

基于 RSSI 测距的三维无线传感网络萤火虫定位算法的研究^①

张立国^② 马 鹏 金 梅 范美常 刘宇浩

(燕山大学河北省测试计量技术与仪器重点实验室 秦皇岛 066004)

摘 要 研究了三维空间无线传感器网络(WSN)未知节点的定位。针对目前三维空间无线传感网络定位算法精度低的问题,提出一种基于接收信号强度指示(RSSI)测距的萤火虫定位的算法。该算法首先利用 RSSI 进行测距,建立信号强度随传播距离衰减的模型,然后在已知一定数量的锚节点的情况下,通过萤火虫算法寻找目标函数最优值,进而对未知节点进行定位。仿真验证结果表明,该定位算法相比其他算法定位精度有很大提高,在 $30\text{m} \times 30\text{m} \times 30\text{m}$ 的空间内定位误差仅为 0.59m 。

关键词 无线传感网络(WSN),三维空间,接收信号强度指示(RSSI),萤火虫算法(FA)

0 引 言

无线传感网络(wireless sensor networks, WSN)节点位置对整个网络的应用有重要影响,因此节点定位技术在无线传感网络中有着举足轻重的地位。迄今为止无线传感网络的定位方法大都针对二维空间,而在实际应用中网络节点主要分布在三维空间,因而较好的三维空间定位方法有着良好的发展前景。节点定位不但要考虑定位精度,而且要考虑定位成本和复杂度。目前国内外针对此问题提出了多种算法。例如胡伟等人提出了一种基于四面体质心迭代的三维近似三角形内点测试(approximate point-in-triangle test, APIT)定位算法^[1],该算法虽提高了节点覆盖率,减小了计算量,但是实验过程较为复杂,不容易实现;印第安大学的 Zhang 等提出了一种 Landscape-3D 定位算法^[2],此方法节点间无需通讯,减小了功耗,但其存在即时迭代过程计算量大,LA 装置在移动过程中要严格保持同步等缺点;Zhang 等^[3]利用粒子群优化(particle swarm optimization, PSO)算法进行节点定位,但此算法容易陷入局部最

优且计算比较复杂。此外还有多种测距和定位算法^[4]以及 DV-Hop 定位算法和改进的混合蛙跳算法^[5]。

本研究采用接收信号强度指标(received signal strength indicator, RSSI)方法实现测距,并且对所测数据进行统计加权处理来减小环境等因素造成的测量误差,建立起距离与信号强度之间精确的衰减模型,然后用萤火虫算法对未知节点进行定位,并且与传统的四边测量算法^[6]以及交叉粒子群算法^[7]和遗传算法^[8]进行对比,结果表明,萤火虫算法的定位精度明显高于四边测量算法、交叉粒子群算法和模拟退火算法,并且四点定位分八次测量再取均值的方法误差最小,定位精度最优。

1 RSSI 测距

1.1 RSSI 测距原理

无线信号传输过程中,信号强度会随着距离的增大而衰减。因此,接收信号强度指示(RSSI)测距原理是将信号强度的衰减转变成信号的传播距离,利用信号强度的衰减与传播距离之间的函数关系来

① 河北省自然科学基金(F2015203392)和秦皇岛市科技计划(201502A043)资助项目。
② 男,1978年生,博士,副教授;研究方向:惯性导航,三维立体重构及智能信号处理;联系人,E-mail: zlgtime@163.com
(收稿日期:2015-10-20)

估测传播距离^[9,10]。

通常信号衰减模型采用对数——常态分布模型来获得距离信息,但在实际测量时,通常将上述模型转换为阴影模型,如下式所示:

$$RSSI = -10n \times \lg d + A \tag{1}$$

式中, $RSSI$ 为接收信号强度,单位为 dBm; n 为衰减因子,其值与环境有关; d 为接收端与发射端之间的距离,单位为 m; A 为信号传播单位距离时接收到的信号强度,显然, A 与 n 的值决定了 $RSSI$ 与 d 的大小。

1.2 数据的统计加权

由于 RSSI 测量结果容易随环境变化而产生误差,因此有必要对测量数据进行加权处理以减小其误差^[11]。

本文提出了一种利用测量中值对数据进行统计加权处理的方法。设每组数据测量次数为 n , 然后将数据按照从小到大顺序排列,因此数据中值定义如下式所示:

$$M_{RSSI} = \begin{cases} RSSI\left(\frac{n}{2}\right), & n \text{ 为奇数} \\ \frac{1}{2}\left[RSSI\left(\frac{n}{2}\right) + RSSI\left(\frac{n}{2} + 1\right)\right], & n \text{ 为偶数} \end{cases} \tag{2}$$

M_{RSSI} 即为测量中值,再求出各个 $RSSI$ 值和测量中值之间的方差,如下式所示:

$$D_i = (RSSI_i - M_{RSSI})^2 \tag{3}$$

为了避免方差为零,进行加权系数处理, P_i 为加权系数,用下式表示:

$$P_i = \frac{1}{1 + D_i} \tag{4}$$

对加权系数进行求和处理, F_i 为归一化加权系数,用下式表示:

$$F_i = \frac{P_i}{\sum_{i=1}^n P_i} \tag{5}$$

由式(3)、(4)、(5)可以看出,测得的 $RSSI$ 值与测量中值 M_{RSSI} 相差越大,加权系数 F_i 的值越小,即所占比重越小;当 $RSSI$ 值与 M_{RSSI} 值相等时,加权系数 F_i 值最大,此时 $RSSI$ 值被赋予最大权重。

将测得的多组数据分别与对应的加权系数相乘,再求和,所得结果即为两节点间 RSSI 信号的衰

减值: $\sum_{i=1}^n F_i \times RSSI_i$ 。

此方法的优点在于以测量中值为基础计算权值,将包含粗大误差结果的 $RSSI$ 值赋予很小的权值,累加时可以忽略,这样既没有简单的删掉粗差数据,又能滤除一部分粗大信号,使得 RSSI 测距方法能够适应更加复杂的环境。

2 萤火虫算法

2.1 萤火虫算法简介

萤火虫算法 (firefly algorithm, FA) 是由剑桥学者 Yang 在 2009 年提出的一种基于生物群智能的随机优化算法^[12],随后应用在寻优问题上,并能够成功寻找到函数最优解。

2.2 萤火虫算法实现定位

萤火虫之间均可以相互吸引,且由亮度低的向亮度高的靠近,发光强度由目标函数决定^[13,14]。

设未知节点坐标和锚节点坐标分别为 (x, y, z) 和 (x_i, y_i, z_i) ,未知节点的估计坐标为 $(\hat{x}, \hat{y}, \hat{z})$,未知节点和锚节点之间的真实距离为

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \tag{6}$$

设未知节点与锚节点的测量距离为 $\hat{d}_i$,因此可得到目标函数

$$f(x, y, z) = \sum_{i=1}^n | \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - \hat{d}_i | \tag{7}$$

其中每一个萤火虫都是目标函数的一个解。当目标函数取最小值时,误差最小,定位精度最高,即位置最优,因此把定位问题转变为对目标函数求解优化问题。最终,大多数萤火虫都聚集在最优解附近。其中 n 为锚节点个数,得到的输出值即为未知节点估算坐标 $(\hat{x}, \hat{y}, \hat{z})$,因此定位误差为

$$\eta = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2 + (z - \hat{z})^2} \tag{8}$$

程序执行步骤如下:

步骤 1: 获取锚节点与未知节点间距离。

步骤 2: 初始化萤火虫种群,在指定范围内随机分布萤火虫。

步骤 3:计算目标函数值,目标函数越优,则自身亮度越高,萤火虫所处位置越好。

步骤 4:比较萤火虫之间的发光亮度,更新萤火虫位置产生新解,如果迭代次数到达设定值,跳至步骤 6。

步骤 5:计算目标函数值,跳至步骤 4。

步骤 6:输出结果,程序结束。

3 实验分析

3.1 RSSI 测距

测距实验采用 TI 公司生产的 CC2530 开发板来实现测量两节点之间的 RSSI 值。一般情况下,无线传感网络部署地区大多数在户外进行,因此本实验也选择在户外测取数据。实验一共测得 50m 内的 RSSI 值,由于距离较短时衰减变化比较明显,当距离较远,到达一定程度时,衰减变化不大,因此前 30m 每隔 1m 测量一次,后 20m 每隔 4m 测量一次。实验硬件和测量现场如图 1、图 2 所示。



图 1 zigbee CC2530 硬件开发板



图 2 测量现场图

实验时,每个位置测量 30 组数据,然后按照上述方法对数据进行统计加权处理,得到表 1 所示的数据。

表 1 距离-RSSI 值数据表					
距离(m)	RSSI (dBm)	距离 (m)	RSSI (dbm)	距离 (m)	RSSI (dBm)
1	-33.4	13	-58.4	25	-67.3
2	-38.5	14	-59.0	26	-68.8
3	-43.2	15	-60.5	27	-69.0
4	-45.3	16	-61.3	28	-69.4
5	-48.5	17	-61.4	29	-70.0
6	-50.0	18	-62.0	30	-70.5
7	-54.4	19	-63.2	34	-71.2
8	-55.3	20	-65.6	38	-71.6
9	-55.4	21	-65.0	42	-71.9
10	-56.8	22	-65.7	46	-72.6
11	-57.4	23	-66.4	50	-74.2
12	-57.6	24	-66.4	-	-

将所得数据带入式(1),利用最小二乘法求出 A 与 n 的值为: $A = -31.1949$, $n = 2.5552$,因此得到 RSSI 值与距离 d 之间的关系如下所示:

$$RSSI = -10 \times 2.5552 \times \lg d - 31.1949 \quad (9)$$

得到的原始数据和拟合曲线如图 3 所示。

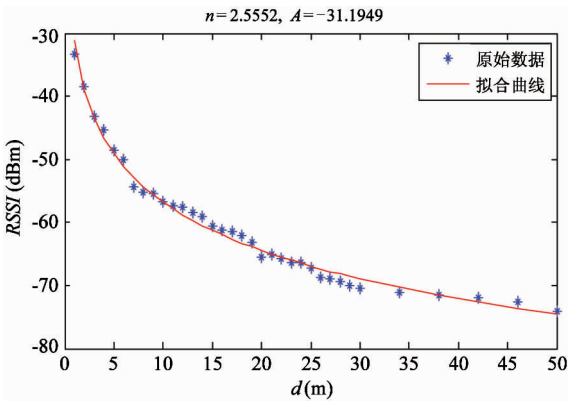


图 3 拟合曲线图

由图 3 拟合曲线图可以看出,得到的拟合曲线基本上和原数据点一致,并且拟合曲线斜率随着距离的增大而减小,说明 RSSI 值在距离较短阶段变化十分明显,而随着距离的增加,斜率变小,拟合曲线

趋于平缓, $RSSI$ 值也趋于稳定。图 4 是相对误差图,从图 4 可以看出,原始数据和得到的拟合曲线上的数据,误差基本上都在 5% 以内。因此采用上述统计中值加权的处理方法,能够有效地减小误差,适应环境变化。

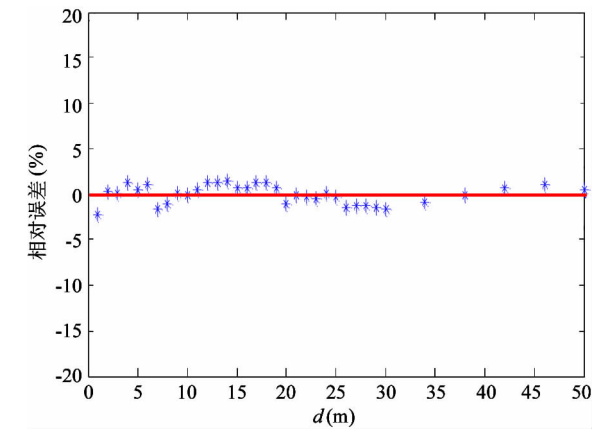


图 4 相对误差图

使用 zigbeeCC2530 节点进行测距实验。该方法简单,成本低廉,利用干电池能够工作半年以上,并且经过仿真验证,精度较高,因此,用此方法进行测距实验能够在较低成本的情况下,获得较高的精确度,是一种很不错的定位测距方法。

3.2 萤火虫算法定位

由于高度受限,太高的地方无法进行节点设置,因此本实验在 $30\text{m} \times 30\text{m} \times 3\text{m}$ 的长方体区域内进行,如图 5 所示。实验现场图如图 6、图 7 所示。

本实验一共设置 8 个锚节点,即 $A(0,0,0)$ 、 $B(0,0,3)$ 、 $C(30,0,0)$ 、 $D(30,0,3)$ 、 $E(0,30,0)$ 、 $F(0,30,3)$ 、 $G(30,30,0)$ 、 $H(30,30,3)$,分别位于

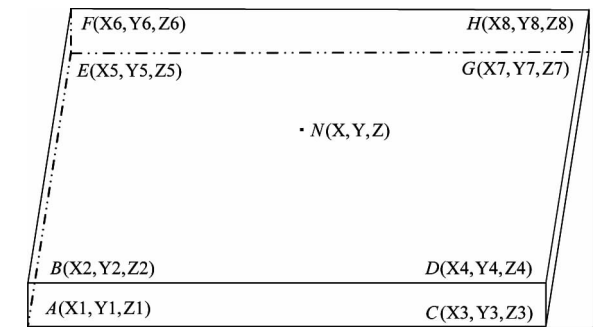


图 5 实验节点设置图



图 6 节点高度设置图



图 7 现场整体布局图

长方体区域的 8 个端点,未知节点 N 坐标为 $(15, 20, 2)$ 。通过测量 8 个锚节点到未知节点的 $RSSI$ 值得到测量距离 $\hat{d}$ 如表 2 所示。

表 2 定位数据表

锚节点	坐标(m)	$RSSI$ (dBm)	测量距离(m)
A	$(0,0,0)$	-67.4	26.12
B	$(0,0,3)$	-67.2	25.64
C	$(30,0,0)$	-66.6	24.32
D	$(30,0,3)$	-67.3	25.88
E	$(0,30,0)$	-63.1	17.74
F	$(0,30,3)$	-62.9	17.41
G	$(30,30,0)$	-63.3	18.03
H	$(30,30,3)$	-62.6	16.94

首先协调器节点与电脑相连,建立无线传感器网路,随后,作为终端发射信号的未知节点和作为路由器的 8 个锚节点打开,开始加入网络。未知节点发射信号,8 个锚节点接收信号,将自己接收到的信号强度传递给协调器节点并在电脑上显示^[15]。

本文先选取 A、E、F、H 4 个点(选取点的原则为 4 点不共面,且未知节点在 4 点所构成的区域内)对未知节点 N 进行定位,将萤火虫数量设置为 30,迭代次数为 200,用 MATLAB 仿真的效果如图 8 所示。

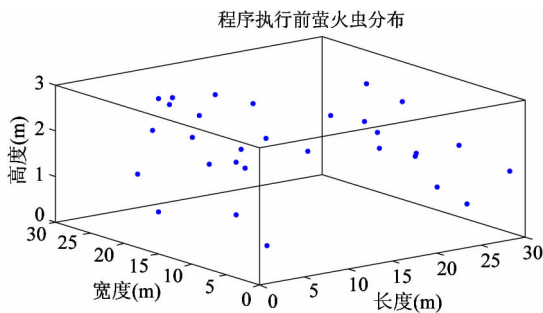


图 8 萤火虫初始分布图

如图 9 所示,大多数萤火虫基本上都聚集在未知节点坐标附近,程序输出结果为 $(\hat{x}, \hat{y}, \hat{z}) = (15.31, 21.13, 1.19)$, 根据式(8)得到定位误差为 $\eta = 1.42 \text{ m}$ 。

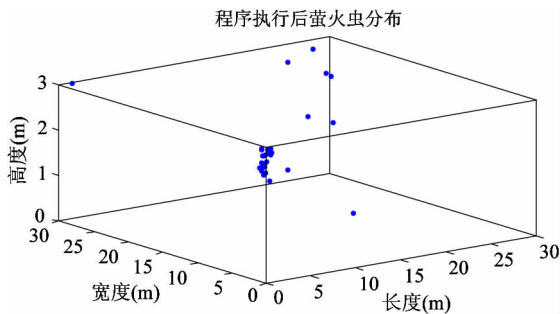


图 9 萤火虫最终分布图

接下来,用 8 个锚节点共同为未知节点定位,本次将萤火虫数量设置为 60,迭代次数设置为 300,仿真效果图如图 10 所示。

如图 11 所示,大部分萤火虫基本上都聚集在未知节点坐标附近,程序输出结果为 $(\hat{x}, \hat{y}, \hat{z}) = (15.12, 21.16, 2.40)$, 根据式(8)得到定位误差为 $\eta = 1.23 \text{ m}$ 。

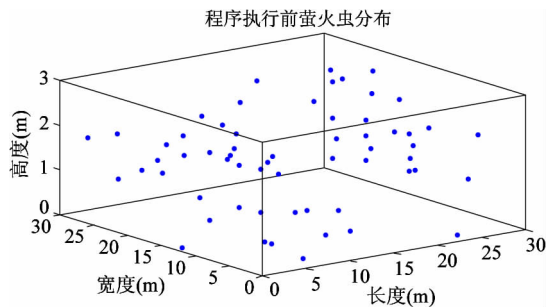


图 10 萤火虫初始分布图

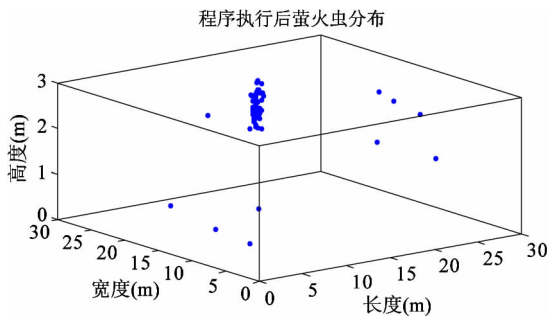


图 11 萤火虫最终分布图

长方体共有 8 个端点,选取其中 4 个不共面的端点对未知节点进行定位,从中任选 8 组包含未知节点区域进行定位,定位之后,取 8 组定位结果平均值。其中 8 组端点最终输出结果分别为 ABFH (15.03, 21.23, 2.39)、AEFH (15.31, 21.13, 1.19)、BFGH (14.83, 20.91, 2.25)、CDFH (15.45, 21.38, 2.04)、ABEH (15.17, 21.13, 2.40)、BEFH (14.98, 18.86, 2.40)、AFGH (15.03, 21.27, 2.02)、CDFG (14.72, 18.64, 2.35)。最后取平均值 $(\bar{\hat{x}}, \bar{\hat{y}}, \bar{\hat{z}}) = (15.07, 20.57, 2.13)$, 根据式(8)得到定位误差 $\eta = 0.59 \text{ m}$ 。

3.3 对比实验分析

依据上述仿真实验,当锚节点个数分别设置为 4,5,6,7,8 时(每个锚节点只利用一次,一起进行定位;当锚节点个数取 8 时,区别于上述 8 组四个节点平均值的方法),萤火虫定位算法与交叉粒子群算法和遗传算法相比较,如图 12 所示。

从图 12 可以看出,萤火虫算法相比于这两种智能算法有着更精确的定位精度,将 8 组 4 个节点萤火虫定位算法与传统的四边定位算法加入对比,得到的比较结果如表 3 所示。

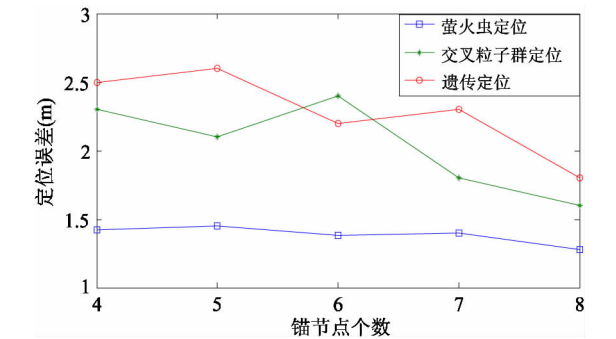


图 12 三种算法对比图

表 3 几种算法比较表

算法名称	锚节点数	定位坐标	定位误差(m)
四边测量定位法	4	(12.15,15.55,0.95)	3.09
交叉粒子群算法	8	(15.34,18.72,1.07)	1.62
遗传算法	8	(15.28,18.16,2.23)	1.88
单次四个节点萤火虫定位算法	4	(15.31,21.13,1.19)	1.42
八个节点萤火虫定位算法	8	(15.12,21.26,2.4)	1.23
八组四个节点萤火虫定位算法平均值	8	(15.07,20.57,2.13)	0.59

经比较,采用 8 组四个节点萤火虫定位算法平均值,不但充分利用上了全部 8 个锚节点,并且定位精度要远远高于传统的四边定位算法以及交叉粒子群算法和遗传算法的智能算法,相比于单次四个节点萤火虫定位方法和八个节点萤火虫定位方法,能够有效地将定位误差控制在 1 m 以内,因此效果最好。

4 结 论

本文主要提出了一种基于三维空间无线传感网络的萤火虫定位算法,打破了二维空间定位算法的局限性。该算法首先用 RSSI 方法对锚节点和未知节点之间进行测距,得到稳定的拟合曲线,然后利用加权算法处理误差,最后在此基础上利用测距结果使用萤火虫算法对未知节点进行定位。经过仿真验证,该算法相比于传统的四边定位算法以及交叉粒子群算法和遗传算法,在定位精度上有了很大的提

高,在 30m × 30m × 3m 的空间内定位误差仅为 0.59m,对于分布在三维空间上的无线传感网络节点的定位具有很好的理论和实际意义。

参考文献

[1] 胡伟,朱西平,文红等. 基于四面体质心迭代的三维 APIT 定位算法研究. 传感技术学报, 2013, 26(10): 1432-1436

[2] Zhang L, Zhou X, Cheng Q. Landscape-3D: a robust localization scheme for sensor networks over complex 3D terrains. *Local Computer Network*, 2006, 31st: 239-246

[3] Zhang Z, Cui H. Localization in 3D sensor networks using stochastic particle swarm optimization. *Wuhan University Journal of Natural Sciences*, 2012, 17(6): 544-548

[4] Dalce R, Val T, Bossche A. Comparison of indoor localization systems based on wireless communications. *Wireless Engineering and Technology*, 2011, (2):240-256

[5] 冯晨,张玲华. 改进混合蛙跳算法在无线传感网定位中的应用. 计量学报, 2013, 34(004): 360-365

[6] 文举,金建勋,袁海. 一种无线传感器网络四边测距定位算法. 传感器与微系统, 2008, 27(5): 108-110

[7] 王俊,李树强,刘刚. 无线传感器网络三维定位交叉粒子群算法. 农业机械学报, 2014, 45(5): 233-238

[8] 王金鑫,赖旭芝,吴敏. 基于遗传算法的三维无线传感器网络定位新算法. 高技术通讯, 2008, 18(6): 579-584

[9] 徐世武,王平. ZigBee 技术中基于 RSSI 测距的定位算法研究. 单片机与嵌入式系统应用, 2012, 12(1): 1-3

[10] Zhu X, Feng Y. RSSI-based algorithm for indoor localization. *Communications and Network*, 2013, 5(02): 37-42

[11] 陶金龙,康志伟,骆坚. 基于 RSSI 测距的加权概率定位算法. 电子测量与仪器学报, 2014, 28(10): 1123-1129

[12] Yang X S. Firefly algorithms for multimodal optimization. *Stochastic algorithms: foundations and applications*. Springer Berlin Heidelberg, 2009. 169-178

[13] 高伟明. 萤火虫算法的研究与应用:[硕士学位论文]. 兰州:兰州大学, 2013. 18-22

[14] 王吉权,王福林. 萤火虫算法的改进分析及应用. 计算机应用, 2014, 34(9): 2552-2556

- [15] 王瑛辉, 石理碧, 孙坚. 基于 ZigBee 技术的无线传感器网络在能源计量系统中的应用. 计量技术, 2013, 11: 30-32

Research on a firefly localization algorithm based on RSSI ranging for 3D wireless sensor networks

Zhang Liguang, Ma Peng, Jin Mei, Fan Meichang, Liu Yuhao

(Measurement Technology and Instrumentation Key laboratory of Hebei Province,
Yanshan University, Qinhuangdao 066004)

Abstract

The techniques of localizing the unknown nodes of 3D wireless sensor networks (WSN) were studied. Aiming at the problem of low localization accuracy of current localization algorithms for 3D wireless sensor networks, a firefly localization algorithm based on the ranging using received signal strength indication (RSSI) for 3D wireless sensor networks was proposed. The algorithm uses RSSI to measure the signal-propagating distance and models the attenuation of signal strength with the signal-propagating distance, and then, uses the firefly algorithm to find the optimal value of the objective function to locate unknown nodes under the condition that a certain number of anchor nodes are known. The simulation results showed that the localization accuracy of the new localization algorithm was greatly higher than current algorithms, and its localizing error in the space of $10\text{m} \times 10\text{m} \times 10\text{m}$ was just 0.59m.

Key words: wireless sensor network (WSN), three-dimensional, received signal strength indication (RSSI), firefly algorithm (FA)