

# 基于测距修正和秃鹫优化的 DV-Hop 改进算法

刘龙飞 倪晓军

(南京邮电大学计算机学院 南京 210023)

**摘要:** 针对传统距离矢量跳(DV-Hop)算法在各向异性网络中定位误差大的问题,提出一种基于测距修正和秃鹫优化的 DV-Hop 改进算法(AVOADV-Hop)。该算法在双通信半径的基础上依据锚节点间的跳数信息引入测距修正阈值,根据锚节点的坐标信息计算基于跳数改进的平均跳距,实现节点间距离估算的分段处理;融合动态调整和加权思想改进非洲秃鹫优化算法取代最小二乘法求解节点坐标。仿真结果表明,在3种不规则网络下,AVOADV-Hop 算法相较于传统 DV-Hop 算法及相关改进算法,定位误差分别下降了43.6%、21.1%和10.5%,定位精度有效提高。

**关键词:** 各向异性网络; DV-Hop 算法; 非洲秃鹫优化算法; 节点定位

**中图分类号:** TP393 **文献标识码:** A **国家标准学科分类代码:** 510.5

## Improved DV-Hop algorithm based on range correction and vulture optimization

Liu Longfei Ni Xiaojun

(School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

**Abstract:** To solve the problem of large positioning error of traditional DV-Hop algorithm in anisotropic network, an improved DV-Hop algorithm based on range correction and vulture optimization (AVOADV-Hop) is proposed. Based on the double communication radius, the algorithm introduces the distance correction threshold according to the hop number information between anchor nodes, and calculates the average hop distance based on the hop number improvement according to the position information of anchor nodes, and realizes the segmentation processing of the distance estimation between nodes. Combining dynamic adjustment and weighted ideas, the African vulture optimization algorithm replaces the least square method to solve the node coordinates. The simulation results show that the positioning errors of AVOADV-Hop algorithm are reduced by 43.6%, 21.1% and 10.5%, respectively, compared with the traditional DV-Hop algorithm and the improved algorithm, and the positioning accuracy is effectively improved in three irregular networks.

**Keywords:** anisotropic networks; DV-Hop algorithm; African vultures optimization algorithm; node localization

### 0 引言

无线传感器网络(wireless sensor network, WSN)因其具有低成本、自主式、动态拓扑等优点,被广泛应用于国防军工、医疗防护、环境监测和智能家居等众多领域。在大多数场景中,传感器节点采集到的数据只有与其位置信息相结合才有意义<sup>[1]</sup>,因此节点定位技术一直是 WSN 中的研究热点之一<sup>[2]</sup>。目前可以将当前提出的 WSN 节点定位算法分为基于测距的定位算法和无需测距的定位算法<sup>[3]</sup>两类。基于测距的定位算法主要是根据到

达角(angle of arrival, AOA)<sup>[4]</sup>、到达时间(time of arrival, TOA)<sup>[5]</sup>、到达时间差(time difference of arrival, TDOA)<sup>[6]</sup>和接收信号强度(received signal strength indication, RSSI)<sup>[7]</sup>等信息估计未知节点(未知坐标信息的节点)到多个锚节点(已知坐标信息的节点)的距离,然后使用三边定位法或最小二乘法获取未知节点的位置信息。这种方法的定位精度很高,但由于需要为网络中的传感器节点配备相应的设备,使得网络的成本和能耗也随之增高<sup>[8]</sup>。

距离矢量跳(distance vector hop, DV-Hop)算法是由

Niculescu 等<sup>[9]</sup>提出的一种无需测距的节点定位算法。该算法实现简单、对传感器节点的硬件要求低,满足无线传感器网络中传感器节点体积小、成本低、能量和算力有限的特点。但该算法的定位精度不高,尤其在网络中存在各种各样的障碍物时,定位误差较大。近年来,随着智能优化算法的发展,许多基于智能优化算法改进 DV-Hop 的方案被提出。文献[10]通过锚节点之间的精确距离计算跳数修正值,并利用改进的模拟退火算法优化平均跳距,降低了未知节点使用同一平均跳距求解距离时由于局部区域的网络拓扑存在结构差异而造成的误差;文献[11]为了提高不规则网络中 DV-Hop 算法的定位精度,引入跳距校正正值来修正平均跳距,并使用基于禁忌思想和加权思想改进的粒子群算法来优化坐标计算方法;文献[12]在双通信半径的基础上利用最小均方差准则优化平均跳距,并引入归一化加权因子调整不同锚节点的权重,最后通过使用改进的麻雀搜索算法计算未知节点坐标,相较于传统 DV-Hop 算法具有更高的定位精度、更好的稳定性;文献[13]提出一种群智能优化算法改进 DV-Hop 算法第 3 阶段的方案,通过区域限定和扰动频率来提高算法的收敛速度和局部搜索能力,相较于其他的改进算法在复杂度和定位精度方面都有一定的改进;文献[14]通过分析原算法的误差从 3 方面进行改进,引入跳数调整因子修正节点间跳数、基于加权思想改进平均跳距以及结合禁忌搜索和遗传算法替代最小二乘法求解节点坐标。

上述算法采用智能优化的方式改进平均跳距和节点坐标的计算方法,提高了节点定位精度,但在各向异性网络中仍存在着复杂度高、收敛速度慢以及受不规则障碍物影响较大等问题。为进一步提高 DV-Hop 算法在各向异性网络中的定位精度,本文结合上述算法的优缺点,从距离估计和坐标计算两个阶段进行改进。基于双通信半径引入测距修正阈值  $\lambda_{hop}$  以及平均跳距  $Hopsiz_{hop}$ , 实现未知节点与锚节点间距离计算的分段处理;使用非洲秃鹫优化算法(African vultures optimization algorithm, AVOA)取代最小二乘法求解未知节点坐标,相较于其他的优化算法,非洲秃鹫优化算法迭代速度快、收敛效率高并且基于饥饿度动态调整更新策略,在各向异性网络中具有更强的寻优能力。仿真结果表明,本文提出的改进算法在各向异性网络中具有更好的稳定性和更高的定位精度。

## 1 相关工作

### 1.1 DV-Hop 算法及误差分析

传统 DV-Hop 算法的定位过程可以分为 3 个阶段:1) 计算节点间的最小跳数值,网络中的传感器节点通过接收和转发锚节点广播的数据包,获得其与该锚节点间的跳数和位置信息;2) 估计未知节点到锚节点的距离,网络中的锚节点通过式(1)计算平均跳距,然后未知节点利用最近锚节点计算的跳数通过式(2)估算出其到网络中所有

锚节点间的距离;3) 计算未知节点坐标,网络中的未知节点利用其估算出的距离和该锚节点的坐标信息,使用最小二乘法计算自身的位置坐标。网络中的传感器节点通过这 3 个阶段可以实现节点自定位。

$$Hopsiz_e = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} hop_{ij}} \quad (1)$$

$$d_{ui} = Hopsiz_e \times hop_{ui} \quad (2)$$

式中:  $(x_i, y_i)$ 、 $(x_j, y_j)$  分别为锚节点  $i$  和  $j$  的坐标;  $hop_{ij}$  表示锚节点  $i$  和  $j$  之间的最小跳数;  $Hopsiz_e$  为距离未知节点  $u$  最近的锚节点计算的跳数;  $hop_{ui}$  为未知节点  $u$  与锚节点  $i$  之间的最小跳数。

DV-Hop 作为一种非测距的定位算法,算法的定位精度受网络连通度、网络密度以及网络中的障碍物影响很大。在各向异性网络中,造成节点定位误差大的主要原因如下。

1) 最小跳数带来的误差。DV-Hop 算法根据通信半径  $R$  将距离小于  $R$  的两个节点统一归为一跳,锚节点与其邻居节点之间的实际距离可以是  $(0, R]$  中的任意实数,但它们的估计距离只能是该锚节点的平均跳距  $Hopsiz_e$ 。根据式(2)可知,当节间的跳数很小时,其估计距离与实际距离偏差很小,但随着节点间跳数的增加,其估算距离与实际距离的偏差将呈线性增大。

2) 路径弯曲带来的误差。DV-Hop 算法是依据距离矢量路由协议进行数据包的转发,对网络的连通性依赖很高,在不规则的网络环境中,由于受到障碍物的影响,数据包的路由路径将与理想状态下的路径发生严重偏离,造成两个节点之间的跳数增加,误差累积更大。如图 1 所示,锚节点  $A$  与锚节点  $B$  受矩形障碍物的影响,两节点间的跳数从理想环境下的 2 跳变成 5 跳,假设  $d_{ab}$  为锚节点  $A$  与  $B$  之间的实际距离,则锚节点  $A$  计算的跳数与实际平均跳距相差  $3d_{ab}/10$ ,未知节点  $C$  测得与锚节点  $A$  的实际距离与理想环境下的距离相差  $(3d_{ab}/2 - 4d_{ab}/5) = 7d_{ab}/10$ ,且这个误差将随着跳数的增加而增大。

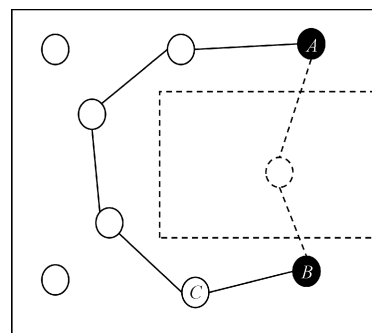


图 1 路径弯曲示意图

Fig. 1 Path bending diagram

3)最小二乘法带来的误差。DV-Hop 算法使用最小二乘法来估算未知节点坐标,使用此类方法的未知节点需满足至少与3个锚节点连通且3个锚节点不在同一直线上的要求<sup>[15]</sup>,在各向异性网络中,未知节点易受到障碍物的影响而无法满足此要求,从而导致无法定位;此外,根据上述的误差分析可知,节点间的跳数越大其估计距离与实际距离的误差越大,若未知节点使用最小二乘法求解坐标,网络中所有的锚节点具有相等的权重,这将使不可靠的锚节点对定位结果造成很大干扰。

针对上述的误差分析,本文在双通信半径的基础上提出一种测距修正和秃鹫优化的 DV-Hop 改进算法,分别针对路径弯曲和最小二乘法带来的误差进行改进,最终达到定位精度提高的目的。

## 1.2 非洲秃鹫优化算法

非洲秃鹫优化算法是由 Abdollahzadeh 等<sup>[16]</sup>提出的一种群智能优化算法,该算法通过模拟自然界中秃鹫的捕食策略来达到搜索解空间、寻找最优解的目的。非洲秃鹫优化算法寻找最优解的具体流程可以分成4个阶段,分别为确定种群最优、计算饥饿度、探索和开发。算法迭代开始时,首先根据适应度函数计算出每只秃鹫的适应度值,判断出当前种群中最好的秃鹫  $\mathbf{X}_{b1}$  和次好的秃鹫  $\mathbf{X}_{b2}$ , 随机选择其中一个秃鹫作为当前最强壮的秃鹫  $\mathbf{X}_s$ ; 然后,根据式(3)计算出每只秃鹫的饥饿度,秃鹫根据饥饿度来选择性的进入探索阶段和开发阶段。

$$F = (2\beta_1 + 1)\beta_2 \left(1 - \frac{t}{T}\right) + \beta_3 \left(\sin^{2.5} \left(\frac{\pi t}{2T}\right) + \cos\left(\frac{\pi t}{2T}\right) - 1\right) \quad (3)$$

式中:  $F$  表示秃鹫饥饿度;  $t$  表示当前的迭代次数;  $T$  表示最大的迭代次数;  $\beta_1$  为  $[0,1]$  内的随机数;  $\beta_2$  为  $[-1,1]$  内的随机数;  $\beta_3$  为  $[-2,2]$  内的随机数。

探索和开发阶段是整个算法的核心,分别影响着秃鹫的全局搜索能力和局部搜索能力,对求解全局最优解至关重要。当秃鹫的饥饿度为  $|F| \geq 1$  时,秃鹫进入探索阶段,在探索阶段中,秃鹫以概率  $1 - P_1$  进入全局搜索模式并以概率  $P_1$  向种群中最强壮秃鹫靠拢。更新策略如下:

$$\mathbf{X}_{(i+1)} = \begin{cases} \mathbf{X}_{S(i)} - |2r_3 \mathbf{X}_{S(i)} - \mathbf{X}_{(i)}| F, & r \leq P_1 \\ \mathbf{X}_{S(i)} + r_1((u-l)r_2 + l) - F, & r > P_1 \end{cases} \quad (4)$$

式中:  $\mathbf{X}_{(i+1)}$  和  $\mathbf{X}_{(i)}$  分别表示第  $i+1$  次和第  $i$  次迭代的秃鹫;  $\mathbf{X}_{S(i)}$  表示第  $i$  次迭代后最强壮的秃鹫;  $u$  和  $l$  为解空间的上限和下限;  $r, r_1, r_2$  和  $r_3$  为  $[0,1]$  内的随机数。

当秃鹫的饥饿度  $|F|$  为  $[0.5,1)$  时,秃鹫进入第1开发阶段,秃鹫以概率  $P_2$  向种群中最强壮的秃鹫移动并以概率  $1 - P_2$  进行螺旋飞行。更新策略如下:

$$\mathbf{X}_{(i+1)} = \begin{cases} |2r_3 \mathbf{X}_{S(i)} - \mathbf{X}_{(i)}| (F + r_4) - \mathbf{X}_{S(i)} + \mathbf{X}_{(i)}, & r < P_2 \\ \mathbf{X}_{S(i)} - (\mathbf{Z}_1 + \mathbf{Z}_2), & r \geq P_2 \end{cases} \quad (5)$$

$$\mathbf{Z}_1 = \mathbf{X}_{S(i)} \times \left(\frac{r_5 \times \mathbf{X}_{(i)}}{2\pi}\right) \times \cos \mathbf{X}_{(i)} \quad (6)$$

$$\mathbf{Z}_2 = \mathbf{X}_{S(i)} \times \left(\frac{r_6 \times \mathbf{X}_{(i)}}{2\pi}\right) \times \sin \mathbf{X}_{(i)} \quad (7)$$

式中:  $r_4, r_5, r_6$  和  $r$  为  $[0,1]$  内的随机数;  $\mathbf{Z}_1$  和  $\mathbf{Z}_2$  表示秃鹫的螺旋飞行方式。

当秃鹫的饥饿度  $|F| < 0.5$  时,秃鹫进入第2开发阶段,秃鹫以概率  $P_3$  进行莱维飞行随机搜索周围空间,以  $1 - P_3$  的概率抢占食物。更新策略如下:

$$\mathbf{X}_{(i+1)} = \begin{cases} \frac{\mathbf{T}_1 + \mathbf{T}_2}{2}, & r \geq P_3 \\ \mathbf{X}_{S(i)} - F |\mathbf{X}_{S(i)} - \mathbf{X}_{(i)}| ly(d), & r < P_3 \end{cases} \quad (8)$$

$$\mathbf{T}_1 = \mathbf{X}_{b1(i)} - \frac{\mathbf{X}_{b1(i)} \mathbf{X}_{(i)}}{\mathbf{X}_{b1} - \mathbf{X}_{(i)}} F \quad (9)$$

$$\mathbf{T}_2 = \mathbf{X}_{b2(i)} - \frac{\mathbf{X}_{b2(i)} \mathbf{X}_{(i)}}{\mathbf{X}_{b2(i)} - \mathbf{X}_{(i)}} F \quad (10)$$

式中:  $r$  为  $[0,1]$  内的随机数;  $ly(d)$  表示莱维飞行;  $\mathbf{X}_{b1(i)}$  与  $\mathbf{X}_{b2(i)}$  分别表示第  $i$  次迭代后当前种群中的最好秃鹫与次好秃鹫。

## 2 本文改进算法

针对最小跳数带来的误差,本文采用双通信半径<sup>[17]</sup>改进最小跳数,网络中所有锚节点以  $0.5R$  和  $R$  为通信半径依次广播数据包,对于通信半径为  $0.5R$  的数据包传感器节点只进行接收不转发,且保存到该锚节点的跳数为  $0.5$ ; 对于通信半径为  $R$  的数据包传感器节点在比较自身保存的跳数信息后,将最小跳数保存后加1转发出去。最终,所有节点将得到与各锚节点的跳数信息。

### 2.1 测距修正

基于上述的误差分析,网络中节点间的通信会出现路径曲折问题,即两个节点间的跳数远远大于理想环境下节点间的跳数,此时通过锚节点计算的平均跳距将远远小于理想环境下的平均跳距,未知节点使用最近锚节点的平均跳距去计算与网络中所有锚节点的距离的方法将产生巨大误差。为此,本文构建跳数与平均跳距的关系式,对距离计算采用分段处理。

1) 计算测距修正阈值  $\lambda_{hop}$ 。网络中的锚节点根据式(11)计算其与网络中其他锚节点的最大跳数  $MaxHop_a$ 。

$$MaxHop_a = \max_{c \in N_a, c \neq a} (hop_{ac}) \quad (11)$$

式中:  $N_a$  为网络中所有锚节点集合。根据式(12)计算测距修正阈值  $\lambda_{hop}$ 。

$$\lambda_{hop} = \min_{a \in N_a} (MaxHop_a) \quad (12)$$

2) 计算修正系数  $\alpha$ 。假设未知节点与锚节点  $A$  之间由于路径曲折而使跳数拉大,为了更准确地估算这两节点的距离,本文使用由相同跳数下的锚节点计算的平均跳距作为该未知节点的平均跳距去计算与锚节点  $A$  的距离,因此构建跳数与平均跳距关系(式(13)),针对特定的跳数

使用不同的平均跳距。

$$Hopsize_{hop} = \alpha_1 hop^2 + \alpha_2 hop + \alpha_3 \quad (13)$$

令：

$$\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} \quad (14)$$

则有：

$$\alpha = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (15)$$

其中：

$$\mathbf{A} = \begin{pmatrix} hop_{i_1}^2 & hop_{i_1} & 1 \\ hop_{i_2}^2 & hop_{i_2} & 1 \\ \vdots & \vdots & \vdots \\ hop_{i_n}^2 & hop_{i_n} & 1 \end{pmatrix} \quad (16)$$

$$\mathbf{B} = \begin{pmatrix} Hopsize_{i_1} \\ Hopsize_{i_2} \\ \vdots \\ Hopsize_{i_n} \end{pmatrix} \quad (17)$$

式中： $hop_{i_1}, hop_{i_2}, \dots, hop_{i_n}$  为网络中锚节点间跳数大于  $\lambda_{hop}$  的值； $Hopsize_{i_n}$  为锚节点间的跳数为  $hop_{i_n}$  下的平均跳距。

$$Hopsize_{i_n} = aver \left( \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{hop_{i_n}} \right) \quad (18)$$

式中： $(x_i, y_i)$  与  $(x_j, y_j)$  是跳数为  $hop_{i_n}$  的两个锚节点。网络中的锚节点计算出修正系数  $\alpha$ ，若求解的系数唯一，网络中的锚节点将计算的修正系数  $\alpha$  广播到整个网络的未知节点；若系数不唯一，则选取  $\alpha_1 < 0$  的一组数据进行广播。

3) 根据式(11)求解未知节点到锚节点的估计距离。

$$Dist_{ui} = \begin{cases} hop_{ui} \times Hopsize_a, & hop_{ui} < \lambda_{hop} \\ hop_{ui} \times Hopsize_{hop_{ui}}, & hop_{ui} \geq \lambda_{hop} \end{cases} \quad (19)$$

式中： $hop_{ui}$  为锚节点  $i$  与未知节点  $u$  之间的最小跳数； $Hopsize_a$  为未知节点  $u$  保存的平均跳距。

## 2.2 求解节点坐标改进

针对最小二乘法带来的误差问题，本文使用 AVOA 算法来求解节点位置坐标。本文将求解节点坐标的问题模拟成秃鹫寻找猎物而进行移动的过程，秃鹫在捕食过程中根据饥饿度调整移动策略，秃鹫所处位置的优劣通过适应度函数来衡量。为了更准确地评估各向异性网络中的定位精度，本文采用加权的锚节点与未知节点的估计距离与经过改进的距离之差作为适应度函数，锚节点到未知节点的跳数越小则权重值  $w_i$  越大，反之，权重值越小。具体适应度函数建模如下：

$$Fitness = \sum_{i=1}^n w_i \times |Dist_{ui} - \sqrt{(x^* - x_i)^2 + (y^* - y_i)^2}| \quad (20)$$

其中：

$$w_i = \frac{\lambda_{hop}}{hop_{ui}} \quad (21)$$

式中： $(x^*, y^*)$  表示当前秃鹫的位置； $(x_i, y_i)$  表示锚节点  $i$  的位置； $Dist_{ui}$  为经过改进后计算的未知节点  $u$  到锚节点  $i$  的距离； $n$  为锚节点的个数； $hop_{ui}$  为改进后的未知节点  $u$  到锚节点  $i$  的最小跳数。

AVOA 算法在探索阶段使用定值参数  $P_1$  确定搜索策略，在整个探索阶段都受到种群中最强壮秃鹫的影响，这会导致算法的全局搜索能力受限，使收敛结果陷入局部最优解陷阱，为了解决这个问题，本文在非洲秃鹫优化算法的探索阶段引入动态调整因子，用于改进探索阶段的全局搜索能力。调整因子的计算公式如下：

$$\omega = \sigma \cdot e^t + \epsilon \quad (22)$$

其中：

$$\sigma = \frac{\omega_{\max} - \omega_{\min}}{e^T - e} \quad (23)$$

$$\epsilon = \frac{e \cdot \omega_{\max} - e^T \cdot \omega_{\min}}{e^T - e} \quad (24)$$

式中： $\sigma$  和  $\epsilon$  为自适应调整常数； $t$  为当前迭代次数； $T$  为最大迭代次数； $\omega_{\max}$  和  $\omega_{\min}$  为调整因子的上下界，本文中设置为 0.8 和 0.2。

根据式(22)可知，在算法迭代初期调整因子  $\omega$  很小，随着迭代次数的增加调整因子  $\omega$  呈指数增加，在最大迭代次数时达到 0.8。为了调整最强壮秃鹫对秃鹫位置更新的影响，引入加权思想，让迭代初期秃鹫的位置更新更注重全局搜索，迭代后期更依赖于最强壮秃鹫的位置。此时，在探索阶段秃鹫的更新策略为：

$$\mathbf{X}_{(i+1)} = \begin{cases} \omega \mathbf{X}_{S(i)} - |2r_3 \mathbf{X}_{S(i)} - \mathbf{X}_{(i)}| F, & r \leq \omega \\ \omega \mathbf{X}_{S(i)} + (1 - \omega)r_1((u - l)r_2 + l) - F, & r > \omega \end{cases} \quad (25)$$

式中： $u$  和  $l$  为解空间的上限和下限； $r, r_1, r_2$  和  $r_3$  为  $[0, 1]$  内的随机数； $F$  为秃鹫的饥饿度。通过动态调整最强壮秃鹫和全局搜索的权重，最终加强了算法的全局搜索能力。

## 2.3 改进算法流程

改进的定位算法的流程如图 2 所示。

1) 根据 DV-Hop 算法的第 1 阶段，在采用双通信半径的基础上，计算网络中所有锚节点与未知节点间的最小跳数。

2) 通过使用式(1)计算网络中的锚节点平均每跳距离  $Hopsize$ ，并使用式(11)计算出该锚节点  $MaxHop$ 。

3) 网络中的锚节点广播一个数据包，该数据包中包含如下 3 部分：锚节点 ID、锚节点计算的  $MaxHop$  以及该锚节点计算的平均跳距  $Hopsize$ ；未知节点只保存第 1 次收到的数据包，然后向网络转发自己收到的数据包，每个锚节点保存并转发来自其他锚节点的数据包。

4) 每个锚节点根据式(12)计算出网络中测距修正阈值  $\lambda_{hop}$ ，通过式(13)~(18)计算出测距修正系数  $\alpha$ ，锚节



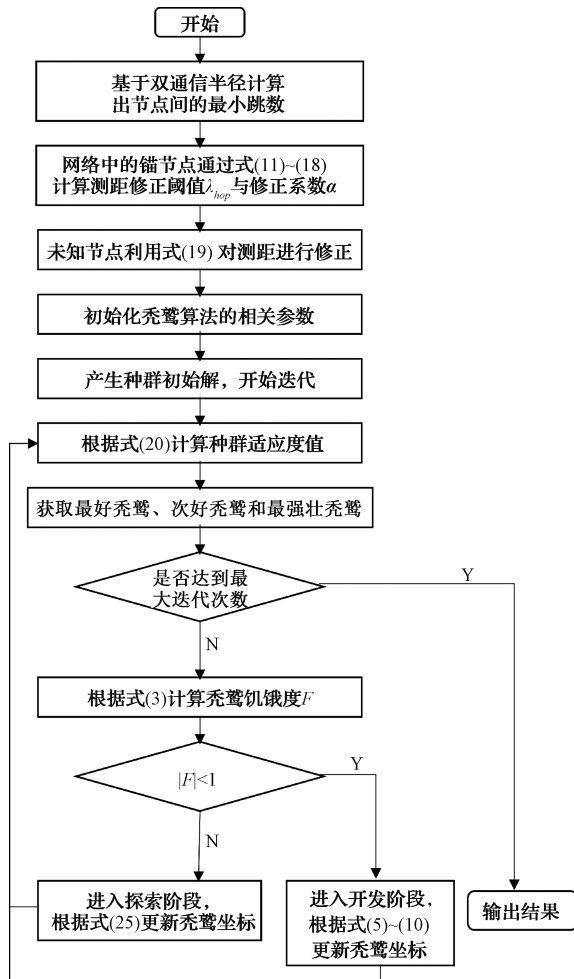


图2 改进算法流程

Fig. 2 Improved algorithm flow chart

点将  $\lambda_{hop}$  和  $\alpha$  压缩成一个数据包后向网络中广播,未知节点只接收并转发一次该类型的数据包。

5)未知节点根据式(19)计算其与锚节点间的修正后的距离;

6)初始化非洲秃鹫优化算法的相关参数:种群数量  $pop$ , 维度  $dim$ , 最大迭代次数  $T$ , 位置上界  $u$  和位置下界  $l$ , 产生种群初始解, 当前迭代  $t = 1$ 。

7)根据式(20)计算每只秃鹫的适应度值,选择最好秃鹫  $X_{b1}$ 、次好秃鹫  $X_{b2}$  以及最强壮秃鹫  $X_s$ 。

8)判断  $t$  是否等于最大迭代次数  $T$ ,若是,执行步骤10);否则,执行步骤9)。

9)根据式(3)计算秃鹫饥饿度  $F$ ,依据秃鹫当前的饥饿度选择性的进入探索阶段和开发阶段更新秃鹫位置,然后跳转到步骤7)。

10)输出最好秃鹫  $X_{b1}$ ,定位结束。

### 3 仿真与分析

本文通过对 DV-Hop 算法的第 2 阶段和第 3 阶段进

行改进,提出一种基于测距修正和秃鹫优化的 DV-Hop 改进算法。为了验证本文的改进算法在各向异性网络中的定位效果,使用 MATLAB R2021a 软件将  $100\text{ m} \times 100\text{ m}$  的仿真区域设置成 3 种不规则网络,分别为“C”型网络、“H”型网络和“L”型网络(图 3),在其内随机部署未知节点和锚节点进行仿真实验。为了更好地分析 AVOADV-Hop 算法的性能,本文选择差分进化算法改进的 DV-Hop 算法(DEIDV-Hop 算法)<sup>[18]</sup>、螭螂算法改进的 DV-Hop 算法(DBODV-Hop 算法)<sup>[19]</sup>以及传统的 DV-Hop 算法作为参照,从通信半径、锚节点占比和节点总数 3 个角度分析它们的定位精度。选择平均定位误差作为定位精度的评价指标:

$$err = \frac{\sum_{i=1}^n \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2}}{n \times R} \quad (26)$$

式中:  $(x_i, y_i)$  与  $(x'_i, y'_i)$  分别为未知节点的估计坐标与真实坐标;  $n$  为网络中未知节点的个数;  $R$  为节点的通信半径。

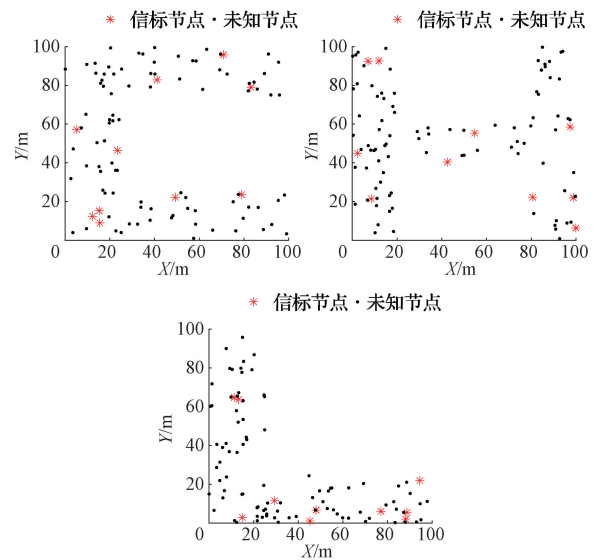


图3 3种不规则区域下的无线传感器网络

Fig. 3 Wireless sensor networks in three irregular regions

为了保证实验结果可靠,本文取相同参数下 50 次实验结果的平均值作为最终定位结果。AVOA 算法的相关参数设置为如表 1 所示。

表 1 AVOA 参数设置

Table 1 AVOA parameter Settings

参数	值
秃鹫种群数 $pop$	10
维度 $dim$	2
秃鹫位置上界 $u$	100
秃鹫位置下界 $l$	0
最大迭代次数 $T$	50

### 3.1 通信半径对定位误差的影响

图4所示是在3种不规则网络下随机部署100个传感器节点,锚节点占比10%,通信半径从20 m增加到50 m,4种定位算法的定位误差图。从图4可以看出,随着通信半径增加,4种定位算法的定位误差均有不同程度的降低,其中尤以传统DV-Hop算法的定位误差下降幅度最大,这是因为受障碍物的影响,传感器节点的通信半径越小网络的连通度越低且节点间的跳数越大,使得累积误差较大。在通信半径较小时,AVOADV-Hop算法的定位误差相较于DEIDV-Hop算法和DBODV-Hop算法有明显的下降,这主要是因为本文基于测距修正的改进,使算法在连通度低的环境下也有着良好的定位精度。当通信半径增大到40 m时,不规则障碍物对网络连通度的影响较小,4种定位算法的定位误差相差不大。在3种不规则网络下,本文算法相较于其他3种算法在通信半径20~50 m范围内的平均误差分别下降了26.7%、21.6%和5.1%。

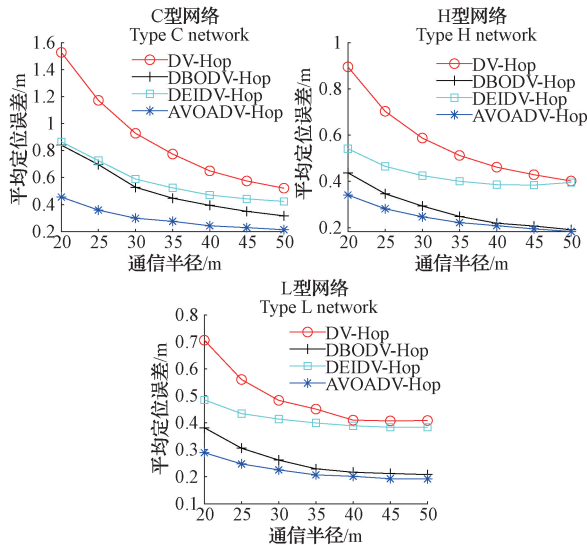


图4 不同通信半径下的定位误差  
Fig. 4 Positioning error diagram under different communication radius

### 3.2 锚节点占比对定位误差的影响

图5所示是在3种不规则网络下随机部署100个传感器节点,通信半径为25 m,锚节点占比从10%~40%,4种定位算法的定位误差图。随着锚节点比例的增大,4种定位算法的定位误差均出现小幅度的下降,这是因为网络中锚节点比例的增大会使未知节点周围的锚节点增多,未知节点会获得更多的未受障碍物影响的锚节点信息,从而使定位更准确。相较于通信半径的影响,锚节点比例的增大对定位误差的影响很小,这是因为锚节点比例的增大并不会改变网络的连通度,节点间的跳数也不会减小,反而在未知节点定位时所参考的不可靠锚节点数增多,受路径弯曲的影响程度增大,因此可以看到传统DV-

Hop算法的定位误差在下降的同时存在上下浮动的现象,且定位误差下降不明显。DEIDV-Hop算法、DBODV-Hop算法以及本文所提出的AVOADV-Hop算法都有基于加权思想来修改不同锚节点在求解未知节点坐标中的权重,使未知节点定位时受不可靠锚节点的影响程度较小,因此它们的定位误差下降的比较平滑。本文提出的改进算法受路径弯曲现象影响较小,在3种不规则网络下,本文算法相较于其他3种算法在锚节点占比10%~40%范围内的平均误差分别下降了54.1%、20.5%和14.1%。

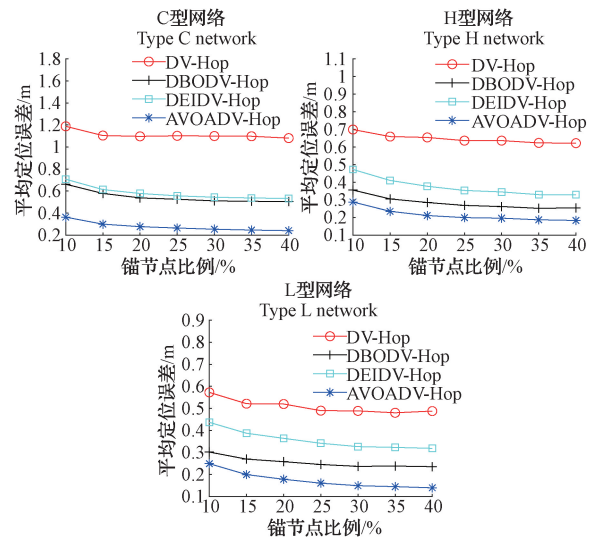


图5 不同锚节点比例下的定位误差  
Fig. 5 Positioning error diagram under different anchor node proportions

### 3.3 节点总数对定位误差的影响

图6所示是在3种不规则网络下随机部署传感器节点,所有传感器的通信半径为25 m,锚节点占比10%,节点总数从100个增加到200个,4种定位算法的定位误差图。从图6可以看出,随着节点总数的增加,4种定位算法的定位误差均有一定程度的下降,这是由于节点总数的增加会使整个网络的节点密度增大,网络连通度更好,节点间跳数与跳距的乘积更趋于真实值,使得定位误差下降。但由于节点总数的增加,未知节点的个数也在增加,网络中出现路径弯曲现象的概率会更大,因此DV-Hop算法、DEIDV-Hop算法和DBODV-Hop算法的定位误差的下降趋势相对平缓。AVOADV-Hop算法受障碍物的影响较小,定位误差呈现单调递减,在3种不规则网络下,本文算法相较于其他3种算法在节点总数100~200个范围内的平均误差分别下降了49.9%、21.3%和12.4%。

### 3.4 改进算法有效性验证

为了验证本文提出的测距修正和秃鹫优化两者各自的改进效果,本节选择基于测距修正的DV-Hop算法(IDV-Hop1)、基于秃鹫优化的DV-Hop算法(IDV-Hop2)与传统的DV-Hop算法在“C”型网络中进行实验仿真。

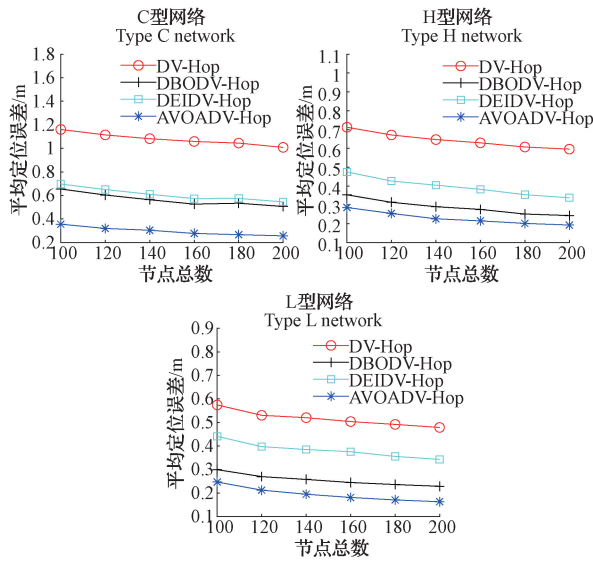


图6 不同节点总数下的定位误差  
Fig. 6 Positioning error diagram under different total number of nodes

其中,传感器节点100个,锚节点占比20%,通信半径为25 m,每个未知节点的定位误差如图7所示。在“C”型网络中,传统DV-Hop算法的平均定位误差在0.111 4~3.857 5 m范围内,其中定位误差在1 m以下的未知节点占有未知节点的52.5%。本文改进算法在“C”型网络下的定位误差相较于传统DV-Hop算法有明显的下降,对于IDV-Hop1算法和IDV-Hop2算法,定位误差在1 m以下的未知节点占有未知节点的88.75%和97.5%,定位精度和定位稳定性明显优于传统DV-Hop算法。

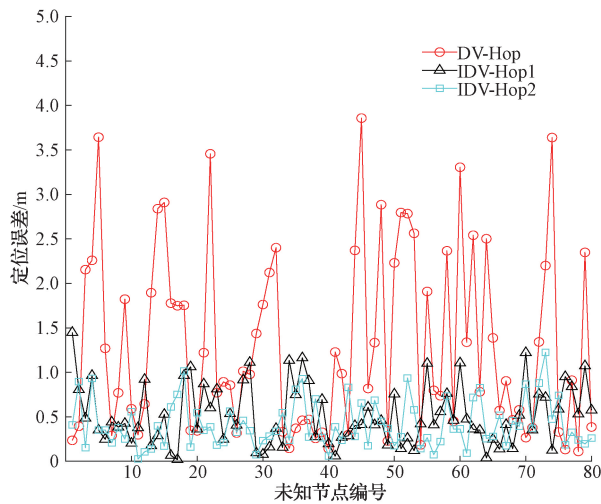


图7 未知节点定位误差  
Fig. 7 Unknown node positioning error diagram

#### 4 结论

传统DV-Hop算法在各向异性网络中由于受障碍物

影响使得节点间距离计算误差较大、定位精度不高,本文在双通信半径的基础上面向各向异性网络对DV-Hop算法的两个方面进行改进。首先针对不规则网络中的路径弯曲现象,本文引入测距修正阈值 $\lambda_{hop}$ ,对于跳数大于 $\lambda_{hop}$ 的两个节点,使用修正的平均跳距 $Hopsiz_{hop}$ 来计算距离;对于跳数小于 $\lambda_{hop}$ 的两个节点,使用未知节点保存的平均跳距计算距离。然后,针对最小二乘法的误差累积等问题,本文引入基于动态调整因子 $\omega$ 和加权思想改进的AVOA算法,并结合 $\lambda_{hop}$ 构造加权适应度函数来求解未知节点坐标。最后,在3种不规则网络环境下对本文提出的改进算法进行仿真实验。从结果可以看出,AVOADV-Hop算法的定位误差相较于传统的DV-Hop算法有显著的下降。

#### 参考文献

- [1] 宋晓东,孙丽君,陈天飞. DV-Hop优化算法的性能分析与比较[J]. 电子测量与仪器学报, 2019, 33(5): 125-133.  
SONG X D, SUN L J, CHEN T F. Performance analysis and comparison of DV-Hop optimization algorithm[J]. Journal of Electronic Measurement and Instrumentation, 2019,33(5): 125-133.
- [2] 刘晓悦,尤晓芳. 融合跳数划分与蝙蝠优化的DV-Hop改进算法[J]. 国外电子测量技术, 2022, 41(2): 26-32.  
LIU X Y, YOU X F. Improved DV-Hop algorithm combining hop division and bat optimization [J]. Foreign Electronic Measurement Technology, 2022, 41(2): 26-32.
- [3] 曹鹏飞,王秀英,孟庆斌. 融合测距修正和 Harris 鹰优化的DV-Hop定位算法[J]. 电子测量技术, 2023, 46(11): 166-172.  
CAO P F, WANG X Y, MENG Q B. DV-Hop localization algorithm combining ranging correction and harris hawks optimization [J]. Electronic Measurement Technology, 2023, 46(11): 166-172.
- [4] 朱霞青. 一种减小偏差的三维AOA定位方法[J]. 无线通信技术, 2020, 29(1): 1-5.  
ZHU X Q. A bias reduced method for 3-D AOA localization[J]. Wireless Communication Technology, 2020, 29(1): 1-5.
- [5] 杨晓倩. 基于TOA-AOA数据融合的定位算法研究[J]. 电子测量技术, 2020, 43(16): 104-108.  
YANG X Q. Research on localization algorithm based

- on TOA-AOA data fusion [J]. *Electronic Measurement Technology*, 2020, 43(16): 104-108.
- [6] 马一鸣,石志东,赵康,等. 基于改进哈里斯鹰优化算法的TDOA定位[J]. *计算机工程*, 2020, 46(12): 179-184.
- MA Y M, SHI ZH D, ZHAO K, et al. TDOA localization based on improved harris hawk optimization algorithm [J]. *Computer Engineering*, 2020, 46(12): 179-184.
- [7] 刘松旭,张大鹏,乌云娜,等. 基于RSSI模型的无线传感器网络定位算法[J]. *计算机仿真*, 2022, 39(1): 427-431.
- LIU S X, ZHANG D P, WU Y N, et al. Wireless sensor network positioning algorithm based on RSSI model [J]. *Computer Simulation*, 2022, 39(1): 427-431.
- [8] 董玉,张治中,冯姣. 基于测距修正和蝙蝠优化的改进DV-Hop定位算法[J]. *电子测量技术*, 2023, 46(7): 110-116.
- DONG Y, ZHANG ZH ZH, FENG J. Improved DV-Hop localization algorithm based on ranging correction and bat optimization [J]. *Electronic Measurement Technology*, 2023, 46(7): 110-116.
- [9] NICULESCU D C S, NATH B. DV based positioning in ad hoc network [J]. *Telecommunication Systems*, 2003, 22(1): 267-280.
- [10] 王国武,陈元琰. 基于跳数修正和遗传模拟退火优化DV-Hop定位算法[J]. *计算机科学*, 2021, 48(S1): 313-316.
- WANG G W, CHEN Y Y. Improvement of DV-Hop location algorithm based on hop correction and genetic simulated annealing algorithm [J]. *Computer Science*, 2021, 48(S1): 313-316.
- [11] 吴建锋,徐振宇,蒋震. 无线传感器网络中改进粒子群优化DV-Hop算法的研究[J]. *传感技术学报*, 2022, 35(6): 825-830.
- WU J F, XU ZH Y, JIANG ZH. Improved particle swarm optimization DV-Hop algorithm based on heterogeneous wireless sensor networks [J]. *Chinese Journal of Sensors and Actuators*, 2022, 35(6): 825-830.
- [12] 印雷,顾德,刘飞. 基于改进麻雀搜索算法优化的DV-Hop定位算法[J]. *传感技术学报*, 2021, 34(5): 670-675.
- YIN L, GU D, LIU F. Improved sparrow search algorithm based DV-Hop localization in WSN [J]. *Chinese Journal of Sensors and Actuators*, 2021, 34(5): 670-675.
- [13] 刘燕,高丽. 改进的人工蜂群优化DV-Hop定位算法[J]. *激光与光电子学进展*, 2020, 57(19): 240-245.
- LIU Y, GAO L. Optimized DV-Hop localization algorithm based on improved artificial bee colony [J]. *Laser & Optoelectronics Progress*, 2020, 57(19): 240-245.
- [14] 余修武,黄露平,刘永,等. 遗传-禁忌搜索优化的三维DV-Hop定位算法[J]. *北京邮电大学学报*, 2021, 44(4): 75-81.
- YU X W, HUANG L P, LIU Y, et al. Three-dimensional DV-Hop location algorithm based on genetic-tabu search optimization in WSN [J]. *Journal of Beijing University of Posts and Telecommunications*, 2021, 44(4): 75-81.
- [15] 张弛,张保衡,张静,等. 水下无线传感器网络定位技术综述[J]. *计算机应用*, 2022, 42(S1): 226-235.
- ZHANG CH, ZHANG B H, ZHANG J, et al. Overview of underwater wireless sensor network positioning technology [J]. *Journal of Computer Applications*, 2022, 42(S1): 226-235.
- [16] ABDOLLAHZADEH B, GHAREHCHOPOGH F S, MIRJALILI S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems [J]. *Computers & Industrial Engineering*, 2021, 158: 107408.
- [17] 张明军,刘增力,杨迪. 基于改进灰狼优化算法的DV-Hop传感器定位算法[J]. *通信技术*, 2023, 56(2): 148-155.
- ZHANG M J, LIU Z L, YANG D. DV-Hop sensor positioning algorithm based on improved gray wolf optimization algorithm [J]. *Communications Technology*, 2023, 56(2): 148-155.
- [18] HAN D Z, YU Y P, LI K C, et al. Enhancing the sensor node localization algorithm based on improved



DV-Hop and DE algorithms in wireless sensor networks[J]. *Sensors*, 2020, 20(2): 343.

- [19] 潘志远, 卜凡亮. 基于蜣螂算法优化的 DV-Hop 定位算法[J]. *电子测量与仪器学报*, 2023, 37(7): 33-41.  
PAN ZH Y, BU F L. DV-Hop localization algorithm optimized based on dung beetle optimizer[J]. *Journal of Electronic Measurement and Instrumentation*, 2023, 37(7): 33-41.

#### 作者简介

刘龙飞(通信作者), 硕士研究生, 主要研究方向为无线传感器网络、嵌入式系统设计与应用。

E-mail: llf\_122@163.com

倪晓军, 副教授, 硕士生导师, 主要研究方向为嵌入式系统在无线传感器网络领域的技术及应用。

E-mail: nixj@njupt.edu.cn