•计算机科学与技术•

DOI:10.12454/j.jsuese.202400614

本刊网刊

融合测距修正和蜘蛛蜂优化的WSN定位算法

余修武,肖　林*,刘　永,叶　莱

(南华大学 资源环境与安全工程学院,湖南 衡阳 421001)

摘　要:针对无线传感器网络节点定位中非测距DV–Hop算法存在显著定位误差的问题,提出了一种融合测距修正和蜘蛛蜂优化的DV–Hop算法。首先,选用杰卡德系数作为跳数计算的度量标准,以提高跳数的准确性。在获取跳数信息后,引入可信度计算以调整跳距,更准确地反映节点间的实际距离。然后,为了进一步优化节点位置计算、提高DV–Hop算法的精度,引入并改进了蜘蛛蜂优化算法。在种群初始化阶段采用Circle映射反向学习策略,以保证种群的分布状态更加均匀。在位置更新过程中引入自适应权重,以优化算法的收敛速度。在交配操作完成后,将柯西–高斯变异扰动应用于蜘蛛蜂群中最优个体位置,以防陷入局部最优。仿真结果显示,针对不同区域面积、区域形状、锚节点数量、通信半径和节点总数5种情况下,改进后的算法相较于传统DV–Hop定位算法,定位误差分别减小了30.0%、33.0%、37.2%、38.9%和45.9%,同时算法运行时间也减少了0.73 s。

关键词:无线传感器网络;DV–Hop;测距修正;定位误差;蜘蛛蜂优化算法

中图分类号:TP393　　**文献标志码:**A　　**文章编号:**2096-3246(2025)05-0333-11

无线传感器网络(WSN)是由一些独立工作的无线传感器节点组成的分布式网络[1–3]。随着智能化技术的快速发展,WSN已被广泛应用于公共安全[4]、生态环保[5]、应急指挥[6]、智能交通[7]、智能家居[8]等诸多领域。在利用WSN进行数据采集和监测时,数百个无线传感器节点被随机分布在待监测区域。为确保无线传感器监测信息发挥作用,准确获取这些未知节点的位置信息至关重要,所以,定位技术已经成为WSN中必不可少的核心技术[9–12]。根据是否需要节点间的距离或相对方向信息,定位算法可分为两类:基于测距和无须测距[13–16]。DV–Hop属于无须测距的定位算法,其定位依赖跳数和平均跳距,具备硬件成本低和计算简单等优势。然而,随着实际应用需求的提高,DV–Hop算法在某些场景下可能面临定位精度不足的挑战[17]。例如,在智慧城市的建筑密集区域中,由于建筑物阻挡和多径效应,无线信号传播路径复杂,跳数计算可能出现偏差,进而影响节点定位的准确性。此外,在灾害监测和应急救援等场景中,节点分布通常较为稀疏,跳数和平均跳距的估计可能不够精确,导致定位误差增加。

现代启发式算法,亦称作智能优化算法,具备全局优化能力和高度的通用性,并且适用于并行计算。典型的算法包括粒子群优化(PSO)[18]、和声搜索[19]、灰狼优化(GWO)[20]、花授粉算法(FPA)[21]等。随着智能优化算法的发展,学者对其进行了改进,并将其应用于WSN中的定位问题[22–24]。Pu等[25]设计了一种基于马氏距离和距离误差的种群多样性表达式,该表达式的计算结果可量化种群个体间的差异程度。根据这一多样性指标,算法对种群进行分组与交互,大大提高了布谷鸟搜索算法的搜索能力,同时提出了一种种群更新漂移策略,以提高算法在全局搜索中的优化能力。基于种群分组和漂移策略的改进布谷鸟搜索算法(ICS–GD)在解决WSN节点定位误差问题方面显示出显著的优势,缺点是计算烦琐且局部搜索能力有待提升。Jia等[26]提出了一种名为CAFOA–DV–Hop的自适应步长调整混沌果蝇优化算法,其核

收稿日期:2024 – 08 – 05　　**修回日期:**2024 – 11 – 25　　**网络出版日期:**2025 – 07 – 10

基金项目:湖南省自然科学基金项目(2024JJ5338);国家自然科学基金项目(11875164)

作者简介:余修武(1976—),男,教授,博士. 研究方向:无线传感器网络与智能安全监控. E-mail: yxw@usc.edu.cn

* **通信作者:**肖　林,E-mail: 408179601@qq.com

心亮点在于创新的自适应搜索步长机制。此机制不仅大幅提升了算法的全局探索性能，还成功地在全局与局部优化间构建了有效的平衡，因此，CAFOA-DV-Hop算法在收敛速度及定位精确度两方面均实现了显著的性能跃升。Ouyang等[27]提出了一种改进的自适应遗传算法（IAGA），该算法切换了交叉和变异操作的顺序，动态调整了交叉和变异算子的概率，成功地提高了DV-Hop算法的定位精度，然而并未减少由于跳数、跳距等原因导致的定位误差。Ou等[28]提出了一种基于模糊逻辑和高斯–柯西策略的改进布谷鸟搜索算法（ICS-FG），利用模糊逻辑更新参数并平衡全局探索的搜索能力，还提出了一种基于高斯分布和柯西分布的高斯–柯西策略，以提高布谷鸟搜索算法的局部搜索能力，但该方法同样忽略了由于跳数、跳距等导致的定位误差。Bhat等[29]介绍了一种利用哈里斯鹰优化（HHO）算法实现的定位方法。该方法首先致力于将搜索区域缩减至最小范围，随后在该限定区域内应用HHO算法进行求解，旨在提升不规则WSN的定位精确度。方旺盛等[30]首先在细化节点间的跳数阶段利用杰卡德系数，然后借助差分误差系数对各节点间的平均跳距进行修正，成功改进了DV-Hop算法的定位精度。然而，当节点数量较少、锚节点覆盖率较低时，该算法的定位精度仍有待进一步提高。

因此，本文提出了一种融合测距修正和蜘蛛蜂优化的改进DV-Hop算法（DV-Hop localization algorithm integrated with range correction and spider wasp optimization，SWODV-Hop），先利用杰卡德系数有效控制节点间的跳数信息，减小节点间跳数不合理对定位结果造成的误差，再通过可信度计算来调整跳距，以更准确地反映节点间的实际距离。此外，引入蜘蛛蜂优化算法并利用多种策略对其进行改进，计算节点位置，从而提升算法寻优效率和定位精度。

1 DV-Hop定位算法及误差分析

1.1 DV–Hop定位算法介绍

DV–Hop定位算法分为3个步骤：首先估算未知节点与锚节点之间的最小跳数，接着估算未知节点与锚节点之间的平均跳距，最后估算未知节点的位置[19]。

1.1.1 估算最小跳数

WSN中所有锚节点遵循距离矢量协议，通过广播消息的方式，每个锚节点将自己的跳数信息传递给邻居节点，每个节点接收到邻居节点的跳数信息后，将自己的跳数加1，并将该信息传递给自己的邻居节点。这样，通过不断传递跳数信息，最终每个节点都能够得到其他节点的最小跳数。

1.1.2 估算平均跳距

各锚节点确定自身位置之后，就可根据式（1）确定自身的平均跳距：

$$H_i=\frac{\sum\limits_{i\neq j}\sqrt{\left(x_i-x_j\right)^2+\left(y_i-y_j\right)^2}}{\sum\limits_{i\neq j}^{N}h_{ij}} \tag{1}$$

式中，坐标(x_i,y_i)和(x_j,y_j)分别为锚节点i和j的位置，h_{ij}为这两个锚节点之间的最小跳数。

锚节点会广播其计算出的平均跳距，每个未知节点一旦接收到首个此类数值，即将其作为自身的平均跳距。当未知节点u接收到来自锚节点i的平均跳距H_i时，使用式（2）计算其与该锚节点之间的距离d_{iu}：

$$d_{iu}=H_i\times h_{iu} \tag{2}$$

式中，h_{iu}为未知节点u到锚节点i的最小跳数。

1.1.3 估算未知节点位置

根据记录到的每个锚节点的跳转距离，利用三边测量法或最大似然估计法确定未知节点坐标。图1为三边测量法示意图。

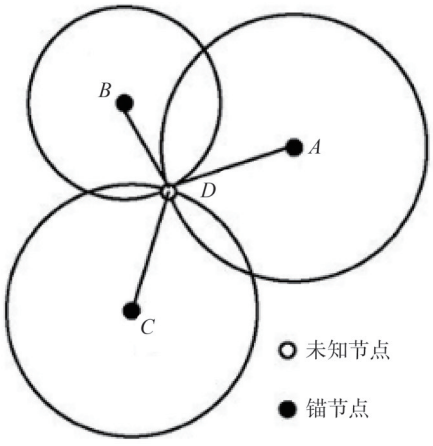

图1 三边测量法示意图

Fig. 1 Schematic diagram of trilateration method

1.2 误差分析

1.2.1 跳数误差

DV-Hop算法的第一阶段基于传统的距离矢量协议来确定跳数，却未考量相邻节点间距的差异。在实际物理环境中，节点在网络中的分布并不均匀，部分区域节点密集，而其他区域则相对稀疏，节点密度各异，导致每一跳的实际距离并不一致。这种不一致会引发跳数误差，因为跳数并不能准确反映节点间的实际物理距离。在节点稀疏的区域，跳数可能会被高估，

而在节点密集区域，跳数可能会被低估。

1.2.2　平均跳距误差

DV-Hop算法依据未知节点至最近锚节点的平均跳距来估算它们之间的距离。然而，在复杂的WSN环境中，因为节点数量众多，单个锚节点难以全面反映整个WSN的节点信息。如果未知节点与该锚节点之间的平均跳距误差较大，计算结果可能存在显著误差，从而影响节点定位的准确性。

1.2.3　计算误差

在节点位置的计算中，节点定位一般采用三边测量法或最大似然估计法。然而，三边测量法依赖于节点到至少3个锚节点的距离测量，信号传播的不稳定性和障碍物对信号的影响都可能导致三边测量法中的距离测量误差。在最大似然估计法中，通常使用信号传播模型来建模节点之间的距离，矩阵误差偏移、信号传播的复杂性以及概率模型的假设等都可能导致节点位置计算误差。

2　蜘蛛蜂优化算法

2.1　初始种群的生成

每只蜘蛛蜂（雌性）代表当前一代中的一个解，使用式（3）对其在搜索空间中的位置进行随机初始化：

$$\boldsymbol{S}_i^t=\boldsymbol{L}+\boldsymbol{r}\times(\boldsymbol{H}-\boldsymbol{L}) \tag{3}$$

式中：$\mathbf{S}_i^t$为第i个个体在搜索空间中的位置向量，上标t为代的索引，下标i为种群索引（i=1,2,⋯,M；M为种群规模）；$\boldsymbol{r}$为D维向量，其各分量独立服从[0,1]的均匀分布；$\boldsymbol{L}$为下界向量，表示每个维度参数的最小取值范围；$\boldsymbol{H}$为上界向量，表示每个维度参数的最大取值范围。

2.2　搜索阶段

雌性蜘蛛蜂通过随机搜索空间找到适合它们后代的蜘蛛，式（4）中的两种情况相互补充以探索搜索空间并定位最有希望的区域，随机生成下一个位置。

$$\boldsymbol{S}_i^{t+1}=\begin{cases}\boldsymbol{S}_i^t+\mu_1\times(\boldsymbol{S}_a^t-\boldsymbol{S}_b^t), r_1<r_2;\\ \boldsymbol{S}_c^t+\mu_2\times(\boldsymbol{L}+\boldsymbol{r}_3\times(\boldsymbol{H}-\boldsymbol{L})), \text{其他}\end{cases} \tag{4}$$

式中：r_1和r_2为[0,1]中的2个随机数；$\boldsymbol{r}_3$为[0,1]中的随机数构成的向量；$\boldsymbol{S}_a^t$、$\boldsymbol{S}_b^t$和$\boldsymbol{S}_c^t$为第t代迭代中种群中第a、b、c个蜘蛛蜂个体的D维位置向量，下标a、b、c为从种群中随机选择的3个索引；μ_1、μ_2可通过式（5）～（6）确定。

$$\mu_1=|r|\times r_4 \tag{5}$$

$$\mu_2=\frac{1}{1+\mathrm{e}^l}\times\cos(2\pi l) \tag{6}$$

式（5）～（6）中，r_4为[0,1]中的随机数，r为服从正态分布的随机数，l为[−2,1]中的随机数。

2.3　跟踪和逃逸阶段

蜘蛛受到攻击后掉落到地面并逃跑，蜘蛛蜂跟踪掉落的蜘蛛，在蜘蛛逃跑时捕捉蜘蛛，在这种情况下，使用式（7）来更新蜘蛛蜂的位置，同时设置一个距离因子Q来模拟蜘蛛躲避蜘蛛蜂，Q通过式（8）确定：

$$\boldsymbol{S}_i^{t+1}=\boldsymbol{S}_i^t+Q\times\left|2\times\boldsymbol{r}_5\times\boldsymbol{S}_a^t-\boldsymbol{S}_i^t\right| \tag{7}$$

$$Q=\left(2-2\times\left(\frac{t}{T}\right)\right)\times r_6 \tag{8}$$

式（7）～（8）中，下标a为从总体中随机选择的索引，t和T分别为当前和最大迭代次数，$\boldsymbol{r}_5$为[0,1]中的随机数构成的向量，r_6为[0,1]中的随机数。

2.4　筑巢行为

蜘蛛蜂把捕捉到的蜘蛛拉进预先准备好的巢穴。因为蜘蛛蜂的筑巢行为多种多样，所以采用两个不同的方程来模拟这些行为，如式（9）所示。

$$\mathbf{S}_i^{t+1}=\begin{cases}\boldsymbol{S}^*+\cos(2\pi l)\times(\boldsymbol{S}^*-\boldsymbol{S}_i^t), r_1<r_2;\\ \boldsymbol{S}_a^t+r_7\times|\gamma|\times(\boldsymbol{S}_a^t-\boldsymbol{S}_i^t)+(1-r_3)\times\boldsymbol{U}\times\\ \quad(\boldsymbol{S}_b^t-\boldsymbol{S}_c^t), \text{其他}\end{cases} \tag{9}$$

式中：$\boldsymbol{S}^*$为迄今为止最好的解决方案；r_7为[0,1]中的随机数；γ为根据Levy飞行生成的数；$\boldsymbol{U}$用于确定何时应用步长以避免在相同位置构建两个嵌套的二进制向量，通过式（10）确定。

$$\boldsymbol{U}=\begin{cases}1, \boldsymbol{r}_8>\boldsymbol{r}_9;\\ 0, \text{其他}\end{cases} \tag{10}$$

式中，$\boldsymbol{r}_8$和$\boldsymbol{r}_9$为[0,1]中的随机数构成的向量。

2.5　交配行为

蜘蛛蜂优化算法的主要特征之一是可以确定蜘蛛蜂的雌雄。在这种方法中，每只蜘蛛蜂代表当前世代中的一个可能解，蜘蛛蜂卵代表该世代中新生成的潜在解。本文算法中，潜在解的生成方式如下。

根据式（11）生成与雌性蜘蛛蜂不同的雄性蜘蛛蜂子代：

$$\boldsymbol{S}_m^{t+1}=\boldsymbol{S}_i^t+\mathrm{e}^l\times|\beta|\times\boldsymbol{v}_1+(1-\mathrm{e}^l)\times|\beta_1|\times\boldsymbol{v}_2 \tag{11}$$

式中，β和β_1为根据正态分布生成的随机数，$\boldsymbol{v}_1$和$\boldsymbol{v}_2$根据式（12）～（13）计算。

$$\boldsymbol{v}_1=\begin{cases}\boldsymbol{x}_a-\boldsymbol{x}_i, f(\boldsymbol{x}_a)<f(\boldsymbol{x}_i);\\ \boldsymbol{x}_i-\boldsymbol{x}_a, \text{其他}\end{cases} \tag{12}$$

$$\boldsymbol{v}_2=\begin{cases}\boldsymbol{x}_b-\boldsymbol{x}_c, f(\boldsymbol{x}_b)<f(\boldsymbol{x}_c);\\ \boldsymbol{x}_c-\boldsymbol{x}_b, \text{其他}\end{cases} \tag{13}$$

式（12）～（13）中：$\boldsymbol{v}_1$和$\boldsymbol{v}_2$为由种群中不同潜在解之间

的差值计算得到的方向向量，它们利用已有解的相对位置，引导算法在搜索空间中向更优区域移动，帮助生成新的候选解，从而提高优化效果；$\boldsymbol{x}_a$、$\boldsymbol{x}_b$、$\boldsymbol{x}_c$为在种群中随机选取的不同个体的位置向量，下标a、b和c满足$a\neq i\neq b\neq c$，用于重新组合双亲蜘蛛蜂的遗传材料，以产生一个共享其父母特征的后代（卵）。

3 SWODV-Hop定位算法

3.1 跳数改进

为了降低节点间跳数对定位精度的影响，引入杰卡德系数对DV-Hop算法的跳数计算方式进行改进[30]。

杰卡德系数是一种衡量集合相似度的指标，表示两个集合的重叠程度，其计算公式如下：

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|}=\frac{|A\cap B|}{|A|+|B|+|A\cap B|} \tag{14}$$

式中，$J(A,B)$为集合A和B的杰卡德系数，$|A\cap B|$为集合A和B的交集元素数量，$|A\cup B|$为集合A和B的并集元素数量。

与杰卡德系数相关的杰卡德距离则用于衡量集合间的差异程度，其数值接近0表示集合高度相似，接近1则表明二者相差较大，其表达式如下：

$$d_{\mathrm{j}}(A,B)=1-J(A,B) \tag{15}$$

式中，$d_{\mathrm{j}}(A,B)$为集合A和B的杰卡德距离。

在WSN中，杰卡德系数不仅可以衡量集合的相似性，还可以用于优化跳数计算。具体而言，在邻居节点的通信范围内，可将其作为修正因子τ_i，并结合相交区域的节点数量，对跳数计算进行优化。这样能够更精确地估算节点之间的跳数，进而提高定位精度。

如图2所示，锚节点A的通信半径内包含多个未知节点。以未知节点B和C为例，若假设其通信半径相等，则可分别用N_A、N_B、N_C表示锚节点与未知节点的通信范围内的节点数量，可视为不同的集合。

因此，节点A和B通信半径内相交区域的杰卡德系数为

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|}=\frac{N_{A\cap B}}{N_A+N_B}=\frac{9}{18+17}\approx 0.257,$$

杰卡德距离为

$$d_{\mathrm{j}}(A,B)=1-J(A,B)=0.743,$$

同理可得，$J(A,C)=0.353$，$d_{\mathrm{j}}(A,C)=1-J(A,C)=0.647$。

通过分析杰卡德距离，可以评估未知节点与锚节点的相对距离。随着未知节点远离锚节点，杰卡德距离逐渐增大。例如，当$d_{\mathrm{j}}(A,B)=0.743$，说明节点$A$与$B$之间的距离接近单跳。

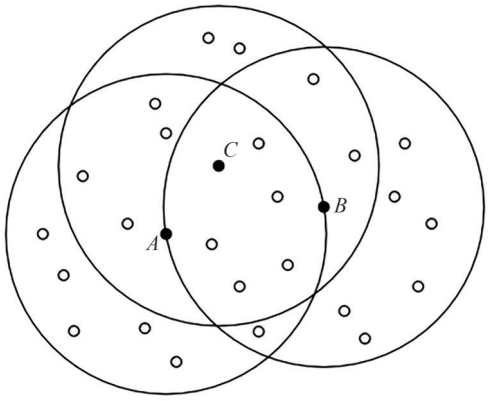

图2 节点通信半径内相交区域示意图

Fig. 2 Schematic diagram of intersecting areas within the communication radius of nodes

为了更精确地描述跳数关系，本文引入基于修正因子τ_i的优化策略。该修正因子由杰卡德系数构造，并用于调整通信范围内的跳数计算，修正后的最小跳数h'_{ij}的表达式如下：

$$h'_{ij}=1-\tau_i \tag{16}$$

式中，$\tau_i=J(A,B)$为锚节点A与未知节点B在通信范围内的相交区域的杰卡德系数。

该修正方法能够更加精确地反映节点间的跳数关系，从而提升DV-Hop算法的定位精度。

3.2 平均跳距改进

在计算锚节点间的距离时，采用直线距离而非曲线距离，然而，由于网络拓扑的复杂性，直线距离与实际路径可能存在一定偏差，从而影响定位精度。为减小该误差，引入跳距可信度$P(i,j)$以提高距离估算的准确性[31]。

$$P(i,j)=\frac{d_{ij}}{R\times h'_{ij}} \tag{17}$$

式中，d_{ij}为锚节点i和j之间的距离，R为节点的通信半径，h'_{ij}为基于杰卡德系数计算的最小跳数。

平均跳距通常通过锚节点对之间的距离与跳数的比值计算。然而，网络环境的复杂性导致锚节点对未知节点的影响力并不均衡。因此，在计算平均跳距时，需要引入权重分配机制，根据跳距可信度$D(i,j)$为不同的锚节点对分配相应的权重。权值$W(i,j)$的具体计算方式如下：

$$W(i,j)=1-(1-P(i,j))^2 \tag{18}$$

进一步地，由于未知节点u在多个锚节点的影响下，其跳数估算需考虑不同锚节点对的权重。因此，引入权重因子δ_i来调整各锚节点对未知节点的影响力，其计算方式如下：

$$\delta_i=\frac{\dfrac{1}{h'_{iu}}}{\sum\limits_{j=1}^{m}\dfrac{1}{h'_{ij}}} \tag{19}$$

式中，h'_{iu}为根据杰卡德系数计算所得的未知节点u到周围锚节点i的最小跳数，m为所有锚节点个数。

基于权值$W(i,j)$和δ_i改进后的平均跳距计算如下：

$$H'_i=\delta_i\times W(i,j)\times\frac{\sum\limits_{i\neq j}^{N}\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}}{\sum\limits_{i\neq j}^{N}h'_{ij}} \tag{20}$$

式中，N为未知节点个数。

最终，利用修正后的平均跳距可估算未知节点u与各锚节点之间的距离d'_{iu}。

$$d'_{iu}=H'_i\times h'_{ij} \tag{21}$$

通过引入跳距可信度和权重分配策略，该方法有效降低了直线距离替代曲线距离所带来的误差，能够更精确地反映锚节点与未知节点之间的真实距离关系，从而提升WSN中DV-Hop算法的定位精度。

3.3　改进的蜘蛛蜂优化算法

3.3.1　Circle映射反向学习初始化

蜘蛛蜂优化算法采用随机方式进行种群初始化，导致蜘蛛蜂个体在搜索空间内分布不均。因此，在迭代初期，算法的优化效果不佳，搜索速率下降。为解决这一问题，提出了一种结合Circle映射和反向学习的方法用于种群初始化，以确保蜘蛛蜂个体的初始位置分布更加均匀。以下是具体实施步骤：

首先，利用Circle映射生成N个蜘蛛蜂初始位置$\boldsymbol{S}_i^{t+1}$：

$$\boldsymbol{S}_i^{t+1}=\mathrm{mod}\left(\boldsymbol{S}_i^t+0.2-\frac{\sin(2\pi\boldsymbol{S}_i^t)}{4\pi}\right) \tag{22}$$

然后，利用式(23)将式(22)生成的N个初始解生成相对应的反向初始解$\boldsymbol{S}_{i,n}^t$：

$$\boldsymbol{S}_{i,n}^t=\mathrm{rand}(u_\mathrm{b}-l_\mathrm{b})-\boldsymbol{S}_i^t \tag{23}$$

式中，u_b和l_b为当前蜘蛛蜂所在位置的上下界。

最后，对生成的$2N$个$\boldsymbol{S}_i^{t+1}$和$\boldsymbol{S}_{i,n}^t$的蜘蛛蜂位置，进行函数适应度排序，并选取顺序排列的前N个值作为初始蜘蛛蜂群。

3.2.2　自适应权重位置更新

引入自适应权重$w(t)$用于动态更新蜘蛛蜂的位置以提高算法的性能，$w(t)$随迭代次数t变化。在算法的初始阶段，通过减小自适应权重来降低个体对整个种群位置调整的影响，促使算法更加注重全局搜索。自适应权重的大小随着迭代次数的增加而逐步增加，增强了最优蜘蛛蜂位置的影响力，并使其他蜘蛛蜂能够更快靠近最优蜘蛛蜂位置，加快整个算法的收敛速度。由迭代次数t形成的自适应权重如下：

$$w(t)=0.2\cos\left(\frac{\pi}{2}\left(1-\frac{t}{T}\right)\right) \tag{24}$$

自适应权重$w(t)$呈现出非线性的动态变化，其取值范围限定在[0,1]。由于cos函数在[0,π/2]内初始阶段变化率较大，因此在算法的初始阶段，$w(t)$值较小但变化迅速；而在算法的后期，$w(t)$值逐渐增大，但其变化速度逐渐减缓。这种设计的目的在于充分确保算法的收敛性。经过改进的蜘蛛蜂优化算法的位置更新公式如下：

$$\boldsymbol{S}_i^{t+1}=\begin{cases}w(t)\boldsymbol{S}_i^t+\mu_1\times\left(\boldsymbol{S}_a^t-\boldsymbol{S}_b^t\right), r_1<r_2;\\ w(t)\boldsymbol{S}_c^t+\mu_2\times\left(\boldsymbol{L}+\boldsymbol{r}_3\times(\boldsymbol{H}-\boldsymbol{L})\right),\text{其他}\end{cases} \tag{25}$$

$$\boldsymbol{S}_i^{t+1}=w(t)\boldsymbol{S}_i^t+Q\times\left|2\times\boldsymbol{r}_5\times\boldsymbol{S}_a^t-\boldsymbol{S}_i^t\right| \tag{26}$$

$$\boldsymbol{S}_i^{t+1}=\begin{cases}w(t)\boldsymbol{S}^*+\cos(2\pi l)\times(\boldsymbol{S}^*-\boldsymbol{S}_i^t), r_1<r_2;\\ w(t)\boldsymbol{S}_a^t+r_7\times|\gamma|\times(\boldsymbol{S}_a^t-\boldsymbol{S}_i^t)+(1-r_3)\times\\ \quad\boldsymbol{U}\times(\boldsymbol{S}_b^t-\boldsymbol{S}_c^t),\text{其他}\end{cases} \tag{27}$$

3.3.3　柯西–高斯变异

随着算法运行过程中迭代次数的增加，蜘蛛蜂种群的位置差异逐渐减小，导致算法容易陷入局部最优，因此，引入柯西–高斯变异，加大算法跳出局部最优的概率，提升算法的寻优精度。柯西–高斯变异方法如式(28)所示：

$$\boldsymbol{S}_\mathrm{new}^*=\boldsymbol{S}^*\times(1+k\times\boldsymbol{G}_{(0,1)}) \tag{28}$$

式中，$\boldsymbol{S}^*$为蜘蛛蜂的最优个体位置，k为[0,1]中的随机数，$\boldsymbol{G}_{(0,1)}$为服从均值为0、方差为1的高斯分布的随机向量，$\boldsymbol{S}_\mathrm{new}^*$为变异后的位置。

3.3.4　算法流程

SWODV-Hop算法通过上述方法，利用杰卡德系数计算跳数、利用可信度修正跳距，以及运用蜘蛛蜂优化算法来提高定位精度，具体步骤如下。

步骤1：进行网络初始化。将初始区域尺寸设定为$M\times M$，定义N_1为锚节点数，N_2为未知节点数，R为通信半径。

步骤2：利用杰卡德系数计算未知节点和锚节点之间的最小跳数，利用跳距可信度计算节点间平均跳距，计算未知节点到锚节点的距离。

步骤3：生成适应度函数值$f(x,y)$，并利用Circle映射反向学习对蜘蛛蜂种群位置进行初始化。

步骤4：引入自适应权重，根据式(25)～(27)进行迭代，更新蜘蛛蜂的位置并计算种群个体的适应度值。

步骤 5：根据式（11），生成一定数量的子代个体。

步骤 6：通过柯西–高斯变异，对最优蜘蛛蜂个体位置进行扰动操作。

步骤 7：判断当前迭代次数 t 是否大于最大迭代次数 T。若 $t<T$，则继续进行迭代；否则，输出最优解。

SWODV–Hop 算法流程图如图 3 所示。

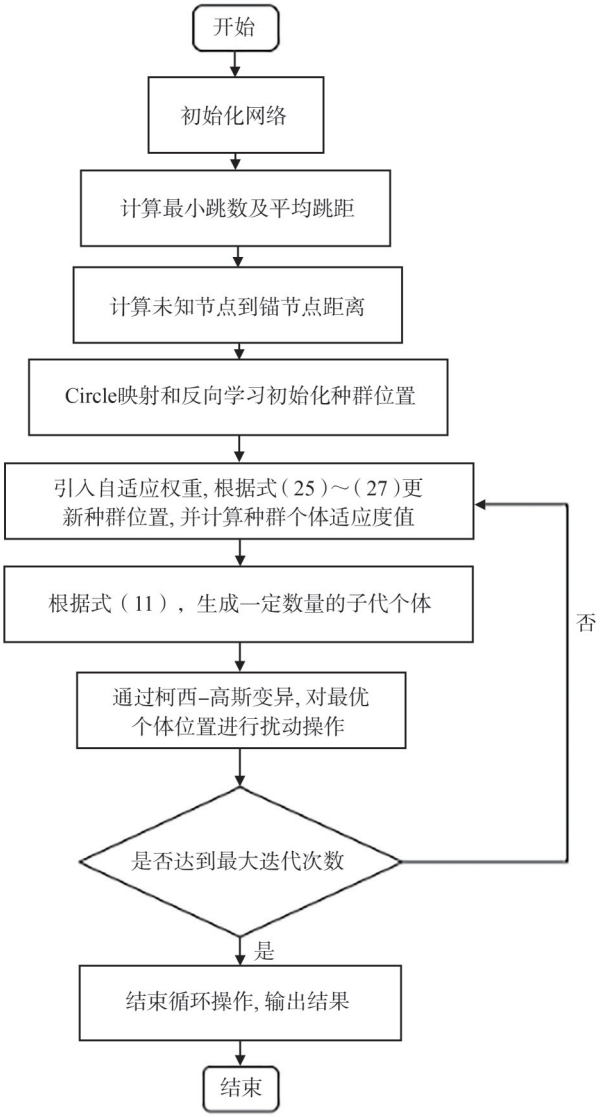

图 3 SWODV–Hop 算法流程图

Fig. 3 SWODV–Hop algorithm flowchart

3.3.5 时间复杂度

时间复杂度是衡量算法性能的一项关键指标，通过对时间复杂度的分析，能够直观地评估其性能表现。考虑一个 WSN 场景，其节点总数记为 n，锚节点数量记为 m。在此场景下，利用杰卡德系数计算节点间跳数的时间复杂度为 $O(n^3)$，而计算节点间实际物理距离的时间复杂度则为 $O(n)$。此外，计算未知节点至各锚节点距离的时间复杂度为 $O(n\times m)$。利用改进的蜘蛛蜂优化算法计算节点位置，定义最大迭代次数 T、种群规模 N、问题维度 D，则改进的蜘蛛蜂优化算法的复杂度为 $O(T\times D\times N)$。综上所述，整个算法的复杂度为 $O(n^3)+O(n)+O(n\times m)+O(T\times D\times N)$。

采用 Routray 等[20]提出的适应度函数作为 SWO-DV–Hop 算法的适应度函数，具体表达式如下：

$$f(x,y)=\sum_{i=1}^{m}\left(d_i-\sqrt{(x_u-x_i)^2+(y_u-y_i)^2}\right) \tag{29}$$

式中，m 为锚节点的数量，(x_u,y_u) 为未知节点坐标，d_i 为未知节点到锚节点的实际距离。

4 仿真结果与分析

使用 MATLAB（R2021a）软件进行仿真测试，用于测试的计算机配置为英特尔®酷睿™ i7–8565U 处理器，基本频率 1.80 GHz，配备 Windows 10 家庭版中文操作系统。参与比较的算法有 DV–Hop 算法、HPSODV–Hop 算法[32]、ICSDV–Hop 算法和 IAGADV–Hop 算法。仿真实验主要观察锚节点数量、通信半径和节点总数对定位误差的影响。性能评价指标采用归一化平均定位误差，具体表达式如下：

$$E_u=\frac{\sum_{u=1}^{N}\sqrt{\left(x_u-x_u^*\right)^2+\left(y_u-y_u^*\right)^2}}{NR} \tag{30}$$

式中，(x_u^*,y_u^*) 为计算出的节点 u 的坐标。

实验中，WSN 中传感器节点随机分配，部署范围为 100 m×100 m，如图 4 所示。

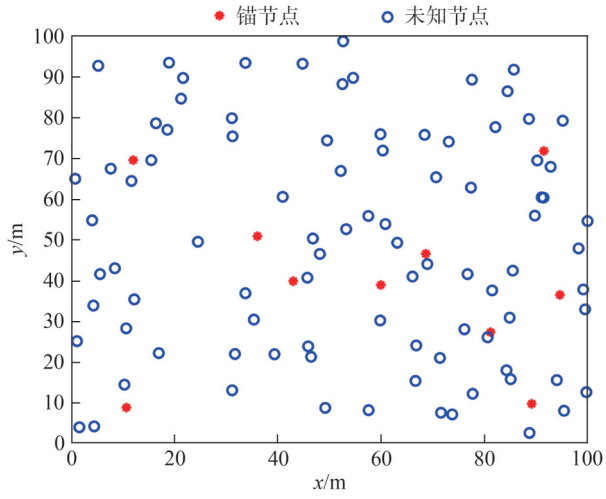

图 4 WSN 中传感器节点分布图

Fig. 4 Distribution of sensor nodes in WSN

4.1 不同网络结构下节点定位误差

4.1.1 不同区域面积节点定位误差

图 5 和 6 为分别为 100 m×100 m 和 80 m×80 m 网络区域未知节点定位误差对比。图 5 和 6 其他参数相同：锚节点数量为 10 个、未知节点数量为 90 个、通信半径为 30 m。仿真结果表明，在 100 m×100 m 网络区

域中，SWODV-Hop算法相较于DV-Hop算法，在相同的条件下节点的定位误差能够显著减小38.0%（图5）。在80 m×80 m网络区域中，SWODV-Hop算法相较于DV-Hop算法，在相同的条件下节点的定位误差能够显著减小30.0%（图6）。由此可见，SWODV-Hop算法在提高节点定位准确性方面表现更为优越。

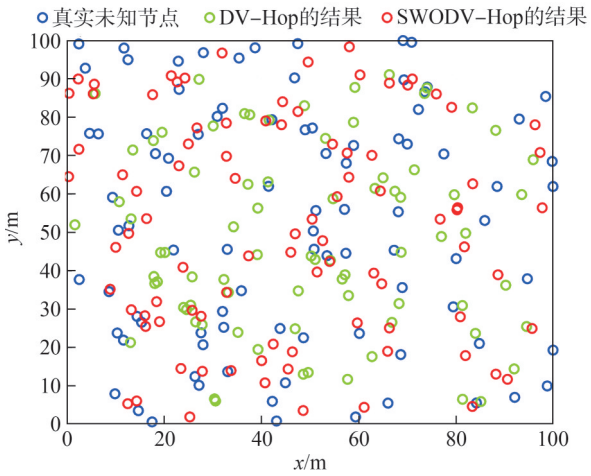

图5　100 m×100 m网络区域未知节点定位误差对比

Fig. 5　Comparison of positioning errors of unknown nodes in 100 m × 100 m network area

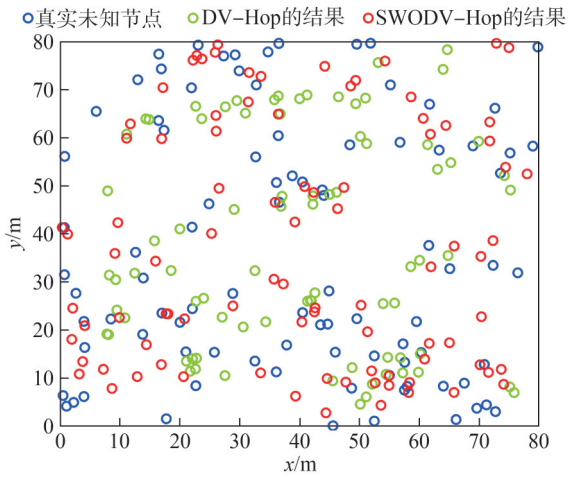

图6　80 m×80 网络区域未知节点定位误差对比

Fig. 6　Comparison of positioning errors of unknown nodes in 80 m × 80 m network area

4.1.2　不同区域形状下节点定位误差

图7为圆形网络区域未知节点定位误差对比。图7与图5网络区域形状不同，其他参数相同。SWODV-Hop算法与DV-Hop算法在节点定位上的误差对比。仿真实验表明，在与100 m×100 m正方形面积相同的圆形网络区域中，SWODV-Hop算法相较于DV-Hop算法，在相同条件下节点的定位误差能够显著减小33.0%。这同样表明了SWODV-Hop算法在提高节点定位准确性方面表现更为优越。

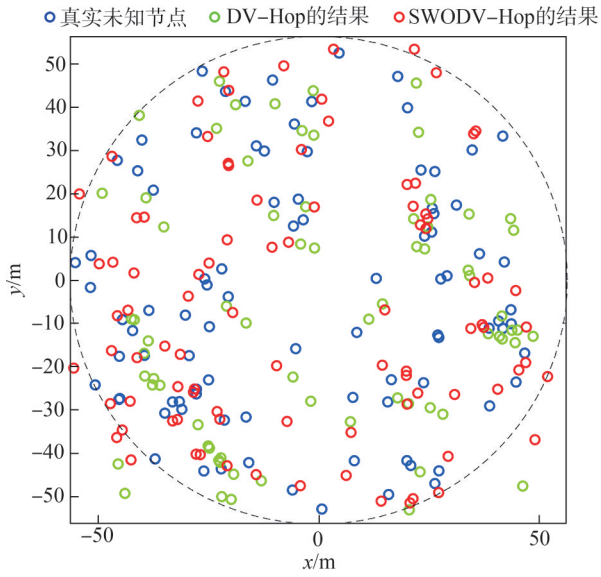

图7　圆形网络区域未知节点定位误差对比

Fig. 7　Comparison of positioning errors of unknown nodes in circular network areas

4.2　锚节点数量对定位精度的影响

在涵盖100个随机节点的网络区域内，以R=30 m为通信半径，研究5种算法在锚节点数量从10个变化至40个的情况下，归一化平均定位误差的变化，结果如图8所示。总体而言，随着锚节点数量的增加，这5种算法的定位误差呈下降趋势。在锚节点数量为10～40个时，SWODV-Hop算法的定位误差均低于其他算法。具体而言，在锚节点数量不同的情况下，相较于DV-Hop、HPSODV-Hop、ICSDV-Hop和IAGADV-Hop算法，SWODV-Hop算法的平均定位误差分别减小了37.2%、17.4%、12.8%和7.8%。这进一步表明SWODV-Hop算法在多种条件下都能提供更为准确的节点定位。

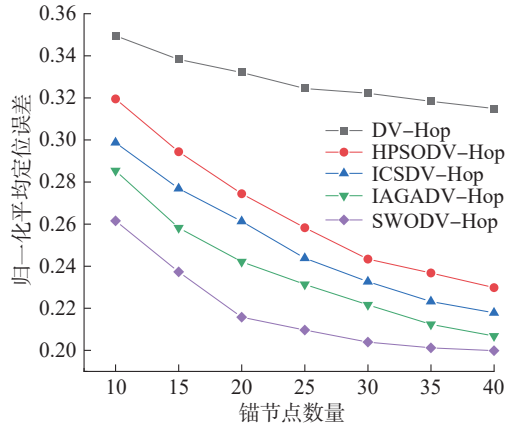

图8　不同锚节点数量定位误差对比

Fig. 8　Comparison of positioning errors for different numbers of anchor nodes

4.3　通信半径对定位精度的影响

图9为不同通信半径定位误差对比。由图9可知，随着通信半径从20 m增至40 m，所有5种算法在节点定

位方面均呈现误差逐步减小的趋势。在这个过程中，SWODV-Hop算法表现出比其他4种算法更为显著的优势，其定位误差明显小于其他算法。具体而言，相较于DV-Hop、HPSODV-Hop、ICSDV-Hop和IAGADV-Hop算法，SWODV-Hop算法的平均定位误差分别减小了38.9%、25.8%、18.2%和8.3%。这表明，针对不同的通信半径，本文提出的SWODV-Hop算法在节点定位方面具有更为优越的性能表现。

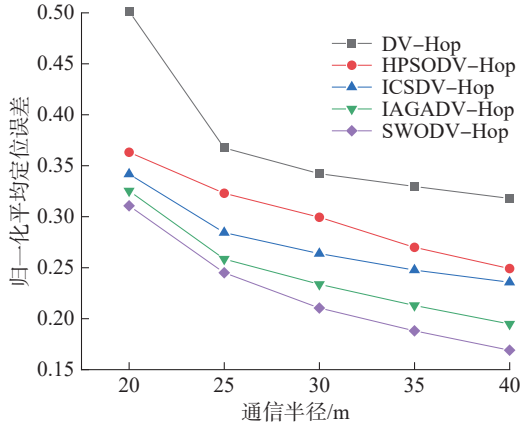

图9 不同通信半径定位误差对比

Fig. 9 Comparison of positioning errors with different communication radius

4.4 节点总数对定位精度的影响

图10展示了在网络区域内，锚节点数量为10个，通信半径为30 m的情况下，5种算法在节点总数分别设置为100、150、200、250、300个时的归一化平均定位误差的变化曲线。由图10可见，随着总节点数的增加，5种定位算法的误差总体呈减小趋势。特别是在不同节点总数的条件下，明显可见SWODV-Hop算法相较于其他4种算法表现出显著的优势。与DV-Hop、HPSODV-Hop、ICSDV-Hop和IAGADV-Hop算法相比，SWODV-Hop算法的定位误差分别减小了45.9%、33.6%、25.9%和13.1%。

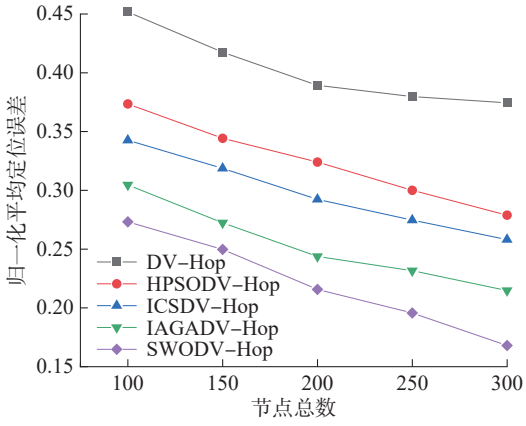

图10 不同节点总数定位误差对比

Fig. 10 Comparison of positioning errors for different numbers of total nodes

4.5 算法运行时间

通过仿真分析，研究算法的效率是否也得到了提高。不同节点总数算法运行时间对比如图11所示。在相同的节点总数条件下，SWODV-Hop算法的运行时间明显少于其他4种参与比较的算法，表明其优化效率最高。SWODV-Hop、IAGADV-Hop、ICSDV-Hop、HPSODV-Hop和DV-Hop算法的平均运行时间分别为0.41、0.50、0.65、0.74和1.14 s。与IAGADV-Hop、ICSDV-Hop、HPSODV-Hop和DV-Hop算法相比，SWODV-Hop算法的平均运行时间分别减少了0.09、0.24、0.33和0.73 s。

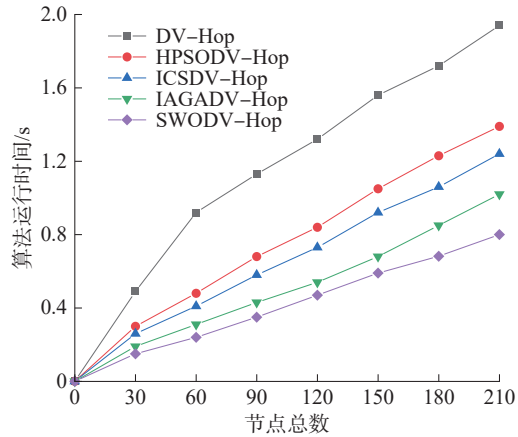

图11 不同节点总数算法运行时间对比

Fig. 11 Comparison of algorithm running time for different numbers of total nodes

5 结 论

本文通过引入杰卡德系数和跳距可信度对DV-Hop定位算法的跳数和误差问题进行了优化。杰卡德系数作为修正因子τ_i，用于更精确地估算节点与其邻居节点之间的跳数，从而提高定位精度；而跳距可信度$D(i,j)$则有效减少了直线距离代替曲线距离引起的误差，并为锚节点对分配不同权重，优化平均跳距的计算。此外，为解决计算误差问题，采用SWODV-Hop算法替代传统的三边测量法或最大似然估计法。在种群初始化阶段，引入Circle映射反向学习策略以保证种群分布均匀；在位置更新中采用自适应权重优化算法收敛速度；交配操作后，对最优个体位置施加柯西-高斯变异扰动，避免陷入局部最优。仿真实验的结果表明，与DV-Hop、HPSODV-Hop、ICSDV-Hop和IAGADV-Hop算法相比，SWODV-Hop算法显著提升了节点定位精度，同时无须额外增加硬件成本。

尽管本文提出的SWODV-Hop算法表现出良好的定位精度和效率，但仍存在局限性。例如，实验基于理想化的网络模型，未充分考虑复杂环境中的传感器故障和信道干扰；对节点移动及计算所引发的能耗优化研究不足；验证手段主要依赖仿真，缺乏实际部署测试。未来

研究可重点关注复杂环境下的鲁棒性测试、低能耗优化设计以及实际场景中的应用验证。同时，可结合深度学习与博弈论等跨学科方法，探索多目标优化在WSN中的应用潜力。改进后的算法在智慧城市目标追踪、农业监测、灾害预警和军事应急等领域具有广阔的应用前景，尤其在物联网和边缘计算等新兴技术的推动下，其在复杂网络环境中的应用价值将进一步提升。

参考文献：

[1] Abd El Ghafour M G, Kamel S H, Abouelseoud Y. Improved DV–Hop based on Squirrel search algorithm for localization in wireless sensor networks[J]. Wireless Networks,2021,27(4):2743–2759.

[2] Kumar S, Kumar S, Batra N.Optimized distance range free localization algorithm for WSN[J].Wireless Personal Communications,2021,117(3):1879–1907.

[3] Kanwar V,Kumar A.DV–Hop localization methods for displaced sensor nodes in wireless sensor network using PSO [J].Wireless Networks,2021,27(1):91–102.

[4] Messous S,Liouane H,Cheikhrouhou O,et al.Improved recursive DV－Hop localization algorithm with RSSI measurement for wireless sensor networks[J].Sensors,2021,21 (12):4152.

[5] Mirsadeghi E, Khodayifar S. Hybridizing particle swarm optimization with simulated annealing and differential evolution[J].Cluster Computing,2021,24(2):1135–1163.

[6] Zhang Zhixia,Cao Yang,Cui Zhihua,et al. A many-objective optimization based intelligent intrusion detection algorithm for enhancing security of vehicular networks in 6G[J].IEEE Transactions on Vehicular Technology, 2021,70(6): 5234－5243.

[7] Cui Zhihua,Zhao Yaru,Cao Yang,et al.Malicious code detection under 5G HetNets based on a multi-objective RBM model[J].IEEE Network,2021,35(2):82–87.

[8] Cai Xingjuan, Geng Shaojin,Wu Di, et al.A multicloud-model-based many–objective intelligent algorithm for efficient task scheduling in Internet of Things[J].IEEE Internet of Things Journal,2021,8(12):9645–9653.

[9] Cai Xingjuan,Cao Yihao,Ren Yeqing,et al.Multi-objective evolutionary 3D face reconstruction based on improved encoder-decoder network[J]. Information Sciences, 2021, 581:233–248.

[10] Jin Yong,Zhou Lin,Zhang Lu, et al.A novel range-free node localization method for wireless sensor networks[J]. IEEE Wireless Communications Letters,2022,11(4):688–692.

[11] Mohanta T K, Das D K. Multiple objective optimization-based DV–Hop localization for spiral deployed wireless sensor networks using non-inertial opposition-based class topper optimization (NOCTO)[J].Computer Communications, 2022,195:173–186.

[12] Chen Tianfei, Hou Shuaixin, Sun Lijun. An enhanced DV－Hop positioning scheme based on spring model and reliable beacon node set[J].Computer Networks,2022,209:108926.

[13] Cai Xingjuan, Geng Shaojin, Wu Di, et al. Unified integration of many-objective optimization algorithm based on temporary offspring for software defects prediction[J]. Swarm and Evolutionary Computation,2021,63:100871.

[14] Liu Wenyan,Luo Xiangyang,Wei Guo,et al.Node localization algorithm for wireless sensor networks based on static anchor node location selection strategy[J].Computer Communications,2022,192:289–298.

[15] Chen Yun, Wang Zidong, Yuan Yuan, et al. Distributed H_∞ filtering for switched stochastic delayed systems over sensor networks with fading measurements[J]. IEEE Transactions on Cybernetics,2020,50(1):2–14.

[16] Cui Zhihua, Zhao Peng, Hu Zhaoming, et al. An improved matrix factorization based model for many-objective optimization recommendation[J]. Information Sciences, 2021, 579:1–14.

[17] Yang Xiaoying,Zhang Wanli,Tan Chengfang,et al.A novel localization technology based on DV–Hop for future internet of things[J].Electronics,2023,12(15):3220.

[18] Piotrowski A P,Napiorkowski J J,Piotrowska A E.Population size in particle swarm optimization[J]. Swarm and Evolutionary Computation,2020,58:100718.

[19] Mahmoudi S M,Rad M M,Ochbelagh D R.Hybrid of the fuzzy logic controller with the harmony search algorithm to PWR in-core fuel management optimization[J].Nuclear Engineering and Technology,2021,53(11):3665–3674.

[20] Routray A, Singh R K, Mahanty R. Harmonic reduction in hybrid cascaded multilevel inverter using modified grey wolf optimization[J].IEEE Transactions on Industry Applications,2020,56(2):1827–1838.

[21] Singh P,Mittal N.An efficient localization approach to locate sensor nodes in 3D wireless sensor networks using adaptive flower pollination algorithm[J]. Wireless Networks,2021,27(3):1999–2014.

[22] Ge Chunpeng,Susilo W, Liu Zhe,et al. Secure keyword

search and data sharing mechanism for cloud computing [J].IEEE Transactions on Dependable and Secure Computing,2021,18(6):2787–2800.

[23] Ren Yongjun,Leng Yan,Qi Jian,et al.Multiple cloud storage mechanism based on blockchain in smart homes[J]. Future Generation Computer Systems,2021,115:304–313.

[24] Almalki K J,Jabbari A,Ayinala K,et al.ELSA:Energy-efficient linear sensor architecture for smart city applications[J]. IEEE Sensors Journal,2022,22(7):7074–7083.

[25] Pu Yuanyuan,Song Junfang,Wu Meng,et al.Node location using cuckoo search algorithm with grouping and drift strategy for WSN[J].Physical Communication,2023,59: 102088.

[26] Jia Wenxian,Qi Guohong,Liu Menghan,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.

[27] Ouyang Aijia,Lu Yinsheng,Liu Yanmin,et al.An improved adaptive genetic algorithm based on DV–Hop for locating nodes in wireless sensor networks[J]. Neurocomputing, 2021,458:500–510.

[28] Ou Xianfeng,Wu Meng,Pu Yuanyuan,et al.Cuckoo search algorithm with fuzzy logic and Gauss–Cauchy for minimizing localization error of WSN[J]. Applied Soft Computing,2022,125:109211.

[29] Bhat S J,Venkata S K.An optimization based localization with area minimization for heterogeneous wireless sensor networks in anisotropic fields[J].Computer Networks, 2020,179:107371.

[30] Fang Wangsheng,Yang Geng,Hu Zhongdong.Improved DV–Hop algorithm with Jaccard coefficient and differential error based on hop distance correction[J]. Computer Engineering and Applications,2018,54(23):57–63.[方旺盛,杨庚,胡中栋.杰卡德系数差分误差跳距修正的DV–Hop改进算法[J].计算机工程与应用,2018,54(23):57–63.]

[31] Zhu Zihang,Chen Hui,Wang Xu.DV–Hop localization algorithm based on adaptive differential particle swarm optimization[J].Journal of Fuyang Normal University(Natural Science),2023,40(1):72–78.[朱子行,陈辉,王旭.基于自适应差分粒子群优化的DV–Hop定位算法[J].阜阳师范大学学报(自然科学版),2023,40(1):72–78.]

[32] Jawad H M,Jawad A M,Nordin R,et al.Accurate empirical path-loss model based on particle swarm optimization for wireless sensor networks in smart agriculture[J].IEEE Sensors Journal,2020,20(1):552–561.

WSN Localization Algorithm Integrating Ranging Correction and Spider Wasp Optimization

YU Xiuwu, *XIAO Lin**, *LIU Yong*, *YE Lai*

(School of Resource Environment and Safety Engineering, University of South China, Hengyang 421001, China)

Abstract:

Objective In the context of node localization in wireless sensor networks (WSNs), the non-range-based DV–Hop algorithm exhibits significant localization errors due to inaccuracies in hop count estimation and the neglect of actual node distances. This study proposes a modified DV–Hop algorithm that incorporates distance correction and a spider wasp optimization (SWO) algorithm to address these limitations. The objective is to improve the accuracy of node localization and enhance the overall performance of the DV–Hop algorithm, making it more reliable in practical deployment scenarios.

Methods The proposed algorithm comprised two main components: distance correction and optimization using SWO. First, the traditional hop count calculation was improved by adopting the Jaccard coefficient as the metric to enhance the accuracy of hop count estimation. The Jaccard coefficient, a well-established similarity measure, ensured that the hop count reflected a more accurate estimate of the network's topology. After acquiring the hop count information, a credibility calculation was introduced to adjust the hop distances, enabling a more accurate representation of the actual distances between nodes.

SWO was incorporated to refine the node position calculation and improve the precision of the DV–Hop algorithm. The initialization of the SWO population was enhanced by a chaos mapping-based reverse learning strategy, which ensured that the population was more uniformly distributed across the search space. During the position update process, adaptive weighting was applied to optimize the convergence speed. Following the mating operation, Cauchy–Gaussian mutation disturbance was introduced to the positions of the best individuals in the Spider Wasp swarm to prevent premature convergence to local optima.

Results and Discussions The proposed algorithm significantly outperformed the conventional DV−Hop algorithm and other related methods in terms of localization accuracy and energy efficiency. The use of the Jaccard coefficient for hop count estimation improved the precision of distance calculation, while the credibility adjustment further enhanced the accuracy of node localization. The integration of SWO, with its improved population initialization and adaptive weighting mechanism, contributed to faster convergence and more precise localization results.

In the simulation experiments, the proposed algorithm reduced the localization error by 30.0%, 33.0%, 37.2%, 38.9%, and 45.9%, respectively, compared to the traditional DV−Hop localization algorithm under five different conditions: varying area size, region shape, number of anchor nodes, communication radius, and total number of nodes. At the same time, the algorithm's running time improved by 0.73 seconds. The incorporation of the chaos mapping strategy in the SWO initialization phase helped achieve a more evenly distributed population, reducing the risk of suboptimal solutions. The introduction of Cauchy－Gaussian mutation after the mating operation prevented the algorithm from becoming trapped in local optima, ensuring better exploration of the solution space.

Conclusions This study presents a novel hybrid localization algorithm that combines distance correction with spider SWO to enhance the DV−Hop algorithm. The integration of the Jaccard coefficient to improve hop count accuracy, combined with the application of SWO using chaos mapping-based initialization, adaptive weighting, and Cauchy−Gaussian mutation, significantly enhances localization precision and energy efficiency. The proposed algorithm exhibits robustness to environmental variations and network dynamics, establishing its effectiveness for real-world wireless sensor network localization tasks. The findings indicate that the algorithm provides a promising solution for increasing the accuracy and reliability of node localization in large-scale WSNs.

Key words: wireless sensor network; DV−Hop algorithm; range correction; localization accuracy; spider wasp optimization algorithm

（编辑　陈　雪）

引用格式: Yu Xiuwu,Xiao Lin,Liu Yong,et al.WSN localization algorithm integrating ranging correction and spider wasp optimization[J].Advanced Engineering Sciences,2025,57(5):333–343.[余修武,肖林,刘永,等.融合测距修正和蜘蛛蜂优化的WSN定位算法[J].工程科学与技术,2025,57(5):333–343.]