

# 基于改进麻雀搜索算法的 3DDV-Hop 定位算法

高翔宇<sup>1,2</sup>, 韩慧子<sup>3</sup>, 孟亚男<sup>1</sup>, 刘美<sup>2</sup>

(1. 吉林化工学院 信息与控制工程学院, 吉林 吉林 132022;

2. 广东石油化工学院 自动化学院, 广东 茂名 525000; 3. 香港理工大学 工程学院, 香港 999077)

**摘要:** 针对经典 3DDV-Hop 算法定位精度不高的问题, 提出一种基于改进麻雀搜索算法优化的改进的 3DDV-Hop 算法; 算法首先通过多通信半径优化传感器节点之间跳数, 并且在平均跳距计算过程中引入动态加权因子提高平均跳距计算精度, 其次在麻雀搜索算法的基础上融合反向学习策略与萤火虫算法分别对麻雀搜索算法的种群与位置更新迭代进行优化, 最后将未知节点坐标计算问题转化成改进后的麻雀搜索算法寻优问题, 利用改进后的麻雀搜索算法替代最小二乘法计算未知节点坐标, 进一步提高未知节点位置计算精度; 经过 MATLAB 仿真验证, 改进算法相比于经典 3DDV-Hop 算法和相关算法, 定位精度得到有效提高。

**关键词:** 无线传感器网络; 3DDV-Hop; 麻雀搜索算法; 节点定位; 萤火虫算法

## 3DDV-Hop Localization Algorithm Based on Improved Sparrow Search Algorithm

GAO Xiangyu<sup>1,2</sup>, HAN Huizi<sup>3</sup>, MENG Yanan<sup>1</sup>, LIU Mei<sup>2</sup>

(1. School of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin 132022, China;

2. School of Automation, Guangdong Institute of Petrochemical Technology, Maoming 525000, China;

3. School of Engineering, The Hong Kong Polytechnic University, Hongkong 999077, China)

**Abstract:** Aimed at the problem that the positioning accuracy of the classical 3DDV-Hop algorithm is not high, an improved 3DDV-Hop algorithm based on the optimization of the improved sparrow search algorithm is proposed. Firstly, the algorithm optimizes the hop count between sensor nodes through multiple communication radii optimization, and a dynamic weighting factor is introduced to improve the calculation accuracy of average hop distance during the average hop distance calculation; Secondly, on the basis of the sparrow search algorithm, the reverse learning strategy and firefly algorithm are integrated to optimize the population and position update iteration of the sparrow search algorithm, respectively. Finally, the unknown node coordinate calculation is transformed into the improved sparrow search algorithm optimization, and the improved sparrow search algorithm is used to replace the least squares method to calculate the unknown node coordinates, further improve the unknown node position calculation accuracy. Through the simulation verification of MATLAB, the results show that compared with the classical 3DDV-Hop algorithm and other related algorithms, the improved algorithm effectively improves the positioning accuracy.

**Keywords:** wireless sensor network; 3 DDV-Hop; sparrow search algorithm; node location; firefly algorithm

## 0 引言

近年来, 随着无线传感器技术的发展<sup>[1]</sup>, 无线传感器网络在军工安全, 森林防火, 环境监测等领域得到了大量的应用<sup>[2]</sup>。在众多领域的应用中传感器网络感知的信息很重要, 但是在更多时候人们更需要的是信息的发生位置<sup>[3]</sup>。然而二维 WSN 定位方式在复杂环境中无法精确反映环境地形等空间上的差异<sup>[4]</sup>, 在信息获取上产生较大差异, 因此无线传感网络中三维定位算法在近些年随之兴起<sup>[4]</sup>。

在无线传感器网络中无线定位技术可以实现有毒有害气体泄漏源定位、人员定位的功能, 如石化厂区有毒气体泄漏源定位、石化厂区人员定位等, 通过微型传感器节点的搭建可以极大地节省定位系统建造成本。现阶段无线传

感器网络中的定位技术主要分为两类<sup>[5]</sup>: 一类为基于测距的定位方式, 该类算法具有定位精度较高的优点但是相对应的也需要较好的硬件技术支持并且能耗较高<sup>[6]</sup>, 如基于信号强度的定位方式 (RSSI, received signal strength indicator)<sup>[7]</sup>、基于到达时间的定位方式 (TOA, time of arrival)<sup>[8]</sup>等; 另一类为与距离无关的定位算法, 该类算法具有对硬件设施要求不高且能耗相对较低的优势<sup>[9]</sup>, 如质心算法<sup>[10]</sup>, DV-Hop 定位算法、APIT 定位算法<sup>[11]</sup>等。其中 DV-Hop 算法作为典型的无需测距的定位方式得到众多学者青睐并应用到实际当中去。文献 [12] 提出了一种以优化跳数和跳距为核心的三维 DV-Hop 定位改进算法提高了传统算法的精确度, 但是没有考虑在未知节点坐标计算时

收稿日期: 2023-08-24; 修回日期: 2023-12-22。

作者简介: 高翔宇(1997-), 男, 硕士。

通讯作者: 韩慧子(1998-), 女, 硕士。

引用格式: 高翔宇, 韩慧子, 孟亚男, 等. 基于改进麻雀搜索算法的 3DDV-Hop 定位算法[J]. 计算机测量与控制, 2024, 32(5): 246-252.

的误差; 文献 [13] 提出一种神经动力学的改进 DV-Hop 算法通过误差分析计算未知节点与锚节点之间距离误差的波动范围, 将传统的定位模型转化为距离和坐标随时间变化的代数方程, 而后通过 ND 算法对未知节点坐标进行求解, 降低了定位误差; 文献 [14] 提出一种自适应步长变化混沌果蝇优化算法 DV-Hop (CAFOA-DV-Hop), 该算法的核心在于其创新的自适应搜索步长策略, 这一策略不仅极大增强了算法的全局搜索能力, 而且在全局和局部优化之间找到了一个有效的平衡点。因此, DV-Hop 算法在收敛速度和定位精度上都有了显著的提升。文献 [15] 利用 PSO 算法优化信标节点的平均跳距并且以 DE 算法优化未知节点的计算坐标, 实现对 DV-Hop 算法精度的有效提高。本文针对环境监测中复杂的三维环境针对经典三维 DV-Hop 算法缺陷提出了一种基于改进麻雀搜索算法的三维 DV-Hop 算法。

在本文中主要贡献分为以下两点:

1) 本文算法与经典 3DDV-Hop 算法相比大大减小了未知节点的定位误差。并且在广泛的仿真场景中, 我们评估了信标节点比例, 通信半径, 信标节点数量的变化对算法影响。根据仿真结果可知, 本文改进算法在不同场景下均优于经典 3DDV-Hop 算法。

2) 提出一种基于反向学习策略与萤火虫扰动的麻雀搜索算法, 麻雀搜索算法的全局搜索能力得到了显著提升, 有效避免了算法陷入全局最优解的困境。同时, 算法的局部极值逃逸能力也得到了加强, 从而进一步改善了算法的寻优效果。

## 1 三维 DV-Hop 定位算法

### 1.1 算法原理

由 DragosNiculescu 提出的 DV-Hop 算法<sup>[16]</sup>是经典的无需测距的定位算法之一, 而三维 DV-Hop 定位算法是以 DV-Hop 算法为理论基础实现的, 其定位步骤主要为以下三步。

#### 1) 计算最小跳数:

信标节点在网络中进行广播, 向其通信范围内的其他节点传递自身跳数 (初始设为 0) 信息。随后, 其他节点将接收到的跳数信息进行比较, 仅保留最小的跳数值, 并在此基础上加一, 再向其他节点进行转发。

#### 2) 计算平均跳距:

通过广播在每个信标节点能够确定自身位置之后, 信标节点  $i(x_i, y_i, z_i)$  即可通过式 (1) 计算自身平均每跳距离:

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

式中, 信标节点  $i, j$  之间的最小跳数为  $h_{ij}$ , 信标节点自身平均每跳距离为  $h_i$ 。信标节点  $j$  的坐标为  $(x_j, y_j, z_j)$ 。

#### 3) 计算节点坐标:

设未知节点  $u$  坐标为  $(x, y, z)$ , 其中信标节点  $i$  能与  $u$  直接通信并且其坐标为  $(x_i, y_i, z_i)$ , 跳数为  $h_i$  ( $i=1,$

$2, 3, \dots, n$ ) 则未知节点到信标节点之间距离  $d_i$  可通过式 (2) 得出。

$$d_i = Hopsize_i \times h_i \quad (2)$$

根据四边定位法可得出式 (3)

$$\begin{cases} (x_1 + x)^2 + (y_1 + y)^2 + (z_1 - z)^2 = d_1^2 \\ (x_2 + x)^2 + (y_2 + y)^2 + (z_2 - z)^2 = d_2^2 \\ \vdots \\ (x_n + x)^2 + (y_n + y)^2 + (z_n - z)^2 = d_n^2 \end{cases} \quad (3)$$

令

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) & 2(z_1 - z_n) \\ 2(x_2 - x_n) & 2(y_2 - y_n) & 2(z_2 - z_n) \\ \vdots & \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) & 2(z_{n-1} - z_n) \end{bmatrix} \quad (4)$$

$$x = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5)$$

$$B = \begin{bmatrix} d_1^2 - d_n^2 + x_n^2 - x_n^2 + y_n^2 - y_n^2 + z_n^2 - z_n^2 \\ d_2^2 - d_n^2 + x_n^2 - x_n^2 + y_n^2 - y_n^2 + z_n^2 - z_n^2 \\ \vdots \\ d_{n-1}^2 - d_n^2 + x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + z_{n-1}^2 - z_n^2 \end{bmatrix} \quad (6)$$

根据最小二乘法可得出未知节点做标式如式 (7):

$$x = (A^T A)^{-1} A^T B \quad (7)$$

### 1.2 误差分析

#### 1) 最小跳数误差:

针对算法第一阶段, 发现跳数通过节点间通信完成, 相邻节点间跳数为 1, 定义通信半径为  $R$ 。但图 1 所示, 跳数为 1 的实际距离不等于  $R$ , 导致跳数误差。信标节点  $A$  与  $B, C, D$  通信时,  $h_{AB}=1, h_{AC}=1, h_{AD}=1$ , 但  $AB$  的实际距远大于  $AC, AD$  却与  $AE$  相近, 而  $h_{AE}=2$ , 影响  $A$  的平均跳距。因此, 跳数计算机制易产生误差。

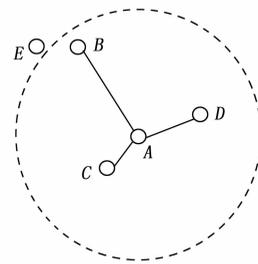


图 1 跳数误差示意图

根据图 1 信息, 我们发现跳数为 1 时,  $B, C, D$  节点与信标节点  $A$  间距离差异大。  $B$  与  $E$  间距离相近, 但跳数差异明显。这种不一致性和差异是经典 DV-Hop 算法误差的关键原因之一。

#### 2) 平均跳距计算误差:

在经典的三维距离向量-跳数 (3DDV-Hop) 定位算法中, 对于未知节点的平均跳距计算, 通常采用与其自身跳数最小的信标节点的平均跳距作为参考。然而, 在实际应

用中,无线传感器网络节点的分布具有随机性,这导致在节点分布密集的区域,平均跳距相对较小,而在节点分布稀疏的区域,平均跳距则相对较大。因此,若仅依赖单一的平均跳距作为整个系统中节点计算的关键要素,则无法准确反映整个系统的实际分布状况。

3) 节点坐标计算误差:

在经典 3DDV-Hop 算法中最小二乘法计算节点坐标带来累积误差。

为解决上述问题本文对经典 DV-Hop 算法进行改进。

## 2 基于改进麻雀搜索算法的 3DDV-Hop 定位算法

针对上述对于经典三维 DV-Hop 算法误差分析,本文提出以下优化方案。

### 2.1 跳数细化

根据误差分析得知,跳数误差为算法误差来源之一,因此本文引入多通信半径的方法以此细化节点跳数。假设信标节点通信半径为  $R$ ,节点间实际距离记为  $d$ ,跳数为  $h$ 。通信半径分为  $m$  份,通过不同通信半径广播数据之后,各节点之间跳数  $h$  为:

$$h = \begin{cases} \frac{1}{m} 0 < d \leq \frac{R}{m} \\ \frac{i}{m} \frac{(i-1)R}{m} < d \leq \frac{iR}{m} i \in [1, m) \\ 1 \frac{(m-1)R}{m} < d \leq R \end{cases} \quad (8)$$

在引入多通信半径后,跳数与实际距离之间的误差呈现出逐渐减小的趋势,这一特点使得在后续计算中能够更精确地确定节点与邻居节点之间的距离。然而,在实际应用中,随着跳数细化的增加,也带来了一些负面影响。首先,跳数的细化导致节点间的通信次数显著增加,进而大幅提升了节点的能耗,不利于其长期稳定运行。其次,较高的跳数细化要求无线传感器具备更高的灵敏度,但在复杂的实际使用环境中,无线传感器网络容易受到电磁干扰等外部因素的影响,从而产生新的误差。综合考虑以上因素,本文选择采用 4 个通信半径。

### 2.2 信标节点平均跳距修正

对于平均跳距的计算在经典 DV-Hop 算法中不能够完全反映节点在网络中的实际情况,导致在计算中会引入累积误差。在式 (2) 的计算过程中可以发现信标节点与未知节点之间的距离越小通过计算得到的距离就越精确。因此,在上述误差分析的基础上本文引入加权因子  $\delta$  来提高平均跳距精度:

$$\delta_i = \frac{1}{h_i} \quad (9)$$

$$\sum_{j=1}^k \frac{1}{h_j}$$

$$Hopsize_u = \sum_{i=1}^k (\delta_i \times Hopsize_i) \quad (10)$$

其中:  $\delta_i$  主要由信标节点与未知节点之间的  $\frac{1}{h_i}$  决定,

并且与之成正比;由式 (10) 可求出修正之后的未知节点平均跳距,并通过  $Hopsize_u$  与跳数求出未知节点到信标节点之间的计算距离。

$$d_i = Hopsize_u \times h_i \quad (11)$$

### 2.3 计算优化

经过上述改进后,未知节点的估计位置与真实位置仍存在误差。如果未知节点  $A(x, y, z)$  接收了来自  $n$  个信标节点的数据,则设  $n$  个信标节点位置分别为  $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), \dots, (x_n, y_n, z_n)$ 。未知节点与信标节点之间的估计距离  $d_i$  可通过优化后的跳数与平均跳距求出,然而两者之间实际距离为  $d'_i$ ,因此产生误差  $\alpha_i$  如式 (12) 所示:

$$\alpha_i = |d'_i - d_i| \quad (12)$$

那么将未知节点定位问题转化为求解坐标方程问题如式 (13):

$$\begin{cases} d_1^2 - \alpha_1^2 \leq (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 \leq d_1^2 + \alpha_1^2 \\ d_2^2 - \alpha_2^2 \leq (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 \leq d_2^2 + \alpha_2^2 \\ \vdots \\ d_n^2 - \alpha_n^2 \leq (x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2 \leq d_n^2 + \alpha_n^2 \end{cases} \quad (13)$$

求解未知节点  $A(x, y, z)$ ,只需使  $F(x, y)$  最小,得到的误差也就最小。 $F(x, y)$  可通过式 (14) 求得。

$$F(x, y) = \sum_{i=1}^m | \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} - d_i | \quad (14)$$

即式 (14) 为优化目标函数式中  $F(x, y)$  为麻雀个体适应度值。

### 2.4 改进麻雀算法

针对 SSA 的不足,我们首先利用反向学习策略对 SSA 的初始化过程进行改进,改善初始种群在搜索空间上的分布,增强全局搜索能力,避免算法陷入局部最优;其次,融合萤火虫算法,增强局部极值逃逸能力,改善寻优效果。

#### 2.4.1 标准麻雀搜索算法

麻雀搜索算法<sup>[17]</sup>作为一种新型群智能优化算法,具有寻优能力强,结构简单等优点。理想麻雀算法如下:

1) 初始化种群:

确定麻雀种群中个体数量  $N$ 、所优化问题维度  $dim$ 、搜索区域  $[lb, ub]$ 、最大迭代次数  $MaxIter$ 。种群初始位置矩阵如式 (15) 所示:

$$x = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,dim} \\ x_{2,1} & x_{2,2} & \dots & x_{2,dim} \\ \vdots & \vdots & \dots & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,dim} \end{bmatrix} \quad (15)$$

式中,  $dim$  为所优化问题的变量维度,  $n$  为麻雀种群中个体数量。则所有麻雀的适应度值可表示如式 (16) 所示:

$$f_x = \begin{bmatrix} f([x_{1,1} & x_{1,2} & \dots & x_{1,dim}]) \\ f([x_{2,1} & x_{2,2} & \dots & x_{2,dim}]) \\ \vdots & \vdots & \dots & \vdots \\ f([x_{n,1} & x_{n,2} & \dots & x_{n,dim}]) \end{bmatrix} \quad (16)$$

式中,  $f_x$  为适应度值,  $f$  为所求适应度值的函数。

2) 寻找者位置更新:

在整个麻雀群体中, 寻找者占据主导作用。其位置更新如式 (17) 所示:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t \times \exp\left(\frac{-1}{\alpha \times t_{\max}}\right) & \text{if } R_2 < ST \\ x_{i,j}^t + M \times L & \text{if } R_2 \geq ST \end{cases} \quad (17)$$

式中,  $t$  为当前迭代次数;  $j=1, 2, 3, 4, \dots, dim$ 。  $t_{\max}$  为最大迭代次数;  $x(i, j)^t$  为第  $i$  个麻雀在  $j$  维度的位置信息;  $\alpha \in (0, 1]$  是一个随机数;  $R_2$  为警戒者发现捕食者发出的示警信号值;  $ST$  为预设的安全阈值;  $M$  为服从正态分布的随机数;  $L$  为一个  $1 \times dim$  的矩阵且其中元素均为 1。

3) 加入者位置更新:

加入者在寻找者的引领下发现食物来源, 为了提升自身的捕食效率, 甚至可能与寻找者产生竞争。其位置更新如式 (18) 所示:

$$x_{i,j}^{t+1} = \begin{cases} M \times \exp\left(-\frac{x_{\text{worst}}^t - x_{i,j}^t}{i^2}\right) & \text{if } i < \frac{n}{2} \\ x_p^{t+1} + |x_{i,j}^t - x_p^{t+1}| \mathbf{A}^+ L & \text{if } i \geq \frac{n}{2} \end{cases} \quad (18)$$

$$\mathbf{A}^+ = \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1} \quad (19)$$

其中:  $\mathbf{A}$  为一个  $1 \times dim$  的矩阵且矩阵中每个元素随机赋值为 1 或 -1;  $x_{\text{worst}}$  为当前适应度值最低的位置;  $x_p$  为当前寻找者占据的最优位置。

4) 警戒者位置更新:

警戒者一般在每代种群中随机选取  $SD$  (一般为 10%~20%) 其位置更新如式 (20) 所示:

$$x_{i,j}^{t+1} = \begin{cases} x_{\text{best}}^t + \beta |x(i, j)^t - x_{\text{best}}^t| & \text{if } f_i > f_g \\ x_{i,j}^t + K \times \frac{|x_{i,j}^t - x_{\text{worst}}^t|}{(f_i - f_w) + \epsilon} & \text{if } f_i = f_g \end{cases} \quad (20)$$

其中:  $x_{\text{best}}$  为整个种群中迭代  $t$  次中的最优解;  $K \in [-1, 1]$  是随机数代表种群中麻雀移动方向;  $f_g$  为全局最优麻雀适应度值;  $f_w$  为全局最差适应度值;  $\beta$  为步长控制参数。

2.4.2 反向学习初始化种群

麻雀搜索算法以随机初始化的策略起始搜索过程, 但这种方法易造成初始种群分布不均, 进而限制了算法的寻优效能。鉴于此, 本文提出引入反向学习策略, 旨在优化种群初始化过程, 提升初始种群的多样性, 从而增强全局搜索能力。

反向学习策略<sup>[18]</sup>在群智能算法中得到广泛应用并取的较好的实验效果<sup>[19]</sup>。其定义如下。

反向解定义: 假设在  $[l, u]$  中存在数  $x$ , 则  $x$  的反向点定义为式 (21)

$$x' = l + u - x \quad (21)$$

将反向点的定义扩展到  $D$  维空间之中, 设  $m = (x_1, x_2, \dots, x_D)$  为  $D$  维空间中的一个点, 其中  $x \in [l_i, u_i]$ ,  $i=1, 2, \dots, D$ , 则其反向点  $m' = (x'_1, x'_2, \dots, x'_D)$ , 其中  $x'_1 = l_1 + u_1 - x_1$ 。

2.4.3 萤火虫算法

在萤火虫算法<sup>[20]</sup>中萤火虫发出光亮的主要目的是作为指向标, 以此吸引其他个体。其中萤火虫的相对荧光亮度如式 (22) 所示:

$$I = I_0 \times e^{-r_{i,j}^\gamma} \quad (22)$$

式中,  $I_0$  为萤火虫最大亮度, 与目标函数值有关且成正比;  $\gamma$  为光强吸收系数, 荧光会随着距离增加逐渐减弱;  $r_{i,j}$  为萤火虫个体  $i, j$  之间空间距离。

萤火虫吸收度如式 (23) 所示:

$$\beta = \beta_0 \times e^{-r_{i,j}^\gamma} \quad (23)$$

式中,  $\beta_0$  为最大吸引度;  $\gamma$  为光强吸收系数;  $r_{i,j}$  为萤火虫个体  $i, j$  之间空间距离。

萤火虫位置更新如式 (24) 所示:

$$x_i = x_i + \beta \times (x_j - x_i) + \alpha \times \left(\text{rand} - \frac{1}{2}\right) \quad (24)$$

式中,  $x_i, x_j$  为萤火虫  $i, j$  所处空间位置;  $\alpha \in [0, 1]$  为步长因子;  $\text{rand}$  为  $[0, 1]$  上服从均匀分布的随机数。

2.5 基于改进麻雀搜索算法的 3DDV-Hop 算法流程

FASSA3DDV-Hop 算法步骤如下:

1) 信标节点在通信半径得到细化后, 会按照顺序进行信息广播。所有接收到这些信息的节点, 都会存储信标节点的位置信息和最小跳数, 并且进一步转发这些信息。

2) 利用公式 (10), 计算信标节点的平均跳距, 并将结果向所有节点广播。

3) 我们将求解未知节点坐标的问题, 转化为改进后的麻雀搜索算法的优化问题, 并构建相应的适应度函数, 如公式 (14) 所示。

4) 将未知节点到信标节点的距离及各自的位置信息输入适应度函数, 利用改进后的麻雀搜索算法进行寻优。当满足寻优条件或达到预设的最大迭代次数时, 将输出最佳位置, 即所求未知节点的坐标。

5) 当所有未知节点的坐标均求解完成后, 算法将终止。

流程如图 2 所示。

3 实验与仿真

为了验证本文提出的改进算法的有效性, 我们使用 MATLAB 2020b 软件进行了仿真试验, 对照组包括经典 3DV-Hop 算法、SSA-3DDV-Hop 算法和文献 8 中的算法。试验涵盖了节点总数、通信半径以及信标节点比例这三个关键因素。图 3 展示了节点分布的具体情况, 而仿真环境的具体参数则详列于表 1 中。为了评估定位精度, 我们采用了式 (25) 来计算归一化定位误差, 并以此作为评估各算法性能的重要指标。

$$Accuracy = \frac{\sum_{i=1}^n \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}{n \times r} \quad (25)$$

式中,  $(x_i, y_i, z_i), (x_j, y_j, z_j)$  分别为位置节点  $i$  的实际坐标与计算坐标,  $n$  为未知节点个数,  $r$  为通信半径。

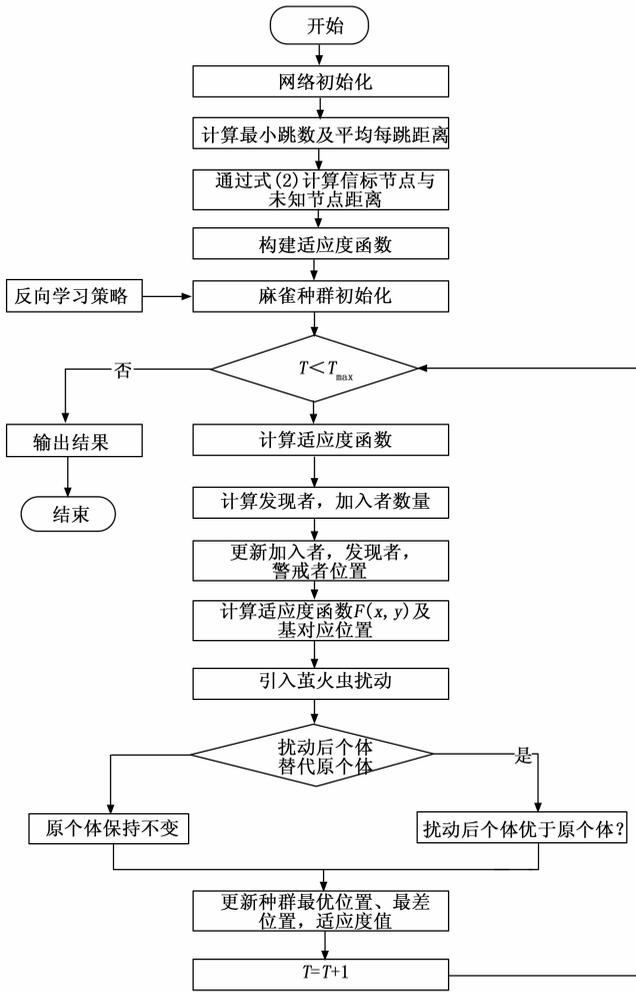


图 2 算法流程图

表 1 仿真参数

参数	取值
网络区域面积	100100100
节点总数	200
通信半径	40
信标节点比例	0.30
实验次数	100
麻雀种群数量	100
迭代次数/T	100
安全值/ST	0.6

根据图 4 所示的数据分析, 经典 DV-Hop 算法在节点定位误差方面表现出一定的波动范围, 其中最小值为 3.73; 最大值达到 35.8, 平均定位误差为 19.812。相比之下, 本文提出的改进算法在节点定位误差方面有了显著的优化。具体而言, 改进算法的最小误差为 1.372, 最大误差为 11.413, 平均定位误差为 4.808。这一数据表明, 通过本文的改进算法, 定位误差减小了 15.004, 从而有效提升了经典三维 DV-Hop 算法的定位精度。

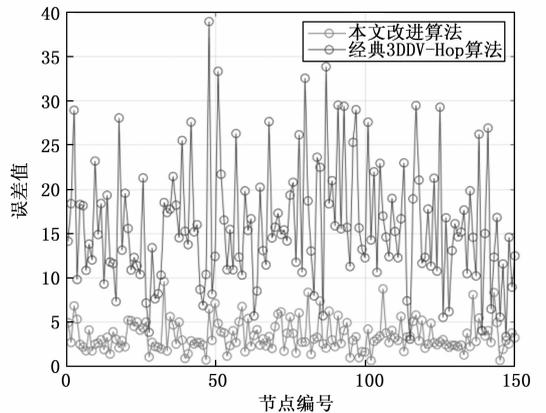


图 4 两种算法定位误差对比图

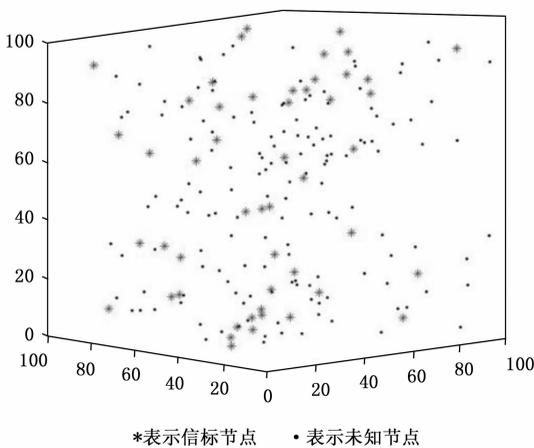


图 3 节点分布图

### 3.1 节点定位误差

图 3 为经典 DV-Hop 算法与本文改进算法定位未知节点坐标的误差对比。两种算法各项参数统一: 信标节点数为 50, 未知节点数 150, 通信半径 30 m。

### 3.2 信标节点比例对定位误差影响

本研究对比经典 3DDV-Hop 算法、SSA-3DDV-Hop 算法以及文献 [8] 中提出的算法, 系统研究了信标节点比例对定位精度的影响, 设定节点总数为 200, 通信半径为 40 米, 全面评估了不同信标节点比例下的定位精度。在实验中选择了 0.15、0.20、0.25、0.30、0.35、0.40、0.45 和 0.50 作为信标节点的比例进行了实验。每个比例都进行了 100 次重复实验, 并计算了归一化定位误差的平均值。通过对比分析实验结果, 能够得出信标节点比例对节点定位精度的影响, 如图 5 所示。

经过对图 5 的深入分析, 可以清晰地观察到四种算法的平均定位精度与锚节点比例之间呈现出明显的正相关关系, 即随着锚节点比例的增加, 平均定位精度也相应提升。进一步对比发现, 本文所提出的改进算法在定位平均误差方面显著优于其他三种算法, 具体的数值变化详见表 2。这一结果表明, 本文改进算法在提升定位精度方面具有显著优势。

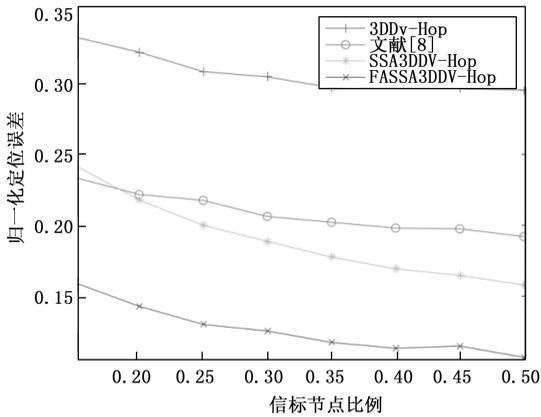


图 5 信标节点比例对定位误差影响

表 2 不同信标比例的算法平均误差

算法	信标节点比例			
	0.15	0.25	0.35	0.5
三维 DV-Hop	0.33	0.30	0.29	0.29
文献[8]	0.23	0.21	0.20	0.193
DSSADV-Hop	0.24	0.19	0.17	0.15
本文算法	0.15	0.12	0.11	0.10

由表 2 可知, 当信标节点比例为 0.15 时, 本文改进算法相较于其他定位算法定位平均误差分别提高了 0.170 49, 0.077 42, 0.083 036; 在锚节点比例为 0.5 时相对于其他定位算法分别提高为 0.188 983, 0.085 453, 0.052 767。由此可知本文算法无论信标节点比例如何相较于其他三种定位算法都具有优越性。

### 3.3 通信半径对定位的影响

本研究对比经典 3DDV-Hop 算法、SSA-DV-Hop 算法以及文献 [8] 中提出的算法, 系统分析了通信半径对节点定位精度的影响。在实验中, 设定节点总数为 200, 信标节点比例为 0.30。为了全面评估信标节点比例对定位精度的影响, 我们选择了 30、35、40、45、50、55、60 作为通信半径进行了实验。每个通信半径下, 我们都进行了 100 次重复实验, 并计算了归一化定位误差的平均值。通过对比分析实验结果, 我们得出了通信半径对节点定位精度的影响, 如图 6 所示。

根据图 6 的展示, 随着通信半径的不断扩大, 未知节点的定位误差呈现出与通信半径成反比的趋势。特别地, 当通信半径从 30 增加至 45 的区间内, 定位误差的下降速度最为显著, 之后逐渐趋于平稳。详细的变化情况可参见表 3 中的数据记录。

表 3 不同通信半径的算法平均误差

算法	通信半径			
	30	40	50	60
三维 DV-Hop	0.38	0.30	0.28	0.28
文献[8]	0.34	0.21	0.17	0.143
DSSADV-Hop	0.26	0.19	0.19	0.19
本文优化算法	0.22	0.10	0.06	0.05

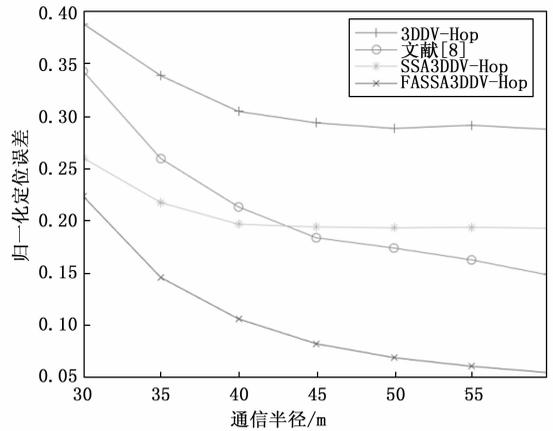


图 6 通信半径对定位误差影响

根据表 3 中的数据, 我们可以观察到, 当通信半径设定为 30 时, 本文所提出的改进算法相较于其他三种算法, 平均定位误差分别降低了 0.164 519, 0.119 27, 0.036 573。而当通信半径扩大至 60 时, 本文的改进算法相较于其他三种算法, 平均定位误差则分别提高了 0.232 025, 0.092 07, 0.137 795。这些数据清晰地表明, 无论通信半径如何变化, 本文的改进算法相较于其他三种定位算法, 其平均定位误差均得到了显著的提升。

### 3.4 节点总数对定位误差影响

本研究对比经典 3DDV-Hop 算法、SSA-3DDV-Hop 算法以及文献 [8] 中提出的算法, 系统分析了节点总数对节点定位精度的影响。在实验中, 设置信标节点比例为 0.30, 节点通信半径为 40。为了全面评估信标节点比例对定位精度的影响, 我们选择了 100、150、200、250、300、350、400、450 作为节点总数进行了实验。每个节点总数下, 我们都进行了 100 次重复实验, 并计算了归一化定位误差的平均值。通过对比分析实验结果, 我们得出了节点总数对节点定位精度的影响, 如图 7 所示。

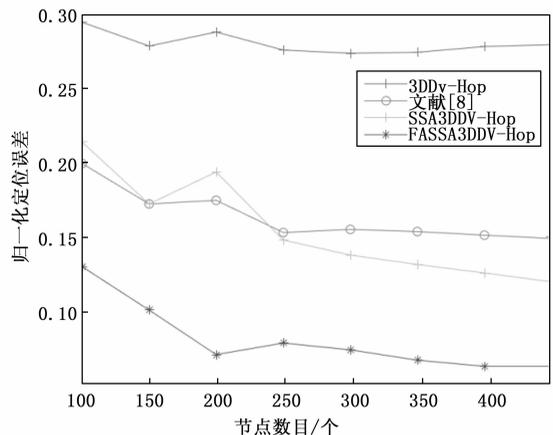


图 7 节点总数对定位误差影响

如图 7 所示, 在各种节点数量的条件下, 本文所提算法的平均定位误差相较于其他三种算法均表现出优越性。具体而言, 本文改进算法的平均定位误差为 0.080 087, 相比之

下,经典的三维 DV-Hop 算法的平均定位误差为 0.280 905,文献 [8] 中的算法平均定位误差为 0.163 317,而 3DSSADV-Hop 算法的平均定位误差为 0.155 057。各种算法的具体定位误差变化情况可参见表 4。

表 4 不同节点总数的平均定位误差

算法	节点总数				
	100	200	300	400	450
三维 DV-Hop	0.29	0.28	0.27	0.27	0.28
文献[8]	0.19	0.17	0.15	0.15	0.143
DSSADV-Hop	0.21	0.19	0.13	0.12	0.11
本文优化算法	0.12	0.06	0.07	0.06	0.06

经过对表 4 的细致分析,可以明确观察到在节点总数达到 100 个时,本文所提算法相较于其他三种算法,其平均定位误差分别实现了 0.165 638、0.069 855 和 0.084 416 的降低。而当节点总数提升至 450 个时,本文算法的平均定位误差相较于其他三种算法,分别实现了 0.218 087、0.086 697 和 0.057 319 的进一步降低。这些数据充分证明了无论是在平均定位误差还是最大定位误差上,本文所提算法均表现出了优于其他三种算法的性能。

#### 4 结束语

本文分析了经典三维 DV-Hop 算法中跳数获取,平均跳距计算,极大似然法计算未知节点坐标 3 个方面的不足,针对性的提出采用四通信半径优化节点间跳数,融入加权思想,通过节点之间距离对平均跳距进行改进,最后在细化节点之间跳数与优化平均跳距的基础上,通过改进后的麻雀算法对未知节点坐标进行求解。仿真实验结果表明,本文提出的算法相比于经典三维 DV-Hop、3DSSADV-Hop、文献 [8] 三种算法从不同角度对比有更高的精度,但也提高了增加了一定的算法运行时间,下一步工作将着手于降低算法的计算量与复杂度。

#### 参考文献:

- [1] Sensor Research; Study Results from University of Malaya Update Understanding of Sensor Research (A Survey on Underwater Wireless Sensor Networks: Requirements, Taxonomy, Recent Advances, and Open Research Challenges) [EB/OL]. 2023.
- [2] CHITHALURU P, AL-TURJMAN F, STEPHAN T, et al. An optimized bio-inspired localization routing technique for sustainable IIoT networks & green cities [J]. Sustainable Cities and Society, 2023, 97: 104722.
- [3] YANG X, ZHANG W, TAN C, et al. A novel localization technology based on DV-Hop for future internet of things [J]. Electronics, 2023, 12 (15): 3220.
- [4] XUE D. Research of localization algorithm for wireless sensor network based on DV-Hop [J]. EURASIP Journal on Wireless Communications and Networking, 2019 (1): 218.
- [5] 吕淑芳. 无线传感器网络节点定位研究综述 [J]. 传感器与微系统, 2016, 35 (5): 1-3.
- [6] GUPTA A, SINGH U. Optimal performance evaluation of localization of sensor nodes in wireless sensor networks [J]. Wireless Personal Communications, 2023, 131 (2): 941-954.
- [7] 张兴会, 张志辉, 邓志东. 基于 RSSI 测距技术的三角形面积和定位算法 [J]. 电子测量技术, 2008 (11): 92-94.
- [8] HATAMI A, PAHLAVAN K, HEIDARI M, et al. On RSS and TOA based indoor geolocation-a comparative performance evaluation [C] //IEEE Wireless Communications and Networking Conference, USA: IEEE, 2006: 2267-2272.
- [9] LALAMA ZAHIA, FOUZI S, SAMRA B. Node localization optimization in WSNS by using cat swarm optimization meta-heuristic [J]. Automatic Control and Computer Sciences, 2023, 57 (2): 177-184.
- [10] WANG Y. Research on indoor wireless positioning method based on center-of-mass algorithm [J]. Journal of Physics: Conference Series, 2023, 2560 (1): 012038.
- [11] YING ZHANG, QIMAN ZHANG. Research on APIT localization algorithm in wireless sensor networks [C] //第 30 届中国控制与决策会议论文集 (4), 东北大学、中国自动化学会信息物理系统控制与决策专业委员会, 2018: 1335-1339.
- [12] 时雨农, 刘海隆. 基于跳数与跳距优化的三维 DV-Hop 定位算法研究 [J]. 传感技术学报, 2022, 35 (8): 1080-1085.
- [13] LIU J, LIU M, DU X, et al. An improved DV-Hop algorithm for wireless sensor networks based on neural dynamics [J]. Neurocomputing, 2022, 491: 172-185.
- [14] JIA W, QI G, LIU M, et al. A high accuracy localization algorithm with DV-Hop and fruit fly optimization in anisotropic wireless networks [J]. Journal of King Saud University - Computer and Information Sciences, 2022, 34 (10): 8102-8111.
- [15] MOHANTA T K, DAS D K. Improved wireless sensor network localization algorithm based on selective opposition class topper optimization (SOCTO) [J]. Wireless Personal Communications, 2023, 128 (4): 2847-2868.
- [16] DV based positioning in ad hoc networks. pdf [Z].
- [17] XUE J, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm [J]. Systems Science & Control Engineering, 2020, 8 (1): 22-34.
- [18] TIZHOOSH H R. Opposition-Based learning: a new scheme for machine intelligence [C] //International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06): Austria: IEEE, 2005 (1): 695-701.
- [19] 孙小琳, 季伟东, 王旭. 基于余弦相似度反向策略的自然计算方法 [J]. 信息与控制, 2022, 51 (6): 708-718.
- [20] KRISHNANAND K N, GHOSE D. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics [C] //Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005. Pasadena, CA, USA: IEEE, 2005: 84-91.