

# 蝗虫群优化和极限学习机相结合的 RFID 室内定位算法

王 哲 郑嘉利 李 丽 袁 源 石 静

(广西大学计算机与电子信息学院 南宁 530004) (广西多媒体通信与网络技术重点实验室 南宁 530004)

**摘 要** 随着室内定位技术的飞速发展,射频识别(Radio Frequency Identification,RFID)技术以其非接触、快速识别等优点成为解决问题的首选方案。针对目前室内定位算法的精度容易受到标签密度和算法效率的影响及对动态环境适应性不足的问题,文中提出了一种蝗虫群优化(Grasshopper Optimization Algorithm,GOA)和极限学习机(Extreme Learning Machine,ELM)相结合的 RFID 室内定位算法。该算法通过蝗虫群优化对极限学习机随机产生的输入层权值和隐含层阈值进行选择,以提升极限学习机的性能,从而在离线阶段减少学习时间;利用蝗虫群算法对极限学习机参数进行优化,有效克服环境以及信号强度值变化对定位精度的影响。通过实验研究了影响算法性能的因素,并验证了算法的有效性。与 BP 神经网络算法(NN-Based)和非度量多维尺度算法(NMDS-RFID)相比,所提算法的定位平均误差分别降低了 22.32%和 20.06%,平均执行时间分别减少了 58.7%和 7.55%。仿真和实验结果表明,所提算法在获得更精确的定位结果的同时降低了时间成本,并对环境变化具有较好的适应性。

**关键词** 射频识别,室内定位,接受信号强度值,蝗虫群优化算法,极限学习机

中图分类号 TP301.6 文献标识码 A DOI 10.11896/jsjcx.181202381

## RFID Indoor Positioning Algorithm Combining Grasshopper Optimization Algorithm and Extreme Learning Machine

WANG Zhe ZHENG Jia-li LI Li YUAN Yuan SHI Jing

(School of Computer,Electronics and Information,Guangxi University,Nanning 530004,China)

(Guangxi Key Laboratory of Multimedia Communications and Network Technology,Nanning 530004,China)

**Abstract** With the rapid development of indoor positioning technology,radio frequency identification (RFID) technology has become the preferred solution due to its advantages of non-contact and rapid identification. However, the accuracy of existing RFID indoor positioning algorithms is easily affected by the tag density and algorithm efficiency, and environmental adaptation of existing algorithms is not strong enough. Therefore, this paper introduced an RFID indoor positioning algorithm based on the grasshopper optimization algorithm (GOA) fused with extreme learning machine (ELM). The algorithm is proposed to tune the input layer weight and hidden layer threshold biases randomly generated by the extreme learning machine, so that it can reduce learning time in the offline phase. At the same time, the algorithm can effectively resist the environmental interference and overcome the change of signal strength value on the positioning accuracy. Experiments are carried out to study the influence factors and validate the performance. Both the simulation and test experiment results show that compared with NN-based algorithm and NMDS-RFID algorithm, the average positioning error of the proposed algorithm is reduced by 22.32% and 20.06% respectively, and the average execution time is reduced by 58.7% and 7.55% respectively. GOA-ELM indoor positioning algorithm can achieve more accurate positioning results and has certain adaptability to the changes of the environment.

**Keywords** Radio frequency identification, Indoor positioning, Received signal strength indicator, Grasshopper optimization algorithm, Extreme learning machine

## 1 引言

近年来,随着基于物联网技术的各种智能设备的不断发

展,精确定位技术成为了人们研究和关注的焦点。在精确定位技术中,室内精确定位技术的应用需求不断增加,潜力巨大。而 RFID 以其通过射频信号进行非接触、快速、自动识别

收到日期:2018-12-13 返修日期:2019-04-26 本文受国家自然科学基金项目(61761004)资助。

王 哲(1992-),男,硕士生,主要研究方向为多媒体通信网络理论与技术;郑嘉利(1979-),男,教授,主要研究方向为多媒体通信、物联网技术,E-mail:zjl@gxu.edu.cn (通信作者);李 丽(1994-),女,硕士生,主要研究方向为多媒体通信网络理论与技术;袁 源(1995-),女,硕士生,主要研究方向为多媒体通信网络理论与技术;石 静(1992-),女,硕士生,主要研究方向为多媒体通信网络理论与技术。

的优点成为室内精确定位问题的首选解决方案<sup>[1]</sup>。

现有的 RFID 室内定位方法主要有基于测距的定位方法和基于非测距的定位方法两类<sup>[2]</sup>。其中,基于测距的定位方法主要有到达时间 TOA<sup>[3]</sup>、到达时间差 TDOA<sup>[4]</sup>、到达角 AOA<sup>[5]</sup>等。基于非测距的定位方法主要是基于接受信号强度值 RSSI<sup>[6]</sup>,该方法又分为距离路径衰减模型法<sup>[7]</sup>和指纹匹配法<sup>[8]</sup>。室内指纹匹配定位技术以其易部署、高精度、非视距等优点备受关注<sup>[9]</sup>。Gao 等<sup>[10]</sup>提出一种新的非度量多维尺度 RFID 算法,它结合了 NMDS 算法和指纹定位算法,实现了室内 RFID 多标签的协同定位。Kung 等<sup>[11]</sup>提出了一种 NN-Based 算法,采用 BP 神经网络对 Landmarc 算法进行改进,在线阶段计算标签坐标并将其作为神经网络的输入,以增强定位效果。Dwiyasa 等<sup>[12]</sup>提出了一种以 RSSI 值作为输入,将 ELM 运用到位置指纹定位的 RFID 算法,其凭借 ELM 的随机特征映射和紧密的网络结构获得了较快的学习速度和优良的泛化性能,使定位更加快速、精确。

然而,对于 ELM 室内定位算法而言,由于其输入层神经元和隐含层神经元的权值  $w$  及隐含层神经元阈值  $b$  是在构建定位模型时随机选择的,因此权值  $w$  和阈值  $b$  不会在训练过程中得到纠正。研究表明,采用不同形式的优化算法对 ELM 的输入权值和阈值进行优化,可以得到更精确、更稳定的定位模型。文献[13]采用标准差分进化算法选择并优化 ELM 神经元和隐含层神经元的权值  $w$  及隐含层神经元阈值  $b$ 。文献[14]使用各种变异策略来改进标准差分进化算法,以获得更优良的模型。文献[15]的提出基于半监督的极限学习机减少了带标记的训练数据,降低了训练成本,但是构建模型过程中数据样本的重复训练会降低学习速度并增加累计误差。文献[16]在优化过程中考虑了 ELM 模型的复杂性,利用 PSO 算法<sup>[17]</sup>优化了 ELM 的输入权值和隐含层的偏差。

本文提出了一种基于 GOA<sup>[18]</sup>和 ELM 的 RFID 室内定位算法,称为 GOA-ELM 算法。首先,采用改进的高斯滤波算法对 RFID 参考标签的信号强度值样本数据进行预处理。然后,利用 GOA 算法对输入层神经元和隐含层神经元的权值  $w$  及隐含层神经元阈值  $b$  进行优化,并结合带有位置信息的参考标签样本数据对初始 ELM 定位模型进行训练。最后,处理用户的定位请求。该算法能在不同的条件下快速获得最优的输入层神经元和隐含层神经元的权值  $w$ ,以及隐含层神经元阈值  $b$ ;同时,优化后的 ELM 凭借自身的随机特征映射和紧密的网络结构,可以获得极快的学习速度,从而减少离线学习时间,并且能够有效克服环境变化以及 RSSI 的时变性对定位精度的影响。

## 2 GOA-ELM 室内定位算法

面对大量含位置信息的参考标签数据时,在 ELM 算法的离线学习阶段,输入层与隐含层之间的连接权值  $w$  以及隐含层神经元阈值  $b$  随机生成。由于有个别隐含层节点的作用非常小,最终会产生定位精度低和对动态环境适应性不足等问题。如需改善上述问题,ELM 需要设置大量隐含层节点以达到理想精度,这会大幅降低 ELM 算法的性能。

本文针对以上问题,用蝗虫群算法来优化极限学习机的

连接权值  $w$  和阈值  $b$ ,提出了一种蝗虫群优化与极限学习机相结合的算法,称为 GOA-ELM 算法。与其他优化算法相比,蝗虫群优化的最大特点是:它只有一个位置矢量,所有目标都将参与每一个目标的位置更新。这样不仅可以在最大程度上避免算法陷入局部最优解和过快收敛,还可以针对 ELM 的参数找到更好的全局最优解,从而使得 ELM 定位模型获得更好的定位精度,提升其对动态环境的适应性。

### 2.1 参考标签信号强度值的预处理

使用改进的高斯滤波算法对带有位置标记的参考标签数据进行预处理。

1)第  $i$  个阅读器对同一参考标签的信号强度重复读取  $K$  次,并将第  $k$  次读取到的信号强度记录为  $RSSI_{ik}$  ( $i=1,2,\dots,M$ )。

2)计算 RSSI 信号强度值的方差  $\delta^2$ :

$$\delta^2 = \frac{1}{K-1} \sum_{k=1}^K (RSSI_{ik} - \bar{A})^2 \quad (1)$$

其中,  $\bar{A} = \frac{1}{K} \sum_{k=1}^K RSSI_{ik}$ 。

3)对于第  $i$  个阅读器对同一标签读到的第  $k$  次信号强度值  $RSSI_{ik}$ ,若  $|RSSI_{ik} - \bar{A}| > 3\delta$ ,则删除相对应的样本数据。处理完成后,可得到一个大小为  $m$  的 RSSI 集合,计算 RSSI 集合的平均值,并将其记为平均信号强度。

$$\overline{RSSI} = \frac{1}{m} \sum_{k=1}^m RSSI_{ik} \quad (2)$$

4)对于得到的  $\overline{RSSI}$ ,采用对数-正态路径损耗模型来模拟信号在室内环境中的衰减变化。RFID 的信号强度值衰减变化可表示为:

$$RSSI(d) = RSSI(d_0) - 10n \log_{10} \frac{d}{d_0} + X_\delta \quad (3)$$

其中,  $d$  是参考标签与阅读器之间的距离,  $d_0$  是参考距离(一般取为 1m),  $RSSI(d_0)$  是近距离时的参考路径损耗,  $n$  是路径损耗因子。本文设定  $n=2, \delta=2, RSSI(d_0) = -45$  dBm。根据式(3)可以对  $\overline{RSSI}$  进行修正,以提高 RFID 参考标签样本集的准确性。

5)对定位区域内(含有  $M$  个阅读器和  $N$  个参考标签)收集到的参考标签信号强度值进行上述预处理,会得到一个大小为  $N \times M$  的 RSSI 训练样本集  $RSSI_{NM}$  和一个对应的大小为  $N \times 2$  的参考标签位置标记集  $P$  (其中包含参考标签的位置信息)。

### 2.2 蝗虫群算法优化极限学习机参数

在利用 GOA 为 ELM 选择参数最优解的过程中,需要根据 ELM 的特性,设计集群优化算法中解空间内的食物源和个体适应度评价函数,以获取更准确的全局最优解,也就是获取 ELM 所需的输入层神经元和隐含层神经元的权值  $w_{i,j}$  及隐含层神经元阈值  $b_j$ 。

1)设计解空间内的食物源

在所有解空间内,每一个食物源都包含以下信息:

$$[w_{i,j}, b_j]^T, i=1,2,\dots,N; j=1,2,\dots,L \quad (4)$$

根据 ELM 定位算法,每个食物源的信息随机生成。  $w_{i,j}$  为连接第  $j$  个隐含层神经元和第  $i$  个输入层节点的权值,取值为  $[-1,1]$  之间的随机数;  $b_j$  为第  $j$  个隐含层神经元的阈

值,取值为 $[0,1]$ 之间的随机数; $N$ 和 $L$ 分别为输入层节点数(即参考标签样本个数)和隐含层神经元数。

## 2)设计个体适应度评价函数

采用均方根误差来反映预测结果的精确性:

$$R = \sqrt{\frac{\sum_{i=1}^N [(x_j - x_j')^2 + (y_j - y_j')^2]}{N}} \quad (5)$$

其中, $x_j, y_j$ 为参考标签的真实位置值; $x_j', y_j'$ 为定位预测值; $N$ 为输入层节点数(即参考标签样本的个数)。均方根误差较小的解对应较大的适应度,因此取均方根误差的倒数作为适应度值:

$$fitness = \frac{1}{R} \quad (6)$$

蝗虫群算法选择全局最优解的具体步骤如下。

1)在解空间内随机产生 $K$ 个食物源,初始化蝗虫群个体 $X_i (i=1,2,\dots,K)$ ,将每一个蝗虫个体随机分布在解空间内的所有食物源位置上;设置 $T_{\max} (T_{\max} > 1)$ 为最大迭代次数, $t=1$ 为当前迭代次数。

2)根据每个蝗虫个体所在的食物源,获取该食物源的信息,即输入层神经元和隐含层神经元的权值 $w_{i,j}$ 及隐含层神经元阈值 $b_j$ 。结合预处理之后的RSSI训练样本集 $RSSI_{NM}$ 和参考标签的位置标记集 $P$ ,在ELM模型中计算出定位预测值 $(x_j', y_j')$ 。

3)根据式(5)、式(6)计算食物源的适应度,并设置 $\hat{T}_{fitness}$ 为当前最佳适应度食物源。

4)查看当前迭代次数 $t$ 是否大于最大迭代次数 $T_{\max}$ 。若未达到,重复执行步骤5)直至达到最大迭代次数;若迭代次数到达设定 $T_{\max}$ ,从最佳适应度食物源 $\hat{T}_{fitness}$ 中提取出ELM所需的连接权值 $w_{i,j}$ 和阈值 $b_j$ 。

5)根据食物源对蝗虫个体进行位置更新:

$$X_i = S_i + G_i + A_i \quad (7)$$

其中, $X_i$ 为蝗虫群中第 $i$ 只蝗虫的位置, $S_i$ 为第 $i$ 只蝗虫受到种群的作用力, $G_i$ 为第 $i$ 只蝗虫受到的重力影响, $A_i$ 为第 $i$ 只蝗虫受到的风力水平影响。由于此数学模型无法直接用于优化本文的ELM定位模型,因此这里不考虑重力因素(无 $G_i$ 分量),并且假定蝗虫的移动方向总是与风向相同(无 $A_i$ 分量)。

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^K s(d_{ij}) \hat{d}_{ij} \quad (8)$$

其中, $d_{ij} = |x_j - x_i|$ 表示第 $i$ 只蝗虫与第 $j$ 只蝗虫的距离。 $\hat{d}_{ij} = \frac{x_j - x_i}{d_{ij}}$ 表示第 $i$ 只蝗虫到第 $j$ 只蝗虫的单位向量。 $s$ 函数定义了蝗虫受到种群作用力的影响,其表达式如下:

$$s(r) = \frac{1}{2} e^{-\frac{3}{2}r} - e^{-r} \quad (9)$$

为了更好地获得全局最优解,使整个种群收敛到唯一食物源,参考粒子群算法中的惯性权值,在位置更新公式中加入自适应系数 $c = 1 - t \frac{1}{T_{\max}}$ ,它平衡了整个种群在解空间对食物源的搜算能力。因此,蝗虫的个体位置更新函数可被替换为:

$$X_i^d = c \left\{ \sum_{\substack{k=1 \\ k \neq i}}^K \frac{u_d - l_d}{2} s(|x_j^d - x_i^d|) \frac{x_j^d - x_i^d}{d_{ij}} \right\} + \hat{T}_{fitness} \quad (10)$$

其中, $u_d$ 和 $l_d$ 表示第 $k$ 只蝗虫第 $d$ 维变量的上界和下界; $\hat{T}_{fitness}$ 为蝗虫群移动的目标食物源,也就是上一次迭代计算出的最佳适应度食物源。

将更新位置后的每个蝗虫个体彼此之间的距离限制在 $[1,4]$ 范围,并重新计算更新位置后蝗虫个体所在食物源的适应度,更新 $\hat{T}_{fitness}$ ,迭代 $t = t + 1$ 次。

在式(9)中,当 $s(r) = 0$ 时,蝗虫个体之前的互相作用力为0,此时距离 $r$ 为舒适距离。当 $s(r) > 0$ 时,蝗虫个体间的吸引力会大于排斥力,个体之前相互吸引,此时 $r$ 的大致取值范围 $[2,10]$ 为吸引区。当 $s(r) < 0$ 时,蝗虫个体间的吸引力会小于排斥力,个体之前相互排斥,此时 $r$ 的大致取值范围 $[0,2]$ 为排斥区。当距离大于10时,此函数的返回值接近于零。因此,在式(9)中, $s(r)$ 不能在蝗虫个体之间设置过大的距离,规定蝗虫之间的距离在 $[1,4]$ 之内。

## 2.3 构建ELM定位模型

对预处理之后获得的训练样本集 $RSSI_{NM}$ 和位置标记集 $P$ 进行归一化处理,得到RSSI训练样本集 $R_i$ 和 $P_i (i=1,2,\dots,N)$ 。根据 $R_i$ 和 $P_i$ ,具有 $L$ 个隐含层神经元的ELM定位模型可表示为:

$$t_i = \sum_{j=1}^L \beta_j g(w_{i,j} \cdot R_i + b_j), i=1,2,3,\dots,N \quad (11)$$

其中, $w_{i,j}$ 和 $b_j$ 由上述蝗虫群优化得到; $g(x)$ 为隐含层神经元的激活函数; $g(w_{i,j} \cdot R_i + b_j)$ 为第 $i$ 个参考标签样本在隐含层第 $j$ 个神经元的输出; $w_{i,j} \cdot R_i$ 为向量的内积; $\beta_j$ 为隐含层第 $j$ 个神经元和输出层之间的连接权值。在以 $g(x)$ 为激活函数的神经网络中,定位预测值 $t_i$ 能以零误差接近标签的真实位置 $P_i$ ,即:

$$\sum_{i=1}^N \|P_i - t_i\| = 0 \quad (12)$$

则 $t_i = \sum_{j=1}^L \beta_j g(w_{i,j} \cdot R_i + b_j)$ 可以表述为: $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$ 。其中, $\mathbf{H}$ 为隐含层的输出矩阵, $\boldsymbol{\beta}$ 为输出权重, $\mathbf{T}$ 为期望输出。

$$\mathbf{H}(\omega_1, \dots, \omega_L, b_1, \dots, b_L, R_1, \dots, R_N) = \begin{bmatrix} g(\omega_1 \cdot R_1 + b_1) & \dots & g(\omega_L \cdot R_1 + b_L) \\ \vdots & & \vdots \\ g(\omega_1 \cdot R_N + b_1) & \dots & g(\omega_L \cdot R_N + b_L) \end{bmatrix}_{N \times L} \quad (13)$$

其中, $\mathbf{H} = \{h_{ij}\}, h_{ij} = g(\omega_{i,j} \cdot R_i + b_j)$ 。根据 $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$ ,通过求其最小二乘解来得到 $\hat{\boldsymbol{\beta}}$ :

$$\min_{\boldsymbol{\beta}} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| \quad (14)$$

其解为: $\hat{\boldsymbol{\beta}} = \mathbf{H}^+ \mathbf{T}$ , $\mathbf{H}^+$ 为隐含层输出矩阵的广义逆。根据求解出的 $\hat{\boldsymbol{\beta}}$ ,即可完成对ELM定位模型的构建。

在线预测阶段:当待测目标标签进入具有 $M$ 个阅读器的定位区域时,收集所有阅读器收到的信号强度值并对信号强度值进行预处理,从而得到一个目标标签的信号强度值集合 $T_i (i=1,2,\dots,M)$ 。对 $T_i$ 进行归一化处理,得到该标签的处理值 $T$ ,将其输入已经构建成功的ELM定位模型之中,根据式(11)即可得出 $t$ 。对 $t$ 进行反归一化处理,即可得到该待测目标标签的具体位置坐标 $(x, y)$ 。

GOA-ELM 室内定位算法的流程如图 1 所示。

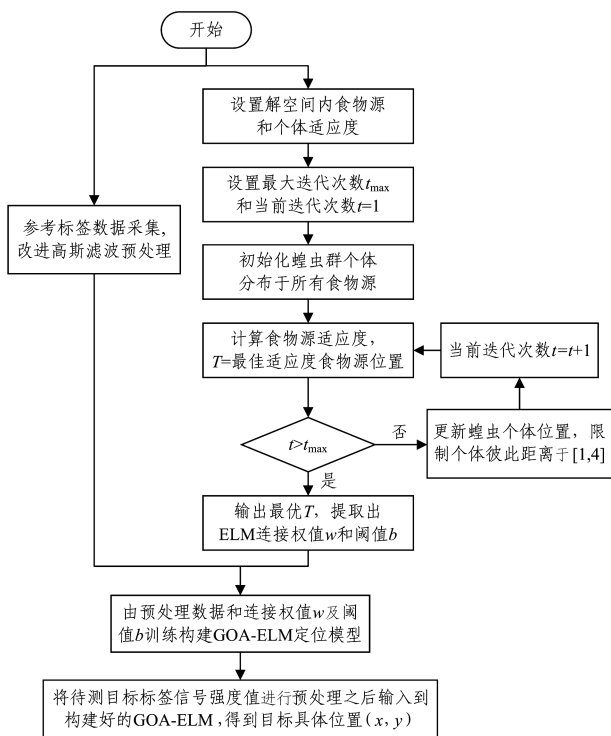


图 1 GOA-ELM 室内定位算法的流程图

Fig.1 Flowchart GOA-ELM indoor positioning algorithm

### 3 仿真测试与实验结果分析

本文使用 Matlab2014a 进行仿真实验。定位指纹库的覆盖面积为  $8\text{m} \times 8\text{m}$ , RFID 阅读器放置在角落和边缘, 间隔不小于  $2.5\text{m}$ , RFID 参考标签分布在该区域, 具体布置方式如图 2 所示。标签数量会根据标签密度的不同进行调整。仿真实验中所选取的 RFID 标签为抗金属 UHF 超高频标签, 其在阅读器天线增益为  $30\text{dBm}$  时的有效读写距离为  $8\text{m}$ , 通信协议为 EPC Class1Gen2(ISO18000-6C)。仿真实验的目的是分析影响因素, 比较不同条件下的定位效果。

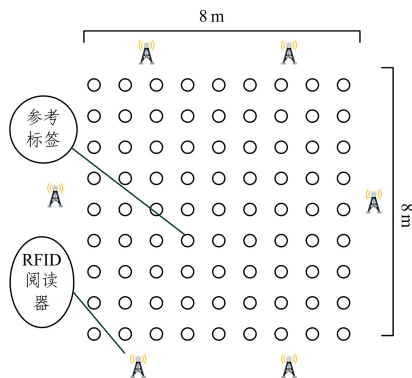


图 2 仿真实验布局图

Fig.2 Layout of simulation experiment

在仿真实验的离线训练阶段, 将 GOA-ELM 模型的隐含层神经元个数设为  $L=100$ , 将激活函数  $g(x)$  设为  $Sig$  函数, 将蝗虫群优化的迭代次数设为 100。将样本随机分成两个子集, 一个子集作为测试集, 另一个子集作为训练集。

### 3.1 仿真实验的影响因素

本节主要探讨参考标签的密度 ( $\Delta$ )、参考阅读器的数量 ( $N$ )、带位置信息的 RFID 参考标签数据的预处理结果对室内定位结果的影响。下文将分析每个因素对定位精度的影响。

#### 3.1.1 标签密度

本实验中, 将离线阶段的阅读器数设为  $N=6$ , 布置方式如图 2 所示; 将标签密度(参考标签的相互距离)设为  $2.0\text{m}$ ,  $1.8\text{m}$ ,  $1.6\text{m}$ ,  $1.5\text{m}$ ,  $1.2\text{m}$ ,  $1.0\text{m}$  和  $0.8\text{m}$ 。在在线预测阶段, 在定位区域共单独测试 500 个标签。不同标签密度下的参考标签数量如表 1 所列。

表 1 不同密度下的参考标签数量

Table 1 Number of reference labels under different densities

标签密度/m	数量/个
2.0	16
1.8	20
1.6	28
1.5	36
1.2	64
1.0	100
0.8	150

如图 3 所示, 当参考标签的密度从  $2.0\text{m}$  变化到  $1.8\text{m}$ , 以及从  $1.8\text{m}$  变化到  $1.5\text{m}$  时, 定位误差显著减小; 但当参考标签的密度从  $1.5\text{m}$  变化到  $0.8\text{m}$  时, 定位误差变化不大。这是因为单位面积内的参考标签数量随着参考标签密度的增加而增加, 从而减小了定位误差, 并且获得了更精确的定位结果; 但当参考标签达到相对较高的密度 ( $0.8\text{m}$ ) 时, 其比低密度情况 ( $1.5\text{m}$ ) 包含了更多的冗余信息, 不会大幅增加定位精确度, 并且参考标签的数量增加还会增大标签碰撞的概率, 导致 RSSI 值波动较大或读取不到标签。

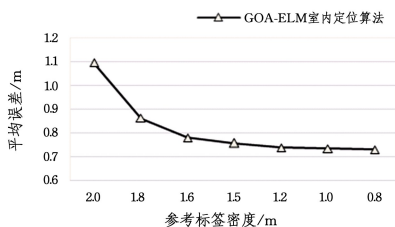


图 3 参考标签密度对本文定位算法的影响

Fig.3 Influence of density of reference tags on localization algorithm in this paper

#### 3.1.2 RFID 阅读器数量

本实验中, 设定参考标签的密度为  $\Delta=1.5\text{m}$ , 数量为 36 个。在离线训练阶段, 将阅读器放置在定位区域的角落和边缘, 阅读器的数量从 4 个依次递增至 10 个, 阅读器间隔控制在  $2\text{m}$  以上。在在线预测阶段, 在定位区域共单独测试 500 个标签。

如图 4 所示, 随着阅读器数量的增加, 算法的定位误差不断减少。在阅读器的数量从 4 个增加到 8 个的过程中, 定位精度得到显著提高。然而, 当数量增加到一定程度 (10 个) 后, 阅读器会提供大量重复的定位标签数据, 对定位误差的影响很小, 而阅读器数量的增加将显著增加整个定位系统的成本, 并且会增加阅读器碰撞的概率, 导致获取的标签的信号强度值不准确。

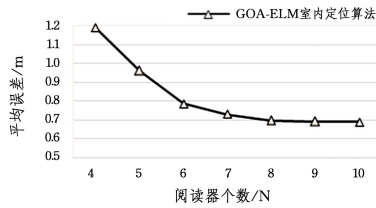


图4 阅读器数量对本文定位算法的影响

Fig. 4 Influence of number of readers on positioning algorithm in this paper

### 3.1.3 预处理的影响

本实验研究了改进的高斯滤波预处理算法对定位精度的影响。在离线训练阶段,将阅读器的数量设置为  $N=6$ ,布置方式如图 2 所示;设置参考标签密度为  $\Delta=1.5\text{m}$ ,数量为 36。在在线预测阶段,在定位区域共单独测试 500 个标签。

如表 2 所列,GOA-ELM 算法的定位精度通过预处理过程得到了明显改善,平均误差由无预处理的 0.8635 m 下降到 0.7745 m,标准差从 0.9945 m 下降到 0.7987 m。

表2 数据预处理对本文定位算法的影响

Table 2 Influence of preprocessing on localization algorithm in this paper

是否预处理	最小误差	最大误差	平均误差	标准差
无预处理	0.2358	2.4212	0.8635	0.9945
预处理	0.2196	1.9865	0.7745	0.7987

(单位:m)

### 3.2 与其他算法的比较

根据 3.1 节的实验结果,在离线训练阶段将实验条件设置为:参考标签的密度  $\Delta=1.5\text{m}$ ,数量为 36,阅读器数目  $N=6$ 。在在线预测阶段,在定位区域共测试 500 个标签。实验面积为  $8\text{m}\times 8\text{m}$ ,RFID 阅读器放置在角落和边缘,最小间隔为 2.5m,RFID 参考标签分布在该区域,布置方式如图 2 所示。

本节将所提出的 GOA-ELM 算法与 NN-Based 算法、NMDS-RFID 算法在相同的仿真空间中进行实验,以比较 3 种算法的定位性能。实验中 3 种算法所需的样本集相同,实验数据均经过 2.1 节的参考标签信号强度值预处理之后得到。表 3 列出了各种算法的参数设置。

表3 各算法的参数设置

Table 3 Parameters setting for each method

算法	参数设置
GOA-ELM	激活函数 sigmoid, $L=100$ , 迭代次数=100
NN-Based	激活函数 sigmoid, $L=100$ , $K=4$
NMDS-RFID	最佳阈值 $\epsilon=10^{-4}$ , 迭代次数=100

对 500 个测试标签的定位误差进行统计,以 0.2 m 作为采样间隔,分别统计 3 种算法的误差落在每个误差区间内的个数(频数),结果如图 5 所示。可以看出,本文提出的 GOA-ELM 算法在定位准确度上优于 NN-Based 算法和 NMDS-RFID 算法。实验结果显示,NN-Based 算法的定位误差小于 1m 的概率为 53.4%,NMDS-RFID 算法为 57.5%,本文算法为 64.4%。相比 NN-Based 和 NMDS-RFID 算法,GOA-ELM 算法的定位精度分别提升了 11%和 6.9%。

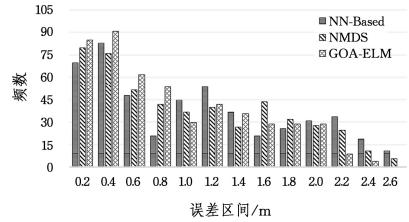


图5 误差区间的频数

Fig. 5 Frequency of error interval

结合表 4 和图 5 可以看出,GOA-ELM 算法在整体平均误差和稳定性方面均优于其他算法。与 NN-Based 算法和 NMDS-RFID 算法相比,GOA-ELM 算法的整体平均误差分别降低了 22.32%和 20.06%。此外,实验结果显示,本文算法的最大误差为 2.3686 m,这是因为当目标标签落入边缘和角落时,参考标签的减少将导致误差增加。

表4 仿真环境下 3 种算法的比较结果

Table 4 Comparison results under simulation environment

(单位:m)

算法	最小误差	最大误差	平均误差	标准差
GOA-ELM	0.1147	2.3686	0.6658	0.7356
NN-Based	0.1328	2.5186	0.8572	0.8689
NMDS	0.1305	2.5687	0.8329	0.7987

各算法的平均执行时间如表 5 所列,可以看出,GOA-ELM 算法的执行速度更快。GOA-ELM 的平均执行时间为 32.8601 s,这得益于 GOA 蝗虫群算法通过迭代次数对 ELM 参数进行了优化。GOA-ELM 的平均执行时间比 NN-Based 算法和 NMDS-RFID 算法分别减少了 58.7%和 7.55%。

表5 仿真环境下各算法的平均执行时间

Table 5 Average execution time of each algorithm under simulation environment

(单位:s)

算法	平均执行时间
GOA-ELM	32.8601
NN-Based	79.5638
NMDS	35.5434

**结束语** 本文提出了一种 RFID 室内定位算法,称为 GOA-ELM 算法,它是一种基于蝗虫群优化和极限学习机的算法。针对现有 RFID 室内定位算法存在的标签需求大、算法效率低、环境适应性差等问题,本文提出的算法通过蝗虫群优化算法对极限学习机随机产生的输入层权值和隐含层阈值选择全局最优解来提升极限学习机的性能,能获得更精确的定位结果。实验结果表明,与 NN-Based 算法和 NMDS-RFID 算法相比,本文算法的定位平均误差分别降低了 22.32%和 20.06%,平均执行时间分别减少了 58.7%和 7.55%。实验结果证明,该算法在室内定位精确度和定位时间方面具有较大的优势。

考虑到 RFID 标签之间、阅读器之间的冲突干扰会导致 RFID 室内定位系统性能的下降,我们在后续研究工作中将加入快速有效的“多阅读器-多标签”防碰撞算法来改善这一问题;同时,我们还将分析环境和天线位置对定位准确度的影

响,加入半监督和在线增量学习机制来减少参考标签的数量,从而进一步提高算法对真实定位环境变化的适应性。

### 参 考 文 献

- [1] POTGANTWAR A D, WADHAI V M. Improved Indoor Positioning Using RSS and Directional Antenna Integrating with RFID and Wireless Technology[C]// Proceedings of International Conference on ICT for Sustainable Development. Singapore: Springer, 2016: 319-328.
- [2] SHEN G H, MA Y T, LIU K H, et al. An indoor RFID tag localization algorithm in NLOS environment[J]. Computer Engineering and Science, 2016, 38(3): 454-459. (in Chinese)  
沈郭浩, 马永涛, 刘开华, 等. 非视距环境下室内 RFID 标签定位算法研究[J]. 计算机工程与科学, 2016, 38(3): 454-459.
- [3] TOMIC S, BEKO M. Exact Robust Solution to TW-ToA-Based Target Localization Problem With Clock Imperfections [J]. IEEE Signal Processing Letters, 2018, 25(4): 531-535.
- [4] AI Z, LIU Y. Research on the TDOA Measurement of Active RFID Real Time Location System [C]// IEEE International Conference on Computer Science & Information Technology. IEEE, 2010: 410-412.
- [5] AZZOUZI S, CREMER M, DETTMAR U, et al. New measurement results for the localization of UHF RFID transponders using an Angle of Arrival (AoA) approach[C]// IEEE International Conference on Rfid. IEEE, 2011: 91-97.
- [6] SONG X, LI X, TANG W, et al. A hybrid positioning strategy for vehicles in a tunnel based on RFID and in-vehicle sensors [J]. Sensors, 2014, 14(12): 23095-23118.
- [7] XU K, LIU H L. Characteristics of path loss for wireless sensor networks in indoor environments[J]. Transducer and Microsystem Technologies, 2016, 35(12): 11-13. (in Chinese)  
徐琨, 刘宏立. 室内环境下无线传感网络路径衰减特性[J]. 传感器与微系统, 2016, 35(12): 11-13.
- [8] ZHANG X Q, FANG F, YING K Z, et al. Design and Implement of an Interpolation Indoor Fingerprint-Based Localization System[J]. Chinese Journal of Sensors and Actuators, 2017, 30(4): 596-602. (in Chinese)  
章晓强, 方飞, 应可珍, 等. 一种基于插值的室内指纹定位系统设计与实现[J]. 传感技术学报, 2017, 30(4): 596-602.
- [9] TANG Y, BAI Y, MA Y, et al. Research of WiFi-based Fingerprinting Matching Algorithm in Indoor Positioning[J]. Computer Science, 2016, 43(5): 73-75. (in Chinese)  
唐洋, 白勇, 马跃, 等. 基于 WiFi 的指纹匹配算法在室内定位中的应用研究[J]. 计算机科学, 2016, 43(5): 73-75.
- [10] GAO Z, MA Y, LIU K, et al. An Indoor Multi-tag Cooperative Localization Algorithm Based on NMDS for RFID [J]. IEEE Sensors Journal, 2017, 17(7): 2120-2128.
- [11] KUNG H Y, CHAISIT S, PHUONG N T M. Optimization of an RFID location identification scheme based on the neural network [J]. International Journal of Communication Systems, 2015, 28(4): 625-644.
- [12] DWIYASA F, LIM M H. Extreme Learning Machine for Active RFID Location Classification[C]// Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems. Springer International Publishing, 2015: 657-670.
- [13] ZHU Q Y, QIN A K, SUGANTHAN P N, et al. Evolutionary extreme learning machine [J]. Pattern Recognition, 2005, 38(10): 1759-1763.
- [14] CAO J, LIN Z, HUANG G B. Self-Adaptive Evolutionary Extreme Learning Machine [J]. Neural Processing Letters, 2012, 36(3): 285-305.
- [15] ABDELGAYED T S, MORSI W G, SIDHU T S. Fault Detection and Classification based on Co-Training of Semi-Supervised Machine Learning [J]. IEEE Transactions on Industrial Electronics, 2017, PP(99): 1-1.
- [16] HAN F, YAO H F, LING Q H. An improved evolutionary extreme learning machine based on particle swarm optimization [J]. Neurocomputing, 2013, 116(1): 87-93.
- [17] DONG H B, LI D M, ZHANG X P. Particle Swarm Optimization Algorithm with Dynamically Adjusting Inertia Weight [J]. Computer Science, 2018, 45(2): 98-102. (in Chinese)  
董红斌, 李冬锦, 张小平. 一种动态调整惯性权重的粒子群优化算法[J]. 计算机科学, 2018, 45(2): 98-102.
- [18] SAREMI S, MIRJALILI S, LEWIS A. Grasshopper Optimisation Algorithm: Theory and application [J]. Advances in Engineering Software, 2017, 105(4): 30-47.