

## Localization of Wireless Sensor Network Based on Modified Pigeon Inspired Optimization\*

CUI Huanqing\* ZHANG Na LUO Hanjiang

(College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao Shandong 266590, China)

**Abstract:** Localization is one of the key techniques of wireless sensor networks. In order to improve the localization accuracy a localization algorithm based on Modified Pigeon Inspired Optimization (MPIO) is proposed. MPIO applies adaptive adjustment mechanism to balance the local search and global exploration ability. In further, MPIO combines random walk mechanism to increase the population diversity and enhances algorithm optimization ability. Sensor nodes utilize received signal strength to estimate the distance and utilize MPIO to estimate the position. The simulation results show that MPIO has higher localization accuracy and faster convergence than other algorithms.

**Key words:** wireless sensor network; localization; pigeon inspired algorithm; received signal strength; random walk  
EEACC: 7230 doi: 10.3969/j.issn.1004-1699.2022.03.018

## 基于改进鸽群算法的无线传感器网络定位方法\*

崔焕庆\* 张娜 罗汉江

(山东科技大学计算机科学与工程学院, 山东 青岛 266590)

**摘要:** 定位是无线传感器网络的关键技术之一, 为了提高定位性能, 提出了一种改进鸽群算法的定位方法。在鸽群算法中加入自适应调整机制以平衡局部搜索能力与全局搜索能力, 然后通过种群分组并融入随机游走机制来增加种群多样性, 增强了算法寻优能力。传感器节点使用接收信号强度估计距离, 使用改进的鸽群算法来估计位置。仿真实验表明, 所提出的定位算法具有较高的定位精度和较快的收敛速度。

**关键词:** 无线传感器网络; 定位; 鸽群算法; 接收信号强度; 随机游走

中图分类号: TP393

文献标识码: A

文章编号: 1004-1699(2022)03-0399-06

无线传感器网络 (Wireless Sensor Networks, WSNs) 由大量传感器节点组成, 广泛应用于环境监测、智慧城市、灾害救援和目标追踪等领域。确定传感器节点的位置信息为许多位置感知协议和应用程序提供了基础支持, 因此, 节点定位是无线传感器网络的关键技术之一<sup>[1]</sup>。由于成本因素, 为每个节点配备全球导航卫星系统 (Global Navigation Satellite System, GNSS) 进行定位是不现实的。目前常用的方法是先给部分节点配备 GNSS 以获取其自身位置并辅助其他节点进行定位, 将已知自身位置的节点称为锚节点, 其他节点称为未知节点<sup>[2]</sup>。

目前定位算法可分为无需测距和基于测距两

类<sup>[3]</sup>。前者根据节点间的连通信息估计距离, 后者需要测量节点之间的距离或角度。相比于无需测距的定位算法, 基于测距的定位算法精度更高<sup>[4]</sup>。获得距离后, 使用三边测量、极大似然估计 (Maximum Likelihood Estimation, MLE) 等估计节点位置<sup>[5]</sup>。相比于传统的定位算法, 智能优化算法计算精确、计算复杂度低, 被广泛应用于节点定位中。ISSADV-Hop<sup>[6]</sup> 算法通过修正 DV-Hop 的平均跳距, 并使用佳点集和 Levy 飞行策略对麻雀搜索算法进行改进, 改善了定位精度。AWL-MC<sup>[7]</sup> 算法采用非线性映射曲线距离分析对未知节点进行粗略的相对坐标定位, 再通过线性变换将相对坐标转换成绝对坐标, 最后

项目来源: 国家自然科学基金项目 (62072287); 国家重点研发计划课题项目 (2017YFC0804406); 山东省高等学校青创科技支持计划项目 (2019KJN024)

收稿日期: 2021-08-16 修改日期: 2021-10-28

采用自适应 Levy 飞行优化鲸鱼算法,以提高定位精度。OLSSO<sup>[8]</sup>算法先通过 Bounding-box 方法得到未知节点可能存在的区域以初始化个体,后将加权中心反向学习策略与群居蜘蛛群优化算法相结合,具有较高的定位精度和较快的收敛速度。

目前,鸽群算法(Pigeon Inspired Optimization, PIO)<sup>[9]</sup>已广泛应用于各个领域,PIO 相比于粒子群优化(Particle Swarm Optimization, PSO)<sup>[10]</sup>、蝙蝠算法(Bat Algorithm, BA)<sup>[11]</sup>等,具有收敛速度快、参数较少和稳健性强等优点,但是容易陷入局部最优<sup>[12]</sup>。为此,本文提出了一种基于改进鸽群优化(Modified Pigeon Inspired Optimization, MPIO)的定位算法。本文的主要贡献有:①为了解决 PIO 容易陷入局部最优的问题,设计了一种改进鸽群算法即 MPIO 算法,在保持快速收敛能力的同时,提高了 PIO 的计算精度。②将 MPIO 算法应用于无线传感器网络节点定位以提高性能。

本文组织如下。第一部分介绍了网络模型,第二部分介绍了标准鸽群算法,第三部分介绍了 MPIO 算法及其在定位中的应用,第四部分给出了仿真实验和结果分析,最后第五部分总结了全文。

## 1 网络模型

假设 WSN 由  $N$  个传感器节点组成,节点在部署完后不再移动。锚节点数为  $M$ ,第  $i$  个锚节点的坐标为  $A_i = (x_{ai}, y_{ai})$ 。第  $j$  个未知节点的实际坐标为  $U_j = (x_{uj}, y_{uj})$ ,估计坐标为  $\hat{U}_j = (\hat{x}_{uj}, \hat{y}_{uj})$ 。传感器节点通信范围是一个半径为  $R$  的圆,锚节点向其通信范围内的节点广播信息,所广播的用于辅助未知节点定位的信息称为信标信息,包含锚节点坐标和发送信号强度两个字段。传感器节点使用接收信号强度(Received Signal Strength, RSS)测距, RSS 采用对数-常态分布无线信号传播模型<sup>[13]</sup>,即:

$$P_{rx} = PL(d_0) - 10\eta \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (1)$$

式中: $d_0$  为参考距离,  $PL(d_0)$  为传播距离为  $d_0$  时的接收信号功率。 $d$  为参考节点与未知节点间的距离,  $P_{rx}$  为传播距离为  $d$  时的信号接收功率。 $\eta$  为路径损耗因子,  $X_\sigma$  为均值为 0 标准差为  $\sigma$  的高斯变量。

在实际应用中,由于多径衰弱、信号传播不稳定和噪声等因素,传感器节点的通信范围不是一个规则的圆,所以使用通信不规则度(Degree of Irregularity, DOI)进行建模<sup>[14]</sup>。DOI 值越高,通信范围受环境影响越严重。DOI 定义如下:

$$K_\theta = \begin{cases} 1 & \theta = 0 \\ K_{\theta-1} \pm r \times DOI & 1 \leq \theta \leq 359 \end{cases} \quad (2)$$

式中: $K_\theta$  表示第  $\theta$  个方向上的不规则度,满足  $|K_0 - K_{359}| \leq DOI$   $r \in [0, 1]$  是一个服从均匀分布的随机数。使用 DOI 后的信号传播模型定义为:

$$P_{rx} = PL(d_0) - 10\eta \log\left(\frac{d}{d_0}\right) K_\theta + X_\sigma \quad (3)$$

设锚节点  $A_i$  和未知节点  $U_j$  的实际距离为  $d_{ij}$ ,使用 RSS 测得的距离为  $\hat{d}_{ij}$ 。通过最小化  $d_{ij}$  与  $\hat{d}_{ij}$  之差来达到估计未知节点位置的目的,所以使用智能优化算法定位的适应度函数定义如下:

$$\min f(X) = \frac{1}{n} \sum_{i=1}^n (\|X - A_i\| - \hat{d}_{ij})^2 \quad (4)$$

式中: $n$  为  $U_j$  通信范围内的锚节点数,  $X$  为智能优化算法中的个体位置。

平均定位误差(Average Localization Error, ALE)定义为全部未知节点估计坐标与实际坐标之间欧氏距离的均值,即:

$$ALE = \frac{1}{N-M} \sum_{i=1}^{N-M} \|U_i - \hat{U}_i\| \quad (5)$$

## 2 标准鸽群算法

PIO 是模仿鸽子归巢行为的一种智能优化算法。鸽子归巢可分为两个阶段,在不同阶段使用不同的导航工具。在地图和指南针算子阶段,鸽子通过感知磁场生成地图,把太阳作为指南针调整飞行方向。在飞向目的地的过程中,其对磁场和太阳的依赖性减少。在  $D$  维搜索空间中,第  $i$  只鸽子的位置  $X_i$  和速度  $V_i$  按照下式进行更新:

$$V_i^{t+1} = V_i^t e^{-\alpha} + r(G_{best} - X_i^t) \quad (6)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (7)$$

式中: $t$  为当前迭代次数,  $\alpha \in (0, 1)$  为地图和指南针因子,  $G_{best}$  为全局最优位置。记  $T_{1max}$  为该阶段最大迭代次数,当  $t > T_{1max}$  时,进入下一阶段。

在地标算子阶段,每次迭代时先根据适应度值将鸽子排成非减序,保留前一半鸽子;然后在剩余鸽子中计算加权中心位置作为鸽子飞行的参考方向。鸽子位置更新如下:

$$N_p^t = \frac{N_p^{t-1}}{2} \quad (8)$$

$$X_c^t = \frac{\sum_{i=1}^{N_p^t} X_i^t f(X_i^t)}{N_p^t \sum_{i=1}^{N_p^t} f(X_i^t)} \quad (9)$$

$$X_i^{t+1} = X_i^t + r(X_c^t - X_i^t) \quad (10)$$

式中:  $N_p$  为鸽子总数,  $X_c$  为加权中心位置,  $f(\cdot)$  为适应度函数。记  $T_{2max}$  为该阶段最大迭代次数, 当  $t > T_{2max}$  时, 地标算子停止工作。

### 3 改进鸽群算法

#### 3.1 地图和指南针算子阶段

在该阶段, PIO 算法的全局搜索能力主要由指数权值  $W = e^{-\alpha t}$  决定, 使得鸽群在迭代过程中多样性呈递减趋势并逐步向最优解收敛<sup>[15]</sup>。这种更新方式使算法快速收敛, 但是全局搜索能力较差。地图和指南针因子  $\alpha$  在很大程度上决定了算法全局搜索能力的优劣。图 1 展示了指数权值  $W$  在  $T_{1max} = 50$  时的变化情况。 $\alpha = 0.3$ ,  $t < 15$  时,  $W < 0.75$ , 种群多样性快速减少; 而  $t \geq 15$  时,  $W \approx 0$ , 种群多样性丧失, 算法主要在  $G_{best}$  附近进行局部搜索, 失去了全局搜索能力, 导致过早收敛。 $\alpha = 0.02$ ,  $t = 50$  时,  $W = 0.38$ , 算法不易收敛。 $0.02 < \alpha < 0.3$  时,  $W$  呈指数递减趋势, 种群多样性快速减少, 导致较差的全局搜索能力。

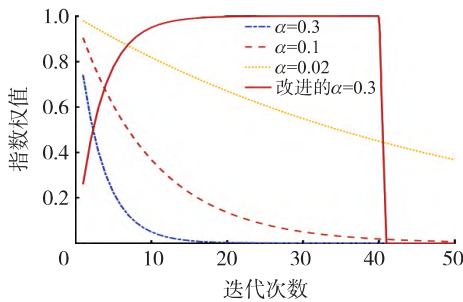


图 1 指数权值随迭代次数变化图

为平衡全局搜索与局部搜索, 改进式(6)为:

$$V_i^{t+1} = \begin{cases} V_i^t (1 - e^{-\alpha t}) + r(G_{best} - X_i^t) & t \leq 0.8T_{1max} \\ V_i^t e^{-\alpha t} + r(G_{best} - X_i^t) & \text{otherwise} \end{cases} \quad (11)$$

$\alpha = 0.3$  时, 改进后的指数权值  $W' = 1 - e^{-\alpha t}$  的变化情况如图 1 所示。 $t \leq 4$  时,  $W' < 0.6$ , 仅在  $G_{best}$  附近搜索, 能够更快地在  $G_{best}$  处收敛;  $4 < t \leq 40$  时,  $0.6 < W' \leq 1$ , 全局搜索能力增强, 保持充分的种群多样性, 能够避免陷入局部最优;  $\omega \approx 0$ , 主要使用局部搜索能力, 能够快速向  $G_{best}$  收敛。

#### 3.2 地标算子阶段

在该阶段, MPIO 不删除鸽子, 而是按照适应度值排序后将鸽子分为两组种群, 在不同种群中用不同方式进行迭代更新。定义:

$$N_p' = \lceil N_p - t \frac{N_p}{2T_{2max}} \rceil \quad (12)$$

前  $N_p'$  只鸽子为一组, 鸽子位置更新公式为:

$$X_c^t = \frac{\sum_{i=1}^{N_p'} X_{pi}^t f(X_{pi}^t)}{\sum_{i=1}^{N_p'} f(X_{pi}^t)} \quad (13)$$

$$X_i^{t+1} = X_{pi}^t + r(X_c^t - X_{pi}^t) \quad (14)$$

式中:  $X_{pi}$  为第  $i$  只鸽子的局部最优位置。

剩余  $N_p - N_p'$  只鸽子分一组, 使用随机游走和捕食机制更新鸽子位置。随机游走是在最优解附近产生局部新解, 在一定程度上能够使算法跳出局部最优, 即:

$$X_i^{t+1} = \begin{cases} G_{best} + \beta E & r > r_i^t \\ X_{pi}^t + r(X_c^t - X_{pi}^t) & \text{otherwise} \end{cases} \quad (15)$$

式中:  $\beta \in [-1, 1]$  是一个随机变量,  $E \in (0, 1)$  为一个常数,  $r_i^t$  为第  $i$  只鸽子在第  $t$  次迭代时的脉冲发射率。

捕食机制中, 鸽子会减少响度  $L_i$ , 增大  $r_i$ , 择优选择鸽子位置并更新  $L_i$  和  $r_i$ , 即:

$$X_{pi}^t = \begin{cases} X_i^t & r < L_i^t \text{ and } f(X_i^t) < f(X_{pi}^t) \\ X_{pi}^{t-1} & \text{otherwise} \end{cases} \quad (16)$$

$$L_i^{t+1} = \varepsilon L_i^t \quad (17)$$

$$r_i^{t+1} = r_i^0 [1 - e^{-\mu t}] \quad (18)$$

式中:  $\varepsilon \in [0, 1]$  为常数, 表示响度衰减系数;  $\mu > 0$  为常数, 表示脉冲增强系数;  $r_i^0 = 0.7$  表示第  $i$  只鸽子的初始脉冲发射率, 设置每只鸽子的响度  $L_i$  初始值为 0.1。

#### 3.3 基于 MPIO 的定位算法

使用 MPIO 进行定位的基本思路是: 锚节点广播信标信息后, 未知节点使用 RSS 测量与邻居锚节点的距离。在测得至少 3 个相邻锚节点的距离、位置后, 便采用 MPIO 算法进行位置计算。具体流程如算法 1 所示。

算法 1: 基于 MPIO 的定位算法

输入: 未知节点  $U$  及其相邻锚节点集合  $A$

输出:  $\hat{U} = (\hat{x}_u, \hat{y}_u)$

1. 初始化  $D, N_p, T_{1max}, T_{2max}, \alpha, E, \varepsilon, \mu$
2. 使用 RSS 估算  $U$  到每个锚节点的距离
3. 随机生成  $N_p$  只鸽子, 第  $i$  只鸽子坐标为  $X_i (i = 1, 2, \dots, N_p)$
4. 使用式(4)计算适应度函数值
5.  $G_{best} \leftarrow$  最佳位置
6. for  $t \leftarrow 1$  to  $T_{1max}$  do
7. 所有鸽子根据式(11)和(7)更新位置
8. 使用式(4)计算适应度函数值
9. 更新  $X_p$  和  $G_{best}$
10.  $t \leftarrow t + 1$
11. end for

12. for  $t \leftarrow 1$  to  $T_{2\max}$  do
13. 根据适应度函数值对鸽子进行排序
14. 根据式 (12) 和 (13) 更新  $N_p$  和  $X_c$
15. 前  $N_p$  只鸽子根据式 (14) 更新位置
16. 其余鸽子根据式 (15) 更新位置
17. 使用式 (4) 计算适应度函数值
18. 更新  $X_p$  和  $G_{\text{best}}$
19.  $t \leftarrow t+1$
20. end for
21. return  $G_{\text{best}}$

算法 1 第 1 行初始化必要参数,第 2 行使用 RSS 测量未知节点与锚节点的距离,第 3~5 行初始化鸽群,第 6~11 行是地图和指南针算子阶段,第 12~20 行是地标算子阶段,每阶段使用不同的公式更新鸽子位置。第 21 行返回  $G_{\text{best}}$  作为未知节点  $U$  的估计位置。

该算法的主体是两个 for 循环,每个 for 循环都是根据上述公式对鸽子位置进行更新,其中第 2 个循环需要对鸽子进行排序,因此其时间复杂度为  $O(N_p T_{1\max} + N_p T_{2\max} \log N_p)$ 。

## 4 仿真实验

本文使用 MATLAB R2016b 进行仿真,所有算法运行 50 次并取均值作为实验结果。在定位精度上将 MPIO 与 MLE、PSO、PIO、BA、GPIO<sup>[16]</sup> 算法相比较。各算法参数设置如表 1 所示,其他实验参数设置如表 2 所示。

表 1 优化算法的参数设置

算法	值
PSO	$N_p = 50$ $T_{\max} = 100$ $\rho_1 = 2$ $\rho_2 = 2$ $\omega = 0.6$
BA	$N_p = 50$ $T_{\max} = 100$ $Q_{\max} = 0.1$ $\varepsilon = 0.9$ $E = 0.1$ $\mu = 0.9$
PIO	$N_p = 50$ $T_{1\max} = T_{2\max} = 50$ $\alpha = 0.1$
GPIO	$N_p = 50$ $T_{1\max} = T_{2\max} = 50$ $\alpha = 0.1$ $d_{\text{low}} = 4$ $\xi = 2$
MPIO	$N_p = 50$ $T_{1\max} = T_{2\max} = 50$ $\alpha = 0.3$ $E = 0.1$ $\varepsilon = \mu = 0.9$

表 2 其他实验参数设置

参数	值
部署区域	100 m × 100 m
$N$	150
$\eta$	3.3
$\sigma$	2
$P_t(d_0)$	55 dB
$d_0$	1 m

### 4.1 定位精度对比

在锚节点数  $M=50$  通信半径  $R=20, 25, 30, 35, 40$  时的结果如图 2 所示。随着  $R$  的增加,平均定位

误差 ALE 随之下降,原因在于  $R$  的增大使得孤立节点的数量大幅减少,距离测量更加准确。BA 算法的 ALE 最大,因为其在较大的部署区域内寻优能力较差;其次是 MLE,它计算简单,但是求解能力较差;其次是 PIO 和 PSO 算法,因为 PSO 和 PIO 都有容易陷入局部最优的局限性;GPIO 通过在距离较近的种群个体上添加高斯变异使种群个体随机发散,并没有有效解决算法陷入局部最优。MPIO 算法的定位精度始终优于其他算法,GPIO、MPIO 相比于 PIO 分别提高了 30%、60%。MPIO 算法具有更优的求解能力,一方面是因为添加了自适应调整机制,使种群个体快速寻找全局最优并提高了种群在最优解附近的局部搜索能力;另一方面是因为种群分组和随机游走机制增加了种群多样性,使算法能够跳出局部最优。

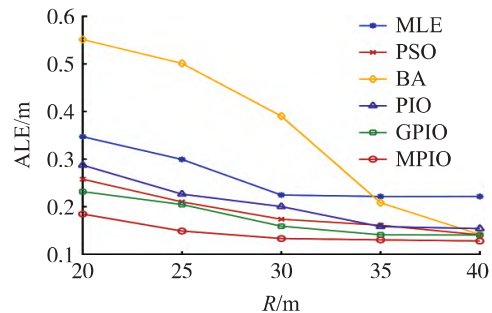


图 2 通信半径对定位误差的影响

在通信半径  $R=30$  锚节点数  $M=40, 45, 50, 55, 60$  时的结果如图 3 所示。随着锚节点数量的增加,未知节点接收到的信标信息更多,平均定位误差 ALE 会随之下降。BA 算法的 ALE 最大,因为其在较大的部署区域内寻优能力较差;其次是 MLE,它计算简单但是求解能力较差;其次是 PIO 和 PSO,因为 PSO 和 PIO 都容易陷入局部最优;GPIO 没有有效改进 PIO 容易陷入局部最优的缺陷。MPIO 算法的定位精度始终优于其他算法,GPIO、MPIO 相比于 PIO 分别提高了 33%、60%。MPIO 算法的自适应调整机制和随机游走机制增加了种群多样性,有效提高了 PIO 算法的求解精度。在相同条件下,MPIO

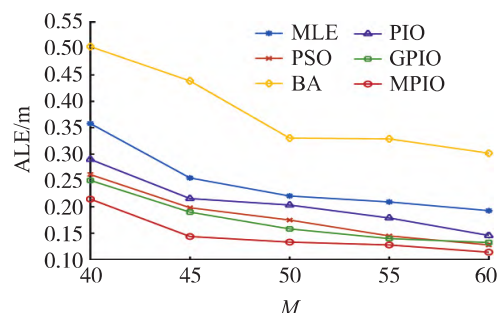


图 3 锚节点数量对定位误差的影响

算法所需的锚节点数量最少,能有效减少网络成本。

在锚节点数  $M=50$ ,通信半径  $R=30$ ,测距误差  $\sigma=2,4,6,8$  时的结果如图 4 所示。随着  $\sigma$  的增加,平均定位误差 ALE 也随之增大。MLE 受测距误差影响最大;其次是 BA 算法,因为其在较大的部署区域内寻优能力较差;其次是 PIO 和 PSO,因为 PSO 和 PIO 都容易陷入局部最优;GPIO 没有有效改进 PIO 容易陷入局部最优的缺陷。MPIO 算法的定位精度始终优于其他算法,GPIO、MPIO 相比于 PIO 分别提高了 27%、49%。MPIO 算法的自适应调整机制和随机游走机制使算法跳出局部最优,有效提高了 PIO 算法的求解精度。相同条件下,MPIO 算法受测距误差的影响最小。

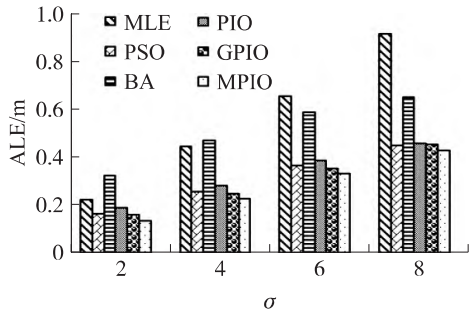


图 4 测距误差对定位误差的影响

在锚节点数  $M=50$ ,通信半径  $R=30$ ,通信不规则度 DOI=0.05,0.1,0.15,0.2,0.25 时的结果如图 5 所示。DOI 越大,节点通信范围越不规则,平均定位误差 ALE 也随之增大。BA 算法受 DOI 影响使得 ALE 最大,因为其在较大的部署区域内求解能力较差;其次是 MLE,它计算简单但是求解能力较差;其次是 PIO 和 PSO,因为 PSO 和 PIO 都容易陷入局部最优;GPIO 没有有效改进 PIO 容易陷入局部最优的缺陷。MPIO 算法的定位精度始终优于其他算法,各算法受 DOI 影响定位误差的波动幅度分别为:MLE 0.79%、PSO 1.12%、BA 4.63%、PIO 1.32%、GPIO 1.34%、MPIO 0.91%。MPIO 算法的自适应调整机制和随机游走机制有效改进了 PIO 算法容易

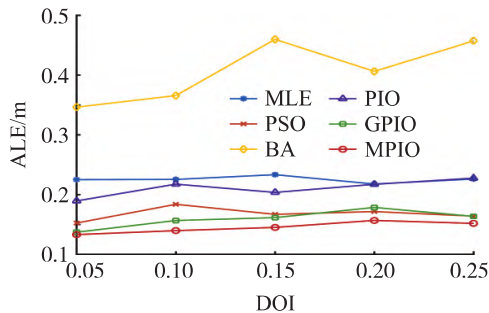


图 5 DOI 对定位误差的影响

陷入局部最优的缺陷。相同条件下,MPIO 算法受 DOI 的影响变化幅度较小。

### 4.2 收敛速度对比

在锚节点数  $M=50$ ,通信半径  $R=30$  时的收敛曲线如图 6 所示。在算法迭代前期 MPIO 算法的收敛最快,其次是 GPIO、PIO、PSO 和 BA。

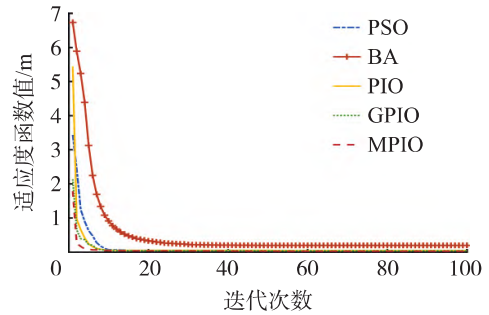


图 6 不同算法收敛曲线

图 7 和表 3 分别显示了在锚节点数  $M=40,45,50,55,60$  时不同定位算法的定位时间和最优适应度值。可以看出随着锚节点数量的增加,定位时间也随之增加,各定位算法平均定位时间依次为 MLE 0.129 s、PIO 0.881 s、PSO 0.92 s、MPIO 0.95 s、BA 1.04 s、GPIO 1.09 s。随着锚节点数量的增加各优化算法的适应度值减小,各优化算法的平均适应度值依次为 MPIO 0.023、GPIO 0.045、PSO 0.049、PIO 0.057、BA 0.323。由于 MLE 算法计算简单所以定位时间最短,但是由 4.1 节分析可知 MLE 算法定位精度较低;BA 具有较长的定位时间和较高的最优适应度值;由于 PIO 收敛速度较快所以定位时间较短,但是由于容易陷入局部最优导致 PIO 的最优适应度值高于 MPIO;由于 GPIO 频繁使用高斯变异增加种群多样性提高算法求解精度,使最优适应度值较低,但是定位时间最长;由于 MPIO 在 PIO 的基础上进行改进来使 PIO 跳出局部最优,所以定位时间较 PIO 长,但是具有最低的适应度值,算法求解精度最高。

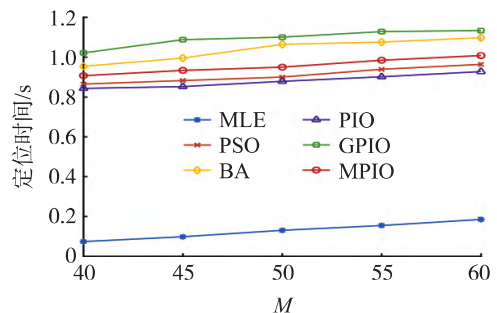


图 7 锚节点数量对定位时间的影响

表 3 各优化算法的最优适应度值

算法	M=40	M=45	M=50	M=55	M=60
PSO	0.057 3	0.052 1	0.050 6	0.046 2	0.039 4
BA	0.493 7	0.380 7	0.292 7	0.268 5	0.182 2
PIO	0.068 2	0.060 5	0.056 9	0.053 7	0.049 1
GPIO	0.053 8	0.049 3	0.045 2	0.039 9	0.038 6
MPIO	0.031 9	0.025	0.023 1	0.020 3	0.018 0

## 5 总结

本文针对 PIO 算法容易陷入局部最优的缺陷,提出了一种基于改进鸽群算法(MPIO)的无线传感网络定位算法。MPIO 的第一阶段通过自适应调整机制来平衡全局搜索与局部搜索能力,并同时保证较快收敛速度;第二阶段保留全部鸽子并分组,使用随机游走等机制增加种群多样性,提高了算法的求解精度。定位过程中首先使用 RSS 进行测距,然后使用 MPIO 进行定位。仿真结果表明,MPIO 算法相比于传统的定位算法和其他的优化算法,具有较高的定位精度和较快的收敛速度。

### 参考文献:

[1] 苗春雨,陈丽娜,吴建军,等.无线传感器网络节点位置验证框架[J].计算机研究与发展,2019,56(6):1231-1243.  
 [2] 石琴琴,徐强,张建平.基于最佳路径匹配的 DV-Hop 改进方法[J].系统仿真学报,2019,31(2):316-323,331.  
 [3] 许佳慧,陈柯宇,程恩.水下传感网络的低复杂度 APIT 算法及 OPNET 仿真实现[J].系统仿真学报,2020,32(1):27-34.

[4] 王群,陈友荣,陈浩,等.基于移动锚点的三维无线传感网节点定位算法研究[J].传感技术学报,2021,34(3):391-399.  
 [5] 高美凤,尹持俊.基于 UAV 的无线传感器网络加权质心定位算法[J].传感技术学报,2021,34(4):539-545.  
 [6] 印雷,顾德,刘飞.基于改进麻雀搜索算法优化的 DV-Hop 定位算法[J].传感技术学报,2021,34(5):670-675.  
 [7] 余修武,李莹,刘永,等.基于映射曲线的自适应莱维鲸鱼无线定位算法[J].北京邮电大学学报,2020,43(5):125-129,136.  
 [8] 余修武,张可,刘永,等.基于反向学习的群居蜘蛛优化 WSN 节点定位算法[J].控制与决策,2021,36(10):2459-2466.  
 [9] Duan H, Qiao P. Pigeon-Inspired Optimization: A New Swarm Intelligence Optimizer for Air Robot Path Planning[J]. International Journal of Intelligent Computing and Cybernetics, 2014, 7(1): 24-37.  
 [10] 徐利锋,黄祖胜,杨中柱,等.引入多级扰动的混合型粒子群优化算法[J].软件学报,2019,30(6):1835-1852.  
 [11] Yang X. A New Metaheuristic Bat-Inspired Algorithm[J]. Computational Intelligence & Technology, 2010, 284: 65-74.  
 [12] 胡耀龙,冯强,海星朔,等.基于自适应学习策略的改进鸽群优化算法[J].北京航空航天大学学报,2020,46(12):2348-2356.  
 [13] 任克强,潘翠敏.融合 RSSI 跳数量化与误差修正的 DV-Hop 改进算法[J].传感技术学报,2020,33(5):718-724.  
 [14] Cui H Q, Liang Y Q, Zhou C A, et al. Localization of Large-Scale Wireless Sensor Networks Using Niching Particle Swarm Optimization and Reliable Anchor Selection[J]. Wireless Communications and Mobile Computing, 2018: 2473875.  
 [15] 陈忠华,刘博,郭瑞,等.基于改进鸽群算法的光伏阵列 MPPT 方法[J].电力系统及其自动化学报,2021,33(8):32-40.  
 [16] He J, Liu Y, Chen B, et al. An Improved Gaussian Pigeon-Inspired Optimization Algorithm [C]//IEEE Congress on Evolutionary Computation, Wellington: IEEE, 2019: 3270-3276.



崔焕庆(1979—),男,汉族,山东泰安人,教授,硕士研究生导师,博士,主要研究领域为无线传感器网络、智能计算等,cuihq@sdust.edu.cn;



张娜(1996—),女,汉族,山东德州人,硕士研究生,主要研究方向为无线传感器网络,1502470685@qq.com;



罗汉江(1968—),男,汉族,山东淄博人,副教授,硕士研究生导师,博士,主要研究方向为物联网、人工智能、海洋物联网、工业互联网等。