

基于坐标优化的 FOA-Amorphous 节点定位算法

彭 铎,张 倩,张腾飞,陈江旭

(兰州理工大学 计算机与通信学院,甘肃 兰州 730050)

摘 要:节点的位置信息在无线传感器网络的定位中起着至关重要的作用,而 Amorphous 算法的节点定位精度低。针对影响 Amorphous 定位精度的主要原因分析,提出了一种基于坐标优化的 FOA-Amorphous 节点定位算法。首先,引入多通信半径的概念细化节点跳数,利用网络平均连通度对节点的平均跳距进行重算;然后,以极大似然估计法得到的未知节点坐标值为果蝇优化算法中各果蝇的初始位置,通过此初始位置产生每个果蝇的初始种群,代入适应度函数求得当前果蝇的最佳位置,引入了个体认知因子 c_1 和群体引导因子 c_2 ,优化了果蝇随机搜索的距离和方向,使得算法快速收敛到全局最优,避免算法早熟,提高了算法的收敛精度,通过迭代找到最佳未知节点位置坐标。与双通信半径算法、PSO-IDV-Hop 算法以及 Amorphous 算法相比,该算法的归一化定位误差分别降低了约 7%、23% 和 44%。

关键词:Amorphous 算法;坐标优化;多通信半径;果蝇优化算法;认知因子;引导因子

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2023)07-0091-07

doi:10.3969/j.issn.1673-629X.2023.07.014

FOA-Amorphous Node Localization Algorithm Based on Coordinate Optimization

PENG Duo,ZHANG Qian,ZHANG Teng-fei,CHEN Jiang-xu

(School of Computer and Communication,Lanzhou University of Technology,Lanzhou 730050,China)

Abstract:The location information of nodes plays a crucial role in the location of wireless sensor networks, and the node location accuracy of Amorphous algorithm is low. Aiming at the main reasons that affect the accuracy of Amorphous location, a FOA-Amorphous node location algorithm based on coordinate optimization is proposed. Firstly, the concept of multi-communication radius is introduced to refine the number of node hops, the average network connectivity is used to recalculate the average distance per hop of the node. Then, the coordinate value of the unknown node obtained by the maximum likelihood estimation method is used as the initial position of each fruit fly in the fruit fly optimization algorithm, the initial population of each fruit fly is generated through this initial position, and the optimal position of the current fruit fly is obtained by substituting the fitness function. Individual cognitive factor c_1 and group guidance factor c_2 are introduced to optimize the distance and direction of the random search of the fruit fly, so that the proposed algorithm quickly converges to the global optimum, avoids its premature maturity, improves its convergence accuracy, and finds the optimal unknown node position coordinates through iteration. Compared with the dual communication radius algorithm, the PSO-IDV-Hop algorithm and the Amorphous algorithm, the normalized positioning errors are reduced by about 7%, 23% and 44%, respectively.

Key words:Amorphous algorithm; coordinate optimization; multi-communication radius; fruit fly optimization algorithm; cognitive factor; guidance factor

0 引 言

无线传感器网络^[1-2] (Wireless Sensor Networks, WSN)是由数量众多的无线传感器节点构成,这些节点以自组织的方式相互协调工作,传感器节点之间可以传输信息并协作感知环境。无线传感器网络在军事侦察、环境监测、抢险救灾、医疗保障等许多领域都有

着广泛的应用^[3]。在这些应用中,获取的数据必须要有相应的位置信息,如果没有位置信息,相互传递的数据就毫无价值,失去了采集数据的意义,因此节点定位也是 WSN 的关键支撑技术之一^[4-5]。近些年来,诸多学者对距离无关的定位算法展开研究,希望提高定位精度,使得未知节点在定位过程中能获得更为准确的

收稿日期:2022-08-23

修回日期:2022-12-27

基金项目:国家自然科学基金资助项目(61663024,62061024);甘肃省高校创新基金项目(2020A-021)

作者简介:彭 铎(1976-),男,副教授,硕导,博士,通信作者,研究方向为无线传感器网络、光纤网络、无线通信理论与技术;张 倩(1999-),女,硕士研究生,CCF 会员(K6068G),研究方向为无线传感器网络。

位置信息^[6]。

Amorphous 算法定位精度不高,诸多学者对其进行了数学优化,或者加入改进后的群体智能算法力求有效降低未知节点的定位误差。文献[7]提出了一种新的算法来优化 Amorphous 定位算法。首先,通过传统算法得到未知节点位置的初始解;然后,利用遗传禁忌搜索算法优化初始解,从而得到未知节点的最优位置。文献[8]分析 Amorphous 定位算法的缺点,对节点间跳数进行修正,引用质心算法加权,提出改进的算法模型,仿真结果表明该算法可获得较为精确的定位结果。文献[9]提出了基于 KNN 算法的指纹室内定位技术,解决了基于距离相关的 RSSI 定位算法中定位节点遗漏问题。文献[10]提出基于 RSS 阈值模型的 Amorphous 算法,在 Amorphous 算法离线计算网络平均连通度的基础上,建立了四种 RSS 阈值模型来抑制 Amorphous 算法在不同通信模型下的定位误差。文献[11]引用了双通信半径对节点之间最小跳数进行细化,修正平均跳距则引入了最小均方差准则以及归一化加权因子,最后,采用优化的麻雀搜索算法估算未知节点的位置。

文献[12]通过增加通信半径,将距离信标节点较近的未知节点获得的最小跳数更新为较小的跳数,并保持距离较远的未知节点的最小跳数信息。该方法以跳数的形式反映了实际距离之间的差距,在一定程度上解决了跳数相同但实际距离差异较大的问题。因此,可以估计更准确的平均跳跃距离,计算出它们之间更精确的距离,得到未知节点更精确的坐标。文献[13]提出了一种基于粒子群优化(PSO)的 DV-Hop 定位算法。在所提出的算法中,锚节点的跳数使用校正因子进行修改,通过更新校正因子,提高了节点之间距离测量的准确性。为了进一步提高准确性,使用 PSO 算法,因为 WSN 中的定位是一个优化问题。在有界种群可行域的帮助下,使用 PSO 对未知节点的定位更加准确,收敛速度相对较高。仿真结果表明,该算法的定位误差显著降低。

上述研究通过不同的方法对传统算法的不足之处进行了不同程度的改进,但改进算法仍然存在定位误差较大或算法复杂度高等问题。针对这些不足,该文提出了一种基于坐标优化的 FOA-Amorphous 节点定位算法。该算法在计算跳数值时用多通信半径细化跳数,利用网络平均连通度对节点的平均跳距进行重算,果蝇优化算法将极大似然估计法计算得到的未知节点坐标作为每个果蝇的初始位置,通过此初始位置产生每个果蝇的初始种群,代入适应度函数对解进行更新迭代,直到找到具有最佳适应度的解,将此解作为未知节点的坐标。

1 相关理论

1.1 Amorphous 算法介绍与定位误差分析

Amorphous 算法^[14]是基于非测距的定位算法,该算法通过以下三步实现对未知节点的定位:

步骤 1:计算未知节点距各锚节点的最小跳数。

每个锚节点将含有其位置坐标信息的数据包广播到全网,锚节点的初始跳数为 0,数据包每经过一个节点,跳数加 1,当一个节点收到多个跳数信息时,保存跳数值最小的一条信息。

步骤 2:计算未知节点到各锚节点的距离。

在第一阶段,每个锚节点记录了到其他锚节点的最小跳数,通过公式(1)计算它和锚节点之间的距离 $d_{i,j}$:

$$d_{i,j} = R \times \text{hop}_{i,j} \quad (1)$$

其中, $d_{i,j}$ 表示锚节点 i 和未知节点 j 之间的距离, R 为节点的通信半径, $\text{hop}_{i,j}$ 表示锚节点 i 和未知节点 j 之间的最小跳数。

步骤 3:利用极大似然估计法计算自身位置。

通过前两个步骤的计算可以得到未知节点到锚节点之间的距离,因此,可以得到以下方程组:

$$\begin{cases} \sqrt{(x-x_1)^2 + (y-y_1)^2} = d_1 \\ \sqrt{(x-x_2)^2 + (y-y_2)^2} = d_2 \\ \dots \\ \sqrt{(x-x_i)^2 + (y-y_i)^2} = d_i \\ \dots \\ \sqrt{(x-x_M)^2 + (y-y_M)^2} = d_M \end{cases} \quad (2)$$

其中, (x,y) 为未知节点的坐标, (x_i,y_i) 为信标节点的坐标, d_i 为估计距离。通过解上式方程组可以得到未知节点的坐标。

Amorphous 算法与 DV-Hop 算法在某些方面具有相似性,均是通过计算出的最小跳数乘以平均跳距来估计未知节点至锚节点的距离。不同之处是,Amorphous 算法将通信半径作为锚节点的平均跳距来估算节点间的距离,而 DV-Hop 算法中锚节点的平均跳距是由两个锚节点间的距离除以跳数得到。

下面将分析 Amorphous 算法定位误差产生的主要原因:

①算法在计算节点间的最小跳数时,结果都是整数,但是现有的实验证明这大约增加了 0.5 个平均跳数的误差,导致算法的定位误差增大。

②由于网络节点分布的不规则性,两节点之间的距离或大或小,以节点通信半径来估算两节点间的距离会导致定位误差过大。

③利用极大似然估计法求解未知节点坐标时,受

初始值测量误差影响较大,小的测量误差就会导致较大的位置估计误差。

1.2 果蝇优化算法

果蝇优化算法 (Fruit Fly Optimization Algorithm, FOA) 是由潘文超博士^[15]提出的一种演化式群智能优化算法,其基本思想来自于果蝇在自然界中的觅食行为。相较于其他物种,果蝇在嗅觉和视觉上更为敏感。因此,在觅食过程中,果蝇可以利用自身独特的嗅觉发现食物气味,并分享给周围的果蝇,比较后可以得到气味最优的果蝇个体位置,同时其他果蝇均会向该位置飞去,通过不断地迭代更新最后得到果蝇群体中的最佳位置。

由文献[15]可知果蝇优化算法是一种更简单、更鲁棒的优化算法;该算法不仅具有易于理解的特点,而且易于写入程序代码。同时,与其他算法相比,程序代码不会太长,很容易用于处理各种优化问题。其算法步骤如下:

(1) 给定果蝇种群规模 (Sizepop)、群体最大迭代次数 (Maxgen), 随机初始化果蝇个体初始位置 X_{axis} 、 Y_{axis} 。

(2) 根据式(3)确定每个果蝇个体寻找食物的随机距离和方向。

$$\begin{cases} X_i = X_{axis} + RValue \\ Y_i = Y_{axis} + RValue \end{cases} \quad (3)$$

式中, RValue 为果蝇随机搜索距离和方向的随机数。

(3) 计算果蝇个体与原点距离 D_i 和果蝇个体的味道浓度判定值 S_i , S_i 为距离 D_i 的倒数。

$$D_i = \sqrt{X_i^2 + Y_i^2} \quad (4)$$

$$S_i = 1/D_i \quad (5)$$

(4) 将味道浓度判定值 S_i 代入适应度函数, 求解出该种群中果蝇个体的味道浓度 $Smell(i)$ 。

$$Smell(i) = \text{Function}(S_i) \quad (6)$$

(5) 找出种群当前味道浓度最优个体 (即味道浓度最小值), 记录其坐标和味道浓度值。

$$[bestSmell, bestIndex] = \min(Smell) \quad (7)$$

(6) 记录最优味道浓度值及对应的横纵坐标, 群体中的其他果蝇均利用视觉向该位置飞去, 形成新的果蝇群体。

$$X_{axis} = X(bestIndex) \quad (8)$$

$$Y_{axis} = Y(bestIndex) \quad (9)$$

$$bestSmell = Smellbest \quad (10)$$

(7) 迭代寻优阶段, 重复执行步骤(2)~步骤(5), 判断当前迭代次数是否小于最大迭代次数, 当前味道浓度是否优于所记录的最优味道浓度, 若成立, 执行步骤(6)。

2 基于多通信半径和 FOA-c 的节点定位算法

2.1 自适应权重因子优化果蝇算法 (FOA-c)

FOA 迭代寻优时, 由式(3)可知, 每只果蝇个体移动的距离是固定范围的随机值, 在接近最优果蝇时, 果蝇的搜索范围具有局限性, 易陷入局部最优, 收敛精度低, 出现早熟收敛。该文借鉴文献[16]中粒子群优化算法的认知因子和引导因子, 在算法迭代过程中引入了个体认知因子 c_1 和群体引导因子 c_2 , 其中, 个体认知因子控制着果蝇个体对自己和其他果蝇个体位置的认知, 群体引导因子控制着整个果蝇群体目前的最优位置对果蝇个体的引导。这将有助于算法迅速收敛于全局最优, 避免算法早熟, 提高了算法的收敛精度。

$$\begin{cases} c_1 = \exp(\text{rand}) - 1 \\ c_2 = 1 - c_1 \end{cases} \quad (11)$$

迭代寻优阶段, 将式(3)可更换为:

$$\begin{cases} X_i = X_{axis} + c_1 \cdot (\text{mean}(X) - X(i, :)) + \\ \quad c_2 \cdot (X_{axis} - X(i, :)) \\ Y_i = Y_{axis} + c_1 \cdot (\text{mean}(Y) - Y(i, :)) + \\ \quad c_2 \cdot (Y_{axis} - Y(i, :)) \end{cases} \quad (12)$$

其中, $\text{mean}(X)$ 和 $\text{mean}(Y)$ 表示上次迭代中所有果蝇个体的平均值, $(X(i, :), Y(i, :))$ 表示上次迭代中果蝇的位置。

在相同实验条件下, 选取了与式(18)维度一致的基测函数对 FOA-c 算法和 FOA 算法进行了寻优性能的对比, 如图 1 所示。从图中不难看出, FOA-c 算法相比较 FOA 算法, 收敛速度更快, 寻优精度更高。

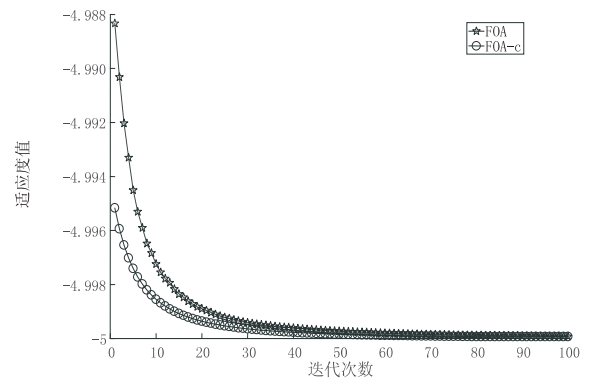


图 1 算法收敛曲线

2.2 FOA-Amorphous 算法

FOA-Amorphous 算法首先通过多通信半径对最小跳数进行细化; 接着采用网络平均连通度对平均每跳距离进行重算; 最后将极大似然估计法计算到的未知节点坐标作为每个果蝇的初始位置, 利用 FOA-c 算法求得最佳未知节点的位置坐标。

2.2.1 多通信半径细化跳数

如图 2 所示, 假设在锚节点 A 的通信半径范围内

有 B 、 C 、 D 三个未知节点,由 Amorphous 算法可知,锚节点 A 到 B 、 C 、 D 之间的跳数均为 1 跳,若用 Amorphous 算法的实现步骤来计算节点间的距离,则 $AB=AC=AD$,但从图中来看,未知节点 B 、 C 、 D 到锚节点 A 的实际距离却不相同。

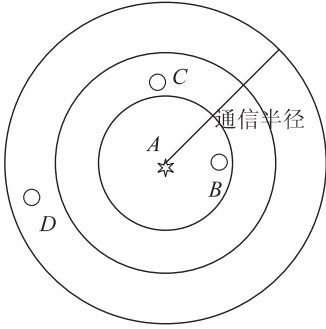


图 2 通信半径内节点分布

因此,该文利用多通信半径来细化跳数:

$$h = \begin{cases} \frac{1}{n} & 0 < d \leq \frac{R}{n} \\ \frac{i}{n} & \frac{(i-1)R}{n} < d \leq \frac{iR}{n} \\ \vdots & \\ 1 & \frac{(n-1)R}{n} < d \leq R \end{cases} \quad (13)$$

其中, R 为节点的通信半径, h 为跳数; d 为锚节点与其邻居节点的实际距离。

各锚节点向整个网络以洪泛的方式广播包含自身位置坐标信息的数据包,锚节点的初始跳数为 0,数据包每经过一个节点,跳数加 $\frac{1}{n}$,当一个节点收到多个跳数信息时,保存跳数值最小的一条信息。经验证,将通信半径细化为 4 跳,对该算法最有益,此时,跳数 h 为:

$$h = \begin{cases} 0.25 & 0 < d \leq \frac{R}{4} \\ 0.50 & \frac{R}{4} < d \leq \frac{R}{2} \\ 0.75 & \frac{R}{2} < d \leq \frac{3R}{4} \\ 1.00 & \frac{3R}{4} < d \leq R \end{cases} \quad (14)$$

2.2.2 重新计算平均每跳距离

由于网络节点分布的随机性,两节点之间的跳段距离通常不是两者的直线距离,用通信半径乘以最小跳数过大地估计了跳段距离。

FOA-Amorphous 算法先用式 (15) 计算网络平均连通度:

$$n_{\text{local}} = N\pi R^2/S \quad (15)$$

式中, N 为网络节点总数, R 为节点的通信半径, S 为

网络区域面积。

用式 (16) 计算平均每跳距离:

$$\text{HopSise} = R(1 + e^{-n_{\text{local}}} - \int_{-1}^1 e^{-\frac{n_{\text{local}}}{\pi}(\arccos t - t\sqrt{1-t^2})} dt) \quad (16)$$

2.2.3 定位计算优化

由于节点间的距离都是通过跳数乘以平均跳距估计出来的,由 Amorphous 算法的误差分析得知估计距离与节点间的实际距离有较大误差。假设未知节点 (x, y) 与锚节点 (x_i, y_i) 之间的距离为 d_i ,估计误差为 τ_i ,则有以下约束条件:

$$\begin{cases} d_1 - \tau_1 \leq \sqrt{(x - x_1)^2 + (y - y_1)^2} \leq d_1 + \tau_1 \\ d_2 - \tau_2 \leq \sqrt{(x - x_2)^2 + (y - y_2)^2} \leq d_2 + \tau_2 \\ \dots \\ d_n - \tau_n \leq \sqrt{(x - x_n)^2 + (y - y_n)^2} \leq d_n + \tau_n \end{cases} \quad (17)$$

越小的 τ_i 将使得未知节点的估计坐标与实际值更加接近。因此,将节点定位问题转化为求最优值问题,则有以下适应度函数:

$$F = \sum_{i=1}^n |\sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i| \quad (18)$$

通过 FOA-c 算法迭代寻优找到使得函数值 F (即估计误差 τ_i) 趋近于 0 的适应度值的坐标,此坐标更接近未知节点的实际坐标,提高了算法的定位精度。

算法将极大似然估计法计算到的未知节点位置坐标作为每个果蝇的初始位置,利用式 (3) 产生每个未知节点的初始种群,代入适应度函数得到初始适应度值,再用式 (12) 产生的新种群进行更新迭代,直到找到具有最佳适应度的解,将此解作为未知节点的位置。

算法的具体步骤如下:

Step1: 利用多通信半径的方法细化跳数;

Step2: 利用公式 (16) 重新计算节点的平均每跳距离;

Step3: 由平均跳距乘以细化跳数得到未知节点与锚节点之间的跳距;

Step4: 利用极大似然估计法计算每个未知节点的坐标,将此坐标作为每个果蝇的初始位置;

Step5: 利用式 (3) 产生每个果蝇 (即每个未知节点) 的初始种群,代入适应度函数得到个体的初始适应度值;

Step6: 利用式 (12) 产生新的种群代入适应度函数,通过迭代找到最佳适应度的解,将输出的最佳解作为未知节点的坐标;

Step7: 循环多次步骤 5 和步骤 6,直到找到所有的最佳未知节点的坐标为止。

3 算法仿真及结果分析

3.1 算法参数设置

为了验证算法的定位性能,利用 MATLAB2016b 对文中算法(FOA-Amorphous)、Amorphous 算法、文献[12]提出的双通信半径算法以及文献[13]提出的 PSO-IDV-Hop 算法,从锚节点个数、通信半径大小、总节点个数以及时间复杂度等四个方面进行仿真实验分析。在 $100\text{ m} \times 100\text{ m}$ 的仿真区域内,随机产生一定数量的网络节点。

算法的参数设置如表 1 所示。

表 1 算法相关参数设置

| 参数 | 值 |
|--------|-----|
| 运行次数 | 100 |
| 种群规模 | 80 |
| 最大迭代次数 | 300 |

算法中每个未知节点的定位误差计算公式如下:

$$E_{\text{err}} = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (19)$$

算法的归一化定位误差计算公式如下:

$$E_{\text{avg}} = \frac{1}{N} \sum_{i=1}^n \frac{\sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}}{R} \quad (20)$$

式中, (x_i, y_i) 与 (\hat{x}_i, \hat{y}_i) 为未知节点 i 的实际坐标和估计坐标, R 为锚节点的通信半径, N 为可定位的未知节点总数。

3.2 定位误差仿真实验分析

设置总节点个数为 100,锚节点个数为 20,通信半径为 30。该文提出的算法与传统算法的定位效果如图 3 所示,从图中不难看出,该文提出的算法估计的节点位置与传统算法相比更加靠近真实位置。

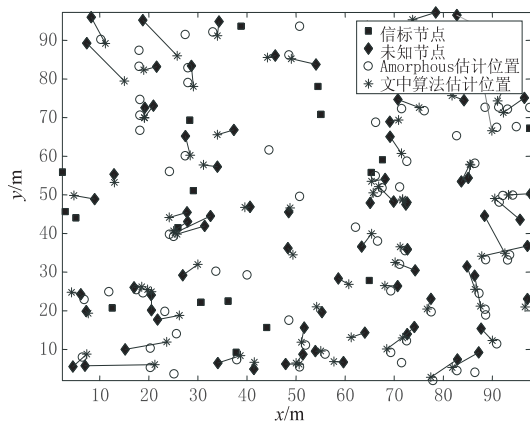


图 3 算法的定位效果

3.2.1 未知节点平均定位误差比较

在监测区域内随机部署 100 个节点,其中锚节点个数为 20,未知节点个数为 80 个,通信半径为 30,四种算法各运行 100 次后的平均定位误差如图 4 所示,PSO-IDV-Hop 算法和文献[12]的算法的平均定位误差大约分别为 13.4 m 和 6.5 m,该文提出算法的未知

节点的平均定位误差大约为 4.3 m,与其他三种算法相比,定位误差更低。

3.2.2 锚节点对定位误差的影响

图 5 是节点总数为 100,通信半径为 30 时,锚节点比例变化对未知节点定位误差的影响。

由图 5 可知,四种算法定位误差均随锚节点所占比例的增大而减小,是因为在总节点个数不变的情况下,锚节点的个数增加,则其密度增大,一个未知节点可以获得的锚节点的位置信息增多,使其定位误差降低。在相同的条件下,文中算法的误差一直都是最小的,与 PSO-IDV-Hop 算法相比,其归一化定位误差降低了 30% 左右,与文献[12]改进的算法相比,其归一化定位误差降低 7% 左右,与 Amorphous 算法相比,归一化误差降低了 53% 左右。从整体来看,无论锚节点的比例是多或少,该文提出的算法都具有更好的定位效果。

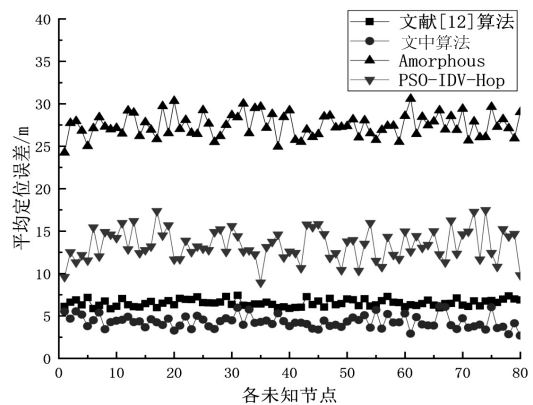


图 4 四种算法每个未知节点的定位误差对比

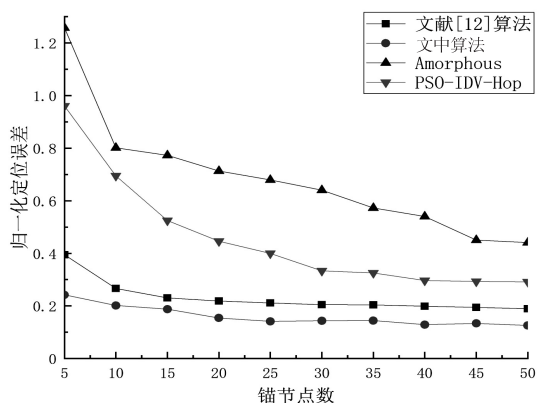


图 5 锚节点个数对定位误差的影响

3.2.3 通信半径对定位误差的影响

图 6 是在总节点个数为 100,锚节点个数为 30 的条件下,通信半径从 25 变化到 50 的误差变化图。

由图 6 可见,文中算法与文献[12]中算法的定位误差均随通信半径的增大而稳定减小,而 PSO-IDV-Hop 算法在通信半径大于 45 之后定位误差略有增加,这是由于随着通信半径增大,一跳范围内的节点数增

多,导致算法的跳数误差增大,而文中算法和文献[12]的算法对跳数进行了细化,使误差随着通信半径的增加平稳减小。与其他三种算法相比,文中算法的归一化定位误差始终最低,相比较文献[12]的算法、PSO-IDV-Hop 算法和 Amorphous 算法归一化定位误差分别降低了 8%、24% 和 44% 左右。

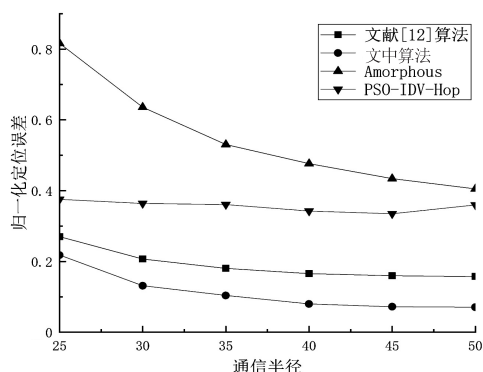


图6 通信半径对定位误差的影响

3.2.4 节点总数对定位误差的影响

图7是在锚节点比例为30%,通信半径为30的条件下,定位误差随节点总数的变化图。

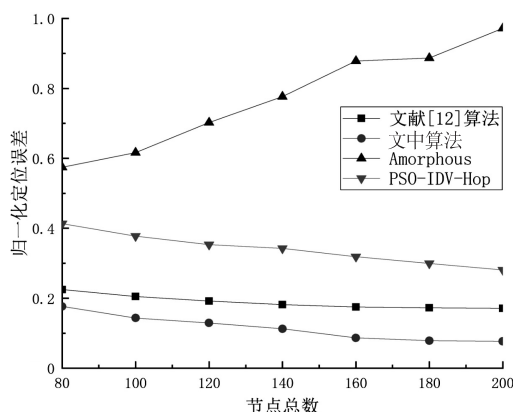


图7 节点总数对定位误差的影响

从图中可以看出,随着节点总数的增加,Amorphous 算法的定位误差呈现上升的趋势,这是因为当仿真区域面积不变的情况下,节点个数增加,节点密度增大,则节点的平均邻居节点数增多,用通信半径代替平均跳距计算距离时本来就有误差产生,当邻居节点增多,这种误差将会累积增大。然而,文中算法与 PSO-IDV-Hop 算法以及文献[12]算法的定位误差稳定下降,是因为算法对跳数进行了优化,且文中算法和 PSO-IDV-Hop 算法加入了智能优化算法,优化了未知节点的坐标。与其他三种算法相比,文中算法的归一化定位误差最低,低于文献[12]算法 7% 左右,低于 PSO-IDV-Hop 算法和 Amorphous 算法分别为 23% 和 66% 左右。

3.3 时间复杂度与计算开销分析

时间复杂度反映了算法执行的时间长短。假设未

知节点个数为 n , 锚节点的个数为 m ($m < n$), 智能优化算法的最大迭代次数为 k 。则 Amorphous 算法的时间复杂度为 $O(n^3)$; 文献[12]算法在传统算法的基础上增加的时间复杂度为 $O(nm)$; PSO-IDV-Hop 算法在修正平均跳距阶段时间复杂度为 $O(n)$, 算法第三阶段加入粒子群算法增加的时间复杂度为 $O(kn^3)$; 文中算法细化跳数阶段的时间复杂度为 $O(4nm)$, 在求估计位置坐标阶段时间复杂度为 $O(kn^2m)$ 。可以看出,文中算法的时间复杂度低于 PSO-IDV-Hop 算法,但高于 Amorphous 算法和文献[12]算法。

将算法的平均运行时间作为算法开销指标进行分析。图8为相同实验条件下所统计的四种算法的运行时间比较。

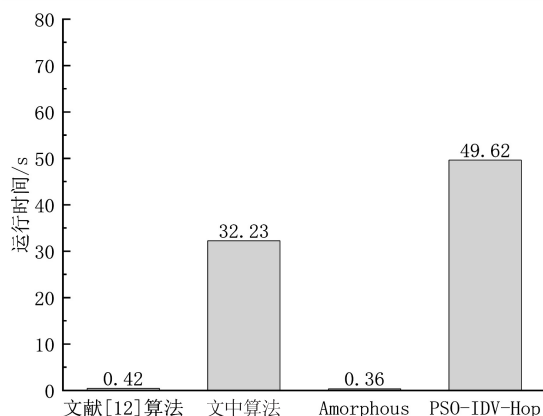


图8 四种算法的平均运行时间

由实验结果可知,文中算法的运行时间比 Amorphous 算法和文献[12]算法的运行时间大,是因为文中算法加入了智能优化算法,使得算法的运行时间增加;但比 PSO-IDV-Hop 算法的运行时间小,这是因为 FOA 算法比 PSO 算法更简单、更鲁棒^[10],且文中算法加入了个体认知因子和群体引导因子,使算法快速收敛到全局最优,提高了算法的收敛精度。从整体来说,文中算法的运行时间略有增加,但算法的定位精度得到了很好的提升。

4 结束语

主要针对 Amorphous 算法存在的误差分析,提出了一种新的 FOA-Amorphous 算法。该算法通过对通信半径划分实现了跳数的细化,利用网络平均连通度对锚节点的平均跳距进行重算,然后用极大似然估计法计算得到的未知节点坐标值作为果蝇算法中每个果蝇的初始位置,通过此初始位置产生每个果蝇的初始种群,代入适应度函数对解进行更新迭代,直到找到具有最佳适应度的值,将此解作为未知节点的最终位置坐标。仿真实验表明,虽然算法的运行时间略有增加,但从整体来看,无论锚节点、通信半径、总节点数如

何变化,该算法都具有更好的定位性能。未来主要的研究热点可能是移动节点、不规则拓扑网络和三维地形中的研究。

参考文献:

- [1] 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.
- [2] 田 昕, 魏国亮, 王甘楠. 无线传感器网络定位综述[J]. 信息与控制, 2022, 51(1): 69-87.
- [3] GE C, LI L, YU Y, et al. Improvement of DV-Hop algorithm based on error analysis[C]//2022 IEEE 6th information technology and mechatronics engineering conference (ITOEC). Chongqing: IEEE, 2022: 1299-1303.
- [4] 李跃飞. 基于锚节点分布特性的跳距修正的 DV-HOP 定位算法[J]. 传感技术学报, 2021, 34(10): 1379-1384.
- [5] 陈业海, 郑日荣, 许 亮. 基于进化算法的多通信半径 DV-Hop 改进定位算法[J]. 计算机工程与应用, 2019, 55(1): 89-95.
- [6] 张 楠, 史建华, 亿 吉, 等. 基于 WSNs 加权质心定位算法的井下移动目标实时跟踪方法[J/OL]. 吉林大学学报(工学版): 1-7[2022-10-25]. DOI: 10.13229/j.cnki.jdx-bgxb20220137.
- [7] 胡 伟, 袁三男. 遗传-禁忌搜索优化的 Amorphous 定位算法[J]. 传感技术学报, 2019, 32(6): 940-944.
- [8] 安文秀, 赵菊敏, 李灯熬. 基于 Amorphous 的无线传感器网络定位算法研究[J]. 传感器与微系统, 2013, 32(2): 33-35.
- [9] 张银蒲. 基于 KNN 算法的指纹室内定位技术研究[J]. 电子制作, 2019(22): 13-15.
- [10] 宋海声, 朱长驹, 吴佳欣, 等. 基于 RSS 阈值模型的 Amorphous 算法定位误差抑制[J]. 计算机工程与科学, 2018, 40(5): 821-828.
- [11] 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.
- [12] LI T, WANG C, NA Q. Research on DV-hop improved algorithm based on dual communication radius [J]. EURASIP Journal on Wireless Communications and Networking, 2020, 2020(1): 1-10.
- [13] SHARMA G, RAJESH A. PSO-based improved DV-hop localization algorithm in wireless sensor networks[C]//International conference on intelligent computing and smart communication 2019. Singapore: Springer, 2020: 95-104.
- [14] NEMER I, SHELTAMI T, SHAKSHUKI E, et al. Performance evaluation of range-free localization algorithms for wireless sensor networks [J]. Personal and Ubiquitous Computing, 2021, 25(1): 177-203.
- [15] PAN W T. A new fruit fly optimization algorithm; taking the financial distress model as an example [J]. Knowledge-Based Systems, 2012, 26: 69-74.
- [16] 张水平, 王丽娜. 果蝇优化算法的进展研究分析[J]. 计算机工程与应用, 2021, 57(6): 22-29.
- [17] 詹京吴, 黄宜庆. 融合安全 A* 算法与动态窗口法的机器人路径规划[J]. 计算机工程, 2022, 48(9): 105-112.
- [18] 路宜松. 基于 Unity 引擎的 2D 角色扮演游戏的设计与实现[D]. 沈阳: 沈阳理工大学, 2021.
- [19] 黄 鹏. 基于 Unity3D 的分布式协同虚拟装配仿真研究及实现[D]. 湘潭: 湘潭大学, 2017.
- [20] DU X, ARYA A. Design and evaluation of a learning assistant system with optical head-mounted display[C]//Learning and collaboration technologies: second international conference 2015. Los Angeles: [s. n.], 2015: 75-86.

(上接第 79 页)

- [8] 张凤军, 戴国忠, 彭晓兰. 虚拟现实的人机交互综述[J]. 中国科学: 信息科学, 2016, 46(12): 1711-1736.
- [9] 姜 薇. 用户界面设计研究[D]. 杭州: 浙江大学, 2006.
- [10] 王 宁, 余隋怀, 周 宪, 等. 人机交互界面中形状特征的视觉显著度计算[J]. 图学学报, 2016, 37(4): 514-518.
- [11] 张 咪. 虚拟仿真实验环境中学习评价指标体系研究[D]. 无锡: 江南大学, 2020.
- [12] 耿官鑫, 张豫鼎, 张晓媛. 摄像技术虚拟实验教学系统的建设[J]. 软件, 2021, 42(4): 12-15.