DV-Hop Location Algorithm Based on Improved Jaya*

PENG Duo, YUN Qi, LI Yingtang, CHEN Hao*
(*School of Computer and Communication of Lanzhou University of Technology, Lanzhou Gansu* 730050, *China*)

Abstract: Aiming at the problem of low positioning accuracy of classical DV-Hop algorithm, a DV-Hop positioning algorithm based on improved Jaya is proposed. The algorithm first adds a correction factor to modify the average hop distance, then selects anchor nodes through the concept of collinearity to reduce the positioning error, then introduces the tent map to generate the initial population, in order to enhance the diversity of the population and improve the convergence speed, finally constructs the objective function, and uses the improved Jaya optimization algorithm to obtain the unknown node coordinates. The simulation results show that compared with the traditional DV-Hop algorithm and the improved algorithm, the average value of positioning error is reduced by 75.00% and 65.83% respectively, and the positioning accuracy is higher.

Key words: wireless sensor network; localization algorithm; Jaya algorithm; DV-Hop algorithm
EEACC: 7230 **doi:** 10.3969/j.issn.1004-1699.2020.08.019

基于改进 Jaya 的 DV-Hop 定位算法*

彭 铎*, 贠 琦, 李英堂, 陈 昊
(兰州理工大学计算机与通信学院, 甘肃 兰州 730050)

摘 要: 针对经典 DV-Hop 算法定位精度较低的问题, 提出一种基于改进 Jaya 的 DV-Hop 定位算法。该算法首先添加修正因子修正平均跳距; 然后通过共线性的概念选择锚节点以减小定位误差; 接着引入 Tent 映射生成初始种群, 以增强种群多样性, 提高收敛速度; 最后构建目标函数, 利用改进的 Jaya 优化算法求得未知节点坐标。仿真结果表明, 提出算法与经典 DV-Hop 算法和改进算法相比, 定位误差平均值分别降低了 75.00%和 65.83%, 定位精度更高。

关键词: 无线传感器网络; 定位算法; Jaya 优化算法; DV-Hop 算法
中图分类号: TP393 **文献标识码:** A **文章编号:** 1004-1699(2020)08-1204-06

无线传感器网络(Wireless Sensor Network, WSN)是由多个节点组成的自组织网络, 它的优点是体积小、成本低、能耗少、对环境适应性强, 能执行环境监控、灾难救援、目标跟踪等任务[1-2]。因此, WSN 定位算法成为科研人员研究的热点之一。WSN 定位算法通常分为距离有关和距离无关的算法[3]。基于接收信号强度指示的 RSSI、基于信号到达时间差的 TDoA 以及基于信号到达角度的 AoA 等属于距离有关的算法, 通常需要额外的硬件支持, 硬件设施能耗成本较高[4-5]。而基于距离向量交换的 DV-Hop、基于多维标度的 MDS-MAP、基于相似距离映射的 PDM/SDM 等属于距离无关的算法, 通常利用网络的连通性进行定位, 不需要额外硬件支持, 因此得到广泛关注[6]。

DV-Hop 算法的复杂性低、成本低, 因此在 WSN 定位中被广泛应用, 许多改进的 DV-Hop 算法也被提出。文献[7]提出的改进 DV-Hop 算法, 通过加入锚节点权重和逃逸因子的改进粒子群算法代替极大似然估计法定位节点坐标, 以减小定位误差; 文献[8]提出一种改进的 DV-Hop 算法, 引入基于跳数细化来计算节点间跳数, 进而细化平均跳距, 以提高定位精度; 文献[9]提出一种基于全局跳数优化与跳距误差修正的 DV-Hop 算法, 先利用节点的相关参数对全局跳数值进行优化, 然后根据最小指数的最小均方差准则计算平均跳距, 以减小累积误差; 文献[10]提出一种基于粒子群优化算法的改进的 DV-Hop 算法, 在计算出平均跳距后, 用粒子群算法修正二维双曲线算法, 使定位结果更接近真实值。上述研究在一定程度上减小了定位误差, 提高了定位精度。但仍存在最小二乘法对初值敏感以及引入智能优化算法增加计算复杂度等问题。本文提出一种基于改进 Jaya 的 DV-Hop 定位算法, 对算法的测量阶段和定位阶段进行改进, 以得到更高的定位精度。

项目来源: 国家自然科学基金项目(61663024, 51667014, 61841107); 甘肃省高校创新基金项目(2020A-021)
收稿日期: 2020-06-08 修改日期: 2020-08-17

1　DV-Hop 算法

1.1　DV-Hop 算法步骤

DV-Hop 定位算法步骤如下[11]：

①计算未知节点与锚节点之间的最小跳数

每个锚节点 A_i 以多跳的方式向网络中广播包含 A_i 的位置以及从 0 开始的跳数字段的报文信息。在这个过程中，每经过一跳，跳数都会增加 1。网络中的节点 J（锚节点或未知节点）记录 A_i 的位置并初始化它和 A_i 之间的最小跳数，记为 $\mathrm{hop}_{i,j}$。当节点 J 接收到的跳数值低于先前存储的跳数 $\mathrm{hop}_{i,j}$ 时，则 J 将根据该最小跳数更新 $\mathrm{hop}_{i,j}$；否则忽略该跳数值。

②计算锚节点的平均每跳距离

第 1 阶段记录其他锚节点的位置信息和相距跳数后，每个锚节点估算平均每跳距离为：

$$\mathrm{Hopsize}_i=\frac{\sum_{i\neq j}\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}}{\sum_{i\neq j}\mathrm{hop}_{ij}} \tag{1}$$

式中，Hopsize 表示锚节点的平均每跳距离；(x_i,y_i) 和 (x_j,y_j) 分别是锚节点 i、j 的坐标；hop_{ij} 表示锚节点 i、j 之间的最小跳数。锚节点将计算的平均跳距广播至网络中，未知节点记录接收到的平均跳距并将其转发。

③计算未知节点的坐标

未知节点接收到平均跳距后，计算它和锚节点 A_i 间的距离，然后根据三边测量法得到未知节点的坐标，

$$d_{it}=\mathrm{Hopsize}_i\times\mathrm{hop}_{it} \tag{2}$$

式中，d_{it} 和 hop_{it} 分别表示锚节点 i 和未知节点 t 之间的距离和最小跳数。

1.2　DV-Hop 算法误差分析

网络拓扑的影响导致节点间的实际每跳距离存在误差，所以当节点间的平均跳距没有达到理想值时，得到的节点间的实际距离和计算距离相比就会存在较大误差，影响定位精度。

节点在网络中是随机分布的。当大于等于 3 个锚节点存在未知节点周围时，可以很快计算出未知节点的坐标。然而，当参与定位的锚节点在一条直线上时，就可能出现定位盲区，无法确定未知节点的坐标，降低算法的定位精度。

2　基于 Tent 映射的 Jaya 优化算法

2.1　Jaya 算法

Jaya 优化算法是一种群体智能优化算法，在迭代过程中只有一个方程。每一次迭代中，每个解通过更新公式在搜索空间中移动并生成新解。它的优点是：同时考虑最优解和最差解对搜索的作用、计算复杂度低、易于实现、鲁棒性强[12-14]。该算法中，首先随机生成初始解构成初始种群（初始解服从过程变量）。然后进入迭代搜索过程，此过程中每个解的每个变量都是随机的，使用式（3）更新，Jaya 算法通过更新变量的值将每个解的目标函数值移向最佳解，变量的值更新后，将更新的新解 X_{i+1} 与相应的旧解 X_i 进行比较，只有优解被下一代考虑保存下来，

$$X_{i+1}=X_i+r_1(X_{\mathrm{best}}-|X_i|)-r_2(X_{\mathrm{worst}}-|X_i|) \tag{3}$$

式中，[0,1] 范围内的随机数 r_1，r_2 作为比例因子，可确保良好的多样化；式中的第 2 项“$r_1(X_{\mathrm{best}}-|X_i|)$”表示个体在更新的过程中不断向最优解靠近，而第 3 项“$r_2(X_{\mathrm{worst}}-|X_i|)$”表示个体在更新过程中远离最差解。所以 Jaya 算法的工作原理是持续改进，倾向于通过向某个方向不断移动来靠近当代最优解同时远离最差解，能够很好地实现搜索过程的多样化，最终，满足终止标准的最优解即为算法搜索的最优解。

2.2　基于 Tent 映射的 Jaya 优化算法（T-Jaya）

在使用 Jaya 算法选择最优解时，易出现收敛速度慢、种群多样性丢失等问题，因此提出 T-Jaya 优化算法来增强其寻优能力和可靠性。在 Jaya 算法中引入 Tent 混沌序列方法生成初始种群。T-Jaya 优化算法执行框图如图 1 所示。

Tent 映射表达式如下：

$$x_{k+1,i}=\begin{cases}2x_{k,i} & 0\leqslant x_{k,i}\leqslant 0.5\\ 2(1-x_{k,i}) & 0.5\leqslant x_{k,i}\leqslant 1\end{cases} \tag{4}$$

式中，$x_{k,i}$ 表示个体 k 第 i 维变量对应的混沌数。

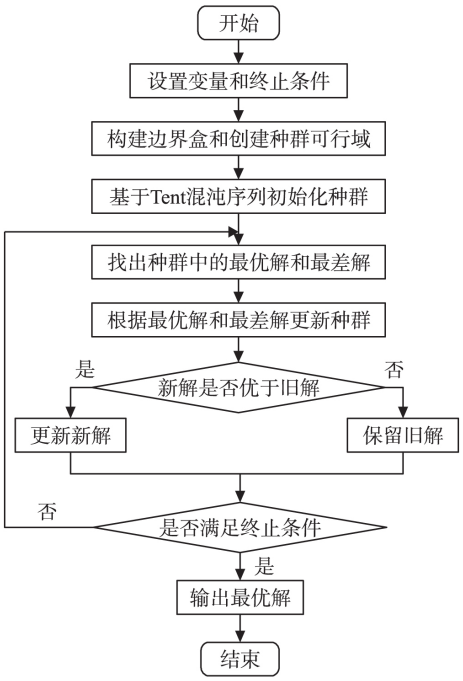

图 1　T-Jaya 算法程序执行框图

个体 k 第 i 维变量更新公式为:

$$X_{k,i}=x_{k,i}(X_{\max,i}-M_{\min,i})+X_{\min,i} \tag{5}$$

生成初始种群的具体方法如下:

Step 1 根据种群的大小 $2M$,随机生成 $2M$ 个 $(0,1)$ 之间的混沌数,同时对生成的随机数按照式(4)进行 5 次试算,如果生成的混沌数出现重复,则重新生成初始混沌值直到该值可用;

Step 2 根据更新式(5)将生成的混沌序列还原到解空间内,生成第 1 个个体,然后按照式(3)生成下一个个体对应的混沌序列,重复步骤 2,直到达到种群数量。

3 基于 T-Jaya 的 DV-Hop 定位算法

本文提出的基于 T-Jaya 的 DV-Hop 定位算法主要包括 3 个步骤。首先,利用修正因子修正锚节点平均每跳距离;然后利用共线性选择锚节点避免出现定位盲区;最后引入 T-Jaya 优化算法寻求未知节点的位置坐标。

3.1 添加修正因子修正锚节点平均每跳距离

锚节点的平均跳距是节点间距离估计不准确的主要原因。因此添加修正因子优化锚节点的平均跳距。

锚节点间的实际距离表示为:

$$D_{ij}'=\sqrt{(x_i-y_j)+(y_i-y_j)^2}\quad(i\neq j) \tag{6}$$

式中,(x_i,y_i) 和 (x_j,y_j) 分别是锚节点 i、j 的坐标;D_{ij}' 是两个锚节点 i、j 之间的实际距离。

两个锚节点之间的距离还可以表示为它们之间的跳数和任意锚节点的平均距离的乘积。令 D_{ij} 为锚节点 i、j 之间的计算距离,则:

$$D_{ij}=\text{Hopsize}\times\text{hop}_{ij} \tag{7}$$

根据式(6)和式(7),锚节点 i、j 之间的距离误差 d_{ij}^e 可表示为:

$$d_{ij}^e=|D_{ij}-D_{ij}'| \tag{8}$$

根据距离误差 d_{ij}^e 计算锚节点 i 的修正因子 c_i 为:

$$c_i=\frac{\sum_{i\neq j}d_{ij}^e}{\sum_{i\neq j}\text{hop}_{ij}} \tag{9}$$

将修正因子添加到式(1)中修正平均跳距,则修正后锚节点和未知节点之间的距离为:

$$d_{it}'=(\text{Hopsize}_i+c_i)\times\text{hop}_{it} \tag{10}$$

式中,d_{it}' 和 hop_{it} 分别表示锚节点 i 和未知节点 t 之间的修正距离和跳数。

3.2 利用共线性选择锚节点

在 WSN 中,如果参与定位的锚节点在一条直线上,未知节点的坐标就可能无法确定,出现定位盲区,降低算法的定位精度。为解决这一问题,引入共线性(DCL)的概念。当 3 个锚节点构成的区域面积为零时,则代表这 3 个点共线,该面积表示为:

$$a=\frac{1}{2}\begin{vmatrix}x_1 & y_1 & 1\\ x_2 & y_2 & 1\\ x_3 & y_3 & 1\end{vmatrix} \tag{11}$$

式中,a 为 3 个锚节点构成的区域面积大小;(x_1,y_1)、(x_2,y_2) 和 (x_3,y_3) 为 3 个锚节点坐标。

DCL 表达式为:

$$\Psi_{\text{DCL}}=\begin{cases}0 & \text{共线}\\ a & \text{不共线}\end{cases} \tag{12}$$

根据上式选择合适的锚节点以计算未知节点坐标,如果 3 个锚节点不共线,则参与未知节点的定位;反之,则不参与未知节点的定位。

3.3 基于 T-Jaya 的定位坐标优化

定位的实质是误差的最小化问题,我们将 Jaya 算法应用到 WSN 定位算法中,以减小定位误差。将估计未知节点位置的问题转换为优化问题,将定位误差作为目标函数,以最小化目标函数作为优化目标。每个未知节点独立地运行优化算法以找到自身坐标。假设 WSN 中共有 N 个传感器节点,k 个锚节点,$N-k$ 个未知节点。

锚节点与未知节点之间的估计距离表示为:

$$d_{it}=\sqrt{(x_t-x_i)^2+(y_t-y_i)^2} \tag{13}$$

式中,(x_i,y_i) $(i=1,2,\cdots,k)$ 表示锚节点的坐标;(x_t,y_t) $(t=k+1,k+2,\cdots,n)$ 表示未知节点的估计坐标;

定位问题的目标函数可以定义为:

$$f(x_t,y_t)=\text{Min}\left(\sum_{\substack{i=1,2,\cdots,k\\ t=k+1,\cdots,n}}|d_{it}-d_{it}'|\right) \tag{14}$$

式中,d_{it}' 是未知节点到锚节点之间的实际距离。

由于锚节点随机分布,一些锚节点可能集中在某一区域,通过这些锚节点求未知节点估计位置,会产生很大的定位误差和很差的稳定性。因此,计算未知节点坐标时,为了最小化定位误差,创建种群可行域,并在可行域中随机部署节点作为初始种群,如图 2 所示。假设 R 为所有节点的通信半径,未知节点 N_t 坐标是 (x_t,y_t),其邻居锚节点 A_1,A_2,A_3 坐标是 (x_1,y_1)、(x_2,y_2)、(x_3,y_3),未知节点 N_t 与锚节点 A_1,A_2,A_3 之间的最小跳数分别是 h_{1,N_t}、h_{2,N_t} 和 h_{3,N_t}。

分别以未知节点 N_t 的 3 个锚节点 A_1,A_2,A_3 为圆心,$R\times h_{i,N_t}$ $(i=1,2,3)$ 为半径构造 3 个圆,并给出 3 个圆的外接正方形。图中阴影为 3 个外接正方形的重叠区域,也是未知节点 N_t 的可行域。在该域内

随机生成种群，该种群根据设定的上、下界搜索未知节点 N_t 的精确坐标，未知节点的坐标处在该可行域内时，误差最小。上界和下界表示为：

$$\begin{cases}\max_{i=1,2,\cdots,k}(x_i-R\times h_{i,N_t})\leqslant x_t\leqslant \\ \quad\min_{i=1,2,\cdots,k}(x_i-R\times h_{i,N_t}) \\ \max_{i=1,2,\cdots,k}(y_i-R\times h_{i,N_t})\leqslant y_t\leqslant \\ \quad\min_{i=1,2,\cdots,k}(y_i-R\times h_{i,N_t})\end{cases} \tag{15}$$

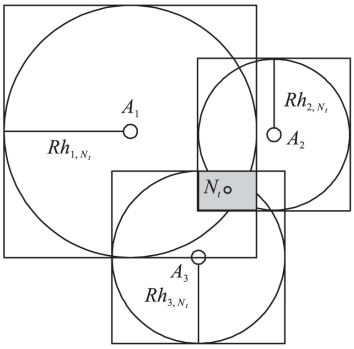

图 2　种群可行域

基于 T-Jaya 的 DV-Hop 算法定位过程：

Step 1　计算未知节点和锚节点之间的最小跳数和锚节点的平均每跳距离；

Step 2　添加修正因子修正平均每跳距离，计算出修正后的锚节点和未知节点间的距离；

Step 3　根据式(12)，利用共线性选择合适的定位锚节点；

Step 4　找出每个未知节点种群可行域，并在可行域内生成初始种群；

Step 5　在可行域内，确定每个未知节点上、下界的值；

Step 6　计算未知节点坐标，寻找最优解，将输出的最优解作为未知节点的坐标。

4　仿真结果分析

4.1　算法参数设置

仿真基于 MATLAB2016b 仿真软件。为了验证提出算法的定位性能，将经典 DV-Hop 算法、文献[10]中改进的 DV-Hop 算法和提出的 Jaya-DV-Hop 算法进行比较分析。在 100 m×100 m 的仿真区域内，随机生成 k 个锚节点和 n 个未知节点，所有节点通信半径均为 R。Jaya-DV-Hop 算法的相关参数设置如表 1 所示。

表 1　算法参数设置

参数	值
运行次数	100
种群规模	100
最大迭代次数	300

定位算法性能通过定位误差和平均定位误差来评估，表达式如下：

定位误差(LE)表示为：

$$\Gamma_{\mathrm{LE}}=\sqrt{(x_t-x_a)^2+(y_t-y_a)^2} \tag{16}$$

式中，(x_t,y_t) $(t=k+1,k+2,\cdots,n)$ 为未知节点估计坐标，(x_a,y_a) 为未知节点实际坐标。

平均定位误差(ALE)表示为：

$$\zeta_{\mathrm{ALE}}=\frac{\sum_{a=1,t=1}^{N}\sqrt{(x_t-x_a)^2+(y_t-y_a)^2}}{n} \tag{17}$$

4.2　仿真结果分析

4.2.1　定位误差比较

在传感区域为 100 m×100 m 的二维平面内，随机部署 100 个节点，其中锚节点个数为 60，未知节点个数为 40 以及通信半径为 25 m 时，3 种算法的 LE 结果如图 3 所示，与经典的 DV-Hop 算法(标记为 DV-Hop)和改进 DV-Hop 算法(标记为 IDV-Hop)相比，本文提出的算法(标记为 Jaya-DV-Hop)效果更好。

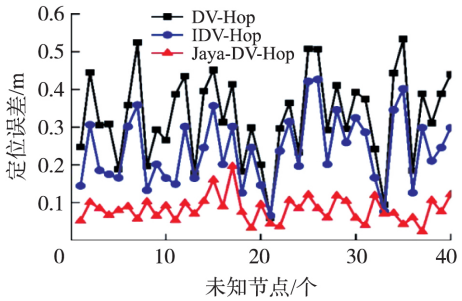

图 3　不同算法未知节点定位误差

表 2 为图 3 中 3 种算法 LE 的最小值、最大值和平均值。可以看出，与另外两种算法相比，Jaya-DV-Hop 算法的最小定位误差和最大定位误差均有明显下降，Jaya-DV-Hop 定位误差平均值为 0.082，DV-Hop 和 IDV-Hop 算法的 LE 平均值分别为 0.328 和 0.240，Jaya-DV-Hop 算法与其他两种算法相比，LE 平均值分别降低了 75.00%和 65.83%，具有更高的定位精度。

表 2　定位误差比较　　单位：m

算法	最小定位误差	最大定位误差	定位误差平均值
DV-Hop	0.058	0.533	0.328
IDV-Hop	0.065	0.426	0.240
Jaya-DV-Hop	0.022	0.121	0.082

4.2.2　锚节点比例对定位性能的影响

在传感区域随机部署 100 个传感器节点，通信半径为 20 m，锚节点数量从 10 增加到 40，在不同锚节点数量的情况下，对比分析 3 种算法的定位性能。

如图 4 所示，随着锚节点比例的增加，3 种算法的 ALE 均逐渐降低，其原因是随着锚节点数量增加意味着锚节点密度增大，未知节点和锚节点间跳数随之变小，从而导致平均跳距更加准确。而本文引入修正因子修正平均跳距也会使平均跳距更准确。也可以看出，在条件相同时，Jaya-DV-Hop 算法的 ALE 一直都最小。当锚节点比例为 35%时，Jaya-DV-Hop 算法的 ALE 最小为 0.102，此时 DV-Hop 与 IDV-Hop 算法的 ALE 分别为 0.344 和 0.261，Jaya-DV-Hop 算法与这两种算法相比，ALE 分别降低了 70.35%和 60.92%。

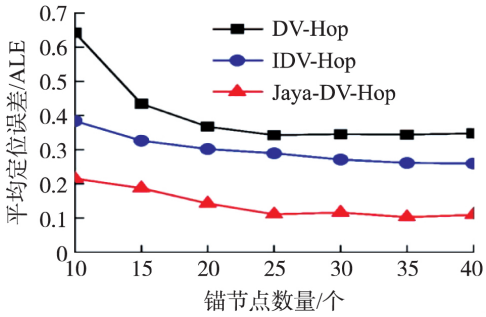

图 4 锚节点比例对定位性能的影响

4.2.3 节点总数对定位性能的影响

在传感区域随机部署传感器节点，通信半径为 25 m，锚节点比例为 10%，节点数目从 60 增加 200，在不同节点总数下，对比 3 种算法的性能。如图 5 所示，3 种算法的 ALE 随着节点总数增加均逐渐降低。其原因是节点总数增加意味着节点密度提高，网络的连通性变好，WSN 网络就可以收集到更多用于节点定位的信息，并且使用 Jaya 优化算法能够提高定位精度。所以当节点总数到达一定值时，3 种算法的 ALE 都逐渐趋于稳定，不会发生较大变化。也可以看出，在条件相同时，Jaya-DV-Hop 算法的 ALE 一直都最小。当节点总数为 200 时，Jaya-DV-Hop 算法的 ALE 最小为 0.123，此时 DV-Hop 与 IDV-Hop 算法的 ALE 分别为 0.275 和 0.238，Jaya-DV-Hop 算法与这两种算法相比，ALE 分别降低了 55.27%和 48.31%。

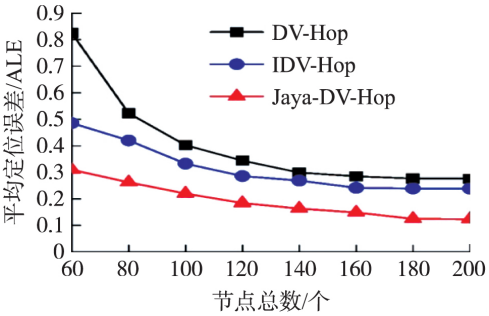

图 5 总节点数量对定位性能的影响

4.2.4 通信半径对定位性能的影响

在传感区域随机部署 100 个传感器节点，锚节点比例为 10%，通信半径从 15 增加到 40，通过改变通信半径，对比分析 3 种算法的定位性能。图 6 为不同通信半径下 3 种算法的 ALE，可以看出，在条件相同时，Jaya-DV-Hop 算法的 ALE 一直都最小。随着通信距离的增加，锚节点与更多节点直接通信，因此 ALE 逐渐趋于稳定，没有较大变化。当通信半径为 20m 时，Jaya-DV-Hop 算法的 ALE 最小为 0.187，此时 DV-Hop 与 IDV-Hop 算法的 ALE 分别为 0.519 和 0.293，Jaya-DV-Hop 算法与这两种算法相比，ALE 分别降低了 63.97%和 36.18%。

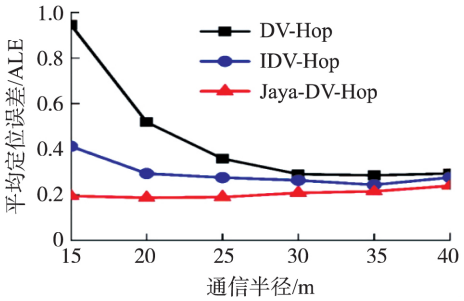

图 6 通信半径对定位性能的影响

5 结论

本文研究了经典 DV-Hop 算法定位精度较低的原因，提出了基于 T-Jaya 的 DV-Hop 定位算法。该算法首先添加修正因子对测量阶段进行改进；然后引入共线性概念和 T-Jaya 优化算法对定位阶段进行改进。减小了平均跳距产生的偏差和计算未知节点位置时产生的误差。与经典 DV-Hop 算法和 IDV-Hop 算法相比，提出的 Jaya-DV-Hop 算法具有更高的定位精度。

参考文献：

[1] 钱开国，卜春芬，王玉见，等. 基于可靠信标和节点度估计距离的无线传感器网络定位算法[J]. 计算机应用，2019，39(3)：817-823.

[2] Shahzad F，Sheltami T R，Shakshuki E M. DV-maxHop：A Fast and Accurate Range-Free Localization Algorithm for Anisotropic Wireless Networks[J]. IEEE Transactions on Mobile Computing，2017，16(9)：2494-2505.

[3] Chang S，Li Y，He Y，et al. Target Localization in Underwater Acoustic Sensor Networks Using RSS Measurements[J]. Applied Sciences，2018，8(2)：225-238.

[4] Nguyen T L N，Shin Y. An Efficient RSS Localization for Underwater Wireless Sensor Networks[J]. Sensors，2019，19(14)：3105-3122.

[5] Tomic S，Beko M，Dinis R，et al. On Target Localization Using

Combined RSS and AoA Measurements [J]. Sensors, 2018, 18(4): 1266-1290.

[6] Faheem M, Abbas M Z, Tuna G, et al. EDHRP: Energy Efficient Event Driven Hybrid Routing Protocol for Densely Deployed Wireless Sensor Networks [J]. Journal of Network and Computer Applications, 2015, 58(12): 309-326.

[7] 李新春,李苏晨,王晓明. 基于粒子群优化的 DV-Hop 定位算法研究 [J]. 测控技术, 2017, 36(1): 84-87.

[8] 胡玉兰,于溪,赵青杉. 一种改进的 DV-Hop 定位算法 [J]. 太原师范学院学报(自然科学版), 2019, 18(3): 40-44.

[9] 邓浪. 无线传感器网络 DV-Hop 定位算法研究与改进优化 [D]. 赣州: 江西理工大学, 2019.

[10] Gao M, Li F. Genetic PSO Improved DV-Hop Localization Algorithm [J]. Chinese Journal of Sensors and Actuators, 2017, 30(7): 1083-1088.

[11] Tomic S, Mezei I. Improvements of DV-Hop Localization Algorithm for Wireless Sensor Networks [J]. Telecommunication Systems, 2016, 61(1): 93-106.

[12] Rao R. Jaya: A Simple and New Optimization Algorithm for Solving Constrained and Unconstrained Optimization Problems [J]. International Journal of Industrial Engineering Computations, 2016, 7(1): 19-34.

[13] Venkata Rao R, Rai D P, Balic J. Optimization of Abrasive Waterjet Machining Process Using Multi-Objective Jaya Algorithm [J]. Materials Today: Proceedings, 2018, 5(2): 4930-4938.

[14] Venkata Rao R, Rai D P. Optimization of Selected Casting Processes Using Jaya Algorithm [J]. Materials Today: Proceedings, 2017, 4(10): 11056-11067.

彭　铎(1976—),男,甘肃省兰州市人,兰州理工大学副教授,硕导,主要研究方向为无线传感器网络,光纤网络,无线通信,pengduo7642@163.com;

贠　琦(1995—),女,黑龙江省伊春市人,兰州理工大学信号与信息处理专业研究生,主要研究方向为无线传感器网络,yunqi9503@163.com。