routings may increase the delivery ratio and reduce the delivery delay as much as possible, because multiple copies of a message exist in the network. However, this scheme will consume network resources and increase the overhead. Epidemic is one of the classic flooding routings for DTNs, which replicates message copies for all contacted nodes until the messages reach their destinations [8]. However, massive network resources are wasted by delivering the redundant copies of messages. This strategy can achieve the highest message delivery rate and the lowest delivery delay when nodes have large message buffer and the network has high bandwidth. The Spray and Wait is an improvement of the Epidemic algorithm, which limits the number of replication copies to reduce the overhead of delivering messages [9]. In Spray and Wait, a binary quota allocation scheme is usually adopted to allocate half of the quota from the current message to a new replicated message in spray phase, which has been proved to be effective for homogeneous DTNs where contact rate between any pair of nodes is nearly equal. All above algorithms are based on the assumption that the movement of the node is completely random. However, in real scenarios, the movement of nodes is often predictable. Prophet routing continuously estimates the probability of successful message forwarding according to the historical information of the encounter and transmission [10]. Copies of the message may be replicated to the optimal neighbor node that has higher delivery probability.

In [11], a Community-based Bus System (CBS) is proposed as routing backbone of VDTN. CBS is composed of a community-based backbone and a two-level (i.e., the inter-community level and the intra-community level) routing scheme over the backbone to support message delivery to both mobile vehicles and specific locations/areas. Despite this, CBS is only for bus system and just considers the movement pattern of buses.

With the boom of machine learning, many machine learning methods have been applied to DTNs. As mentioned above, the node movement is not completely random and predictable. Routings based on the machine learning can make full use of network topology and movement patterns and provide link state information for message delivering.

In [12], a Machine Learning-based Prophet routing protocol (MLProph) is proposed, which makes use of neural network and decision tree to train itself based on past network routing data and some contextual parameters such as hop count, buffer capacity, speed, energy, and number of successful deliveries. Based on these, the probability that a node is selected as next forwarder of a message is determined. However, the decision tree and neural network are prone to the over-fit problem.

Delay Tolerant Reinforcement-Based (DTRB) routing [13] is a replication-based routing protocol for DTNs. It utilizes multi-agent reinforcement learning techniques to learn about routes in the network and forwards or replicates messages that produce the best reward. In the DTRB framework, rewards are associated with the time of the messages reaching to the destination. It is shown that DTRB improves the delivery ratio in dense population areas. However, using Q-learning in DTNs may suffer a large penalty, because it may produce a positive bias by using the maximum Q-value as the approximation of the maximum expected rewards.

Li *et al.* [14] proposed a reinforcement-learning-based hierarchical protocol called QGrid to improve the message deliver ratio with minimum possible delay and hops. Q-learning is used to find the optimal action-selection strategy for Markov Decision Process (MDP) even when the agent does not have prior knowledge about the effect of its actions on the environment. However, the QGrid protocol does not take into account the topology characteristics of the dynamic network and the attributes of the vehicules, which also affect the efficiency of the routing protocol.

Segundo *et al.* [15] proposed an ANN model to predict the next node and the moment of contact. The predictor based on an ANN trained with historical data extracted from synthetic traces provides a high hit rate for next node and next moment of contact. Nevertheless, the initialized weights and threshold of ANN are difficult to be configured, and this model requires repeated training to determine the network structure and various parameters.

Ahmed and Kanhere [16] introduced a Naive Bayesian classifier in DTNs, and the simulation suggests that the NB-based routing algorithm can achieve a great performance. However, NB is based on the conditional independence assumption. It does not take the dependency among attributes into account, but such dependency certainly exits in the real environment.

In [17], an architecture of a machine learning based router is proposed for delay tolerant space networks, and two machine learning methods are used to supplement the routing decisions of the Contact Graph Routing (CGR) algorithm, where Naive Bayesian classifier is used for the prediction of reliability of each candidate relay node and Q-learning routing is used for the estimation of the end-to-end delivery delay of routing. But, the delay tolerant space network is quite different from VDTN, and both these two machine learning methods in CGR need to be improved further.

Liang *et al.* [18] proposed the concept of adaptive routing based on BN according to application characteristics in DTN, which means that before transmitting data, a suitable routing protocol is decided to use in order to meet application requirements and to achieve better network performances. However, their BN model focuses on the choice of different routing protocols, not on the scheme of message forwarding.

In [19], a VDTN routing algorithm is proposed, where Bayesian network is adopted to evaluate the delivery level of nodes. However, the BN structure learning method is based on the mutual information of each pair of attributes, which is greedy algorithm and can only obtain the local optimal structure of BN. The inference method in the paper is based on the joint distribution, which is not feasible when there are many variables in BN. Besides, its reward mechanism may cause the reward value of the source node to be much higher than that of all relay nodes after the algorithm runs for a period of time, which causes the source node will no longer transmit any message to the relay nodes.

Therefore, in this paper, we propose the K2-GA algorithm to search for the global optimal network, utilize JTA inference algorithm to accelerate the inference procedure in large scale BN, and improve the reward mechanism to alleviate the above problems.

III. SYSTEM MODEL

The Bayesian network is a graphical model that encodes the joint probabilistic relationships among a set of random variables. It is comprised of two components. The one is the Directed Acyclic Graph (DAG), i.e., $G = (V, E)$, where V is the set of nodes representing the random variables, and E is the set of ordered pairs of distinct elements of V, which represents the dependent relationship among these variables.

TABLE 1. List of notations.

Symbol	Description	Symbol	Description
X	attributes set of nodes in VDTN	x	an instance of X
X_1	attribute of Region Code	$R_1, R_2, \ldots, R_{k_1}$	discrete values of X_1
X_2	attribute of Time Slot	$T_1, T_2, \ldots, T_{k_2}$	discrete values of X_2
X_3	attribute of Movement Angle	$A_1, A_2, \ldots, A_{k_3}$	discrete values of X_3
X_4	attribute of Average Inter-contact Interval	$I_1, I_2, \ldots, I_{k_4}$	discrete values of X_4
X_5	attribute of Velocity	$V_1, V_2, \ldots, V_{k_5}$	discrete values of X_5
X_6	attribute of Path Id	$U_1, U_2, \ldots, U_{k_6}$	discrete values of X_6
X_7	attribute of Delivery Level	$L_1, L_2, \ldots, L_{k_7}$	discrete values of X_7
n_i	relay node for transmitting messages	n_s	source node of a message
m	the *m*th message delivered through n_i	n_d	destination node of a message
O_m	transmission path of the *m*th message	N_m	total number of nodes in O_m
$\theta_{n_i,m}$	order of n_i in transmission path O_m	ψ_d	delivery reward
$\psi_{n_i}(t)$	reward of n_i at time slot t	ψ_c	contact reward
$r_{n_i}(t)$	number of delivered messages through n_i during time slot t	ρ	ageing coefficient
$c_{n_i}(t)$	number of other nodes encountering with n_i during time slot t	$\hat{X} = < X_{i_1}, X_{i_2}, \ldots, X_{i_7} >$	attribute variables ordering
$\pi(X_{i_k})$	candidate parents set of attribute X_{i_k}	$\pi^*(X_{i_k})$	the best parents set of X_{i_k}
D	historical data set	q_k	number of all possible states of $\pi(X_{i_k})$
d_{kj*}	number of instances where X_{i_k} is any value and $\pi(X_{i_k})$ is in the *j*th state in historical data set D	s_k	number of states of X_{i_k}
d_{kjv}	number of instances where X_{i_k} is v and $\pi(X_{i_k})$ is in the *j*th state in historical data set D	$f(X_{i_k}, \pi(X_{i_k}))$	scoring function of X_{i_k}
$f^*(X_{i_k})$	the best scoring of X_{i_k}	$\hat{X}^*$	the best attribute variables ordering
$G(\hat{X})$	BN structure when attribute variables ordering is $\hat{X}$	$F(\hat{X})$	score of $G(\hat{X})$
w	the maximum number of iterations	z	size of population
α	crossover probability	β	mutation probability
H_j	the *j*th population	$H_{i,j}$	the *i*th chromosome of the *j*th population
$G^*(\hat{X}^*)$	the best BN structure	$F^*(\hat{X}^*)$	the best score of BN structure

The elements of E are called edges (or arcs), and a directional edge indicates the dependency between two variables. The other component is a set of Conditional Probability Tables (CPTs) that quantify the joint probability distributions of nodes in G. As an important method in machine learning, BN is usually used as a classifier. Thus, in this paper, we propose a classification model based on BN to estimate the ability of nodes for delivery messages in VDTN. In order to describe our model and algorithm in clarity, we provide a list of notations as Table 1.

In VDTN, when the source node n_s generates a message and wants to send the message to the destination node n_d, it is obvious that there exists more than one path consisting of a series of nodes through which the message can be transmitted from the node n_s to n_d. Then, the question is how to find the shortest path. It has been seen that Epidemic routing algorithm based on flooding strategy can always find the optimal path for message delivery with the shortest delay, because all available delivery paths are used to transmit the message. However, a large number of redundant message copies consume network resources and increase the overhead. Since the relay nodes belonging to the shortest path often have great ability to transmission messages, in this section, we propose a classifier with BN model to evaluate the ability of node for transmitting messages to destination. Firstly we introduce the attributes selection to determine the variables for constructing the Bayesian network. Besides these observable variables, we also define a variable to measure the transmission ability of nodes, which is updated dynamically based on the ant colony optimization method [24].

The details of attributes selection and classification metric are as follows.

A. ATTRIBUTES SELECTION

The nodes in the BN model are the attributes of nodes in VDTN, which are closely related to the forwarding of messages. Therefore, we need to choose the appropriate attributes before we collect the training data. The selected attributes in this paper can be described by a vector $X = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7\}$

- $X_1 = \textit{Region Code} < R_1, R_2, \ldots, R_{k_1} >$, representing the area in which the node in VDTN is located when its message is forwarded. The entire area is divided into k_1 rectangular grids, and each grid has the same size and distinct identifier. In this paper, the entire area is 35 km × 25 km, then it is divided into 35 grids, and the size of each grid is 5 km × 5 km.
- $X_2 = \textit{Time Slot} < T_1, T_2, \ldots, T_{k_2} >$, indicating the time slot when message is forwarded. This metric has been selected because vehicle has different movement patterns in different time periods of one day. We are supposed to discrete time periods according to the availability of historical data. In this paper, the granularity of the time slots is 30 min. Therefore, there are 24 time slots to distinguish the different periods since we focus on a 12 hours period from 7 am-7 pm.

- $X_3 = Movement\ Angle < A_1, A_2, \ldots, A_{k_3} >$, representing the movement direction of vehicle at the time of message forwarding. Since the vehicle can move in different directions in geography, in this paper, we divide all feasible movement angles into 8 discrete directions (i.e., north, south, east, west, northeast, southeast, northwest and southwest), that is, the granularity of the movement angle is $\frac{\pi}{4}$.
- $X_4 = Average\ Inter\text{-}contact\ Interval < I_1, I_2, \ldots, I_{k_4} >$, indicating the average interval of the encounter among nodes in VDTN. In this paper, based on the historical data set, we let the maximum value of the average inter-contact interval be 3000 s and the granularity be 300 s. The average inter-contact interval is divided into 10 slots. For a vehicle node, the smaller the value is, the better the forwarding ability the node gets.
- $X_5 = Velocity < V_1, V_2, \ldots, V_{k_5} >$, signifying the moving velocity of the node in VDTN. According to the historical data set, we get that the maximum velocity is 120 km/h of all vehicles, and we let the granularity be 24 km/h. Therefore, there are 5 velocity grades.
- $X_6 = Path\ Id < U_1, U_2, \ldots, U_{k_6} >$, indicating the routes on which the vehicle is traveling. It is a common situation that vehicle usually has a fixed route and specific movement patterns in VDTN, e.g. bus routes. Different vehicles may travel along the same route, so they have the same path Id. the attribute can preserve the specific behavior of vehicles.
- $X_7 = Delivery\ Level < L_1, L_2, \ldots, L_{k_7} >$, signifying the ability of transmitting messages to destination. The node with higher delivery level has greater ability to transmit messages successfully.

Among the above-mentioned attributes, X_1, X_2, X_3, X_4, X_5 and X_6 are observable attributes, which are used as the evidence variables. X_7 is an unobservable attribute, representing the ability of transmitting messages to the destination, and is used as the inference key.

B. CLASSIFICATION OF X_7

Ant colony algorithm is a probabilistic algorithm to find the optimal path. In [19], we apply the algorithm to distribute every node a reward value, which plays a role of metric signifying the ability for message delivery in DTNs. A message is generated by a source node, forwarded through the relay nodes, and finally delivered to the destination. At this time, the destination node broadcasts a small ACK message containing all Ids of relay nodes in the delivery path. And the node whose Id is included will get a certain reward named delivery reward, once the ACK message is received.

Let m indicate the mth message successfully delivered through node n_i, O_m and ψ_d be the routing path and the delivery reward of the message, respectively. Moreover, suppose $\theta_{n_i,m}$ denotes the order of node n_i in the routing path O_m and N_m represents the total number of hops of O_m, i.e., $N_m = |O_m| - 1$. In this paper, the delivery reward ψ_d is distributed to each relay node according to the order of the node in O_m. The closer the relay node is to the destination node, that is, the larger the order of the relay node in the routing path, the greater the delivery reward it will obtain. Since $\sum_{n_i \in O_m} \theta_{n_i,m} = 1 + 2 + \cdots + N_m = [N_m(N_m + 1)]/2$, the node n_i should be distributed the certain delivery reward by $2\theta_{n_i,m}\psi_d/[N_m(N_m+1)]$. Therefore, we can see that the sum of delivery reward of all relay nodes is equal to ψ_d. By this method, the delivery reward assigned to the source node n_s is less than that of other nodes in O_m, which alleviates the problem in [19] and increases the robustness of the reward mechanism greatly. In addition, the relay node with higher delivery reward is closer to the destination node, thus has a higher probability for message forwarding in a shorter path.

For example, as shown in Fig. 1, the destination node n_d has received two messages from node n_s at time slot T_1, where the two messages are delivered through the path 1 : $n_s \rightarrow n_1 \rightarrow n_2 \rightarrow n_3 \rightarrow n_d$ and path 2 : $n_s \rightarrow n_4 \rightarrow n_5 \rightarrow n_d$, respectively. Let $\psi_d = 600$. According to path 1, $O_m = \{n_s, n_1, n_2, n_3, n_d\}$ and $N_m = 4$. Thus, $\theta_{n_s,m} = 1$ and the reward of node n_s increases by $(2 \times 1 \times \psi_d)/(4 \times (4 + 1)) = 60$, $\theta_{n_1,m} = 2$ and the reward of node n_1 increases by $(2 \times 2 \times \psi_d)/(4 \times (4 + 1)) = 120$, etc. Similarly, according to path 2, the reward of node n_s increases by $(2 \times 1 \times \psi_d)/(3 \times (3 + 1)) = 100$, the reward of node n_4 increases by $(2 \times 2 \times \psi_d)/(3 \times (3 + 1)) = 200$, etc. From the above, n_s always get the minimum reward in Path 1 and Path 2, and the total reward obtained by n_s (i.e., 160) is still lower than that of n_2, n_3, n_4, n_5, which alleviates the problem in [19] and increases the robustness of the reward mechanism greatly. It can be concluded that the shorter the path of delivering message successfully is, the more reward the nodes can obtain. And the relay node closer to the destination can get more reward.

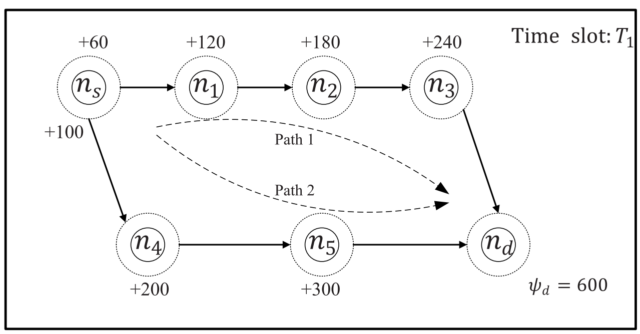


FIGURE 1. An example of the delivery reward.

In addtion, since the contact between nodes is helpful for the message forwarding in VDTN, we introduce another reward named contact reward. When a node encounters with another, they both are supposed to be assigned a certain contact reward ψ_c. Therefore, combining the delivery and contact rewards, we can obtain the reward of a node in VDTN. Let $\psi_{n_i}(t)$ denote the reward of node n_i at time slot t. Then, the update rule of $\psi_{n_i}(t+1)$ can be illustrated by (1).

$$\psi_{n_i}(t+1) = (1-\rho)\psi_{n_i}(t) + c_{n_i}(t)\psi_c + \sum_{m=1}^{r_{n_i}(t)} \left(\frac{2\theta_{n_i,m}}{N_m(N_m+1)}\psi_d\right) \quad (1)$$

where ρ is the ageing coefficient, which makes the reward value of a node age over time as same as the ant colony algorithm and $\rho \in [0, 1]$, $c_{n_i}(t)$ denotes the number of other nodes encountering with node n_i during time slot t, $r_{n_i}(t)$ is the total number of delivered messages through node n_i during time slot t.

Thus, the improved rule of (1) updates the reward value dynamically according to message delivery, encounter frequency and time. Since the reward changes with the movement of nodes, the forwarding of messages, and the passage of time, we need to discretize the reward to apply it into the classifier model. In this paper, we divide the reward into k_7 ranges with equal interval and each of them stands for a different level. Based on this method, nodes in VDTN can be divided into $X_7 = < L_1, L_2, \ldots, L_{k_7} >$ levels, which are called the *delivery level*. The higher the level is, the greater the probability of the nodes succeeding in message delivery is. Therefore, the dynamic reward is utilized as the classification criteria, which can help nodes make superior routing decisions. By predicting the delivery level of nodes, nodes carrying messages decide whether to send these messages to their neighbor nodes or not.

IV. STRUCTURE LEARNING

After the historical data set based on the selected attributes is collected, the next step is how to learn the best BN structure. In general, there are two kinds of method in BN structure learning. The one is Mutual Information (MI) [25] method based on dependency between attributes, where statistical or information theories are adopted to analyze the dependency between variables. The other is scoring search learning method [26], which constructs the BN structure according to certain search strategies and scoring criteria. Scoring search learning method focuses on the creation of BN structure that fits the data set. As a classical scoring search learning algorithm, K2 algorithm is used for finding associations between system components and building the BN model. This algorithm uses a heuristic method to provide efficient and accurate results while searching for associations. Moreover, no human intervention is necessary during the process of BN construction in K2 algorithm. However, the performance of the K2 algorithm is greatly affected by the provided attribute variables ordering. If we provide a proper attribute variables ordering, e.g., all parents variables are ordered prior to their children variables, the algorithm will perform optimally and get an accurate result. Unfortunately, in most cases, the correct attribute variables ordering is unknown. Therefore, the purpose of this section is to solve the problem that how to get a better attribute variables ordering for K2 algorithm.

A. OPTIMIZATION PROBLEM

Given a set of variables, there are many directed acyclic graphs structure of BN can be constructed by these variable as nodes. It is difficult to enumerate all of these graphs, so it is reasonable to select the model of graph with the greatest posterior probability. Let D denote the historical data set based on the selected attributes. Let G denote the structure of BN. G^* represents the best BN structure, that is, $G^* = \arg\max_G(P(D|G))$, where the posterior probability $P(D|G)$ is the scoring function in scoring search learning method. In practice, $P(D|G)$ can take different forms. In this paper, Cooper-Herskovits (CH) [27] scoring function is selected to be a metric to evaluate the BN structure and to measure the quality of the attribute variables ordering. And the K2 algorithm uses CH scoring function to decide the best candidate parents set for node X_i. Now, define the attribute variables ordering as $\hat{X} = < X_{i_1}, X_{i_2}, \ldots, X_{i_7} >$,where $< i_1, i_2, \ldots, i_7 >$ is a permutation of 1 to 7. Let $\pi(X_{i_k}) = \{X_{i_j} | j \in [1, k)\}$ be the candidate parents set of attribute X_{i_k}. Then, the CH scoring function is provided as (2)

$$f(X_{i_k}, \pi(X_{i_k})) = \prod_{j=1}^{q_k} \frac{(s_k-1)!}{d_{kj*} + (s_k-1)!} \prod_{v=1}^{s_k} d_{kjv}! \quad (2)$$

where q_k is the number of all possible states of $\pi(X_{i_k})$, s_k represents the number of states that the node X_{i_k} can take. d_{kj*} represents the number of instances in historic data set where X_{i_k} is any value and the instance $\pi(X_{i_k})$ is in the jth state, and d_{kjv} denotes the number of instances in historic data set where X_{i_k} is v and the instance $\pi(X_{i_k})$ is in the jth state.

For attribute variable X_{i_k} and its candidate parents set $\pi(X_{i_k})$, the best parents set $\pi^*(X_{i_k})$ is determined by K2 algorithm to make $f(X_{i_k}, \pi(X_{i_k}))$ up to the maximum. Then, the best score of X_{i_k} is formulated by (3)

$$f^*(X_{i_k}) = \max_{\pi(X_{i_k})} f(X_{i_k}, \pi(X_{i_k})) \quad (3)$$

The K2 algorithm iterates on the attribute variable X_{i_k} according to the attribute variables ordering. And it is supposed to get best score of each variable after K2 has traversed all the attribute variables. Next, we can construct a Bayesian network $G(\hat{X})$ by adding a directed edge from each attribute variable of $\pi^*(X_{i_k})$ to X_{i_k} . According to [28], the quality of $G(\hat{X})$ can be measured by the total score of all the attribute variable, which is formulated by $F(\hat{X})$ as (4)

$$F(\hat{X}) = \sum_{k=1}^{7} log(f^*(X_{i_k})) \quad (4)$$

It is obvious that the ordering of attribute variables $\hat{X}$ plays an important role in searching for BN structure. Therefore, we can formulate the optimization problem as (5)

$$\max_{\hat{X}} F(\hat{X}) \quad (5)$$

When $\hat{X} = \hat{X}^*$ makes $F(\hat{X})$ take the maximum value, we can get the best BN structure $G^*(\hat{X}^*)$.

B. K2-GA ALGORITHM

Unfortunately, it is unrealistic to iterate all the possible ordering of $\hat{X}$ and find the best $\hat{X}^*$ that makes $F(\hat{X})$ up to the maximum, because it is a NP-hard problem. Thus, in this paper, we consider an appropriate evolutionary method to solve the problem. During the past two decades, Genetic Algorithm (GA) [29] has been successfully and broadly applied to solve constrained optimization problems. Consequently, we propose an improved K2 algorithm based on GA (named K2-GA) to search for the optimal BN structure efficiently.

The details of K2-GA are as follows:

1) INITIALIZATION

In K2-GA, the chromosome is defined as the attribute variables ordering $\hat{X}$. Let z be the size of population. At the beginning of the algorithm, z chromosomes are produced randomly and assigned to the first population. In order to increase the diversity of the population, the initial z chromosomes should be different with each other.

2) FITNESS FUNCTION

Fitness function is a metric to measure the quality of the attribute variables ordering. As described in (5), $F(\hat{X})$ is the metric to evaluate the quality of BN structure in the attribute variables ordering. Therefore, $F(\hat{X})$ is determined to be the fitness function. The Bayesian network with the greatest $F(\hat{X})$ value is the best structure.

3) CROSSOVER OPERATOR

In traditional GA, simple crossover operators such as one-point or two-point crossover are often used to increase the local diversity of crossover chromosomes. However, the traditional crossover operator cannot be used in K2-GA, because it cannot guarantee the uniqueness of the elements in attribute variables ordering. Consequently, we adopt Partial-Mapped Crossover (PMX) [30] to generate new generation. Let $H_{i,j}$ denote the ith chromosome of the jth population.

The process of PMX is as follows:

1) Choose $H_{i,j}$ and $H_{i+1,j}$ randomly with the crossover probability α, and generate randomly two crossover segments with equal length on $H_{i,j}$ and $H_{i+1,j}$ respectively.
2) Swap all attributes in the crossover segments of $H_{i,j}$ and $H_{i+1,j}$ to produce two proto-children.
3) Establish a mapping table according to the exchanged attributes. An attribute that belongs to both crossover segment and non-crossover segment in a chromosome is named a conflict attribute. $H'_{i,j}$ and $H'_{i+1,j}$ are newborn chromosomes after replacing all conflict attributes belonging to non-crossover segments in proto-children according to the mapping table.

Fig. 2 is an example of the PMX. As shown in Fig. 2(a), $H_{1,1}$ and $H_{2,1}$ are the chosen chromosomes to be crossed, where gray variables are crossover segments. It can be seen

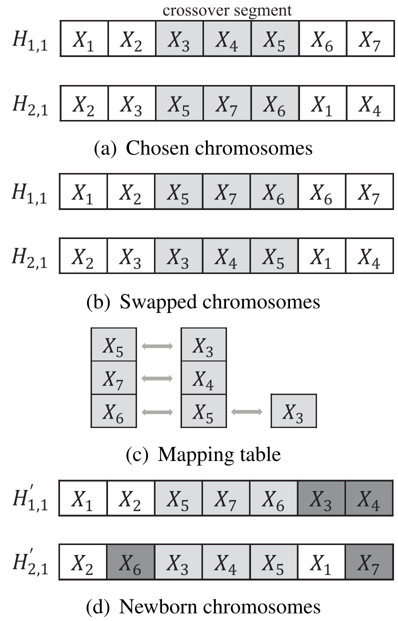


FIGURE 2. An example of the PMX.

from Fig. 2(b), $H_{1,1}$ and $H_{2,1}$ are generated through swapping all attributes in crossover segment. It is obvious that X_6, X_7 are conflict attributes in $H_{1,1}$ and X_3, X_4 are conflict attributes in $H_{2,1}$. The mapping table in Fig. 2(c) is established based on the swapping attributes to solve the confliction. According to the mapping table, the conflict attributes X_6 and X_7 belonging to the non-crossover segment in $H_{1,1}$ are replaced by X_3 and X_4, respectively. And the conflict attributes X_3 and X_4 belonging to the non-crossover segment in $H_{2,1}$ are replaced by X_6 and X_7, respectively. Finally, as shown in Fig. 2(d), $H'_{1,1}$ and $H'_{2,1}$ are the newborn chromosomes without conflicts.

4) MUTATION OPERATOR

In order to search the best BN structure, mutation operator is used to increase the diversity of population and to reduce the rate of convergence. In this paper, each chromosome is supposed to mutate with the mutation probability β. Once a chromosome is determined to mutate, the mutation operator will swap two attributes of the chromosome randomly.

5) SELECT OPERATOR

In order to select outstanding chromosomes from the population, we adopt the roulette wheel election mechanism to choose $z-1$ chromosomes from the current population and add them to the next generation. Moreover, elitism strategy is adopted into the select operator, that is, the excellent chromosome in the current population will be preserved to the next generation.

The pseudo code of the proposed K2-GA algorithm is shown in Algorithm 1, where z is the size of population, α is the crossover probability, β is the mutation probability, and w is the maximum number of iterations. Moreover, $H_j = \{H_{1,j}, H_{2,j}, \ldots, H_{z,j}\}$ is the jth population, and *rand* is used

Algorithm 1 K2-GA Algorithm

Input: z, α, β, w
Output: $\hat{X}^*, F^*(\hat{X}^*), G^*(\hat{X}^*)$
1: $j \leftarrow 1$
2: z chromosomes are produced randomly and assigned to H_j
3: **while** $j < w$ **do**
4:   **for** $i = 1$ to z $step = 2$ **do**
5:     **if** $rand < \alpha$ **then**
6:       $H_{i,j}, H_{i+1,j} \leftarrow PMX(H_{i,j}, H_{i+1,j})$
7:     **end if**
8:   **end for**
9:   **for** $i = 1$ to z **do**
10:     **if** $rand < \beta$ **then**
11:       swap two attributes X_p and X_q in $H_{i,j}$ randomly
12:     **end if**
13:   **end for**
14:   **for** $i = 1$ to z **do**
15:     obtain $G(H_{i,j}), F(H_{i,j})$ by K2 algorithm (4).
16:     **if** $F(H_{i,j}) > F^*(\hat{X}^*)$ **then**
17:       $\hat{X}^* \leftarrow H_{i,j}$, $F^*(\hat{X}^*) \leftarrow F(H_{i,j})$, $G^*(\hat{X}^*) \leftarrow G(H_{i,j})$
18:     **end if**
19:   **end for**
20:   add $\hat{X}^*$ to H_{j+1}
21:   select $z-1$ chromosomes from H_j by roulette wheel selection mechanism, and add them to H_{j+1}
22:   $j \leftarrow j+1$
23: **end while**
24: **return** $\hat{X}^*, F^*(\hat{X}^*), G^*(\hat{X}^*)$

to generate evenly a random number in the range of [0, 1). According to the algorithm, at first, z chromosomes are generated randomly to construct the first generation with initialization operator (lines 1-2). Then the crossover (lines 4-8) and mutation (lines 9-13) operators are performed. The fitness value and corresponding BN structure of each chromosome in the population is calculated by lines 14-19. Afterward, a new generation with z chromosomes is generated using the selection operator (lines 21-22). After w generation of evolutions, at last, we select the best chromosome $\hat{X}^*$ and return the optimal BN structure $G^*(\hat{X}^*)$ and its score $F^*(\hat{X}^*)$ (line 24).

When α is 0.6 and β is 0.2, the BN trained through the above algorithm is shown as Fig. 3. After obtaining the network structure, we can establish the CPT of each attribute by the corresponding data in the historic data set.

V. INFERENCE

Once the optimal BN structure learned, an inferece algorithm is needed to predict the delivery ability of VDTN nodes. As mentioned above, inference based on joint distribution is not feasible when there are too many variables in BN, because its complexity is exponential with the number of variables.

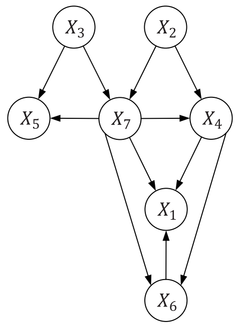


FIGURE 3. The BN structure learned by K2-GA.

Therefore, we use JTA algorithm to calculate the probability according to the BN model [21]. Based on this, we propose a message forwarding strategy for VDTN routing algorithm.

A. JUNCTION TREE ALGORITHM

JTA is used to accelerate the inference process through variable elimination and calculation sharing, which is more suitable for the inference of the large scale Bayesian networks.

The construction of a junction tree consists of the following steps:

Step 1: Form the moral graph. For each node in the BN, add edges between all of its parents, and drop the directions.

Step 2: Triangulate the moral graph. A moral graph is triangulated when every cycle with more than three nodes has an arc between a pair of nonadjacent nodes. Add edges to the moral graph until a triangulated graph is constructed.

Step 3: Determine the cliques and separators. Firstly, the Maximum Cardinality Search (MCS) [31] algorithm is used to determine the elimination order. According to the order, the triangulated graph is divided into cliques and separators. A clique is a maximal complete subgraph of the triangulated graph, and a separator is a set nodes that separates two adjacent cliques.

Step 4: Organize the cliques and separators as a junction tree. Edges between cliques are introduced. This is done such that a tree with the following property results: Whenever two cliques are connected by a path, the intersection of them is a separator on the path. The CPTs in the BN are assigned to each clique in the junction tree for storage.

For example, suppose that Fig. 3 is the optimal BN obtained by K2-GA algorithm. According to Step 1, add an edge between X_2 and X_3, drop all the directions, then we have the moral graph shown as Fig. 4. Fortunately, it is obvious that the moral graph is already a triangulated graph. In Step 3, the elimination order of the triangulated graph is determined as $< X_5, X_3, X_2, X_4, X_6, X_1, X_7 >$ by MCS. According to the order and the triangulated graph, we obtain the cliques and separators, which are given in Table 2. Finally, the junction tree shown as Fig. 5 is constructed through Step 4. The label of each edge between two cliques is the separator. For each probability distribution in the original BN (e.g., Fig. 3), put it into one of the cliques if the domain of the distribution

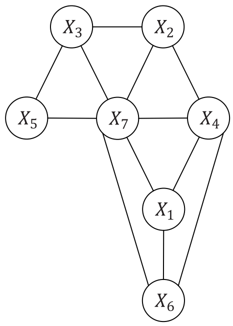


FIGURE 4. Moral graph of Fig. 3.

TABLE 2. Cliques and Separators.

Clique	Separator
$X_5X_7X_3$	X_3X_7
$X_3X_2X_7$	X_2X_7
$X_2X_7X_4$	X_4X_7
$X_4X_7X_1X_6$	

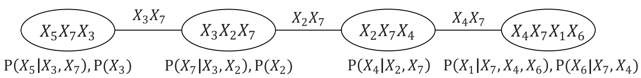


FIGURE 5. Junction tree.

is a subset of the clique. For example, $P(X_5|X_3, X_7)$ is a distribution in Fig. 3, and its domain $\{X_5, X_3, X_7\}$ is a subset of the clique $\{X_5X_7X_3\}$, thus we put $P(X_5|X_3, X_7)$ into clique $\{X_5X_7X_3\}$ in Fig. 5. Similarly, we can put the distribution $P(X_3)$ into clique $\{X_5X_7X_3\}$, $P(X_7|X_3, X_2)$ and $P(X_2)$ into clique $\{X_3X_2X_7\}$, etc.

In JTA, the junction tree can be used to estimate the value of a certain attribute by the observational values of other attributes. In this paper, the probability of a vehicle belonging to a certain delivery level can be calculated according to the junction tree. Let $x = \{R_{j_1}, T_{j_2}, A_{j_3}, I_{j_4}, V_{j_5}, U_{j_6}\}$ denote an instance of the observational attributes set of a vehicle, where $j_l \in [1, k_l]$ and $l \in [1, 6]$. Randomly select a clique containing X_7 to be the inference key and construct its evidence propagation. Supposing clique $\{X_2X_7X_4\}$ is selected, the evidence propagation of the clique is shown in Fig. 6. According to Fig. 5 and Fig. 6, $\delta(X_3) = P(X_3 = A_{j_3})$ denotes the probability that X_3 equals A_{j_3}. $\delta(X_5, X_3, X_7) = P(X_5 = V_{j_5}|X3 = A_{j_3}, X_7)$ is the CPT that X_5 equals V_{j_5} when the value of X3 is $A_{j_3}$and X_7 is unknown. According to the direction of evidences propagation in Fig. 6, $\delta(X_3, X_7) = \delta(X_5, X_3, X_7)\delta(X_3)$ is the information flowing from clique $\{X_5X_7X_3\}$ to clique $\{X_3X_2X_7\}$. Similarly, $\delta(X_2, X_7) = \delta(X_3, X_7)\delta(X_7, X_3, X_2)\delta(X_2)$ is the information flowing from clique $\{X_3X_2X_7\}$ to clique $\{X_2X_7X_4\}$.

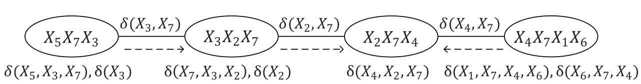


FIGURE 6. Evidence propagation.

$\delta(X_4, X_7) = \delta(X_1, X_7, X_4, X_6)\delta(X_6, X_7, X_4)$ is the information flowing from clique $\{X_4X_7X_1X_6\}$ to clique $\{X_2X_7X_4\}$. Finally, $\delta(X_2, X_7)$ and $\delta(X_4, X_7)$ flow into the key clique $\{X_2X_7X_4\}$, so $\delta(X_7) = \delta(X_2, X_7)\delta(X_4, X_7)\delta(X_4, X_2, X_7)$. The probability $P(X_7 = L_{j_7}|x)$ can be calculated according to (6).

$$P(X_7 = L_{j_7}|x) = \frac{\delta(X_7 = L_{j_7})}{\sum_{j_7=1}^{k_7} \delta(X_7 = L_{j_7})} \tag{6}$$

Then the delivery level of a vehicle node in the state of x can be obtained according to (7).

$$X_7 = \underset{j_7 \in (1,\ldots,k_7)}{\operatorname{argmax}} P(X_7 = L_{j_7}|x) \tag{7}$$

B. MESSAGE FORWARDING

In order to apply the proposed BN model and algorithms into VDTN, the historical data set of all vehicles need to be collected. Then, K2-GA algorithm is used to learn the optimal BN structure according to the data set for entire VDTN, then the JTA is executed to construct a junction tree. In our model, all vehicles share the same BN structure and junction tree. Considering the real situation of VDTN, in order to reduce the cost, we can run the K2-GA and JTA algorithms offline, and just upload the junction tree to each vehicle in VDTN. When a vehicle meets another vehicle, these two vehicles can send messages to each other. There are two cases of message forwarding at this time. When the vehicle carrying the message encounters the destination vehicle, the message will be transmitted to the destination directly. Otherwise, the delivery level, i.e., X_7, of the current vehicle and the encountered vehicle are calculated according to the junction tree and (7). And the inference result will be cached to the node to speed up the inference later. If the delivery level of the encountered vehicle is greater than that of the current vehicle, the current vehicle will transmit the message to the encounter. Otherwise, the current vehicle will retain the message until it encounters a vehicle whose delivery level is greater.

VI. PERFORMANCE EVALUATION

A. SIMULATION SETUP AND DATASET

In this paper, we use the simulation software the ONE (The Opportunistic Network Environment) simulator [32] to simulate the proposed BN model and algorithms in VDTN. The dataset used in the simulation experiment is derived from real mobility traces of buses in Rio de Janeiro, Brasil. The dataset is obtained from CRAWDAD [33], which contains a movement trajectory covering 1,200 square kilometers and involving 17,723 buses. Although the dataset provides data that all buses drives on the area of 1,200 square kilometers per day of a month, we concentrate on a 12 hour period from 7 am-7 pm and 200 buses in the area of 35km × 25km. It should be noted that the proposed BN model and algorithms can be applied to different movement patterns of vehicles in VDTN, and without loss of generality, we use the traces of bus in the simulation. The settings of simulation parameters are shown in Table 3. And in order to evaluate the

TABLE 3. Simulation parameters.

Parameter	Value
Simulate time	47000 s
Terrain size	35 km × 25 km
Transmission range	50 m
Transmission speed	250 Kbytes
Message size	50 Kbytes
Number of nodes	98

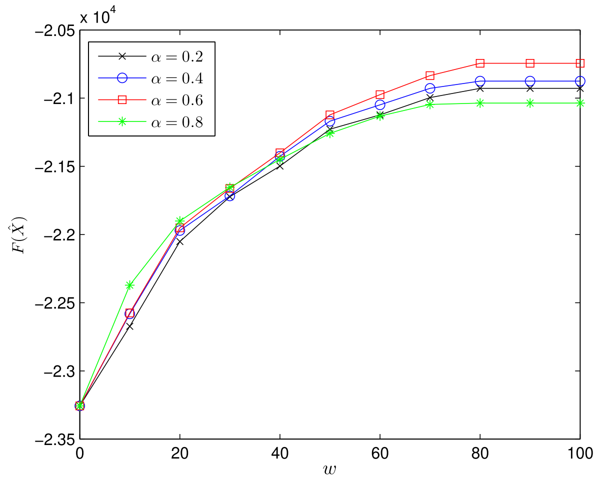
performance of K2-GA, we do the experiment with Matlab Bayesian tools [34].

B. TRAINING DATASET

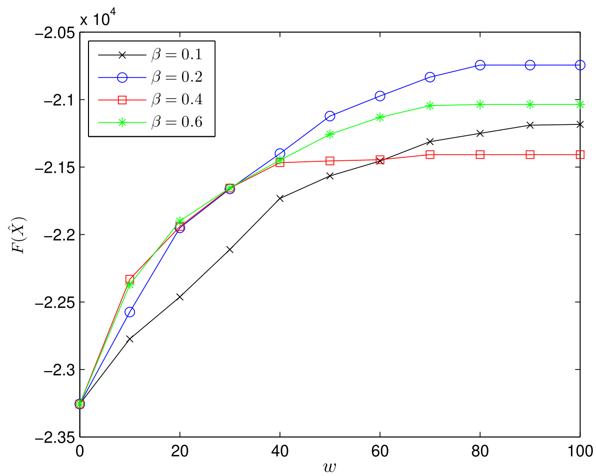
Before conducting this simulation experiment, we have to gather the historic data set. As a result, the data of 14 days are selected. More specifically, 10 day's data are randomly selected as the training set and the rest data are used as the test set. The historic data set of each node is an empty table by default. In the process of generating the historic data set, the nodes adopt the flooding mechanism to deliver the messages, and every time the nodes encounter, the value of the message Id, the destination node, the current region code, time slot, movement angle, velocity, path Id and average inter-contact interval will be recorded into the table. After the experiment is finished, the node level is confirmed according to the reward value and the data set are also stored locally. Finally, we collect all the data of each node to train the BN model.

C. IMPACT OF THE PARAMETERS IN K2-GA

The crossover probability α and the mutation probability β are critical parameters in K2-GA algorithm. In order to evaluate the impact of the two parameters in the K2-GA algorithm, we let the size of population be 50, the maximum iterations be 100, that is, $z = 50$ and $w = 100$. The number of the attribute variables is 7, and the historic dataset of attributes is obtained from the mobility traces. We do some experiments through modifying the crossover probability and mutation probability. The results are shown as Fig. 7. According to Fig. 7(a), with the increase of crossover probability α, K2-GA can obtain a better BN structure, i.e. the score of BN $F(\hat{X})$, when α is greater than 0.2 and less than 0.6. However, when α is 0.8, the BN structure is poor, because the high frequency of crossover leads to that excellent chromosomes cannot be retained to next generation. Therefore, we choose $\alpha = 0.6$ to be applied to K2-GA. According to Fig. 7(b), it is obvious that K2-GA can get the best performance when the mutation probability $\beta = 0.2$. If we let $\beta = 0.1$, the best score of BN structure is poor since the diversity of population is too low. The performance of the K2-GA algorithm is unstable when $\beta > 0.2$, because the high mutation probability increase diversity of population, which is conducive to generate an optimal chromosome. However, the high mutation probability also causes that excellent chromosome mutates to poor chromosomes, which results in a lower average score of the population.



(a) Impact of crossover probability



(b) Impact of mutation probability

FIGURE 7. Performance of K2-GA.

D. COMPARISON WITH OTHER BAYESIAN NETWORK STRUCTURES

In this paper, we evaluate the proposed routing algorithm in VDTN by the following three metrics, which are usually used to indicate the performance of routing algorithms in DTNs.

- Delivery Ratio: the ratio of the number of messages reached at the destination to the number of messages generated.
- Delivery Delay: the average time taken by the message to reach the destination.
- Overhead: the average number of copies of the message in the network at the time of reaching the destination.

In order to evaluate the effectiveness of K2-GA algorithm, we compare our routing algorithm with different BN structures as follows.

- K2-GA: the BN structure is otained by K2-GA algorithm.
- MI: the BN structure is obtained by the mutual information algorithm in [19].

- Rand: the BN structure is obtained by a random sequence.
- Opt: the BN structure is the optimal and obtained by enumerating all possible structures.

Figs. 8-10 show the simulation results of delivery ratio, delivery delay and overhead vs. the deadline of messages (i.e., Time-To-Live, TTL) in different BN structures. As shown in Fig. 8, Rand and Opt show the worst and best performance in delivery ratio, respectively. Because the BN structure depending on a random attribute variables ordering is usually not good, but the BN structure of Opt can be the best by enumerating all possible structures. Comparatively, K2-GA is better than MI and a little worse than Opt in delivery ratio., As shown in Fig. 9, Opt has the lowest delivery delay for the same reason. Besides, the delivery delay of K2-GA is similar to that of MI, and Rand has the higest delivery delay. Causing that Rand can not discriminate the good and poor nodes in VDTN, many messages are transmitted to the poor nodes, which leads to lower delivery ratio, higher delay and larger overhead. According to Fig. 10, K2-GA and Opt have obvious lower overhead than MI and Rand, because K2-GA can learn the vehicle movement patterns efficiently and is better than MI in distinguishing the good nodes and the poor nodes in VDTN. Although Opt shows the best performance, the complexity of the enumeration method is too high to scalable and practical. In addtion, since the performance of K2-GA and Opt are similar, we can see that K2-GA algorithm is effective.

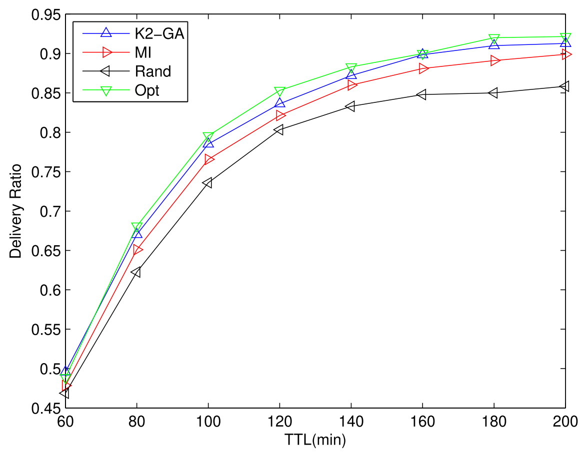


FIGURE 8. Delivery Ratio vs. TTL.

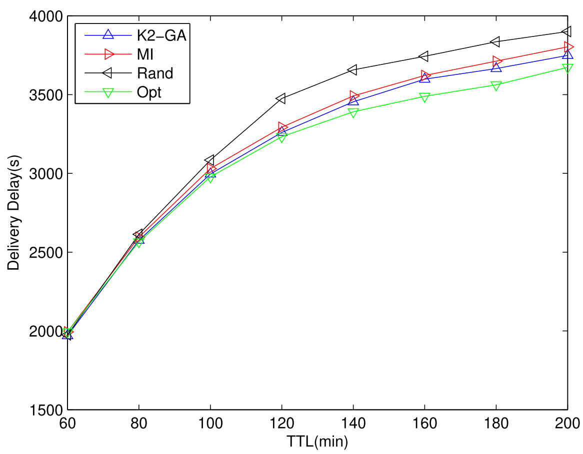


FIGURE 9. Delivery Delay vs. TTL.

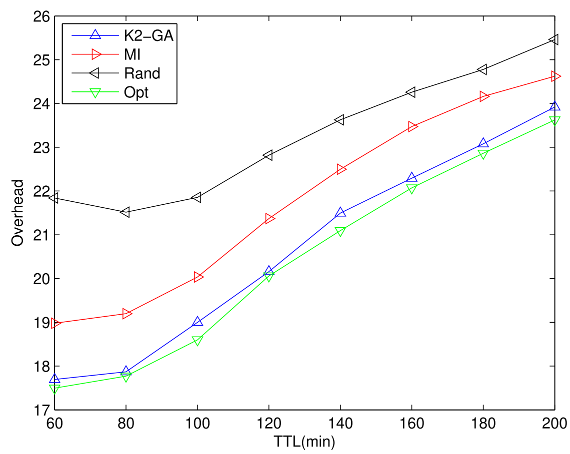


FIGURE 10. Overhead vs. TTL.

E. COMPARISON WITH OTHER ROUTING ALGORITHMS

In this section, we compare the performance of the proposed routing algorithm based on BN with Epidemic, Prophet and NB in delivery ratio, overhead and delivery delay while changing of the TTL of messages and the buffer size of vehicles.

Figs. 11-13 show the simulation results of delivery ratio, delivery delay and overhead vs. TTL. In Fig. 11, the delivery ratio of BN is higher than that of Prophet and NB. Although Prophet determines whether messages are delivered or not based on the historical information, BN not only takes the historical information into account to estimate the transmission ability of vehicle, but also takes advantage of the vehicle movement patterns. Therefore, the performance of BN is better than Prophet. Besides, NB has the same considerations as BN, but the conditional independence assumption of NB limits its performance. When TTL = 100 min, we can find that BN can effectively improve the delivery ratio by

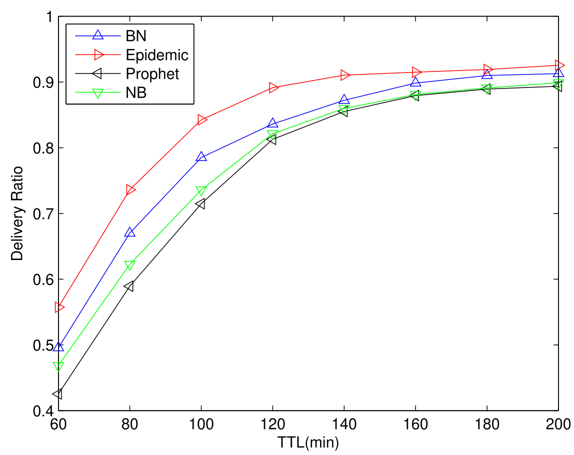


FIGURE 11. Delivery Ratio vs. TTL.

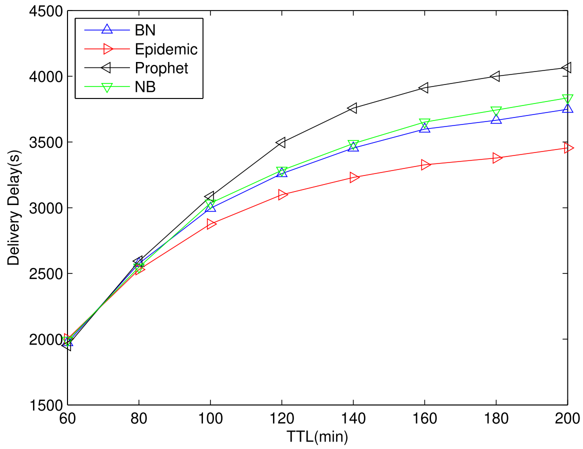


FIGURE 12. Delivery Delay vs. TTL.

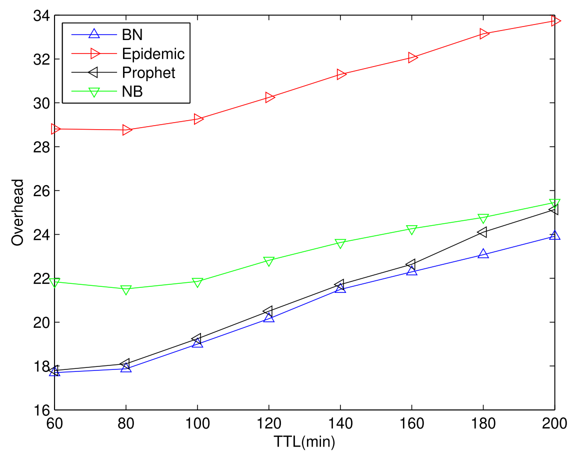


FIGURE 13. Overhead vs. TTL.

about 8% compared to Prophet and 5% compared to NB. However, the delivery ratio of BN is lower than Epidemic, because Epidemic is based on flooding strategy. In Fig. 12, the delivery delay of BN is lower than Prophet and NB, because the transmission ability of vehicles is calculated by improved ant colony algorithm, which makes these vehicles in shortest path have higher delivery level. Consequently, the proposed message forwarding algorithm let messages be delivered to the vehicles with greater ability. Fig. 13 shows the overhead of different algorithms. With the increase of TTL, the overhead increases. However, compared with Epidemic, Prophet and NB, BN has the lowest overhead, because our message forwarding algorithm limits the number of copies.

Figs. 14-16 show the simulation results of delivery ratio, delivery delay and overhead vs. buffer size. As shown in Fig. 14, with the increase of buffer size, the delivery ratio increases. When buffer size is 17 M, BN can effectively improve the delivery ratio by about 5% compared to Prophet and 2% compared to NB. In Fig. 15, it is obvious that BN shows excellent performance in delivery delay. As shown in Fig. 16, BN has the lower overhead than Epidemic, Prophet and NB, and the overhead of BN is decreasing with the increase of buffer size.

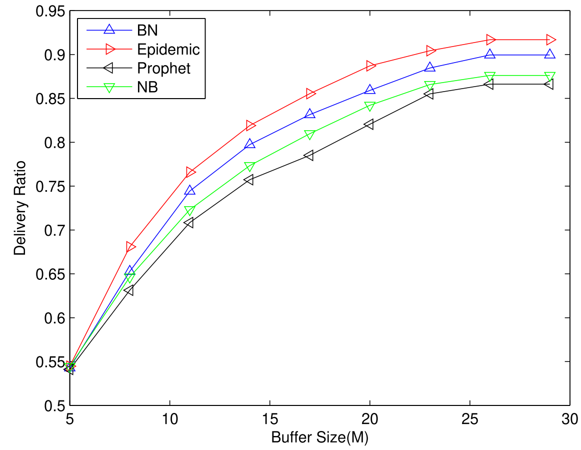


FIGURE 14. Delivery Ratio vs. Buffer Size.

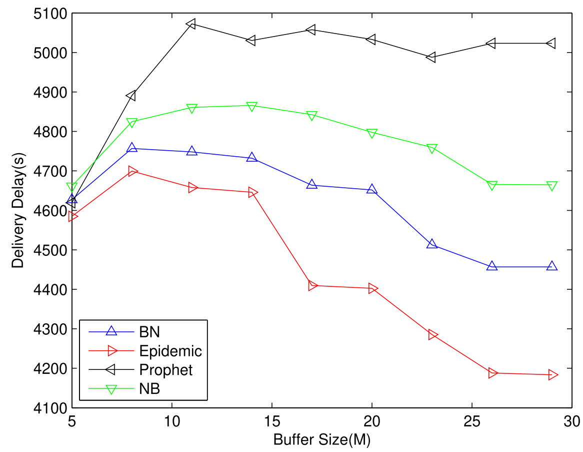


FIGURE 15. Delivery Delay vs. Buffer Size.

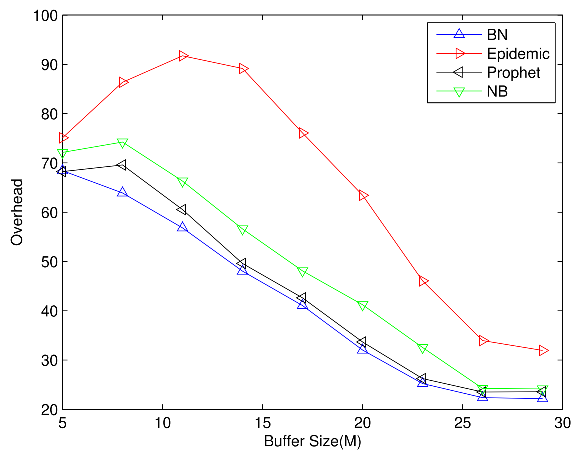


FIGURE 16. Overhead vs. Buffer Size.

VII. CONCLUSION

This paper investigates the messages forwarding problem for VDTN, where a new routing algorithm is proposed and optimized. Considering the vehicle movement patterns, the Bayesian network is utilized to make routing decisions in VDTN. a novel BN structure learning algorithm (K2-GA) is proposed to search the global optimal structure, and JTA is adopted to accelerate the inference process of BN. The experiment of structure learning shows K2-GA has excellent ability to get a optimal BN structure, and the simulations suggest that the proposed routing algorithm based on BN can achieve a satisfactory delivery ratio with a minor forwarding overhead. In the future, we are going to improve the BN model and evaluate our routing algorithm against more VDTN traces.

REFERENCES

[1] N. N. Pal, K. Gaikwad, and D. Das, "Trust calculation and route discovery for delay tolerant networks," in *Proc. IEEE Region 10 Conf. (TENCON)*, Oct. 2018, pp. 1533–1537.

[2] S. E. Alaoui and B. Ramamurthy, "Routing optimization for DTN-based space networks using a temporal graph model," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[3] P. Arora and S. Jain, "Underwater sensor network delay aware routing protocols: A survey," in *Proc. 3rd Int. Conf. Internet Things Connected Technol. (ICIoTCT)*, 2018, pp. 26–27.

[4] S. Xiang and J. Yang, "Performance reliability evaluation for mobile ad hoc networks," *Rel. Eng. Syst. Saf.*, vol. 169, pp. 32–39, Jan. 2018.

[5] J. Tian, M. Gao, and G. Ge, "Wireless sensor network node optimal coverage based on improved genetic algorithm and binary ant colony algorithm," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, p. 104, Dec. 2016.

[6] R. Bocquillon and A. Jouglet, "Robust routing in deterministic delay-tolerant networks," *Comput. Oper. Res.*, vol. 92, pp. 77–86, Apr. 2018.

[7] A. Marwa and M. Imad, "A survey of vehicular ad hoc networks routing protocols," *Int. J. Innov. Appl. Stud.*, vol. 3, no. 3, pp. 829–846, Jul. 2013.

[8] Y. Li, L. Zhao, Z. Liu, and Q. Liu, "N-Drop: Congestion control strategy under epidemic routing in DTN," in *Proc. Int. Conf. Wireless Commun. Mobile Comput., Connecting World Wirelessly*, 2009, pp. 457–460.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay-Tolerant Netw.*, 2005, pp. 252–259.

[10] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *Proc. ACM Int. Symp. Mobilde Ad Hoc Netw. Comput., MobiHoc*, 2003, pp. 1–2.

[11] F. Zhang, H. Liu, Y.-W. Leung, X. Chu, and B. Jin, "Community-based bus system as routing backbone for vehicular ad hoc networks," in *Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst.*, Jun./Jul. 2015, pp. 73–82.

[12] D. K. Sharma, S. K. Dhurandher, I. Woungang, R. K. Srivastava, A. Mohananey, and J. J. P. C. Rodrigues, "A machine learning-based protocol for efficient routing in opportunistic networks," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2207–2213, Sep. 2018.

[13] V. G. Rolla and M. Curado, "A reinforcement learning-based routing for delay tolerant networks," *Eng. Appl. Artif. Intell.*, vol. 26, no. 10, pp. 2243–2250, Nov. 2013.

[14] F. Li, X. Song, H. Chen, X. Li, and Y. Wang, "Hierarchical routing for vehicular ad hoc networks via reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1852–1865, Feb. 2019.

[15] F. R. Segundo, J.-M. Farines, and E. S. E. Silva, "Quasi-opportunistic contact prediction in delay/disruption tolerant network," in *Proc. Global Inf. Infrastruct. Symp.-(GIIS)*. Oct. 2013, pp. 1–6.

[16] S. Ahmed and S. S. Kanhere, "A Bayesian routing framework for delay tolerant networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2010, pp. 1–6.

[17] R. Dudukovich, A. Hylton, and C. Papachristou, "A machine learning concept for DTN routing," in *Proc. IEEE Int. Conf. Wireless Space Extreme Environ. (WiSEE)*. Oct. 2017, pp. 110–115.

[18] J. Liang, Y. Bai, C. Bi, Z. Sun, C. Yan, and H. Liang, "Adaptive routing based on Bayesian network and fuzzy decision algorithm in delay-tolerant network," in *Proc. IEEE 10th Int. Conf. High Perform. Comput. Commun. IEEE Int. Conf. Embedded Ubiquitous Comput.*, Nov. 2013, pp. 690–697.

[19] H. Zhou, J. Wu, L. Shen, and L. Liu, "Vehicle delay-tolerant network routing algorithm based on multi-period Bayesian network," in *Proc. IEEE 37th Int. Perform. Comput. Commun. Conf. (IPCCC)*. Nov. 2018, pp. 1–8.

[20] X.-W. Chen, G. Anantha, and X. Lin, "Improving Bayesian network structure learning with mutual information-based node ordering in the K2 algorithm," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 5, pp. 628–640, May 2008.

[21] A. L. Madsen and F. V. Jensen, "Lazy propagation: A junction tree inference algorithm based on lazy evaluation," *Artif. Intell.*, vol. 113, nos. 1–2, pp. 203–245, Sep. 1999.

[22] S. Jain, K. Fall, and R. Patra, *Routing in Delay-Tolerant Networking*, vol. 34, no. 4. New York, NY, USA: ACM, 2004.

[23] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *Proc. 1st Annu. IEEE Commun. Soc. Conf. Sensor Ad Hoc Commun. Netw. (IEEE SECON)* Apr. 2004, pp. 235–244.

[24] M. Dorigo and T. Stützle, "Ant colony optimization: Overview and recent advances," in *Handbook of Metaheuristics*. Boston, MA, USA: Springer, 2010, pp. 227-263.

[25] R. A. Sutrisnowati, H. Bae, J. Park, and B.-H. Ha, "Learning Bayesian network from event logs using mutual information test," in *Proc. IEEE 6th Int. Conf. Service-Oriented Comput. Appl.*, Dec. 2013, pp. 356–360.

[26] M. Teyssier and D. Koller, "Ordering-based search: A simple and effective algorithm for learning Bayesian networks," 2012, *arXiv:1207.1429*. [Online]. Available: https://arxiv.org/abs/1207.1429

[27] O. Doguc and J. E. Ramirez-Marquez, "A generic method for estimating system reliability using Bayesian networks," *Rel. Eng. Syst. Saf.*, vol. 94, no. 2, pp. 542–550, Feb. 2009.

[28] M. Z. Zgurovskii, P. I. Bidyuk, and A. N. Terent'ev, "Methods of constructing Bayesian networks based on scoring functions," *Cybern. Syst. Anal.*, vol. 44, no. 2, pp. 219–224, Mar. 2008.

[29] O. Kramer, *Genetic Algorithm Essentials*, vol. 679. Cham, Switzerland: Springer, 2017.

[30] S. Varun Kumar and R. Panneerselvam, "A study of crossover operators for genetic algorithms to solve vrp and its variants and new sinusoidal motion crossover operator," *Int. J. Comput. Intell. Res*, vol. 13, no. 7, pp. 1717–1733, 2017.

[31] S. Chakraborty and S. R. Satti, "Space-efficient algorithms for maximum cardinality search, its applications, and variants of BFS," *J. Combinat. Optim.*, vol. 37, no. 2, pp. 465–481, Feb. 2019.

[32] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proc. 2nd Int. Conf. Simul. Tools Techn. (ICST)*, 2009, p. 55.

[33] D. Kotz and T. Henderson, "CRAWDAD: A community resource for archiving wireless data at dartmouth," *IEEE Pervas. Comput.*, vol. 4, no. 4, pp. 12–14, Oct. 2005.

[34] J. R. Gilbert, C. Moler, and R. Schreiber, "Sparse matrices in MATLAB: Design and implementation," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 1, pp. 333–356, Jan. 1992.

JIAGAO WU (Member, IEEE) was born in Nanjing, Jiangsu, China, in 1969. He received the B.S. and M.S. degrees in physics from Nanjing University, Nanjing, in 1992 and 1995, respectively, and the Ph.D. degree in computer science from Southeast University, Nanjing, in 2006.

Since May 2007, he has been an Associate Professor with the School of Computer, Nanjing University of Posts and Telecommunications, Nanjing. His current research interests include mobile computing, machine learning, routing algorithms in mobile ad hoc networks, and delay tolerant networks.

YAHANG GUO received the B.S. degree in material science from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2018, where he is currently pursuing the master's degree in computer science with the School of Computer. His research interests include vehicular delay tolerant networks and routing algorithms.

HONGYU ZHOU received the B.S. degree in applied physics and the M.S. degree in computer science from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2016 and 2019, respectively. His research interests include vehicular delay tolerant networks and routing algorithms.

LU SHEN received the B.S. degree in computer science from the Changshu Institute of Technology, Changshu, China, in 2016, and the M.S. degree in computer science from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2019. His research interests include vehicular delay tolerant networks and machine learning algorithms.

LINFENG LIU received the B.S. and Ph.D. degrees in computer science from Southeast University, Nanjing, China, in 2003 and 2008, respectively. He is currently a Full Professor with the School of Computer, Nanjing University of Posts and Telecommunications, Nanjing. His main research interests include the areas of wireless networks and mobile computing. He has published more than 80 peer-reviewed articles in some technical journals or conference proceedings, such as the IEEE Transactions on Parallel and Distributed Systems, *ACM Transactions on Autonomous and Adaptive Systems*, the IEEE Transactions on Vehicular Technology, *Computer Networks*, the *Journal of Parallel and Distributed Computing*, the *Journal of Network and Computer Applications*, and the IEEE SECON, GLOBECOM, ICC, WCNC, and CSCWD.