

基于自适应蝙蝠算法的室内 RFID 定位算法

谢良波, 李宇洋, 王勇, 周牧, 聂伟
(重庆邮电大学通信与信息工程学院, 重庆 400065)

摘要: 针对传统超高频射频识别室内定位算法中利用几何方法定位耗时较长且定位精度较差的问题, 提出了一种基于自适应蝙蝠算法 (ABA) 的室内 RFID 定位算法。首先, 利用跳频技术获取多频率相位, 基于多重信号分类 (MUSIC) 算法的角度信息和聚类的距离信息建立蝙蝠算法位置评估函数; 其次, 利用 Tent 反向学习初始化蝙蝠位置增加种群的多样性, 并引入自适应权重因子更新蝙蝠位置; 最后, 基于位置评估函数对目标位置进行迭代搜索, 实现快速厘米级定位。实验结果表明, 所提算法的中值定位误差为 7.74 cm, 且实时性比基于中国剩余定理 (CRT) 的传统定位算法提升了 12 倍。

关键词: 射频识别; 室内定位算法; 帐篷反向学习; 自适应蝙蝠算法

中图分类号: TN99

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2022159

Indoor RFID localization algorithm based on adaptive bat algorithm

XIE Liangbo, LI Yuyang, WANG Yong, ZHOU Mu, NIE Wei

School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Abstract: Aiming at the problem that long time-consuming and poor positioning accuracy using geometric method in the traditional UHF RFID indoor localization algorithm, an RFID indoor positioning algorithm based on adaptive bat algorithm (ABA) was proposed. Firstly, the phase of multiple frequency points was obtained by frequency hopping technology, and the location evaluation function of bat algorithm was established based on the angle information of multiple signal classification (MUSIC) algorithm and the distance information of clustering. Secondly, the bat location was initialized by tent reverse learning to increase the diversity of the population, and the adaptive weight factor was introduced to update the bat location. Finally, the target position was searched iteratively based on the position evaluation function to achieve fast centimeter level positioning. Experimental results show that the median localization error of the proposed algorithm is 7.74 cm, and the real-time performance is improved by 12 times compared with the traditional positioning algorithm based on the Chinese remainder theorem (CRT).

Keywords: radio frequency identification, indoor location algorithm, tent reverse learning, adaptive bat algorithm

0 引言

随着物联网技术的迅猛发展, 实时获取人员及重要物品位置的需求日益增大。在现有的射频

识别 (RFID, radio frequency identification) 系统中, 实现快速高精度定位是一项巨大的挑战。目前, 室内 RFID 定位方法主要包括基于接收信号强度指示 (RSSI, received signal strength in-

收稿日期: 2022-05-06; 修回日期: 2022-08-03

通信作者: 王勇, yongwang@cqupt.edu.cn

基金项目: 国家自然科学基金资助项目 (No.62101085); 重庆市教委科学技术研究基金资助项目 (No.KJZD-K202000605); 重庆市研究生科研创新基金资助项目 (No.CYS22473); 重邮信通青创团队支持计划基金资助项目 (No.SCIE-QN-2022-06)

Foundation Items: The National Natural Science Foundation of China (No.62101085), Science and Technology Research Program of Chongqing Municipal Education Commission (No.KJZD-K202000605), Chongqing Graduate Scientific Research Innovation Project (No.CYS22473), Youth Innovation Group Support Program of ICE Discipline of CQUPT (No.SCIE-QN-2022-06)

dication)^[1]、基于到达角度 (AoA, angle of arrival)^[2-3]、基于到达时间 (ToA, time of arrival)^[4]和基于载波相位^[5-6]的定位方法。

基于 RSSI 的定位方法主要是利用电磁波在室内环境下的传输损耗模型进行测距完成定位。文献[7]提出的 LANDMARC 定位算法采用有源标签, 对比目标标签与参考标签的 RSSI, 采用加权算法完成定位受障碍物影响较大, 定位精度不高。文献[8]在 LANDMARC 的基础上提出了 VIRE (virtual reference) 算法, 通过引入大量虚拟标签来增加标签密度, 进而提升定位精度, 其是在 LANDMARC 基础上改进的较为成功的算法, 在降低成本的基础上提升了定位精度, 但是定位精度仍然较低。另一个较成功的改进算法是 SpotON^[9], 其结合有源标签, 使用聚合函数计算标签的信号强度, 通过传输损耗模型计算距离, 结合几何方法实现三维定位, 但是系统较复杂, 计算量大。以上算法定位精度较低, 主要是由于 RSSI 受室内环境影响较大, 很难实现高精度定位。

基于载波相位的定位算法以前主要应用于卫星定位系统中, 由于其易于解算、噪声容错高等特点, 近年来被引入室内定位。文献[10]将跳频与扩展卡尔曼滤波 (EKF, extended Kalman filter) 算法相结合, 实现了高精度快速定位, 但其天线为定向天线, 辐射范围有限。文献[11]通过移动天线构建合成孔径雷达, 实现多点相位信息的收集, 大大提升了定位精度。以上方法虽然实现了高精度定位, 但计算时间相对较长。文献[12]将相位与 RSSI 相结合, 利用 RSSI 做粗估计来约束搜索范围, 再利用相位全息图进一步实现精准定位, 但基于全息图的定位方式计算量大, 耗时较长。文献[13]利用稀疏标签阵列对信号进行角度估计, 结合多个角度实现标签定位, 但标签阵列之间存在耦合问题, 具有一定缺陷。

近年来, 许多工作将室内定位与优化算法相结合, 大大减少了计算复杂度, 缩短了定位时间。文献[14]将粒子群算法与 AoA 相结合, 在提升定位精度的同时大大缩短了定位耗时。文献[15]在合成孔径雷达的基础上使用粒子群算法对目标进行搜索, 弥补了合成孔径雷达计算复杂度大的问题, 大大缩短了定位时间。文献[16]将 2 个标签的信号强度与重叠式年轮定位算法相结合, 增加了搜索区域, 提升了定位精度。文献[17]通过蝙蝠算

法 (BA, bat algorithm) 优化读写器部署位置, 将读写器部署问题类比于平面坐标点的优化问题, 提升了定位的准确性。文献[18]将模糊聚类与猫群算法相结合, 减少了特征匹配的复杂度, 实现了稳定快速的区域化位置搜索。以上研究表明, 传统优化算法能够有效提升定位性能, 但迭代时间长, 容易陷入局部最优, 定位性能仍有待提升。

本文针对传统定位算法时间复杂度高及传统优化算法容易陷入局部最优等问题, 提出了基于自适应蝙蝠算法 (ABA, adaptive bat algorithm) 的室内 RFID 定位算法, 主要研究工作如下。

1) 本文将定位问题转化为优化问题, 融合 MUSIC (multiple signal classification) 测向与聚类测距的原理建立评估函数, 结合自适应蝙蝠算法对定位区域内位置进行评估搜索, 摒弃了传统几何解算的定位思路, 最终实现快速厘米级定位。

2) 相较于其他优化算法, 蝙蝠算法具有收敛速度快、没有较多参数调整的优点, 但容易陷入局部最优, 针对这一问题, 本文提出一种自适应蝙蝠算法。通过引入 2 个自适应权重约束蝙蝠搜索范围, 增强了算法的全局搜索能力, 同时也提升了算法收敛性。

3) 使用帐篷 (Tent) 反向学习增加初始种群的多样性, 提升算法初始搜索速度, 设置自适应终止条件, 在保证定位精度的同时缩短算法耗时。实验结果表明, 与基于中国剩余定理 (CRT, Chinese remainder theorem) 的传统定位算法相比, 本文算法的实时性提升了 12 倍。

1 定位系统结构

1.1 定位平台组成

定位系统由读写器、发射机、接收机、交换机、定位服务器以及标签组成, 结构如图 1 所示。读写器用于激活标签, 与标签建立通信; 发射机发射连续跳频载波信号, 标签利用反向散射特性, 将自身调制信息加到载波信号上; 接收机对标签信号进行接收解调; 交换机负责定位服务器与定位系统相关设备间的数据交互; 定位服务器控制发射机产生跳频信息, 同时从接收机解调的信息中按照标签的通信协议提取出标签信号, 通过标签信号计算出接收信号的相位, 最终结合 ABA 完成定位。

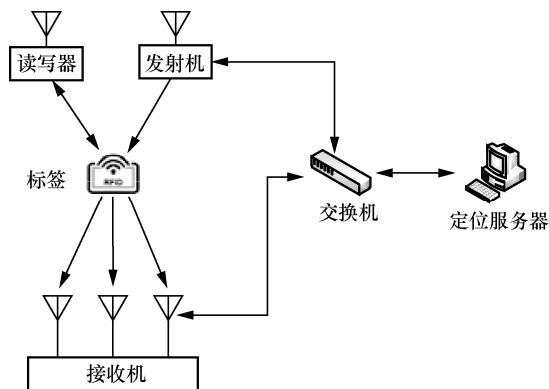


图 1 定位系统结构

1.2 定位算法流程

定位算法流程如图 2 所示。根据定位平台收集的 I/Q 两路信息，从跳频信息中提取每个频率的相位。

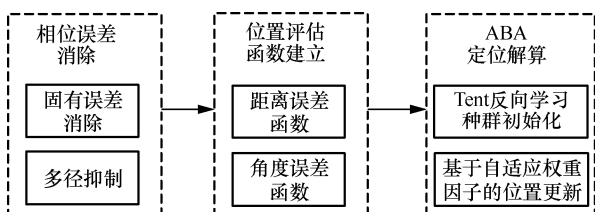


图 2 定位算法流程

接收信号的相位信息包含设备误差和多径误差，即

$$\varphi_r = \left(\frac{2\pi(d+l)}{\lambda} + \varphi_e + \varphi_m \right) \bmod 2\pi \quad (1)$$

其中， d 为标签到接收机的链路距离， l 为发射机到标签的链路距离， φ_e 为设备和天线引入的固有误差， φ_m 为环境造成的多径误差。设备误差通过待测点与参考点的相位相减消除，多径误差采用多径抑制算法进行处理^[19]。

利用误差减小之后的相位建立距离误差函数与角度误差函数，将 2 个函数代入 ABA 中对每次迭代的位置进行评估，找到评估误差最小的坐标，该坐标即最终的定位结果。

2 位置评估函数建立

根据采集到的相位信息建立位置评估函数，用于评估搜索位置。评估函数的输入为待评估的位置，由角度误差函数与距离误差函数组成，即

$$FI(x, y) = P(x, y) + D(x, y) \quad (2)$$

其中， $P(x, y)$ 是角度误差函数，表示搜索位置相对于接收天线阵角度与真实位置相对于天线阵角度的误差； $D(x, y)$ 是距离误差函数，表示搜索位置到接

收天线的距离与真实位置到接收天线的距离的误差。

2.1 角度误差函数

角度误差函数是基于 MUSIC 算法原理建立的。由于信源与天线的距离在数个波长以上，到达天线的信号可以看作远场信号，天线接收到的信号可以看作平面波，如图 3 所示。

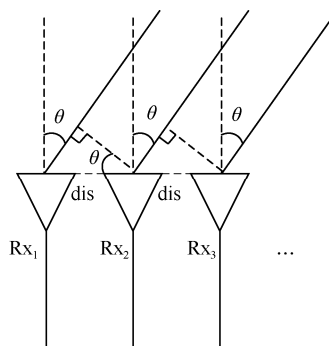


图 3 天线接收到的信号

阵元间的间距为 dis ($dis < \frac{\lambda}{2}$)， λ 为信号的波

长，信号的入射方向与法线夹角为 θ ，天线 i 接收的标签波前信号为 $s_i(t)$ ，则天线 i 接收的信号为

$$x_i(t) = s_i(t)k_i + n_i(t) \quad (3)$$

其中， $n_i(t)$ 为噪声信号； k_i 为信号的方向，表示为

$$k_i = \exp\left(\frac{-j2\pi\omega_0(i-1)dis \sin \theta}{c}\right) \quad (4)$$

其中， ω_0 为信号的中心频率， c 为光速。入射信号向量表示为

$$\mathbf{X} = \mathbf{K}\mathbf{S} + \mathbf{N} \quad (5)$$

天线 i 接收信号的相位为

$$\varphi_i = \bmod\left(\frac{2\pi\omega_0(d_i+l)}{c}, 2\pi\right) \quad (6)$$

其中， $\bmod(A, B)$ 表示 A 对 B 取余， d_i 表示标签到天线 i 的链路距离， l 表示发射天线到标签的距离。根据接收信号的相位，信号的方向向量可以表示为

$$\mathbf{K} = \begin{bmatrix} 1 \\ \exp(-j(\varphi_2 - \varphi_1)) \\ \vdots \\ \exp(-j(\varphi_i - \varphi_1)) \end{bmatrix} \quad (7)$$

根据式(5)构建信号的协方差矩阵为

$$\begin{aligned} \mathbf{R} &= E[\mathbf{X}\mathbf{X}^H] = \\ &= \mathbf{K}E[\mathbf{S}\mathbf{S}^H]\mathbf{K}^H + E[\mathbf{N}\mathbf{N}^H] = \\ &= \mathbf{K}\mathbf{R}_s\mathbf{K}^H + \sigma^2\mathbf{I} \end{aligned} \quad (8)$$

其中, \mathbf{R}_s 为信号相关矩阵, σ^2 为噪声功率, \mathbf{I} 为单位矩阵。根据方向矩阵求解噪声子空间对应的特征向量 \mathbf{v}_i ($i \in [1, k]$), 即

$$\mathbf{K}^H \mathbf{v}_i = 0 \quad (9)$$

利用特征向量 \mathbf{v}_i 组成噪声子空间 \mathbf{U} 。根据采集到的相位提取出信号的噪声子空间之后, 用其对传入的评估位置进行角度评估, 由于传入的为位置坐标, 构建评估位置到接收天线的方向向量 \mathbf{G} 为

$$\mathbf{G} = \begin{bmatrix} 1 \\ \exp\left(-j\text{mod}\left(\frac{2\pi D_{12}}{\lambda}, 2\pi\right)\right) \\ \vdots \\ \exp\left(-j\text{mod}\left(\frac{2\pi D_{li}}{\lambda}, 2\pi\right)\right) \end{bmatrix}^T \quad (10)$$

其中, D_{li} 代表评估位置到天线 1 的距离与评估位置到天线 i 的距离之差, 可以表示为

$$\begin{aligned} D_{li} &= (d_i + l) - (d_1 + l) = d_i - d_1 \\ d_i &= \sqrt{(x - x_i)^2 + (y - y_i)^2} \end{aligned} \quad (11)$$

其中, (x, y) 是评估位置的坐标, (x_i, y_i) 是天线 i 的坐标。根据信号提取的噪声子空间和评估位置构建角度误差函数为

$$P(x, y) = |\mathbf{G}\mathbf{U}^H\mathbf{U}\mathbf{G}^H| \quad (12)$$

由于构建的信号向量是一个 $1 \times N$ 的矩阵, 提取的噪声子空间是 $M \times N$ 的矩阵, 因此式(12)计算结果为一个数值。如果评估位置与真实位置处在以接收机为端点的射线上, 由于信号空间与噪声子空间正交, $P(x, y)$ 将为 0, 但是测试中往往存在误差, 因此在每次迭代中, 寻找其最小值作为结果。

2.2 距离误差函数

接收信号相位的范围为 $[-\pi, \pi]$, 单根天线接收时, 根据相位与距离的关系可得

$$\begin{cases} d + l = \frac{\varphi_1}{2\pi} \lambda_1 + N_1 \lambda_1 \\ d + l = \frac{\varphi_2}{2\pi} \lambda_2 + N_2 \lambda_2 \\ \vdots \\ d + l = \frac{\varphi_j}{2\pi} \lambda_j + N_j \lambda_j \end{cases} \quad (13)$$

其中, N_j 为频率 j 下的整周数, φ_j 为频率 j 下接收信号的相位, λ_j 为频率 j 对应的波长。由于式(13)

是一个欠定方程, 直接求解方程组整周数将会得到无数个解, 不能直接计算整周数得到链路距离, 因此本文根据多频率多天线下传入位置对应的相位与接收信号相位的差值进行聚类, 搜索出差值最小的位置。由于评估函数的输入是蝙蝠位置, 当前蝙蝠位置对应的链路距离 \hat{d} 与标签的真实链路距离的差值可表示为

$$d - \hat{d} = \frac{\varphi_j - \hat{\varphi}_j}{2\pi} \lambda_j + (N_j - \hat{N}_j) \lambda_j \quad (14)$$

其中, $\hat{\varphi}_j$ 为频率 j 下蝙蝠位置对应的相位。当 $d \approx \hat{d}$ 时, $\varphi_j \approx \hat{\varphi}_j$ 、 $N_j = \hat{N}_j$, 则式(14)可以变换为

$$e_j = \hat{\varphi}_j - \varphi_j \quad (15)$$

其中, e_j 为代表第 j 个频率下相位之间的差值。将其扩充到多根天线, 提升测距的准确性, 结合式(15)建立距离误差函数为

$$\begin{aligned} D(x, y) &= \sum_{j=1}^n \sum_{i=1}^m |\hat{\varphi}_{i,j} - \varphi_{i,j}| = \\ &= \sum_{j=1}^n \sum_{i=1}^m \left| \text{mod}\left(\frac{2\pi(d_i + l)}{\lambda_j}, 2\pi\right) - \varphi_{i,j} \right| \end{aligned} \quad (16)$$

其中, n 为天线数目, m 为频率数目, $\varphi_{i,j}$ 为天线 i 在频率 j 下的测量相位, $\hat{\varphi}_{i,j}$ 为频率 j 下发射天线到评估位置再到天线 i 的相位, d_i 为标签与天线 i 的链路距离。当蝙蝠位置坐标与实际位置坐标相同时, 距离误差函数输出为 0。但是, 在实际运算过程中, 由于误差的存在不能搜索到真实位置, 因此在搜索过程中将误差最小值对应的评估位置作为最终结果。

结合式(12)与式(16), 组成式(2)对评估位置进行评估, 得到该位置的评估值, 完成对评估位置的评估, 评估函数算法流程如算法 1 所示。算法 1 中将角度误差函数扩展到了多频率, 同时使用 3δ 准则对奇异值进行了剔除。

算法 1 评估函数 FI(x, y)

输入 待评估位置坐标 $\text{Loc}(x, y)$, 根据 n 个频点下测量相位提取的噪声子空间 $\mathbf{U}_l, l \in [1, n]$, 多径抑制后 k 根天线 n 个频率下的接收相位 $\varphi_{i,l}, i \in [1, k], l \in [1, n]$, 接收天线坐标 $\text{Rx}_i, i \in [1, k]$, 发射天线坐标 Tx , 每个频点对应的波长 $\lambda_l, l \in [1, n]$

输出 评估值 F

- 1) for $l = 1 \rightarrow n$ do
- 2) for $i = 1 \rightarrow k$ do
- 3) $d_i = \text{dis}(\text{Loc}, \text{Rx}_i)$ // $\text{dis}(A, B)$ 表示求 A, B 的欧氏距离
- 4) $\Delta d_i = d_i - d_1$
- 5) $\theta = \text{mod}\left(\frac{2\pi\Delta d_i}{\lambda_i}\right)$ // $\text{mod}(A, B)$ 表示 A 对 B 取模
- 6) $\mathbf{G}_i = [\mathbf{G}_i; e^{-j\theta}]$
- 7) end for
- 8) $\mathbf{G}_i = \mathbf{G}_i^T$ // $(\cdot)^T$ 表示矩阵的转置
- 9) $L = \text{dis}(\text{Loc}, \text{Tx})$
- 10) $\text{temp}_l = |\mathbf{G}_i \mathbf{U}_i^H \mathbf{U}_l \mathbf{G}_i^H|$
- 11) end for
- 12) $\text{aver} = \text{mean}(\text{temp})$ // $\text{mean}()$ 表示求均值
- 13) $\delta = \text{std}(\text{temp})$ // $\text{std}()$ 表示求标准差
- 14) $N = 0, D = 0, P = 0$
- 15) for $i = 1 \rightarrow k$ do
- 16) if $\text{aver} - 3\delta < \text{temp}_i < \text{aver} + 3\delta$ then
- 17) $P = P + \text{temp}_i$
- 18) $N = N + 1$
- 19) end if
- 20) end for
- 21) $P = \frac{P}{N}$
- 22) for $l = 1 \rightarrow k$ do
- 23) for $i = 1 \rightarrow n$ do
- 24) $D = D + \left| \text{mod}\left(\frac{2\pi(d_i + L)}{\lambda_i}, 2\pi\right) - \varphi_{i,l} \right|$
- 25) end for
- 26) end for
- 27) $F = D + P$

3 ABA 定位算法

传统的蝙蝠算法通过模拟蝙蝠搜索猎物的方式，在寻找猎物的过程中，每个蝙蝠会发射出不同的声波频率、响度和脉冲率，每次飞行都会更新这些信息。在蝙蝠算法实现中，虽然其在搜索最优位置时会具有较快的收敛速度，但是蝙蝠的飞行速度更新方式、位置更新方式和跳出局部最优的方式都会受到随机值的影响，使搜索过程很容易陷入局部最优，且定位精度较差^[20]。本文针对蝙蝠算法易陷入局部最优和定位精度较差等问题，提出了基于 Tent 反向学习的自适应蝙蝠算法，降低其陷入局部最优的概率，提升搜索精度。

3.1 Tent 反向学习初始化种群

在搜索过程中，蝙蝠初始位置的多样性能够大幅

提升搜索的速度与精度。传统的初始化种群的算法分为随机算法和确定性算法，随机算法依赖于初始随机种子，初始化后的种群可能分布不均，蝙蝠位置的多样性较差，导致搜索时间增加，搜索精度下降，而确定性算法在种群规模不满足最低要求时，存在与随机算法同样的问题。为了增强蝙蝠种群的多样性，本文提出了基于 Tent 映射和反向学习的种群初始化策略，先使用 Tent 映射初始化种群，之后使用反向学习方法对已初始化的种群进行更新，对 Tent 映射后的种群和反向学习后的种群进行合并选优。Tent 映射表示为^[21]

$$g(x) = \begin{cases} 2x, & 0 \leq x \leq \frac{1}{2} \\ 2x - 1, & \frac{1}{2} < x \leq 1 \end{cases} \quad (17)$$

初始化的蝙蝠位置为

$$\text{loc}_{\text{Tent}} = \text{loc}_{\min} + (\text{loc}_{\max} - \text{loc}_{\min})g(x) \quad (18)$$

其中， loc_{\min} 和 loc_{\max} 分别为搜索范围的下界和上界。得到 Tent 映射的初始化种群之后，对其进行反向学习^[22]，得到

$$\begin{aligned} f(x) &= 1 - g(x) \\ \text{loc}_{\text{reverse}} &= \text{loc}_{\min} + (\text{loc}_{\max} - \text{loc}_{\min})f(x) \end{aligned} \quad (19)$$

将 Tent 映射后的种群和反向学习后的种群作为初始化种群对初始种群各个蝙蝠进行评估，找到最优蝙蝠和最佳评估值。设置种群数为 100，随机位置与 Tent 反向学习后的位置的对比如图 4 所示。

图 4 中，小圆圈表示生成的蝙蝠位置，大圆圈表示对重叠区域进行突出显示。可见，随机产生的种群有较多重叠，而本文算法没有重叠区域且种群分布较均匀、种群多样性较丰富。

3.2 基于自适应权重的位置搜索

初始化种群后，开始对目标进行搜索。为了降低蝙蝠陷入局部最优的概率，提升搜索速度，本文舍弃了蝙蝠速度这一因素，引入 2 个自适应权重 (Q_{1i} 和 Q_{2i}) 来更新蝙蝠位置，即

$$\begin{aligned} Q_{1i} &= e^{-\text{abs}(F_i - F_{\text{best}})} \\ Q_{2i} &= 1 - Q_{1i} \end{aligned} \quad (20)$$

其中， $\text{abs}(\cdot)$ 表示绝对值函数， F_i 表示第 i 个蝙蝠的评估值， F_{best} 表示当前的最优评估值。当前蝙蝠位置的评估值越接近最优评估值时，则 Q_{1i} 越大， Q_{2i} 越小，用这 2 个自适应权重对蝙蝠位置进行更新，即

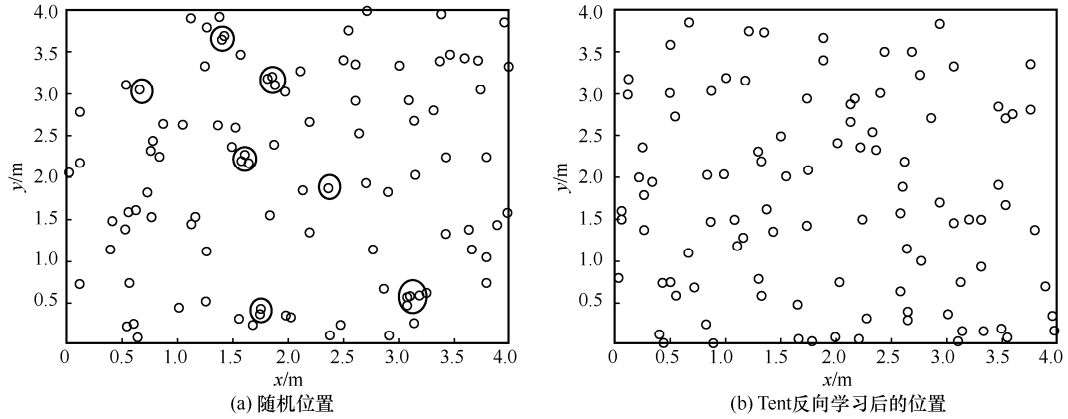


图 4 随机位置与 Tent 反向学习后的位置的对比

$$\text{tem} = \text{Loc}_i + \alpha \text{rand}$$

$$\text{loc} = \text{Loc}_i + (\text{Loc}_{\text{best}} - \text{Loc}_i)Q_{1i} + (\text{tem} - \text{Loc}_i)Q_{2i} \quad (21)$$

其中, tem 为随机位置; Loc_i 为第 i 个蝙蝠当前的位置坐标; α 为一个固定值, 代表搜索范围; rand 为 $[0,1]$ 的随机值; tem 为以当前迭代位置为圆心的范围 α 内随机寻找一个位置, 再通过自适应权重 Q_{1i} 与 Q_{2i} 更新下一次位置; Loc_{best} 为当前搜索过程中的最优位置。每次更新位置后, 判断其是否越界, 如果越界, 则将就近的边界位置作为更新的位置。位置更新后, 对更新后的位置进行评估, 即

$$F_{\text{new}} = \text{FI}(\text{loc}) \quad (22)$$

其中, $\text{FI}(\text{loc})$ 表示对位置使用评估函数进行评估。如果此次更新后的位置的评估值较 F_i 更大, 则需要对此次迭代位置按照式(23)再次进行更新。

$$\begin{aligned} \text{loc} &= \text{Loc}_{\text{best}} + g \text{rand} \\ F_{\text{new}} &= \text{FI}(\text{loc}) \\ \text{s.t. } F_{\text{new}} &> F_i \end{aligned} \quad (23)$$

其中, g 是一个定值, 代表蝙蝠随机飞行范围的大小。如果 $F_{\text{new}} < F_i$, 则对蝙蝠位置和评估值进行更新, 即

$$\begin{aligned} F_i &= F_{\text{new}} \\ \text{Loc}_i &= \text{loc} \\ \text{s.t. } F_{\text{new}} &< F_i \end{aligned} \quad (24)$$

同时, 如果 $F_{\text{new}} < F_{\text{best}}$, 则对最优位置与最优评估值进行更新, 即

$$\begin{aligned} F_{\text{best}} &= F_{\text{new}} \\ \text{Loc}_{\text{best}} &= \text{loc} \\ \text{s.t. } F_{\text{new}} &< F_{\text{best}} \end{aligned} \quad (25)$$

通过不断迭代更新蝙蝠最优位置, 完成对目标位置的搜索, 最终完成定位。

3.3 算法优化

在蝙蝠位置迭代的过程中, 初始的蝙蝠数量和迭代次数是已设定好的, 但是在大量测试中发现, 由于算法收敛性较强, 找到最优位置后的迭代中, 最优位置都不会改变或者位置变化很小; 同时在算法搜索过程后期, 随着搜索范围减小, 过多的蝙蝠数目只会增加算法耗时。

为解决此问题, 本文提出了迭代提前停止和动态蝙蝠数目更新的方法。假设蝙蝠数目和迭代次数都为 W , 在后面的迭代中, 保存第 i 次迭代的最优位置 Loc_{best} , 形成最优位置的一个数组 $\text{Loc}_{\text{best},i} (i < W)$, 每次迭代结束, 将最优位置添加到最优位置数组中, 即

$$\begin{aligned} \text{std}(\text{Loc}_{\text{best},i}) &< \mu \\ \text{s.t. } i &> \tau \end{aligned} \quad (26)$$

其中, $\text{std}(\cdot)$ 表示对值求标准差。式(26)表示在迭代次数大于 τ 次之后, 计算其前 20 次到当前次数的标准差, 如果标准差小于 μ , 则迭代提前停止, 跳出循环, 其中, $\tau = 25$, $\mu = 2$ 。动态蝙蝠数目更新指在每次迭代完成后, 蝙蝠数目动态缩减, 即

$$\begin{aligned} W &= W\beta \\ \text{s.t. } W &\geq 20 \end{aligned} \quad (27)$$

其中, β 是一个 $[0,1]$ 内的定值, 设定 $\beta = 0.95$ 。式(27)表示每次迭代对蝙蝠数目进行更新, 在缩减到 20 个蝙蝠数目时, 停止更新。

上述方法大大缩短了定位时间, 当蝙蝠数为 150 个时, 优化前定位一个目标的时间约为 1.60 s, 优化后定位一个目标的时间能够缩短到约 0.36 s。

ABA 搜索目标位置流程如算法 2 所示。

算法 2 ABA 搜索目标位置

输入 除去待评估位置坐标后算法 1 中评估函数需要的输入参数

输出 定位结果 Loc_{best}

初始化 使用 Tent 映射和反向学习方法初始化蝙蝠位置，使用评估函数对初始位置合并选优，确定边界范围，蝙蝠数目 n ，最大迭代次数 N_{max} ，搜索范围 α ，随机飞行范围 g

```

1) for  $t=1 \rightarrow N_{max}$  do
2)   for  $i=1 \rightarrow n$  do
3)      $Q_{1i} = e^{-|F_i - F_{best}|}$ 
4)      $Q_{2i} = 1 - Q_{1i}$ 
5)      $tem = Loc_i + \alpha rand$ 
6)      $tem_{best} = (Loc_{best} - Loc_i)Q_{1i}$ 
7)      $loc = Loc_i + tem_{best} + (tem - Loc_i)Q_{2i}$ 
8)      $loc = border(loc)$ 
       //border(A)判断A是否越界
9)      $F_{new} = FI(loc)$ 
10)    if  $F_{new} > F_i$  then
11)       $loc = Loc_{best} + grand$ 
12)       $F_{new} = FI(loc)$ 
13)    end if
14)    if  $F_{new} < F_i$  then
15)       $Loc_i = loc$ 
16)       $F_i = F_{new}$ 
17)    end if
18)    if  $F_{new} < F_{best}$  then
19)       $Loc_{best} = loc$ 
20)       $F_{best} = F_{new}$ 
21)    end if
22)  end for
23)   $Best_t = F_{best}$ 
24)  if  $t > 25$  then
25)    if  $std(Best_{(t-20) \rightarrow t}) \leq 2$  then
       //std(B)表示求B的标准差
26)      break //满足条件，跳出循环
27)    end if
28)  end if
29)   $[F, Loc_t] = screen(F, Loc_t)$  //对蝙蝠进行筛选
30) end for
31) return  $F_{best}$ 

```

4 仿真分析与对比

4.1 仿真分析

为验证算法的性能，本节首先对整个系统进

行仿真。执行算法的计算机 CPU 为 AMD R7 4800HS，RAM 为 16 GB，仿真搜索区域大小为 $4\text{ m} \times 4\text{ m}$ ，以区域的中心为坐标原点建立直角坐标系，跳频范围为 830~960 MHz，跳频间隔为 10 MHz，发射天线的坐标为(0,0)，3 根接收天线的坐标分别为(-10, -200)、(0, -200)和(10, -200)，单位为 cm，信号添加信噪比为 10 dB 的高斯白噪声。每次定位迭代次数为 100 次，蝙蝠数 G 分别为 50、100、150、200、300 和 500，探究不同蝙蝠数对定位的影响。进行 500 次定位，每次定位目标在搜索范围内随机分布，不同蝙蝠数定位误差的累积分布函数 (CDF, cumulative distribution function) 如图 5 所示。仿真统计结果如表 1 所示。

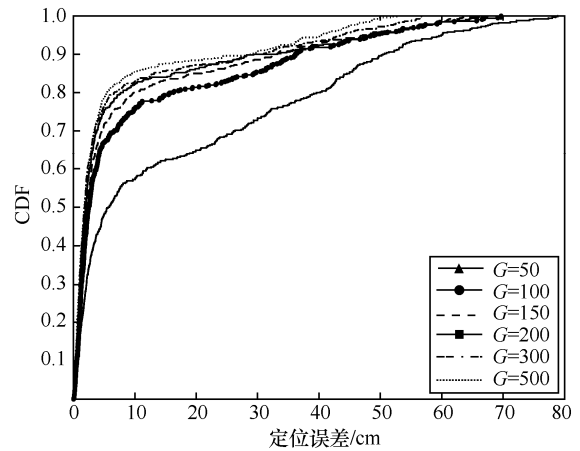


图 5 不同蝙蝠数定位误差的 CDF

表 1 仿真统计结果

蝙蝠数/个	X 轴平均误差/cm	Y 轴平均误差/cm	平均定位误差/cm	平均耗时/s
50	6.01	15.73	17.75	0.11
100	3.34	9.08	10.23	0.23
150	3.41	7.40	8.73	0.36
200	2.78	7.29	8.23	0.48
300	2.61	6.38	7.37	0.74
500	2.59	5.52	6.54	1.19

从图 5 可以看出，不同蝙蝠数下，小于 10 cm 的定位误差的占比都能达到 55%以上。其中，当蝙蝠数为 50、100、150、200、300 和 500 时，小于 10 cm 的定位误差的占比分别为 57.2%、75.8%、80.0%、82.2%、82.8%和 85.4%。从表 1 中可以看出，当蝙蝠数为 50、100、150、200、300 和 500 时，平均定位误差分别为 17.75 cm、10.23 cm、8.73 cm、8.23 cm、7.34 cm 和 6.54 cm。根据平均耗时和平均定位误差对蝙蝠数目进行选择，当 $G=150$ 时，平均耗时为 0.36 s，

定位平均误差为 8.73 cm，具有较低的耗时，且定位精度有较大提升；而当 $G > 150$ 时，耗时的增长和定位精度的增长收益明显不成正比。所以在实测时设置定位的蝙蝠数为 150。

4.2 算法收敛情况对比

为验证本文 ABA 的搜索速度以及局部搜索能力，在定位平台仿真的基础上使用不同的算法进行比较，对比了 BA、粒子群优化 (PSO, particle swarm optimization) 算法^[15]和方向性蝙蝠算法^[23] (DBA, directional bat algorithm) 的收敛情况，所有算法初始种群数都为 150，不同算法的收敛情况如图 6 所示。

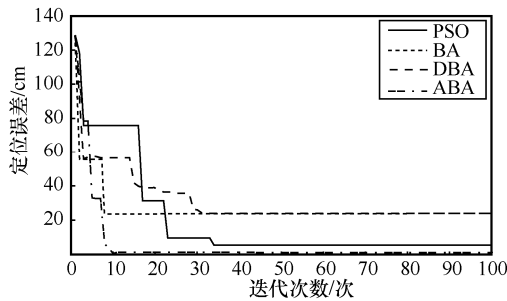


图 6 不同算法的收敛情况

由图 6 可以看出，4 种算法都可收敛。DBA、BA、PSO 与所提算法收敛时的定位误差分别为 24.06 cm、23.77 cm、5.44 cm 和 1.19 cm。传统蝙蝠算法虽然收敛速度很快，但是定位误差较大；本文算法在保证较快的收敛速度下，定位误差较小，具有较高的收敛速度以及定位精度。

5 实测分析

5.1 测试平台

为验证本文算法实测性能，搭建了如图 7 所示的测试场景，室内环境为典型的客厅环境，测试范围为 4 m×4 m，接收天线间距为 10 cm，选用的跳频范围为 830~960 MHz，跳频频率间隔为 10 MHz，跳频时间间隔为 50 ms，标签反射信号处理过程在 GNU Radio 平台上完成，实验设备如表 2 所示。



图 7 测试场景

表 2	实验设备
设备	用途
无源标签 H47	定位目标
全向天线 VERT900	发射信号
Impinj R420	与标签建立通信，频率为 924.125 MHz
USRP N210	共 4 台，一台用于发射跳频信号，3 台用于接收经标签发射的信号
PC	控制收发设备，处理数据
时钟源	型号为 OctoClock-G CDA-2990，用于控制接收机和发射机时钟同步

5.2 实测定位精度

由于噪声和评估函数原理的影响，不同坐标轴的定位精度可能会不同，本节对定位精度进行统计分析。不同坐标轴定位误差的 CDF 如图 8 所示。

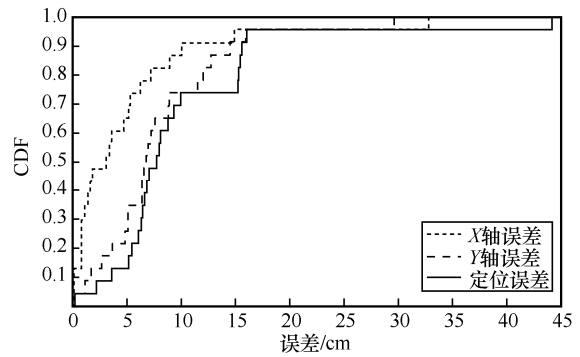


图 8 不同坐标轴定位误差的 CDF

从图 8 可以看出，X 轴误差、Y 轴误差和定位误差 10 cm 以下的占比分别为 91.3%、73.9%和 73.9%。其中，X 轴误差、Y 轴误差和定位误差的中位数分别为 3.08 cm、6.77 cm 和 7.74 cm。因此，X 轴的定位精度优于 Y 轴定位精度。主要原因是评估函数是测距与测向相结合的，测距距离是链路距离，根据椭圆定位原理，在以发射天线和接收天线为焦点的椭圆上的点都满足条件，发射天线与接收天线的坐标在 Y 轴上，形成的椭圆是 Y 轴上的椭圆，其长轴在 Y 轴上，短轴在 X 轴上，如果测向发生偏差，使定位坐标偏差，则 Y 轴误差将大于 X 轴误差。

5.3 不同算法比较

本节选取不同算法对定位精度进行分析。本节对适用于本文 RFID 系统的算法和适用于其他 RFID 系统的算法分别进行了比较。

不同算法定位误差的 CDF 如图 9 所示。DBA、PSO、CRT^[19]、EKF^[10]和 ABA 的定位误差中值分别为 15.95 cm、22.46 cm、9.11 cm、9.35 cm 和 7.74 cm。DBA、PSO、CRT、EKF 和 ABA 的小于 10 cm 的定位误差

占比分别为 26.1%、43.5%、52.1%、52.4%和 73.9%。因此，本文所提 ABA 优于对比算法，可实现更高精度的室内定位。

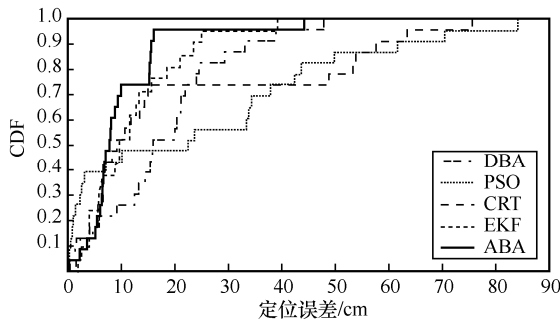


图 9 不同算法定位误差的 CDF

将本文算法与适用于其他 RFID 系统的算法进行对比，结果如表 3 所示。与文献[10]算法相比，本文算法使用天线为全向天线，体积更小，范围更广；与文献[11]算法相比，本文算法天线不需要保持移动，且系统成本较低；与文献[12]算法相比，本文算法天线不需要互成角度，且定位范围更大；与文献[13]算法相比，本文算法不需要定位目标贴附标签阵列，可实现独立目标定位。

表 3 本文算法与适用于其他 RFID 系统的算法对比

定位算法	部署需求	定位范围	中位数误差/cm
文献[10]	定向三天线	4 m×4 m	9.35
文献[11]	移动单天线	室外操场 8 m×8 m	6
文献[12]	移动双天线	1.2 m×2.2 m	9
文献[13]	定向双天线	2.4 m×8 m	5.01
本文算法	单站三天线	4 m×4 m	7.74

5.4 多目标定位分析

为验证多目标情况下算法的性能，在定位区域内，标签间隔 25 cm 平行放置，并均匀移动测量标签位置，多目标定位误差的 CDF 如图 10 所示。从图 10 可以看出，X 轴误差、Y 轴误差和定位误差在 10 cm 以下的占比分别为 88%、84%和 76%。其中，有一个标签定位误差较大，主要是因为测量相位与实际相位偏差过大，导致算法的定位结果与实际标签位置差别过大。从图 10 中可以看出，本文算法在多目标定位时同样具有较高准确性。

5.5 实时性分析

为验证本文算法的实时性，本节对本文算法、CRT、DBA、PSO 和 EKF 进行了实时性对比分析，实时性比较如图 11 所示。从图 11 可以看出，本文算法、EKF、PSO、DBA 和 CRT 的耗时分别为

0.36 s、0.43 s、0.86 s、3.34 s 和 4.35 s。本文算法相较于对比算法实时性更高，其中较 CRT 实时性提升了 12 倍。

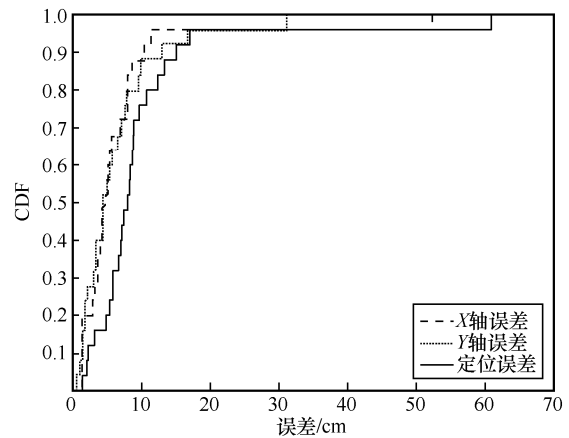


图 10 多目标定位误差的 CDF

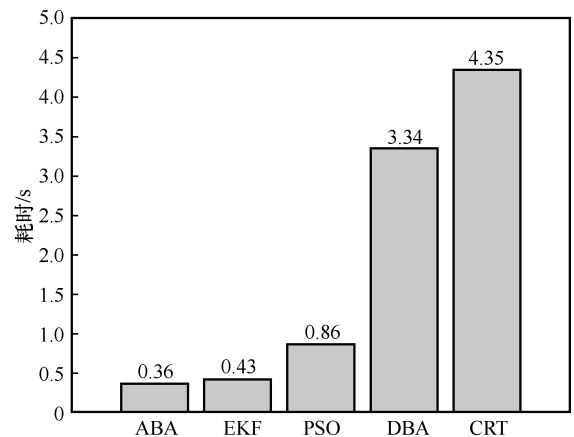


图 11 实时性比较

6 结束语

本文提出了一种基于 Tent 反向学习自适应蝙蝠算法的室内 RFID 定位算法。首先，将采集的相位进行多径抑制，在 MUSIC 测向与聚类测距的基础上建立评估函数；然后，将评估函数与自适应蝙蝠算法相结合，用于评估蝙蝠每次更新的位置；最后，通过不断迭代搜索到目标位置完成定位。实验结果表明，本文提出的自适应蝙蝠算法具有良好的室内定位效果。

参考文献：

[1] ZHANG S G, YANG C W, JIANG D M, et al. Nothing blocks me: precise and real-time LOS/NLOS path recognition in RFID systems[J]. IEEE Internet of Things Journal, 2019, 6(3): 5814-5824.

- [2] PENG C, JIANG H, QU L D. Deep convolutional neural network for passive RFID tag localization via joint RSSI and PDOA fingerprint features[J]. IEEE Access, 2021, 9: 15441-15451.
- [3] SUN Y M, HO K C, WAN Q. Eigenspace solution for AOA localization in modified polar representation[J]. IEEE Transactions on Signal Processing, 2020, 68: 2256-2271.
- [4] CREMER M, PERVEZ A, DETTMAR U, et al. Transmit beamforming for Angle-of-Activation (AoAct) estimation in passive UHF RFID systems[C]//Proceedings of IEEE International Conference on RFID Technology and Applications. Piscataway: IEEE Press, 2015: 1-7.
- [5] ALEMARYEEN A, NOGHANIAN S, MODARESI M. Study of an accurate phase-based localization method for semi-passive RFID tags[C]//Proceedings of IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting. Piscataway: IEEE Press, 2018: 2007-2008.
- [6] JIANG Y, MA Y T, LIU H K, et al. RF-SML: a SAR-based multi-granular and real-time localization method for RFID tags[J]. Electronics, 2020, 9(9): 1447.
- [7] NI L M, LIU Y H, LAU Y C, et al. LANDMARC: indoor location sensing using active RFID[C]//Proceedings of the First IEEE International Conference on Pervasive Computing and Communications. Piscataway: IEEE Press, 2003: 407-415.
- [8] ZHAO Y, NI L M. VIRE: Virtual reference elimination for active RFID-based localization[J]. Ad Hoc & Sensor Wireless Networks, 2013, 17(4): 169-191.
- [9] HIGLITOWRE J, WANT R, BORRIELLO G. SpotON: an indoor 3D location sensing technology based on RF signal strength[J]. Department of Computer Science and Engineering, 2000, 7(2): 1-16.
- [10] 谢良波, 刘西西, 王勇, 等. 基于 RFID 载波相位的室内 EKF 定位算法[J]. 通信学报, 2022, 43(3): 124-134.
- XIE L B, LIU X X, WANG Y, et al. Indoor EKF localization algorithm based on RFID carrier phase[J]. Journal on Communications, 2022, 43(3): 124-134.
- [11] BUFFI A, MOTRONI A, NEPA P, et al. A SAR-based measurement method for passive-tag positioning with a flying UHF-RFID reader[J]. IEEE Transactions on Instrumentation and Measurement, 2019, 68(3): 845-853.
- [12] LI C L, TANGHE E, PLETS D, et al. ReLoc: hybrid RSSI- and phase-based relative UHF-RFID tag localization with COTS devices[J]. IEEE Transactions on Instrumentation and Measurement, 2020, 69(10): 8613-8627.
- [13] YANG C, WANG X Y, MAO S W. SparseTag: high-precision backscatter indoor localization with sparse RFID tag arrays[C]//Proceedings of 16th Annual IEEE International Conference on Sensing, Communication, and Networking. Piscataway: IEEE Press, 2019: 1-9.
- [14] WANG Y. User positioning with particle swarm optimization[C]//Proceedings of International Conference on Indoor Positioning and Indoor Navigation (IPIN). Piscataway: IEEE Press, 2017: 1-5.
- [15] BERNARDINI F, BUFFI A, MOTRONI A, et al. Particle swarm optimization in SAR-based method enabling real-time 3D positioning of UHF-RFID tags[J]. IEEE Journal of Radio Frequency Identification, 2020, 4(4): 300-313.
- [16] 刘捷, 张金艺, 查肇祥, 等. 重叠年轮式无源射频识别定位算法设计[J]. 上海交通大学学报, 2012, 46(11): 1816-1820.
- LIU J, ZHANG J Y, ZHA Z X, et al. Overlapped growth ring passive radio frequency identification positioning algorithm design[J]. Journal of Shanghai Jiao Tong University, 2012, 46(11): 1816-1820.
- [17] YAN H Y, CHU J Z. RFID positioning algorithm based on BA optimization[C]//Proceedings of 5th International Conference on Computer and Communication Systems (ICCCS). Piscataway: IEEE Press, 2020: 854-858.
- [18] 李昂, 付敬奇, 沈华明, 等. 基于模糊聚类和猫群算法的室内定位算法[J]. 仪器仪表学报, 2020, 41(1): 185-194.
- LI A, FU J Q, SHEN H M, et al. Indoor positioning algorithm based on fuzzy clustering and cat swarm optimization[J]. Chinese Journal of Scientific Instrument, 2020, 41(1): 185-194.
- [19] MA Y F, SELBY N, ADIB F. Minding the billions: ultra-wideband localization for deployed RFID tags[C]//Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking. New York: Association for Computing Machinery, 2017: 248-260.
- [20] YANG X S. A new metaheuristic bat-inspired algorithm[C]//Nature Inspired Cooperative Strategies for Optimization, Studies in Computational Intelligence. Berlin: Springer, 2010: 65-74.
- [21] BATRA I, GHOSH S. An improved tent map-adaptive chaotic particle swarm optimization (ITM-CPSO)-based novel approach toward security constraint optimal congestion management[J]. Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 2018, 42(3): 261-289.
- [22] 张强, 李盼池. 自适应分组混沌云模型蛙跳算法求解连续空间优化问题[J]. 控制与决策, 2015, 30(5): 923-928.
- ZHANG Q, LI P C. Adaptive grouping chaotic cloud model shuffled frog leaping algorithm for continuous space optimization problems[J]. Control and Decision, 2015, 30(5): 923-928.
- [23] CHAKRI A, KHELIF R, BENOURET M, et al. New directional bat algorithm for continuous optimization problems[J]. Expert Systems with Applications, 2017, 69: 159-175.

[作者简介]



谢良波(1986-), 男, 四川成都人, 博士, 重庆邮电大学副教授、硕士生导师, 主要研究方向为射频识别技术、室内定位技术等。



李宇洋(1997-), 男, 四川宜宾人, 重庆邮电大学硕士生, 主要研究方向为室内射频识别定位技术。

王勇(1987-), 男, 云南邵通人, 博士, 重庆邮电大学讲师、硕士生导师, 主要研究方向为室内定位、深度学习等。

周牧(1984-), 男, 重庆人, 博士, 重庆邮电大学教授、博士生导师, 主要研究方向为无线定位、参数估计、机器学习等。

聂伟(1987-), 男, 重庆人, 博士, 重庆邮电大学讲师、硕士生导师, 主要研究方向为射频识别技术、室内定位技术、电磁场与微波技术等。