International Journal of Aeronautical and Space Sciences
https://doi.org/10.1007/s42405-023-00580-w

ORIGINAL PAPER

Routing Prediction Strategy for UAV Swarm Network Using Pigeon-Inspired Optimization-Based Neural Network

Yong Li[1] · **Mohong Zheng**[1]

Received: 14 September 2022 / Revised: 30 December 2022 / Accepted: 6 February 2023

Abstract
A routing prediction strategy via a pigeon-inspired optimization (PIO)-based neural network (NN) is designed for UAV swarm networks with highly dynamic topology. The proposed strategy can predict the performance of the neighboring nodes as the next hop. For more precise prediction and less computational complexity, the states of the UAV swarm motion and the network are considered as the prior information, and the PIO-based NN framework is established. Based on the system model, PIO is applied to find the optimal weight matrices of the NN-based routing prediction model. The matrix of the hop count index function is calculated using this prediction model. The proposed strategy can directly determine the next hop based on the prediction results or can be combined with other routing methods to maintain a balance between the stability and the shortest path. Numerical simulations are conducted to demonstrate the effectiveness of the proposed strategy.

Keywords UAV swarm network · High-dynamic topology · Routing prediction strategy · Neural network · Pigeon-inspired optimization

1 Introduction

UAV swarms play an important role in disaster relief, city management, geological reconnaissance and other difficult tasks for humans [1]. Such swarms can work in a complex and dangerous environment with low cost and fewer casualties [2]. The cooperative work of a UAV swarm depends on communication, which requires determining an appropriate routing path [3].

It is difficult for the traditional routing protocol to meet the demands of UAV swarm networks because UAVs move at a high speed and the relative motion among UAV swarms is strenuous [4]. Therefore, the network topology is highly dynamic and cannot be used as prior information when determining the routing path. To address these problems, many studies have focused on routing methods based on machine learning (ML) since they require less prior information and are more flexible [5, 6]. The ML algorithms applied to routing methods include reinforcement learning (RL) [7–9], neural networks (NNs) [5, 10–12], swarm intelligence algorithms [13], and combinations of different ML algorithms [13–17]. Although the ML-based routing method is intelligent, it leads to routing oscillations since the ML algorithms are not robust.

To find the routing path more intelligently and stably, some researchers have employed ML to evaluate the routing information, such as the performance of the neighbor node and the routing path [18–20]. In this case, ML algorithms did not directly determine the next hop, but the predicted results of the routing information were provided as the reference for routing. These ML-based routing methods can estimate the optimal routing path for moving nodes or predict the performance of neighboring nodes through trial-and-error structures, imitating path searching and optimal problem solving [21, 22]. However, some ML algorithms that depend on online learning are difficult to apply. These algorithms occupy much of the computing resources of the UAVs [23, 24]. In addition, some ML-based routing information prediction methods ignore the features of UAV movement. Those features can be the prior information for more precise prediction and can be easily obtained by the navigation and control systems of the UAV [25].

To apply the predicted routing information to the UAV swarm network for routing path determination, a routing prediction strategy based on the combination of PIO and NN (PIONN) is proposed. The strategy includes offline training

✉ Mohong Zheng
mohong_zheng@163.com

1 The 7th Research Institute of China Electronics Technology Group Corporation, Guangzhou 51000, China

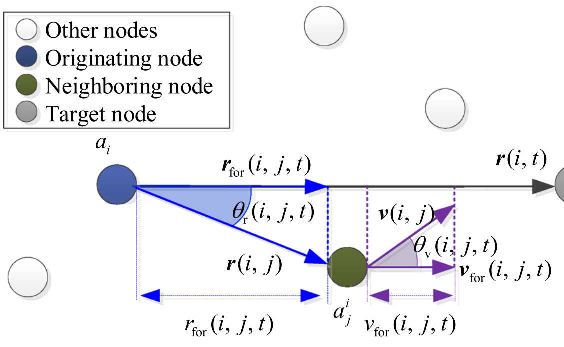


Fig. 1 Motion feature of UAV swarm

and routing prediction. In offline training, the states of the UAV swarm and network are used as the prior information for precise prediction. The PIO method is applied to find the optimal weight matrices of the NN-based routing prediction model, which can reduce the computational complexity and can avoid a differential in the learning process. In routing prediction, the matrix of the hop count index function is calculated according to the PIONN routing prediction (PIONNRP) model.

The remainder of this paper is organized as follows: in Sect. 2, the features of UAV movement are extracted as the prior information for the routing prediction model, and the PIONN framework is designed. In Sect. 3, the routing prediction strategy based on PIONN is presented. In Sect. 4, simulations are conducted to evaluate the performance of PIONNRP. Section 5 presents the conclusions of this paper.

2 System Model

2.1 Feature Extraction of Movement

The UAV swarm includes N_{UAV} nodes and $\mathcal{A} = \{a_1, a_2, \cdots, a_{N_{\text{UAV}}}\}$ denotes the set of the swarm, with node a_i $(i \in [1, N_{\text{UAV}}], \ i \in \mathbb{N}^+)$. Let node $a_j^i \in \mathcal{A}_{\text{NEI}}^i$ be the neighbor of node a_i, in which $\mathcal{A}_{\text{NEI}}^i = \left\{a_j^i \middle| \text{hop}(a_i, a_j) = 1\right\}$ is the set of neighbors and $\text{hop}(a_i, a_j) \in \mathbb{N}^+$ is the function of the minimum hop count from node a_i to node a_j.

Determining a proper next hop can optimize a routing path. The next hop for less $\text{hop}(a_i, a_t)$ may be the neighbor of the originating node a_i, flying toward the target node a_t. Hence, the relative position and relative velocity among the originating node, neighboring node and target node should be considered.

As shown in Fig. 1, to evaluate the performance of node a_j^i as the next hop, the forward distance $r_{\text{for}}(i, j, t)$ and forward speed $v_{\text{for}}(i, j, t)$ are defined as:

$$\begin{cases} r_{\text{for}}(i, j, t) = \|\mathbf{r}_{\text{for}}(i, j, t)\| = \|\mathbf{r}(i, j)\cos[\theta_{\text{r}}(i, j, t)]\| \\ v_{\text{for}}(i, j, t) = \|\mathbf{v}_{\text{for}}(i, j, t)\| = \|\mathbf{v}(i, j)\cos[\theta_{\text{v}}(i, j, t)]\| \end{cases}, \tag{1}$$

where $\mathbf{r}_{\text{for}}(i, j, t) \in \mathbb{R}^3$ and $\mathbf{v}_{\text{for}}(i, j, t)$ are the forward position and forward velocity among the originating node a_i, neighboring node a_j^i and target node $a_t \in \mathcal{A}$, with $t \neq i, j$, respectively. $\mathbf{r}(i, j) \in \mathbb{R}^3$ is the position between node a_i and node a_j^i, and $\mathbf{v}(i, j) \in \mathbb{R}^3$ is the velocity of node a_j^i. $\theta_{\text{r}}(i, j, t) \in \mathbb{R}$ is the angle between $\mathbf{r}(i, j)$ and $\mathbf{r}(i, t)$, and $\theta_{\text{v}}(i, j, t) \in \mathbb{R}$ is the angle between $\mathbf{v}(i, j)$ and $\mathbf{r}(i, t)$.

2.2 Routing Prediction Model Based on the PIONN Framework

2.2.1 Routing Prediction Model

The hop count index function of the neighbor node a_j^i is defined to evaluate the performance of this neighbor as the next hop to the target node a_t. In a network with a known path, the hop count index function $H(a_j^i, a_t) \in \mathbb{R}$ is the minimum hop count from node a_j^i to node a_t. Otherwise, it is obtained by prediction.

$$H(a_j^i, a_t) = \begin{cases} \text{hop}(a_j^i, a_t), & \text{known path} \\ \hat{h}(a_j^i, a_t) & , \text{ else} \end{cases}, \tag{2}$$

where $\hat{h}(a_j, a_t) \in \mathbb{R}$ is the predicted hop count index function of the neighbor node a_j^i.

$\hat{h}(a_j, a_t)$ is predicted using PIONN, which is the fully connected neural network with two hidden layers. The first hidden layer includes 16 neurons, while the second hidden layer includes 4 neurons.

The input $\mathbf{u}(i, j, t) \in \mathbb{R}^4$ and output $\hat{y}(i, j, t) \in \mathbb{R}$ are designed as:

$$\begin{cases} \mathbf{u}(i, j, t) = \left[r_{\text{for}}(i, j, t) \ v_{\text{for}}(i, j, t) \ T_{\text{de}}(i, j) \ B(i, j) \right]^{\text{T}} \\ \hat{y}(i, j, t) = \hat{h}(a_j^i, a_t) \end{cases}, \tag{3}$$

where $T_{de}(i, j) \in \mathbb{R}$ and $B(i, j) \in \mathbb{R}$ are the delay and maximum bandwidth from node a_i to a_j^i, respectively.

And the hop count index function is predicted by:

$$\begin{cases} \mathbf{z}_1(i, j, t) = \tanh[\mathbf{W}_{\text{uh}}\mathbf{u}(i, j, t)] \\ \mathbf{z}_2(i, j, t) = \tanh[\mathbf{W}_{\text{hh}}\mathbf{z}_1(i, j, t)], \\ \hat{y}(i, j, t) = \mathbf{W}_{\text{hy}}\mathbf{z}_2(i, j, t) \end{cases} \tag{4}$$

where $\mathbf{W}_{\text{uh}} \in \mathbb{R}^{16\times 4}$, $\mathbf{W}_{\text{hh}} \in \mathbb{R}^{4\times 16}$ and $\mathbf{W}_{\text{hy}} \in \mathbb{R}^{1\times 4}$ are the weight matrices, and $\mathbf{z}_1(i, j, t) \in \mathbb{R}^{16}$ and $\mathbf{z}_2(i, j, t) \in \mathbb{R}^4$ are the outputs of the first and second hidden layers, respectively.

2.2.2 PIONN Framework

To accurately predict the hop count index function, it is necessary to obtain the optimal weight matrices of the NN-based routing prediction model. To address this problem, the PIO-based NN framework is proposed.

Definition 1 In the learning process of the PIONN, let $\mathcal{W}_{\text{uh}} \in \mathbb{R}^{16\times 4}$, $\mathcal{W}_{\text{hh}} \in \mathbb{R}^{4\times 16}$, and $\mathcal{W}_{\text{hy}} \in \mathbb{R}^{1\times 4}$ be three different pigeon groups:

$$\begin{cases} \mathcal{W}_{\text{uh}} \triangleq \{\mathbf{W}_{\text{uh}}(1),\ \mathbf{W}_{\text{uh}}(2),\ \ldots,\ \mathbf{W}_{\text{uh}}(N_{\text{set}})\} \\ \mathcal{W}_{\text{hh}} \triangleq \{\mathbf{W}_{\text{hh}}(1),\ \mathbf{W}_{\text{hh}}(2),\ \ldots,\ \mathbf{W}_{\text{hh}}(N_{\text{set}})\} \\ \mathcal{W}_{\text{hy}} \triangleq \left\{\mathbf{W}_{\text{hy}}(1),\ \mathbf{W}_{\text{hy}}(2),\ \ldots,\ \mathbf{W}_{\text{hy}}(N_{\text{set}})\right\} \end{cases}, \tag{5}$$

where N_{set} is the number of sample pairs in the database. This means that each pigeon group includes N_{set} individuals.

Definition 2 Pigeon groups $\mathcal{W}_{\text{uh}}$, $\mathcal{W}_{\text{hh}}$, and $\mathcal{W}_{\text{hy}}$ move in the spaces $\mathcal{V}_{\text{uh}} \in \mathbb{R}^{16\times 4}$, $\mathcal{V}_{\text{hh}} \in \mathbb{R}^{4\times 16}$, and $\mathcal{V}_{\text{hy}} \in \mathbb{R}^{1\times 4}$, respectively. At the k^{th} iteration, the positions of the pigeon groups are:

$$\begin{cases} \mathcal{W}_{\text{uh}}(k) \triangleq \{\mathbf{W}_{\text{uh}}(1,\ k),\ \mathbf{W}_{\text{uh}}(2,\ k),\ \ldots,\ \mathbf{W}_{\text{uh}}(N_{\text{set}},\ k)\} \\ \mathcal{W}_{\text{hh}}(k) \triangleq \{\mathbf{W}_{\text{hh}}(1,\ k),\ \mathbf{W}_{\text{hh}}(2,\ k),\ \ldots,\ \mathbf{W}_{\text{hh}}(N_{\text{set}},\ k)\} \\ \mathcal{W}_{\text{hy}}(k) \triangleq \left\{\mathbf{W}_{\text{hy}}(1,\ k),\ \mathbf{W}_{\text{hy}}(2,\ k),\ \ldots,\ \mathbf{W}_{\text{hy}}(N_{\text{set}},\ k)\right\} \end{cases}. \tag{6}$$

Three pigeon groups perform a cooperative flight to decrease the error function of the output layer $e_{\text{y}}(p, k) \in \mathbb{R}$:

$$e_{\text{y}}(p, k) = \tilde{y}(k) - \hat{y}(p, k), \tag{7}$$

with the theoretical output of the output layer $\tilde{y}(k) = \tilde{h}(k)$.

To satisfy $\lim_{k\to k_{\max}} e_{\text{y}}(p, k) = 0$ with the order of the last iteration $k_{\max}$, the center positions $\mathbf{W}^{\text{c}}_{\text{uh}}(k)$, $\mathbf{W}^{\text{c}}_{\text{hh}}(k)$, and $\mathbf{W}^{\text{c}}_{\text{hy}}(k)$ are defined as the guidance. All individuals in the pigeon groups fly toward its center, as shown in Fig. 2 with $N_{\text{set}} = 6$ as an example.

The central positions $\mathbf{W}^{\text{c}}_{\text{uh}}(k)$, $\mathbf{W}^{\text{c}}_{\text{hh}}(k)$, and $\mathbf{W}^{\text{c}}_{\text{hy}}(k)$ are calculated based on the quality of the p^{th} pigeon individual $\text{fitness}_{\text{N1}}(p, k)$, $\text{fitness}_{\text{N2}}(p, k)$, and $\text{fitness}_{\text{y}}(p, k)$:

$$\mathbf{W}^{\text{c}}_{\text{hy}}(k) = \frac{\sum_{p=0}^{N_{\text{set}}} \mathbf{W}_{\text{hy}}(p, k)\text{fitness}_{\text{y}}(p, k)}{N_{\text{set}} \sum_{p=0}^{N_{\text{set}}} \text{fitness}_{\text{y}}(p, k)}, \tag{8a}$$

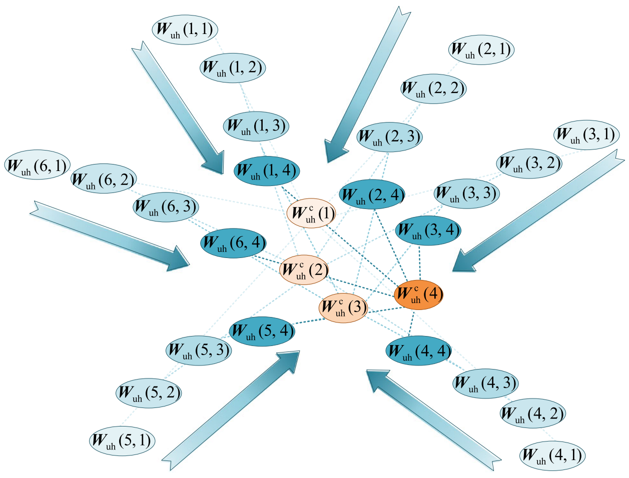


Fig. 2 The movement of pigeon group $\mathcal{W}_{\text{uh}}$

$$\mathbf{W}^{\text{c}}_{\text{hh}}(k) = \frac{\sum_{p=0}^{N_{\text{set}}} \mathbf{W}_{\text{hh}}(p, k)\text{fitness}_{\text{N2}}(p, k)}{N_{\text{set}} \sum_{p=0}^{N_{\text{set}}} \text{fitness}_{\text{N2}}(p, k)}, \tag{8b}$$

$$\mathbf{W}^{\text{c}}_{\text{uh}}(k) = \frac{\sum_{p=0}^{N_{\text{set}}} \mathbf{W}_{\text{uh}}(p, k)\text{fitness}_{\text{N1}}(p, k)}{N_{\text{set}} \sum_{p=0}^{N_{\text{set}}} \text{fitness}_{\text{N1}}(p, k)}, \tag{8c}$$

in which:

$$\begin{cases} \text{fitness}_{\text{y}}(p, k) = \left\| e_{\text{y}}(p, k) \right\| \\ \text{fitness}_{\text{N2}}(p, k) = \|\mathbf{e}_{\text{N2}}(p, k)\| \\ \text{fitness}_{\text{N1}}(p, k) = \|\mathbf{e}_{\text{N1}}(p, k)\| \end{cases}, \tag{9}$$

where $\mathbf{e}_{\text{N1}}(p, k) \in \mathbb{R}^{16\times 1}$ and $\mathbf{e}_{\text{N2}}(p, k) \in \mathbb{R}^{4\times 1}$ are the virtual error function of the first and second hidden layers, respectively:

$$\begin{cases} \mathbf{e}_{\text{N2}}(p, k) = \tilde{\mathbf{z}}_2(p, k) - \mathbf{z}_2(p, k) \\ \mathbf{e}_{\text{N1}}(p, k) = \tilde{\mathbf{z}}_1(p, k) - \mathbf{z}_1(p, k) \end{cases}. \tag{10}$$

Definition 3 In the kth iteration, the virtual target output of the first hidden layer $\tilde{\mathbf{z}}_1(p, k) \in \mathbb{R}^{16\times 1}$ and the virtual target output of the second hidden layer $\tilde{\mathbf{z}}_2(p, k) \in \mathbb{R}^{4\times 1}$ of the pth sample pair are defined as:

$$\begin{cases} \tilde{\mathbf{z}}_2(p,\ k) \triangleq \underset{\mathbf{z}_2^{\#}\in\mathbb{R}^{4\times 1}}{\arg} \left[\mathbf{W}_{\text{hy}}(p,\ k)\mathbf{z}_2^{\#} := \tilde{y}(p)\right] \\ \tilde{\mathbf{z}}_1(p,\ k) \triangleq \underset{\mathbf{z}_1^{\#}\in\mathbb{R}^{16\times 1}}{\arg} \left\{\tanh\left[\mathbf{W}_{\text{hh}}(p,\ k)\mathbf{z}_1^{\#}\right] := \tilde{\mathbf{z}}_2(p,\ k)\right\} \end{cases}. \tag{11}$$

Based on this definition, the virtual error functions $\mathbf{e}_{\text{N1}}(p, k)$ and $\mathbf{e}_{\text{N2}}(p, k)$ are derivate by:

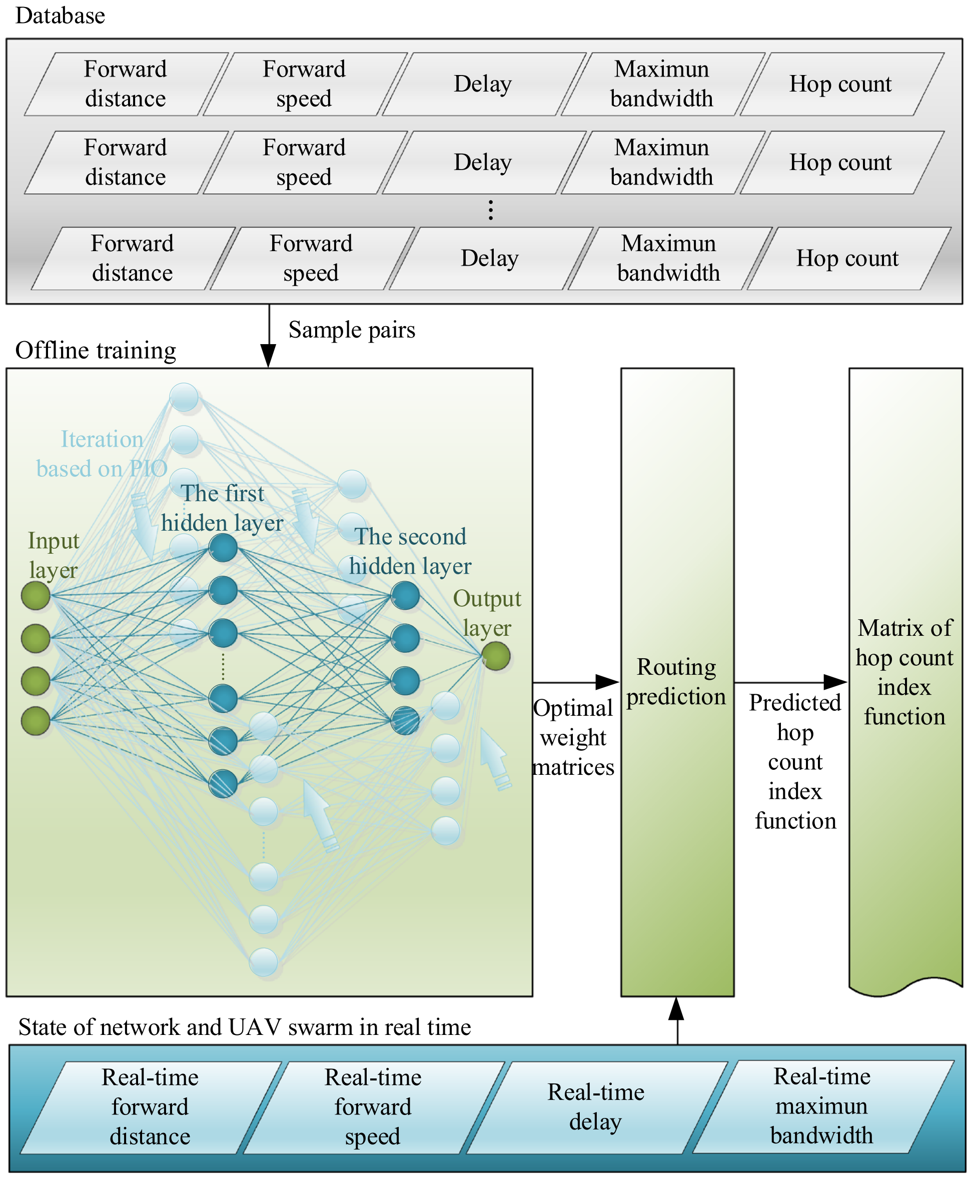


Fig. 3 The structure of the routing prediction based on PIONN

$$
\begin{aligned}
\mathbf{e}_{\mathrm{N2}}(p,k) &= \tilde{z}_2(p,k) - z_2(p,k) \\
&= \mathbf{W}^{+}_{\mathrm{hy}}(p,\ k)\big(\mathbf{W}_{\mathrm{hy}}(p,\ k)\tilde{z}_2(p,k) - \mathbf{W}_{\mathrm{hy}}(p,\ k)z_2(p,k)\big) \\
&= \mathbf{W}^{+}_{\mathrm{hy}}(p,\ k)(\tilde{y}(p,k) - y(p,k)) \\
&= \mathbf{W}^{+}_{\mathrm{hy}}(p,\ k)e_{\mathrm{y}}(p,k),
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
\mathbf{e}_{\mathrm{N1}}(p,k) &= \tilde{z}_1(p,k) - z_1(p,k) \\
&= \mathbf{W}^{+}_{\mathrm{hh}}(p,\ k)(\mathbf{W}_{\mathrm{hh}}(p,\ k)\tilde{z}_1(p,k) - \mathbf{W}_{\mathrm{hh}}(p,\ k)z_1(p,k)) \\
&= \mathbf{W}^{+}_{\mathrm{hh}}(p,\ k)(\tilde{z}_2(p,k) - z_2(p,k)) \\
&= \mathbf{W}^{+}_{\mathrm{hh}}(p,\ k)\mathbf{e}_{\mathrm{N2}}(p,k),
\end{aligned}
\tag{13}
$$

where, $\mathbf{W}^{+}_{\mathrm{hy}}(p,\ k)$ and $\mathbf{W}^{+}_{\mathrm{hh}}$ are the pseudoinverse matrices of $\mathbf{W}_{\mathrm{hy}}(p,k)$ and $\mathbf{W}_{\mathrm{hh}}(p,k)$, respectively.

$\mathbf{W}_{\mathrm{hy}}(p,k)$ and $\mathbf{W}_{\mathrm{hh}}(p,k)$ are not square matrices, the pseudoinverse solutions are used and Eqs. (12)–(13) are rewritten as:

$$
\mathbf{e}_{\mathrm{N2}}(p,k) = \mathbf{W}^{\mathrm{T}}_{\mathrm{hy}}(p,k)\Big(\mathbf{W}_{\mathrm{hy}}(p,k)\mathbf{W}^{\mathrm{T}}_{\mathrm{hy}}(p,k)\Big)^{-1} e_{\mathrm{y}}(p,k),
\tag{14}
$$

$$
\mathbf{e}_{\mathrm{N1}}(p,k) = \mathbf{W}^{\mathrm{T}}_{\mathrm{hh}}(p,k)\big(\mathbf{W}_{\mathrm{hh}}(p,k)\mathbf{W}^{\mathrm{T}}_{\mathrm{hh}}(p,k)\big)^{-1}\mathbf{e}_{\mathrm{N2}}(p,k).
\tag{15}
$$

3 Routing Prediction Strategy Based on PIONN

The routing prediction strategy is designed to predict the hop count index function $H(a^i_j,\ a_t)$, which evaluates the performance of the neighbor node a^i_j as the next hop to the target node a_t. The prediction results are used as a reference when choosing the route. The proposed strategy provides only the predicted hop count index function, and it can directly select the next hop or can be combined with most routing methods. In this case, the routing choice can be more intelligent but still stable.

The structure of the proposed routing prediction strategy is shown in Fig. 3. Before the mission, to find the routing prediction model, offline training based on PIONN is conducted with the database, which is created by the historical state of the network and the UAV swarm and its corresponding hop count index function. Since the routing path is known in the previous missions, the hop count index function is the hop count from the neighbor node to the target node.

During the mission, the hop count index function of each neighbor node is predicted based on the routing prediction model and the state of the network and UAV swarm in real time. The real-time network state includes the real-time delay $T_{de}(i,\ j)$ and maximum bandwidth $B(i,\ j)$ from the originating node a_i to its neighbor node a^i_j, while the real-time state of the UAV swarm includes the real-time forward distance $r_{for}(i,\ j,\ t)$ and forward speed $v_{for}(i,\ j,\ t)$ from the originating node a_i to the target node a_t with node a^i_j as the next hop.

3.1 Offline Training

The offline training procedures based on PIONN are shown in Fig. 4:

Step 1: Importing database

The input and output of the PIONN are obtained based on the database:

$$\begin{cases} \mathbf{u}(p) = \left[r_{for}(p)\ v_{for}(p)\ T_{de}(p)\ B(p) \right]^{T} \\ \tilde{y}(p) = H(p) \end{cases}, \tag{16}$$

where $p \le N_{set}$ $(P \in \mathbb{N}^{+})$ is the order of the sample pair.

Step 2: Initialization of the learning process

The initial order of the learning process is set as $k = 1$, and all initial weight matrices $\mathbf{W}_{uh}(p, 0)$, $\mathbf{W}_{hh}(p, 0)$ and $\mathbf{W}_{hy}(p, 0)$ are chosen.

Step 3: Calculation of the output and error functions of each layer

The output of the first and second hidden layers and the output of the output layer are obtained by:

$$\begin{cases} \mathbf{z}_1(p, k) = \tanh[\mathbf{W}_{uh}(p, k)\mathbf{u}(p, k)] \\ \mathbf{z}_2(p, k) = \tanh[\mathbf{W}_{hh}(p, k)\mathbf{z}_1(p, k)]. \\ \hat{y}(p, k) = \mathbf{W}_{hy}(p, k)\mathbf{z}_2(p, k) \end{cases} \tag{17}$$

The error function of the output layer $e_y(p, k)$ is obtained based on Eq. (7), while the virtual error functions of the first and second hidden layers are obtained based on Eqs. (14), (15).

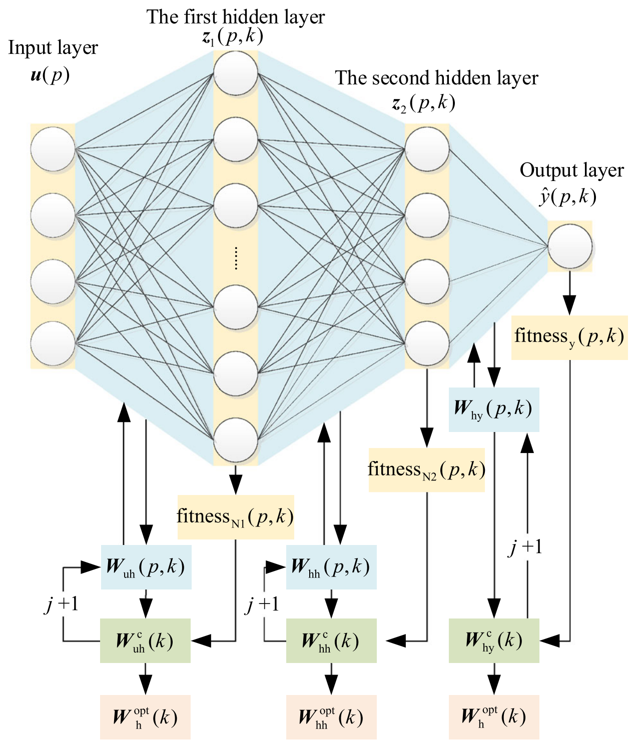


Fig. 4 Offline training

Step 4: Training the pigeon groups

The quality of the p^{th} pigeon individual $\text{fitness}_{N1}(p,\ k)$, $\text{fitness}_{N2}(p,\ k)$, and $\text{fitness}_{y}(p,\ k)$ are calculated based on Eq. (9). The central positions $\mathbf{W}^c_{uh}(k)$, $\mathbf{W}^c_{hh}(k)$, and $\mathbf{W}^c_{hy}(k)$ are obtained based on Eq. (8). Then, each pigeon individual flies toward its central position:

$$\begin{cases} \mathbf{W}_{hy}(p,\ k+1) = \mathbf{W}_{hy}(p,\ k) + \eta_{hy}\left[\mathbf{W}^c_{hy}(k) - \mathbf{W}_{hy}(p,\ k)\right] \\ \mathbf{W}_{hh}(p,\ k+1) = \mathbf{W}_{hh}(p,\ k) + \eta_{hh}\left[\mathbf{W}^c_{hh}(k) - \mathbf{W}_{hh}(p,\ k)\right] \\ \mathbf{W}_{uh}(p,\ k+1) = \mathbf{W}_{uh}(p,\ k) + \eta_{uh}\left[\mathbf{W}^c_{uh}(k) - \mathbf{W}_{uh}(p,\ k)\right] \end{cases}, \tag{18}$$

where η_{hy}, η_{hh}, and η_{uh} are the learning steps.

Step 5: Determination

If $\lim_{k\to k_{max}} e_y(p,\ k) \le \varepsilon$ or $k = k_{max}$ is satisfied with a small positive value ε, the learning process ends. The optimal weight matrices are:

$$\begin{cases} \mathbf{W}^{opt}_{hy} = \mathbf{W}^c_{hy}(k) \\ \mathbf{W}^{opt}_{hh} = \mathbf{W}^c_{hh}(k), \\ \mathbf{W}^{opt}_{uh} = \mathbf{W}^c_{uh}(k) \end{cases} \tag{19}$$

or let $k = k + 1$ and go to Step 3.

Table 1 Routing strategies in simulations

Routing strategy	Topology	Description
PIONNRP	Unknown	Evaluating the performance of PIONNRP without combining with other routing method
PIONNRP + GPSR	Unknown	Evaluating the performance of PIONNRP combining with other routing methods
GPSR	Unknown	For comparison with traditional routing protocol
BPNNRP	Unknown	For comparison with ML-based routing prediction strategy

3.2 Routing Prediction

The hop count index function of j^{th} the neighbor node is predicted by:

$$H(a^i_j,\ a_t) = \hat{h}(a^i_j,\ a_t) = \mathbf{W}^{\text{opt}}_{\text{hy}}\left\{\tanh\left[\mathbf{W}^{\text{opt}}_{\text{hh}}\left(\tanh\left[\mathbf{W}^{\text{opt}}_{\text{uh}}\mathbf{u}(i,\ j,\ t)\right]\right)\right]\right\}, \tag{20}$$

and the matrix of hop count index function is:

$$\boldsymbol{\Gamma}(a_i) = \begin{bmatrix} H(a^i_1,\ a_1) & H(a^i_2,\ a_1) & \cdots & H(a^i_{N_{\text{NEI}}},\ a_1) \\ H(a^i_1,\ a_2) & H(a^i_2,\ a_2) & \cdots & H(a^i_{N_{\text{NEI}}},\ a_2) \\ \vdots & \vdots & \ddots & \vdots \\ H(a^i_1,\ a_{N_{\text{TAR}}}) & H(a^i_2,\ a_{N_{\text{TAR}}}) & \cdots & H(a^i_{N_{\text{NEI}}},\ a_{N_{\text{TAR}}}) \end{bmatrix}, \tag{21}$$

where N_{NEI} and N_{TAR} are the number of the neighbor nodes and target nodes.

4 Performance Evaluation

To evaluate the performance of the proposed routing prediction strategy, four routing strategies are implemented in simulation scenarios, including greedy perimeter stateless routing (GPSR) [26], back-propagation NN-based routing prediction (BPNNRP) strategy [27], PIONNRP, and the combination of PIONNRP and GPSR (PIONNRP + GPSR), as shown in Table 1:

If PIONNRP is implemented without other routing methods, the next hop is selected by the following equation:

$$a_{\text{next}} = \underset{a^j_i}{\arg\min}\left\{\mathbf{W}^{\text{opt}}_{\text{hy}}\left\{\tanh\left[\mathbf{W}^{\text{opt}}_{\text{hh}}\left(\tanh\left[\mathbf{W}^{\text{opt}}_{\text{uh}}\mathbf{u}(i,\ j,\ t)\right]\right)\right]\right\}\right\}. \tag{22}$$

Since PIONNRP is the routing prediction strategy, it can be combined with other routing methods to maintain a balance between stability and less hop count. Therefore, PIONNRP + GPSR is used to determine the next hop. PIONN + GPSR is conducted by a simple strategy:

$$a_{\text{next}} = \begin{cases} \underset{a^j_i}{\arg\min}\left\{\mathbf{W}^{\text{opt}}_{\text{hy}}\left\{\tanh\left[\mathbf{W}^{\text{opt}}_{\text{hh}}\left(\tanh\left[\mathbf{W}^{\text{opt}}_{\text{uh}}\mathbf{u}(i,\ j,\ t)\right]\right)\right]\right\}\right\}, & \text{rand} < \zeta \\ \text{GSRP}(i,\ j,\ t), & \text{else} \end{cases}. \tag{23}$$

The aforementioned equation means that if rand $< \zeta$, the next hop is selected by PIONNRP, or it is selected based on GPSR, where rand $\in [0,\ 1]$ is a random function and $\zeta \in (0,\ 1)$ is a constant.

The parameters of the UAV swarm in the simulation are shown in Table 2, and its position and velocity are shown in Fig. 5. It was flying toward the destination and avoided the forbidden zone. The forbidden zone is shown as a sphere in Fig. 5a.

There are three cases in the simulation, and the half-power beam width (HPBW) among them is different. HPBW is the angular separation in which the magnitude of the radiation pattern decreases by -3 dB from the peak of the main beam. Therefore, there are more neighbors with larger HPBW within limits. The HPBW of case 1 is larger than that of case 2. Similarly, the HPBW of case 2 is larger than that of case 3. Hence, the average number of neighbors of these cases are different. In each case, the number of sample pairs in the database is $N_{\text{set}} = 73500$, and the number of test simulations is 1,225,000. Each case includes four scenarios, and in each scenario, PIONNRP, PIONNRP + GPSR, GPSR, and BPNNRP are implemented. Their parameters are shown in Table 3.

Table 2 Parameters of UAV swarms

Number of nodes	Region	Speed	Transmission distance	Simulation duration
50	$x:\ 0-10$ km $y:\ 0-8$ km $z:\ 8-8.05$ km	0–22 m/s	2 km	500 s

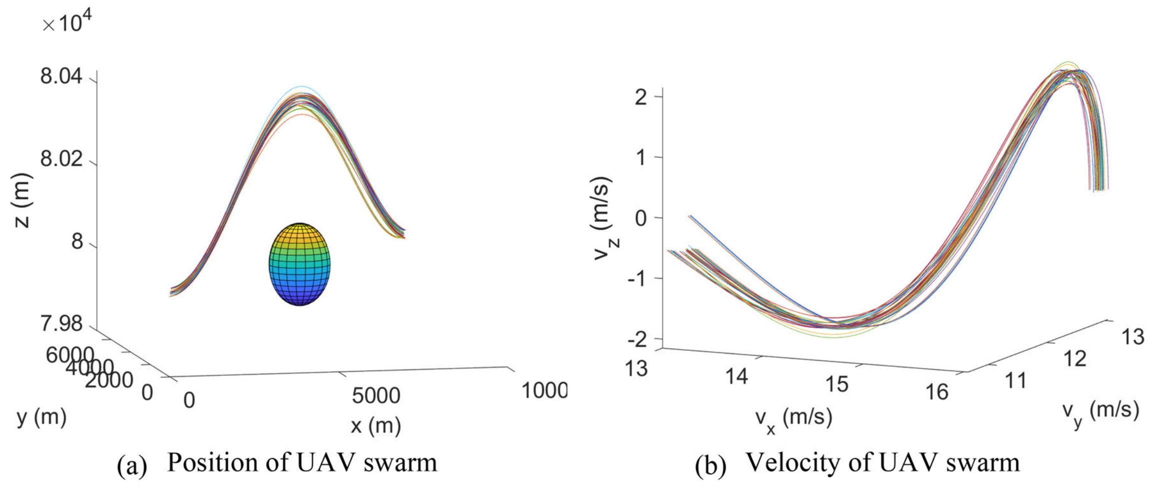


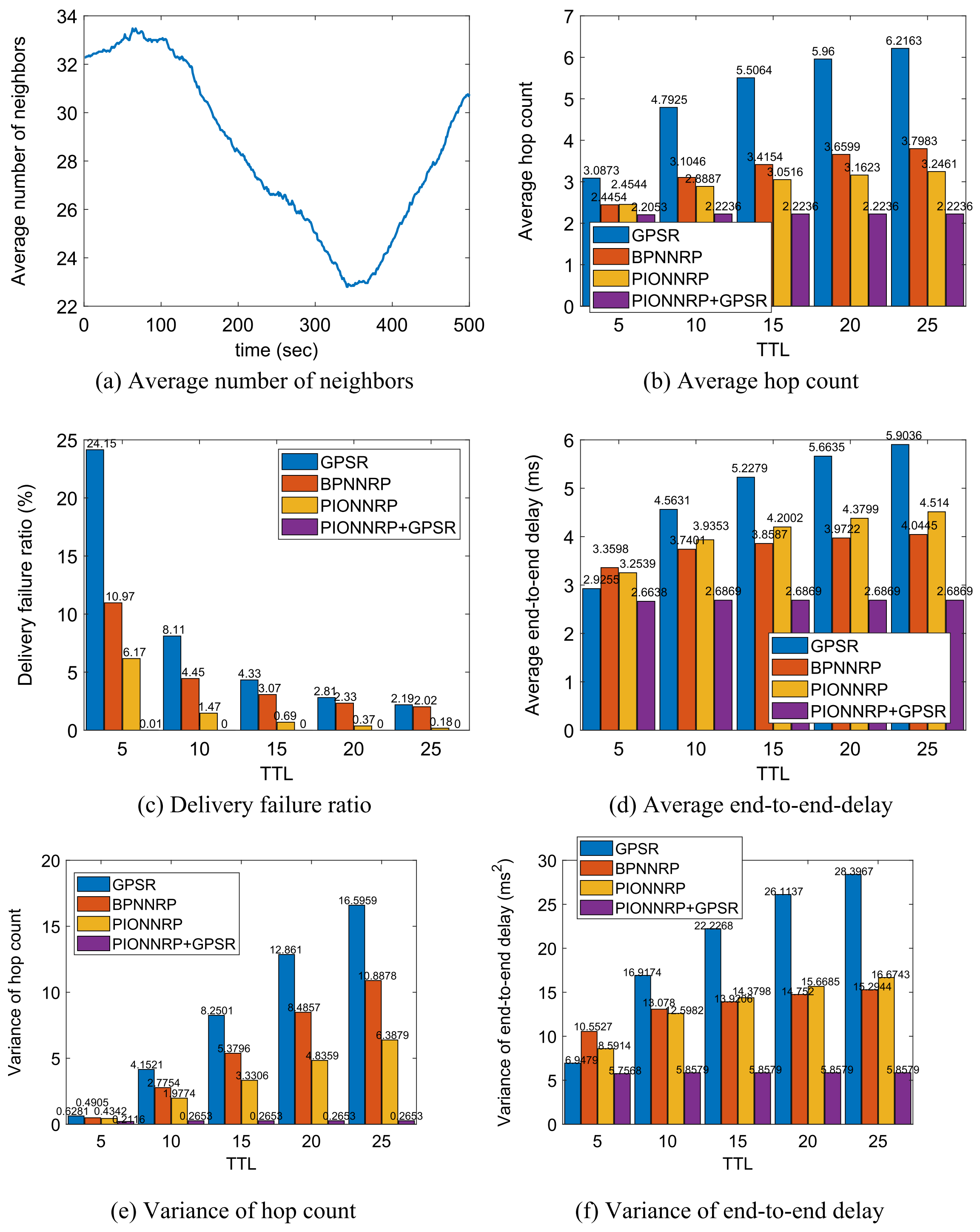
Fig. 5 Movement of UAV swarm

The training times of BPNNRP and PIONNRP were 7311.79 s and 2189.06 s, respectively. Obviously, the training time of PIONNRP was much less than the time of BPNNRP, and the computational complexity of PIONNRP was reduced. For detailed comparison, different time-to-live in networking (TTL) is used in simulations for detailed comparison. TTL is the time limit imposed on the data packet to avoid the problem of circulating forever in the network. It can be regarded as the maximum hop in which the data packet is valid in the network.

The performance evaluations of Case 1 are shown in Fig. 6. In this case, since the average number of neighbors was considerable, as shown in Fig. 6a, it was not difficult to find an appropriate routing path. Therefore, the delivery failure ratios of all routing strategies were less than 10% with TTL $= \{10,\ 15,\ 20,\ 25\}$. PIONNRP and PIONNRP + GPSR performed much better than GPSR and PIONNRP. Compared with those of the traditional routing protocol, the average hop counts of PIONNRP were about approximately 20.5%, 39.7%, 44.6%, 46.9%, and 47.8% less than those of GPSR with varying TTLs because GPSR considered only the distance among nodes and ignored the historical experience. Compared with the ML-based routing strategy, although both PIONNRP and BPNNRP could find a short path, the delivery

Table 3 Parameters of scenarios

Case	Transmission condition	Scenario	Routing strategy	Parameters
1	HPBW: 15°	1	PIONNRP	$\eta_{hh} = \eta_{uh} = 0.1 \times \text{rand}\ \eta_{hy} = 3 \times \text{rand}$
		2	PIONNRP + GPSR	$\eta_{hh} = \eta_{uh} = 0.1 \times \text{rand}\eta_{hy} = 3 \times \text{rand}$ $\zeta = 0.9$
		3	GPSR	–
		4	BPNNRP	Activation function: tanh(·) Number of hidden layers: 2
2	HPBW: 10°	5	PIONNRP	$\eta_{hh} = \eta_{uh} = 0.1 \times \text{rand}\ \eta_{hy} = 3 \times \text{rand}$
		6	PIONNRP + GPSR	$\eta_{hh} = \eta_{uh} = 0.1 \times \text{rand}\eta_{hy} = 3 \times \text{rand}$ $\zeta = 0.9$
		7	GPSR	–
		8	BPNNRP	Activation function: tanh(·) Number of hidden layers: 2
3	HPBW: 5°	9	PIONNRP	$\eta_{hh} = \eta_{uh} = 0.1 \times \text{rand}\eta_{hy} = 3 \times \text{rand}$
		10	PIONNRP + GPSR	$\eta_{hh} = \eta_{uh} = 0.1 \times \text{rand}\eta_{hy} = 3 \times \text{rand}$ $\zeta = 0.9$
		11	GPSR	–
		12	BPNNRP	Activation function: tanh(·) Number of hidden layers: 2



(a) Average number of neighbors

(b) Average hop count

(c) Delivery failure ratio

(d) Average end-to-end-delay

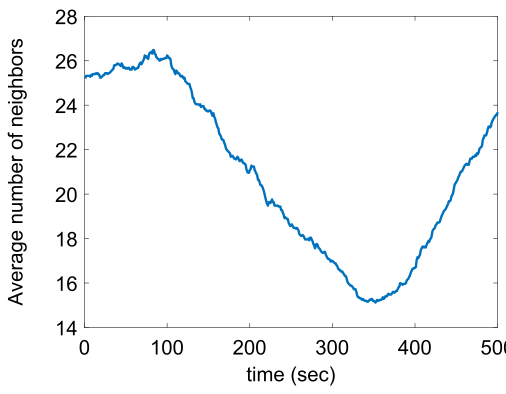
(e) Variance of hop count

(f) Variance of end-to-end delay

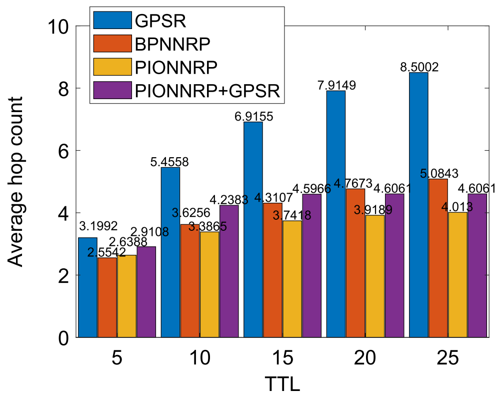
Fig. 6 Performance evaluation with varying TTL of case 1

failure ratios of PIONNRP are 43.8%, 70.0%, 77.5%, 84.1%, and 91.1% less than those of BPNNRP with varying TTL, respectively. In particular, PIONNRP + GPSR performed slightly better than PIONNRP because PIONNRP + GPSR was more stable than PIONNRP with a lower delivery failure ratio.

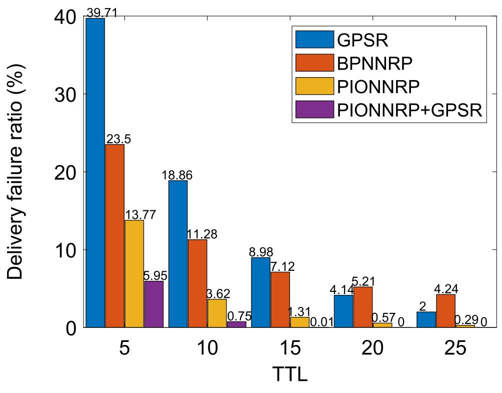
The simulation results of case 2 are shown in Fig. 7. The average number of neighbors is less than that of Case 1 due to the smaller HPBW. Similarly, PIONNRP and PIONNRP + GPSR performed better than GPSR and BPNNRP because the delivery failure ratios, average hop counts, and variance of hop count of GPSR and BPNNRP are larger than those of



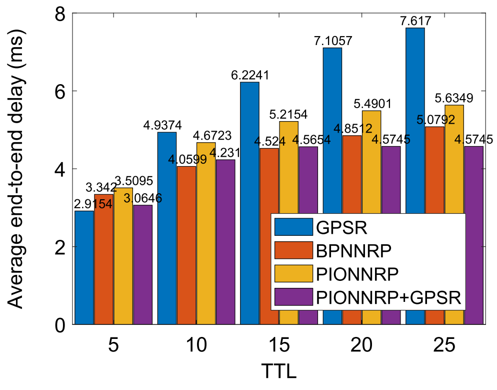
(a) Average number of neighbors



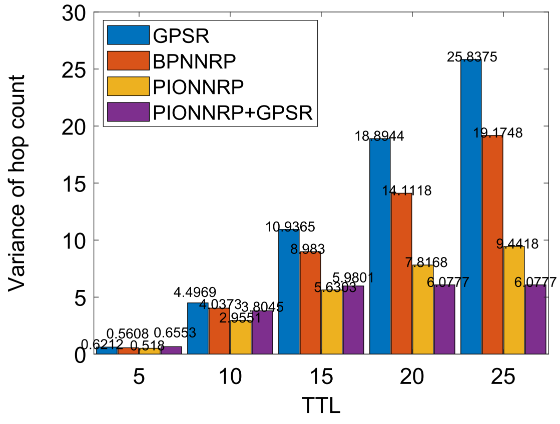
(b) Average hop count



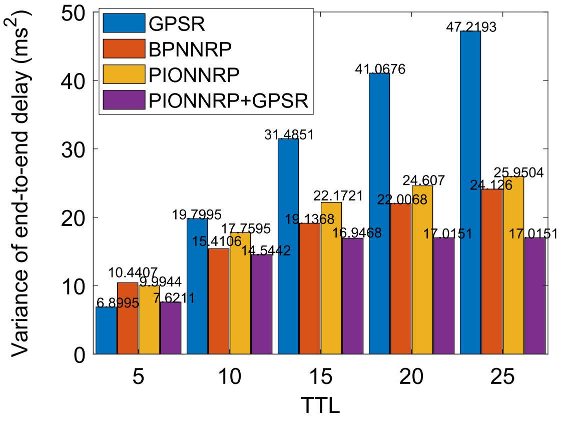
(c) Delivery failure ratio



(d) Average end-to-end-delay



(e) Variance of hop count



(f) Variance of end-to-end delay

Fig. 7 Performance evaluation with varying TTL of case 2

PIONNRP and PIONNRP + GPSR. Contrasting PIONNRP and PIONNRP + GPSR, PIONNRP paid more attention to shorter routing paths, while PIONNRP + GPSR focused on keeping a balance between stability and short paths. In this case, the average hop counts of PIONNRP were 9.3%, 20.1%, 18.6%, 15.0%, and 12.9% less than those of PIONNRP + GPSR with varying TTL, while its delivery failure ratios were 56.8%, 79.3%, and 99.2% larger than those of PIONNRP + GPSR with TTL $= \{5,\ 10,\ 15\}$, respectively. With TTL $= \{20,\ 25\}$, the delivery failure ratios of PIONNRP + GPSR were 0%.

The simulation results of case 3 are shown in Fig. 8. The average number of neighbors decreases due to the smaller HPBW. Compared with that of cases 1–2, the average number

Table 4 Comparisons among routing strategies

Average number of neighbors		11.39	20.83	28.36
Average hop count (TTL = 15)	PIONNRP + GPSR	4.6531	4.5966	2.2236
	PIONNRP	5.3300	3.7417	3.0516
	BPNNRP	6.0023	4.3107	3.4154
	GPSR	8.1341	6.9155	5.5064
Variance of hop count (TTL = 15)	PIONNRP + GPSR	6.1643	5.9801	0.2653
	PIONNRP	11.2581	5.6303	3.3306
	BPNNRP	14.48	8.9830	5.3796
	GPSR	11.5637	10.9365	8.2501
Delivery failure ratio (%) (TTL = 15)	PIONNRP + GPSR	0.20	0.01	0
	PIONNRP	7.06	1.31	0.69
	BPNNRP	28.44	7.12	3.07
	GPSR	19.10	8.98	4.33
Average end-to-end delay (TTL = 15)	PIONNRP + GPSR	4.7345	4.5654	2.6869
	PIONNRP	7.8674	5.2154	4.2002
	BPNNRP	6.8764	4.5240	3.8587
	GPSR	7.3345	6.2241	5.2279
Variance of end-to-end delay (TTL = 15)	PIONNRP + GPSR	19.9536	16.9468	5.8579
	PIONNRP	46.8941	22.1721	14.3798
	BPNNRP	44.2076	19.1368	13.9206
	GPSR	43.6949	31.4851	22.2268

of neighbors in this case decreases and the average delivery failure ratio increases. Similarly, PIONNRP and PIONNRP + GPSR performed better than GPSR and BPNNRP. As shown in Fig. 8c, the delivery failure ratios of GPSR and BPNNRP were unbearable. Comparing PIONNRP and PIONNRP + GPSR, PIONNRP + GPSR still maintained a balance between delivery success ratio and hop count, although the neighbors were not dense.

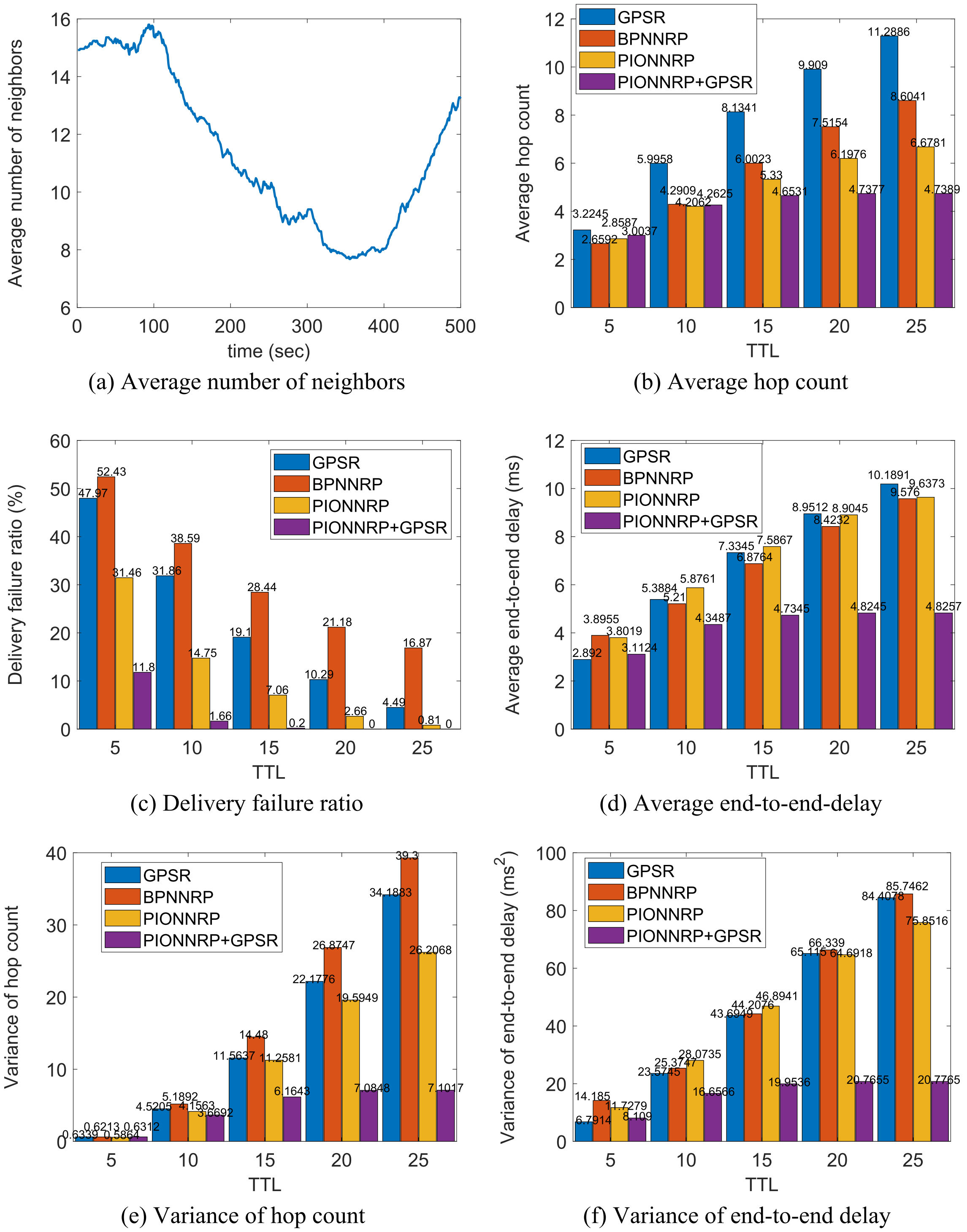
The comparisons among the four routing strategies are shown in Table 4, and the conclusions are given as follows:

(a) The traditional routing protocol, GPSR, failed to find an appropriate routing path for UAV swarm network.
(b) Although BPNNRP could find a shorter path with lower end-to-end delay, compared with GPSR, its delivery failure ratio was still unacceptable.
(c) Compared with GPSR and BPNNRP, PIONNRP could find a shorter path and decrease the delivery failure ratio for the UAV swarm network.
(d) PIONNRP + GPSR performed better than PIONNRP since it could find an appropriate routing path with low end-to-end delay and delivery failure ratio.

Since the combination of PIONNRP and GPSR is better in most cases, it is important to optimally choose ζ. A larger ζ is suggested in the case with higher-dynamic topology because PIONNRP can solve problems with uncertainties.

5 Conclusions

In this paper, a routing prediction strategy is designed based on the combination of PIO and NN for UAV swarm networks. The proposed strategy predicts the performance of neighbors for routing paths through a PIO-based NN framework without the topology as prior information. Therefore, it can be applied to networks with highly dynamic topology. PIONNRP can be implemented to select the next hop according to the prediction results or be combined with other routing methods. Simulation results have demonstrated the efficiency of the proposed strategy. This routing prediction strategy provides a method to apply ML to routing with a balance between intelligence and stability.



(a) Average number of neighbors

(b) Average hop count

(c) Delivery failure ratio

(d) Average end-to-end-delay

(e) Variance of hop count

(f) Variance of end-to-end delay

Fig. 8 Performance evaluation with varying TTL of case 3

Acknowledgements The authors would like to express their acknowledgement for the support by the 2021 Industrial Foundation Reengineering and Manufacturing Development of High Quality (no. tc210h02e).

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Kim J, Oh H, Yu B, Kim S (2020) Optimal task assignment for UAV swarm operations in hostile environments. Int J Aeronaut Space Sci 22:456–467. https://doi.org/10.1007/s42405-020-00317-z
2. Park C, Park S (2021) Directional axis estimation including wind velocity for fixed wing UAV. Int J Aeronaut Space Sci 22:995–1003. https://doi.org/10.1007/s42405-021-00356-0
3. Tan X, Zuo Z, Su S, Guo X, Sun X (2020) Research of security routing protocol for UAV communication network based on AODV. Electron 9:1185–1202. https://doi.org/10.3390/electronics9081185
4. Jung W, Bang H (2021) Fault and failure tolerant model predictive control of quadrotor UAV. Int J Aeronaut Space Sci 22:663–675. https://doi.org/10.1007/s42405-020-00331-1
5. Gharajeh MS (2018) A neural-MCDM-based routing protocol for packet transmission in mobile ad hoc networks. Int J Commun Netw Distrib Syst 21:496–527. https://doi.org/10.1504/IJCNDS.2018.095362
6. Park B, Oh H (2020) Vision-based obstacle avoidance for UAVs via imitation learning with sequential neural networks. Int J Aeronaut Space Sci 21:768–779. https://doi.org/10.1007/s42405-020-00254-x
7. Talekar S, Terdal S (2020) Reinforcement learning based channel selection for design of routing protocol in cognitive radio network. In: 4th international conference on computational systems and information technology for sustainable solution (CSITSS). IEEE, p 19452574. https://doi.org/10.1109/CSITSS47250.2019.9031024
8. Khalid K, Woungang I, Dhurandher SK, Singh J (2021) Reinforcement learning-based fuzzy geocast routing protocol for opportunistic networks. Internet Things 14:100384. https://doi.org/10.1016/j.iot.2021.100384
9. Zhang Y, Zhang Z, Chen L, Wang X (2021) Reinforcement learning-based opportunistic routing protocol for underwater acoustic sensor networks. IEEE Trans Veh Technol 99:2756–2770. https://doi.org/10.1109/TVT.2021.3058282
10. Cardenas LL, Mezher AM, Bautista PB, Leon JPA, Igartua MA (2021) A multimetric predictive ANN-based routing protocol for vehicular ad hoc networks. IEEE Access 99:86037–86053. https://doi.org/10.1109/ACCESS.2021.3088474
11. Gowda CS, Jayasree P (2021) Rendezvous points based energy-aware routing using hybrid neural network for mobile sink in wireless sensor networks. Wirel Netw 27:2961–2976. https://doi.org/10.1007/s11276-021-02630-1
12. Li Y, Zhao W, Cambria E, Wang S, Eger S (2021) Graph routing between capsules. Neural Netw 143:345–354. https://doi.org/10.1016/j.neunet.2021.06.018
13. Li T, Ruan F, Fan Z, Wang J, Kim J (2015) An improved PEGASIS routing protocol based on neural network and ant colony algorithm. Int J Future Gener Commun Netw 8:149–160. https://doi.org/10.1425/ijfgcn.2015.8.6.15
14. Al-Mousawi AJ (2021) Wireless communication networks and swarm intelligence. Wirel Netw 27:1755–1782
15. Akinwande O (2018) Interest forwarding in named data networking using reinforcement learning. Sensors 18:3354–3373. https://doi.org/10.3390/s18103354
16. Mo J, Huang B, Cheng X, Huang C, Wei F (2018) Improving security and stability of ad hoc on-demand distance vector with fuzzy neural network in vehicular ad hoc network. Int J Distrib Sens Netw 14:155014771880619. https://doi.org/10.1177/1550147718806193
17. Liu D, Cui J, Zhang J, Yang C, Hanzo L (2021) Deep reinforcement learning aided packet-routing for aeronautical ad-hoc networks formed by passenger planes. IEEE Trans Veh Technol 70:5166–5171. https://doi.org/10.1109/TVT.2021.3074015
18. Rusek K, Suarez-Varela J, Almasan P, Barlet-Ros P, Cabellos-Aparicio A (2020) RouteNet: leveraging graph neural networks for network modeling and optimization in SDN. IEEE J Sel Areas Commun 38:2260–2270. https://doi.org/10.1109/JSAC.2020.3000405
19. Yao H, Liu H, Zhang P, Wu S, Guo S (2020) A learning-based approach to intra-domain QoS routing. IEEE Trans Veh Technol 69:6718–6730. https://doi.org/10.1109/TVT.2020.2986769
20. Shin C, Lee M (2020) Swarm-intelligence-centric routing algorithm for wireless sensor networks. Sensors 20:5164. https://doi.org/10.3390/S20185164
21. Shao Y, Rezaee A, Liew SC, Chan VWS (2020) Significant sampling for shortest path routing: a deep reinforcement learning solution. IEEE J Sel Areas Commun 38:2234–2248. https://doi.org/10.1109/JSAC.2020.3000364
22. Chen X, Proietti R, Liu CY, Yoo SJB (2021) A multi-task-learning-based transfer deep reinforcement learning design for autonomic optical networks. IEEE J Sel Areas Commun 39:2878–2889. https://doi.org/10.1109/JSAC.2021.3064657
23. Liu D, Cui J, Zhang J, Yang CY, Hanzo L (2021) Deep reinforcement learning aided routing in aeronautical ad hoc networks. IEEE Trans Veh Technol 70:5166–5171. https://doi.org/10.1109/TVT.2021.3074015
24. He Q, Moayyedi A, Dan G, Koudouridis GP, Tengkvist P (2020) A meta-learning scheme for adaptive short-term network traffic prediction. IEEE J Sel Areas Commun 38:2271–2283. https://doi.org/10.1109/JSAC.2020.3000408
25. Chan R, Lim MY, Parthiban R (2021) A neural network approach for traffic prediction and routing with missing data imputation for intelligent transportation system. Expert Syst Appl 171:114573. https://doi.org/10.1016/j.eswa.2021.114573
26. Karp B (2000) GPSR: Greedy perimeter stateless routing for wireless networks. Acm Mobicom. https://doi.org/10.1145/345910.345953
27. Kaur N, Singh K, Kaur H (2014) Back propagation neural network (BPNN) based routing optimization for wireless sensor network. In: International multi track conference on science, engineering and technical innovations