

基于 DV-Hop 测距修正的动态调参差分进化定位算法

李强, 黄霞*, 申东, 邸敬

兰州交通大学电子与信息工程学院, 甘肃 兰州 730070

摘要 针对距离向量-跳距(DV-Hop)定位算法定位误差大的问题,提出一种基于 DV-Hop 测距修正的动态调参差分进化算法(IDV-Hop-DPDEL)。利用多通信半径广播锚节点位置信息,结合节点远近度修正待定位节点与锚节点的测距值;通过动态调整差分进化算法中的变异因子和交叉概率因子,提高搜索精度,得到待定位节点的最优位置。为了验证算法的有效性,在保证其他参数都固定的情况下,令节点总数、通信半径及锚节点占比在一定范围内变化,实验结果显示,IDV-Hop-DPDEL 算法的平均定位误差比 DV-Hop 算法低 51.56%,且与其他两种经典改进 DV-Hop 算法相比,IDV-Hop-DPDEL 算法的性能较优。

关键词 传感器; DV-Hop 算法; 多通信半径; 远近度; 动态调参差分进化

中图分类号 TP393

文献标志码 A

doi: 10.3788/LOP202158.0628001

Improved DV-Hop Ranging-Based Dynamic Parameters Differential Evolution Localization Algorithm

Li Qiang, Huang Xia*, Shen Dong, Di Jing

School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, Gansu 730070, China

Abstract For the problem of larger location error in distance vector-hop (DV-Hop) localization algorithm, an improved DV-Hop ranging-based dynamic parameters differential evolution localization (IDV-Hop-DPDE) algorithm is proposed in this paper. IDV-Hop-DPDEL uses multi-communication radius to broadcast the positions of anchor nodes. Based on near-degree of the nodes, the ranging between unknown nodes and anchor nodes is modified. Finally, to improve searching efficiency and obtain the most optimal position, the scaling factor and the crossover probability of the traditional differential evolution algorithm are dynamically changed. To verify the effectiveness of the algorithm, the localization parameters in the simulation remains unchanged, while the three parameters vary within a certain range, i.e., the total number of nodes in the network, the proportion of the anchor nodes, and the communication radius. The simulation results show that the localization error of the IDV-Hop-DPDEL algorithm was reduced by 51.56%, compared to the DV-Hop algorithm. And the proposed algorithm has better performance than the other two classical improved DV-Hop algorithms.

Key words sensors; DV-Hop algorithm; multi-communication radius; near-degree; dynamic parameters differential evolution

OCIS codes 280.4788; 150.5670; 150.1135

收稿日期: 2020-06-24; 修回日期: 2020-08-04; 录用日期: 2020-08-27

基金项目: 国家自然科学基金(61741113)、甘肃省科技计划(17JR5RA097)、甘肃省高等学校创新能力提升项目(2019B-052)

* E-mail: 19893176679@163.com

1 引言

随着物联网技术的发展,无线传感器网络(WSNs)得到了广泛的应用,如煤矿监测、病人追踪与定位,以及港口集装箱定位等^[1]。通常在 WSNs 监测区域部署大量传感器节点来测量周围环境的物理参数,并通过简单的计算和存储将监测到的信息发送到周边基站,该过程中节点感测到的数据必须与位置信息相结合,其中不含位置信息的数据往往是无用的。定位算法根据是否需要实测可分为基于测距和非测距两类定位算法^[2]。基于测距定位算法的定位精度高,但存在能耗大、成本高^[3]的缺点。非测距定位算法虽定位精度低,但由于其成本低,算法易实现,在大规模的 WSNs 中得到了广泛使用^[4],其中 DV-Hop 算法^[5]作为非测距算法的代表之一,以其简单易用、节能高效等优点,引起了国内外研究者的广泛关注,但仍存在节点定位误差大、节点定位精度低等缺点,因此许多改进的 DV-Hop 算法相继被提出。

文献[6]通过计算节点位置偏差对平均跳距进行重估,将实际和理论跳数比加权最小跳数,降低了定位误差。文献[7]基于位置偏差与距离权值修正平均跳距,将与实际平均跳距的算术平均值作为未知节点平均跳距,最后采用改进的遗传算法优化节点坐标。文献[8]用分数对跳数计数,并根据 WSNs 中锚节点平均跳距跳数加权修正平均跳段距离,最后采用改进的布谷鸟搜索算法优化未知节点坐标。文献[7]与文献[8]虽均有效提高了定位精度,但也增加了算法复杂度。文献[9]利用相近度修正测距值,用最小二乘法估计的未知节点坐标作为遗传模拟退火算法优化初始解;文献[10]提出基于人工蜂群的定位算法;两者都因算法参数多而陷入局部最优。文献[11]提出 HDCDV-Hop 定位算法,该算法利用无线电不规则模型(RIM)根据接收信号强度值(RSS)修正测距值,并利用改进差分进化算法对未知节点坐标进行优化,从而获得最优位置估计,有效提高了定位精度,但如果搜索后期还未寻到满意的值,则算法容易早熟收敛。本文在以上的研究基础上提出基于 DV-Hop 测距修正的动态调参差分进化定位(IDV-Hop-DPDEL)算法,锚节点利用三种通信半径广播消息,利用节点远近度变量比值与改进的广播半径相乘求得修正因子,修正待定位节点的测距值,最后采用动态调参差分进化算法搜寻待

定位节点最优位置,从而提高定位精度。

2 DV-Hop 定位算法及误差分析

DV-Hop 算法包含以下 3 个步骤^[12]。

步骤 1:锚节点广播含有自身位置及初始值为 0 跳的信标列表,逐跳转发,跳数依次递增,节点仅保留最小跳数。

步骤 2:该步计算跳段距离,各锚节点平均跳距 $D_{\text{HopSize},i}$ 为

$$D_{\text{HopSize},i} = \frac{\sum_{j=1, i \neq j}^{N-1} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j} N_{\text{hop},ij}}, \quad (1)$$

式中: N 为锚节点总个数; (x_i, y_i) 和 (x_j, y_j) 为锚节点 i 和 j 的坐标; $N_{\text{hop},ij}$ 为 i, j 间最小跳数。锚节点广播自身平均跳距,广播结束,未知节点 u 以距离最近的锚节点平均跳距作为自身平均跳距计算跳距 d_{ui} ,即

$$d_{ui} = D_{\text{HopSize},i} \times N_{\text{hop},ui} \quad (2)$$

步骤 3:利用两点间距离公式列出方程为

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

式中: (x, y) 表示未知节点坐标; $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ 表示锚节点坐标; d_n 表示未知节点与第 n 个锚节点的跳距估计值。将(3)式改写为矩阵 $\mathbf{AX} = \mathbf{B}$ 的形式,则有

$$\mathbf{A} = 2 \times \begin{bmatrix} x_1 - x_n & y_1 - y_n \\ x_2 - x_n & y_2 - y_n \\ \vdots & \vdots \\ x_{n-1} - x_n & y_{n-1} - y_n \end{bmatrix}, \quad (4)$$

$$\mathbf{B} = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_1^2 - d_n^2 \\ x_2^2 - x_n^2 + y_2^2 - y_n^2 + d_2^2 - d_n^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_{n-1}^2 - d_n^2 \end{bmatrix} \quad (5)$$

由最小二乘法解得未知节点位置坐标 (x, y) 为

$$\mathbf{X} = \begin{bmatrix} x \\ y \end{bmatrix} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (6)$$

在 WSNs 实际应用中,节点随机非均匀部署在监测区域,因此在锚节点通信半径内,未知节点

与锚节点的距离并不相等,然而按照经典 DV-Hop 算法,二者的距离是相等的。如图 1 所示, i, j 和 k 代表锚节点, 数字 1~8 代表未知节点。未知节点 3、4 与锚节点 j 的实际距离并不相等,但均位于锚节点 j 的通信半径 R 之内,因此未知节点 3、4 与锚节点 j 的跳距为 R 。经典 DV-Hop 算法在计算跳距时,用两点间的折线代替直线,中间跳数越多,估算误差也越大。在图 1 中节点 i 与 j 的实际距离略大于通信半径,但在 DV-Hop 算法中却是 4 跳的距离。因此经典的 DV-Hop 算法本身存在较大的定位误差。

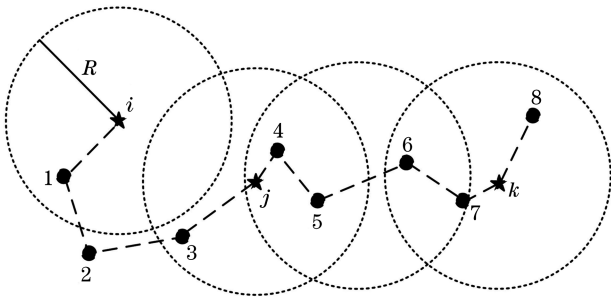


图 1 节点通信示意图

Fig. 1 An example of communication between anchor node and unknown nodes

3 IDV-Hop-DPDEL 算法思想

3.1 测距值的修正

在求解待定位节点与锚节点的距离时,采用距离该未知节点最近的锚节点平均跳距作为自己的平均跳距乘以最小跳数,忽略了锚节点周围网络分布

的差异性,故将带来大的测距误差。本文提出的算法先采用三种通信半径同时广播锚节点位置信息,再依据节点分布情况计算出测距修正因子,最后选取相对应的通信半径修正测距值。

为描述待定位节点与锚节点的远近程度,定义了一个远近度变量 μ_{ij} , 计算公式为

$$\mu_{ij} = \frac{\text{Num}[S_i(r) \cup S_j(r)]}{\text{Num}[S_i(r) \cap S_j(r)]}, \quad (7)$$

式中: $S_i(r)$ 和 $S_j(r)$ 分别为以通信半径为 r 的邻居节点集; $\text{Num}[S_i(r) \cup S_j(r)]$ 和 $\text{Num}[S_i(r) \cap S_j(r)]$ 为 $S_i(r)$ 与 $S_j(r)$ 两个集合的并集与交集所含的未知节点个数。锚节点采用三种通信半径广播,公式为

$$r = \begin{cases} \frac{R}{3}, & 0 < d_{ij} \leq \frac{R}{3} \\ \frac{2R}{3}, & \frac{R}{3} < d_{ij} \leq \frac{2R}{3} \\ R, & \frac{2R}{3} < d_{ij} \leq R \end{cases} \quad (8)$$

远近度变量计算如图 2 所示,表示三种通信半径广播示意图,其中图 2(a)是计算节点 F 远近度变量示意图, $\text{Num}[S_E(\frac{R}{3}) \cup S_F(\frac{R}{3})] = 18$, $\text{Num}[S_E(\frac{R}{3}) \cap S_F(\frac{R}{3})] = 9$, 因此按照 (8) 式可得 $\mu_{EF} = 2$, 该值代表 E, F 两个节点的相近程度,该值越大,表明越接近广播半径的值。在广播半径区域内,找到 μ_{ij} 最大的点,计算可得锚节点的测距修正因子 $\epsilon_{ij} = \frac{\mu_{ij}}{\mu_{i\max}}$, 其中, $\mu_{i\max}$ 为距离锚节点 i 最近的邻居节点,未知节点的测距修正值为 $d_{ij, \text{new}} = \epsilon_{ij} \times r$ 。

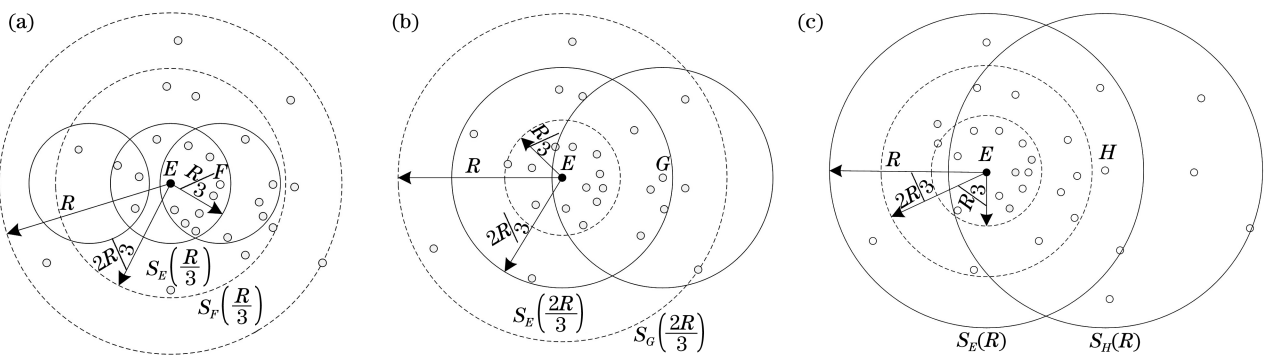


图 2 远近度区域示意图。(a) $r = \frac{R}{3}$; (b) $r = \frac{2R}{3}$; (c) $r = R$

Fig. 2 An example for area of near-degree. (a) $r = \frac{R}{3}$; (b) $r = \frac{2R}{3}$; (c) $r = R$

3.2 动态调参差分进化算法

差分进化算法^[13]是一种基于群体的启发式搜

索算法。进化流程包括变异、交叉和选择三步,该算法主要含有种群规模 N_{NP} 、变异因子(也称缩放

因子) F 和交叉概率因子 C_{CR} 这三个参数。 F 大或 C_{CR} 小, 均有利于全局搜索, 但收敛速度与搜寻精度却相对较低; 反之, 算法收敛速度虽快, 有利于局部寻优, 但易出现早熟收敛现象。为此, 本文算法动态递减变异因子 F , 递增交叉概率因子 C_{CR} [14-16]。在算法初期, 由于种群个体相对分散, 先执行全局搜索, 使种群个体逐渐靠向最优个体。在算法的收敛阶段, 进行局部搜索, 以加快收敛速度的同时提高搜寻精度。提出的动态调参差分进化算法平衡了全局与局部搜索, 使得算法性能表现最优。

以下是动态调参差分进化算法的计算步骤。

步骤 1: 种群初始化。由于 1 个未知节点对应 3 个种群个体, 故随机生成 $3N_{NP}$ 个 D 维个体构成的初始种群(节点定位坐标仅涉及到横纵坐标, 因此 D 为 2), 并对种群进行初始化, 即

$$\mathbf{g}_i = \mathbf{g}_{\min} + \delta(\mathbf{g}_{\max} - \mathbf{g}_{\min}), \quad (9)$$

式中: \mathbf{g}_{\min} 为搜索下限; \mathbf{g}_{\max} 为搜索上限; $\delta \sim U(0, 1)$; 种群中任意个体记为 $\mathbf{g}_i^t = (x_i^t, y_i^t)$, $i = 1, 2, \dots, N_{NP}$, 其中 t 为迭代次数。

步骤 2: 变异操作。变异是差分进化算法的核心, 变异向量为

$$\mathbf{V}_i^t = \mathbf{g}_{r_1}^t + F(\mathbf{g}_{r_2}^t - \mathbf{g}_{r_3}^t), \quad (10)$$

式中: $F(\cdot)$ 为变异因子; $i, r_1, r_2, r_3 \in \{1, 2, \dots, N_{NP}\}$, 且它们互不相同。 F_{new} 为动态变异因子, 其动态调整公式为

$$F_{\text{new}}(t) = (F_{\max} - F_{\min}) \times \frac{\sqrt{T^2 - t^2}}{T} + F_{\min}, \quad (11)$$

式中: T 为最大迭代次数; F_{\max} 为变异因子最大值; F_{\min} 为变异因子最小值; $F_{\text{new}} \in [0.4, 0.9]$, 动态变异因子随迭代次数 t 非线性单调递减。

步骤 3: 交叉操作。将变异向量和目标向量进行交叉操作, 生成实验向量 \mathbf{u}_i^t , 即

$$\mathbf{u}_i^t = \begin{cases} \mathbf{V}_i^t, & \gamma \leq C_{CR} \text{ or } \delta = \delta_{\text{rand}} \\ \mathbf{g}_i^t, & \gamma > C_{CR} \end{cases}, \quad (12)$$

式中: γ 为位于 $[0, 1]$ 间产生的均匀分布的随机数; δ_{rand} 为随机生成数 1 或 2, 1 代表 x_i^t , 2 代表 y_i^t 。 $C_{CR_{\text{new}}}$ 为动态交叉概率因子, 其动态调整公式为

$$C_{CR_{\text{new}}} = (C_{CR_{\max}} - C_{CR_{\min}}) \times \left(1 - \frac{\sqrt{T^2 - t^2}}{T}\right) + C_{CR_{\min}}, \quad (13)$$

式中: $C_{CR_{\max}}$ 为交叉概率因子最大值; $C_{CR_{\min}}$ 为交叉概率因子最小值; $C_{CR_{\text{new}}} \in [0.5, 1]$, 动态交叉概率因子随 t 非线性单调递增。

步骤 4: 选择操作。通过合理的适应度函数比较实验向量 \mathbf{u}_i^t 与当前的目标向量 \mathbf{g}_i^t , 即

$$\mathbf{g}_i^{t+1} = \begin{cases} \mathbf{u}_i^t, & f(\mathbf{u}_i^t) \leq f(\mathbf{g}_i^t) \\ \mathbf{g}_i^t, & f(\mathbf{u}_i^t) > f(\mathbf{g}_i^t) \end{cases}. \quad (14)$$

适应度函数小的向量将在下一代出现。本文设置的适应度函数 f 可表示优化坐标到锚节点的距离与跳距的差值, 差值越小, 定位误差越小, 该优化值就越接近真实位置, 表达式为

$$f_t = \sum_{k=1}^N \left| \sqrt{(u_{i1}^t - x_k)^2 - (u_{i2}^t - y_k)^2} - d_{ik} \right|, \quad (15)$$

式中: (u_{i1}^t, u_{i2}^t) 代表第 t 代种群个体的坐标; (x_k, y_k) 表示第 k 个锚节点的坐标; d_{ik} 为修正测距值。WSNs 锚节点个数, 表示修正后对应于目标节点 \mathbf{g}_i^t 的未知节点的跳距, 对 $3N_{NP}$ 初始种群同时搜索寻优, 结果产生三个最优坐标, 以三个坐标重心作为未知节点的估计位置。

3.3 IDV-Hop-DPDEL 算法流程

将 DV-Hop 算法的未知节点定位问题转换为距离优化问题。先利用改进后的跳数及远近度变量对测距值进行修正, 再利用提出的动态参数差分进化算法优化未知节点的估计位置, 每次迭代更新都要比较目标个体与实验个体的适应度函数值大小, 选取适应度值小的个体作为子代。IDV-Hop-DPDEL 算法流程如图 3 所示, T_{\max} 代表最大迭代次数。

4 算法仿真结果与分析

利用 MATALAB 建立仿真平台, WSNs 的节点分布区域为 $100 \text{ m} \times 100 \text{ m}$, 且所有节点均随机分布, 仿真参数如表 1 所示, 衡量定位算法性能好坏的指标有平均定位误差以及定位精度这两种 [17], 即

$$E_{\text{average_error}} = \frac{\sum_{i=1}^{N-M} \sqrt{(x_i - x_i')^2 - (y_i - y_i')^2}}{N - M}, \quad (16)$$

$$A_{\text{average_accuracy}} = \frac{\sum_{i=1}^{N-M} \sqrt{(x_i - x_i')^2 - (y_i - y_i')^2}}{(N - M)r}, \quad (17)$$

式中: $N - M$ 为未知节点数量; (x_i, y_i) 和 (x_i', y_i')

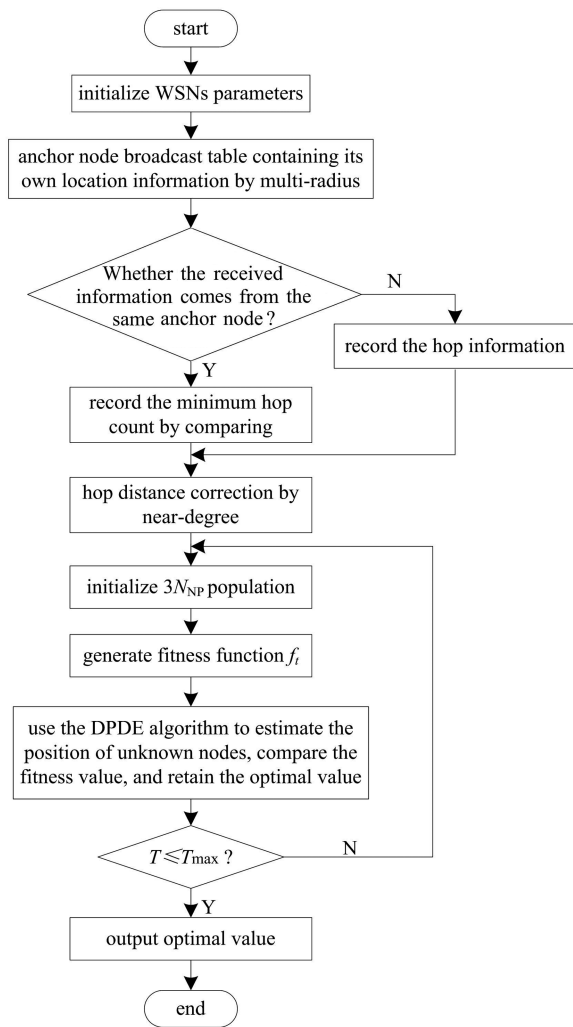


图 3 IDV-Hop-DPDEL 算法流程

Fig. 3 Procedure of IDV-Hop-DPDEL

表 1 算法仿真参数

Table 1 Parameters being used

Parameter	Value
Network size / m ²	100 × 100
Number of nodes	100
Anchor node ratio / %	5, 10, 15, 20, 25, 30, 35, 40
Communication radius / m	45
Variation factor <i>F</i>	[0.4, 0.9]
Crossover probability <i>C_{CR}</i>	[0.5, 1]
Generation counter	100

为真实位置和算法寻优结果。从(16)式和(17)式可以看出,节点总数、锚节点个数以及通信半径均是影响定位精度的因素,因此基于以上三个参数针对本文提出的 IDV-Hop-DPDEL 算法、经典 DV-Hop 算法、文献[8]提出的 MEACS-D 算法以及文献[11]

提出的 HDCDV 算法进行仿真,通过比较定位精度验证 IDV-Hop-DPDEL 算法性能的优劣。

图 4 为 WSNs 节点分布图,所有节点位于 100 m × 100 m 的正方形监测区域内,传感器节点随机部署,未知节点设为 90 个,用圆圈表示,锚节点为 10 个,用五角星表示,节点通信半径均为 45 m。图 5 为不同算法下未知节点的归一化定位误差三维图, DV-Hop 算法用圆圈表示,平均定位误差为 0.64, IDV-Hop-DPDEL 算法则用五角星表示,平均定位误差为 0.31,相比 DV-Hop 算法, IDV-Hop-DPDEL 算法的定位误差下降了 51.56%,由此表明本文提出的 IDV-Hop-DPDEL 算法的定位精度明显提升。

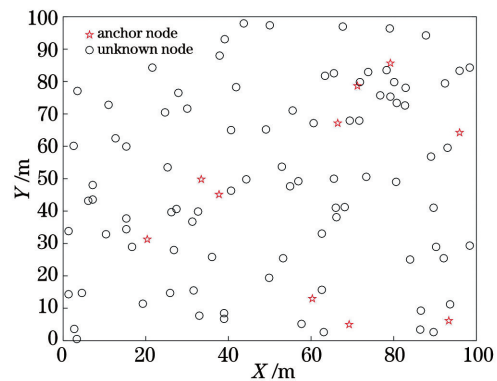


图 4 节点分布图

Fig. 4 Node distribution

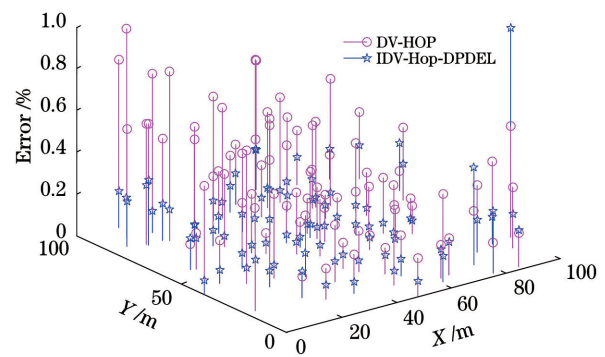


图 5 IDV-Hop-DPDEL 和 DV-Hop 算法下未知节点的定位误差图

Fig. 5 Localization error of each unknown node based on IDV-Hop-DPDEL and DV-Hop

图 6 为仅修正测距值的节点定位误差,其节点最大定位误差为 35 m,最小定位误差为 0.5 m, 90 个未知节点的平均定位误差为 7.19 m。

图 7 为仅在经典 DV-Hop 算法的基础上采用 DPDEL 算法优化的未知节点坐标,其节点最大定位误差为 25 m,最小定位误差为 1 m,平均定位误

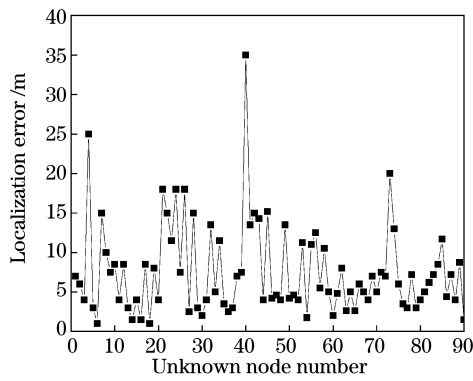


图 6 测距值修正的定位误差

Fig. 6 Localization error of hop distance correction

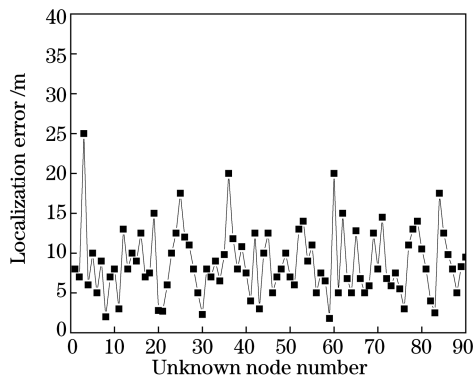


图 7 采用 DPDEL 算法的定位误差

Fig. 7 Localization error of DPDEL algorithm

差为 8.61 m。

图 8 为 IDV-Hop-DPDEL 算法的节点定位误差图,其最大定位误差为 11.58 m,最小定位误差为 0.29 m,平均定位误差为 4.28 m。

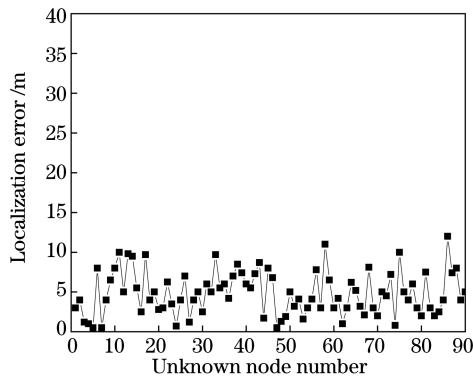


图 8 IDV-Hop-DPDEL 算法的定位误差

Fig. 8 Localization error of IDV-Hop-DPDEL algorithm

图 9 表示监测区域随机部署 100 个传感器节点,锚节点所占比例分别为 5%、10%、15%、20%、25%、30%、35%、40%、45% 时 4 种算法的平均定位误差对比。在保证其他参数不变的情况下,4 种算法曲线走势均随锚节点数量增加而下降,本文提出

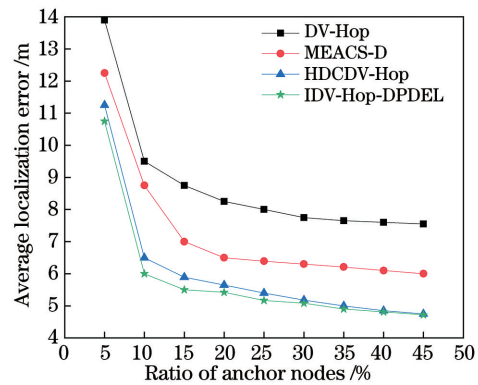


图 9 锚节点对平均定位误差的影响

Fig. 9 Average localization error with variation of ratio of anchor nodes

的 IDV-Hop-DPDEL 算法下降程度明显大于其余 3 种算法,当锚节点占比为 12.5% 的时候, IDV-Hop-DPDEL 算法较经典 DV-Hop 算法定位误差低 38%,比 MEACS-D 算法低 25%,比 HDCDV-Hop 算法低 8%。

图 10 表示网络监测区域随机部署 100 个节点,通信半径从 20 m 以 5 m 为间隔增加至 60 m 时 4 种算法的平均定位误差对比。4 种算法的总趋势都是随着通信半径的增加而降低的。由于通信半径短、网络连通率低,定位误差较大。在节点总数为 100,锚节点占比为 10% 的前提下:当通信半径为 30 m 时,网络连通率达 100%,总体 IDV-Hop-DPDEL 算法的定位误差较其余三种算法都低,这说明本文采用的三种通信半径广播方式取得了良好的定位效果;当半径增加到一定程度时, IDV-Hop-DPDEL 算法呈现平滑趋势,但其余三种算法曲线均有不同程度的上升,这是因为其余三种算法求未知节点跳段距离用的是锚节点平均跳距,半径增加到一定临界点时,锚节点平均跳距误差反而加大,而

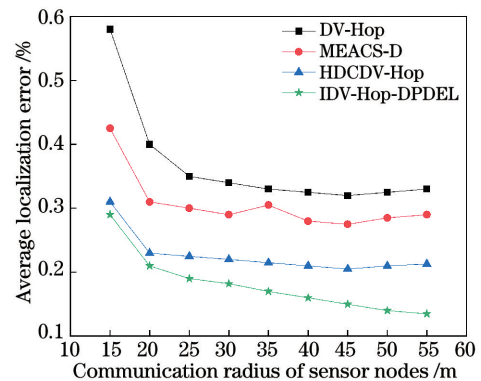


图 10 通信半径对平均定位误差的影响

Fig. 10 Average localization error with variation of communication radius of sensor nodes

IDV-Hop-DPDEL 算法是利用节点远近度求得测距值。

图 11 所示为网络节点总数对定位精度的影响,从图可看出曲线走势随节点数增加,定位误差缓慢降低,经典 DV-Hop 算法受节点数量的影响较大,整体下降 24%,算法稳定性差,而 IDV-Hop-DPDEL 算法稳定性较其余两种经典改进的 DV-Hop 算法更好,曲线仅下降了 7%。

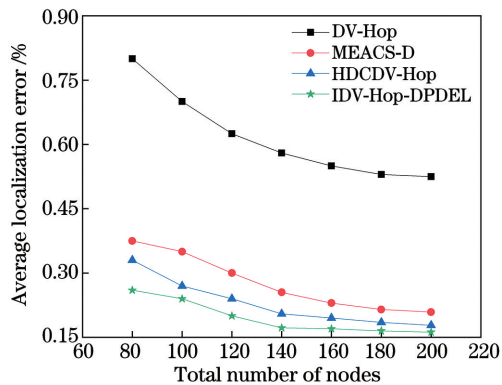


图 11 节点总数对平均定位误差的影响

Fig. 11 Average localization error with variation of total number of nodes

5 结 论

通过分析经典 DV-Hop 算法的主要误差来源,提出基于 DV-Hop 测距修正的动态调参差分进化定位算法。该算法采用具有多通信半径的锚节点广播通信,结合节点远近度修正测距值,利用能够动态调整参数的差分进化算法进行寻优搜索,全局与局部寻优达到动态平衡。实验表明, IDV-Hop-DPDEL 算法更适用于网络节点定位,定位精度较原算法有明显提升,且与其余两种经典改进 DV-Hop 算法相比,性能更优。在后续研究中将结合其他智能优化算法寻求最优位置,降低算法复杂度。高精度低能耗的算法有待进一步研究。

参 考 文 献

- [1] Yang C X. An improved DV-Hop localization algorithm based on differential evolution [D]. Changchun: Jilin University, 2015.
杨楚哲. 基于差分进化的 DV-Hop 定位改进算法 [D]. 长春: 吉林大学, 2015.
- [2] Qian Z H, Sun D Y, Victor L. A survey on localization model in wireless networks[J]. Chinese Journal of Computers, 2016, 39(6): 1237-1256.
钱志鸿, 孙大洋, Victor L. 无线网络定位综述[J]. 计算机学报, 2016, 39(6): 1237-1256.
- [3] Yang C C, Chen J H, Liu D L, et al. Indoor joint localization algorithm based on time and angle of arrival[J]. Laser & Optoelectronics Progress, 2019, 56(22): 222802.
杨超超, 陈建辉, 刘德亮, 等. 基于到达时间和到达角度的室内联合定位算法[J]. 激光与光电子学进展, 2019, 56(22): 222802.
- [4] Cao Y X, Wang Z. Improved DV-Hop localization algorithm based on dynamic anchor node set for wireless sensor networks[J]. IEEE Access, 2019, 7: 124876-124890.
- [5] Niculescu D, Nath B. DV based positioning in ad hoc networks[J]. Telecommunication Systems, 2003, 22 (1/2/3/4): 267-280.
- [6] Fang W S, Hu Y Q, Hu Z D. DV-HOP algorithm to reevaluate hop distance and amend minimum hop count[J]. Journal of Electronic Measurement and Instrumentation, 2018, 32(8): 201-208.
方旺盛, 胡云青, 胡中栋. 对跳距重估与最小跳数修正的 DV-HOP 算法[J]. 电子测量与仪器学报, 2018, 32(8): 201-208.
- [7] Cheng C, Qian Z H, Fu C X, et al. Genetic optimization DV-Hop localization algorithm based on error distance weighted and hop algorithm selection [J]. Journal of Electronics & Information Technology, 2015, 37(10): 2418-2423.
程超, 钱志鸿, 付彩欣, 等. 一种基于误差距离加权与跳段算法选择的遗传优化 DV-Hop 定位算法[J]. 电子与信息学报, 2015, 37(10): 2418-2423.
- [8] Yu X W, Hu M F, Liu Y, et al. Adaptive cuckoo optimization DV-HOP localization algorithm based on multi-communication radius and error correction[J]. Journal of Southwest Jiaotong University, 2018, 53 (6): 1123-1129.
余修武, 胡沐芳, 刘永, 等. 多半径误差修正自适应布谷鸟优化 DV-HOP 定位算法[J]. 西南交通大学学报, 2018, 53(6): 1123-1129.
- [9] Xu H J. Improved DV-Hop ranging-based genetic-simulated annealing localization algorithm in wireless sensor networks[J]. Chinese Journal of Sensors and Actuators, 2018, 31(1): 147-151.
徐慧娟. 基于 DV-Hop 测距修正的遗传模拟退火定位算法[J]. 传感技术学报, 2018, 31(1): 147-151.
- [10] Liu Y, Gao L. Optimized DV-Hop localization algorithm based on improved artificial bee colony[J]. Laser & Optoelectronics Progress, 2020, 57 (19): 192303.
刘燕, 高丽. 改进人工蜂群优化的 DV-Hop 定位算法[J]. 激光与光电子学进展, 2020, 57(19): 192303.
- [11] Liu G Q, Qian Z H, Wang X. An improved DV-Hop localization algorithm based on hop distances

- correction[J]. China Communications, 2019, 16(6): 200-214.
- [12] Fan S P, Luo D, Liu Y L. DV-Hop localization algorithm based on hop-size and improvement particle swarm optimization[J]. Chinese Journal of Sensors and Actuators, 2016, 29(9): 1410-1415.
范时平, 罗丹, 刘艳林. 基于跳距与改进粒子群算法的 DV-Hop 定位算法[J]. 传感技术学报, 2016, 29(9): 1410-1415.
- [13] Storn R, Price K. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.
- [14] Fang X Y, Li X B, Guo Z. Improved hybrid grey wolf optimization support vector machine prediction algorithm and its application [J]. Laser & Optoelectronics Progress, 2020, 57(12): 122801.
方晓玉, 李晓斌, 郭震. 一种改进的混合灰狼优化支持向量机预测算法及应用[J]. 激光与光电子学进展, 2020, 57(12): 122801.
- [15] Yao M L, Wang X J, Zhang F G, et al. Synthesis of sparse rectangular planar arrays with multiple constraints based on dynamic parameters differential evolution algorithm [J]. Journal of Electronics & Information Technology, 2020, 42 (5): 1281-1287.
姚敏立, 王旭健, 张峰干, 等. 基于动态参数差分进化算法的多约束稀疏布矩形面阵优化[J]. 电子与信息学报, 2020, 42(5): 1281-1287.
- [16] Mao Y J, Li S K, Wang X Z, et al. Lithographic tool-matching method based on differential evolution algorithm[J]. Acta Optica Sinica, 2019, 39(12): 1222002.
茅言杰, 李思坤, 王向朝, 等. 基于差分进化算法的光刻机匹配方法[J]. 光学学报, 2019, 39(12): 1222002.
- [17] Shen S K, Yang B, Qian K G, et al. On improved DV-Hop localization algorithm for accurate node localization in wireless sensor networks[J]. Chinese Journal of Electronics, 2019, 28(3): 658-666.