

# 一种具有阶段优势的无锚点定位算法

黄以华, 赵汝威, 陈小若

(中山大学信息科学与技术学院电子与通信工程系, 广东广州 510006)

**摘要:** 经典无锚点定位算法存在定位误差大和节点坐标易陷入局部最优的问题. 本文提出一种包含两个阶段的无锚点定位算法: 第一阶段采用跳数量化方法将节点的通信半径长度量化为多跳, 从而建立更小误差的坐标系; 第二阶段采用自适应质点弹簧优化算法优化节点的坐标, 防止节点坐标在迭代优化时陷入局部最优. 仿真结果表明, 本文的算法具有阶段优势, 比传统算法定位精度高, 收敛速度快.

**关键词:** 无线传感器网络; 无锚点定位算法; 质点弹簧模型

**中图分类号:** TN393      **文献标识码:** A      **文章编号:** 0372-2112 (2015)12-2536-06

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2015.12.029

## An Anchor-Free Localization Algorithm with Phase Advantages

HUANG Yi-hua, ZHAO Ru-wei, CHEN Xiao-ruo

(Dept. of Electronics and Communication Engineering, Sun Yat-Sen University, Guangzhou, Guangdong 510006, China)

**Abstract:** The existing classic anchor-free localization (AFL) algorithms have relatively large localization error and tend to get local optimal value. Therefore, a two-phase improved anchor-free localization (I-AFL) algorithm with phase advantages is proposed. In the first phase of the I-AFL algorithm, a precise coordinate system is established through hop-count quantization, which uses the measured distance information to break the length of node communication radius into multiple hops. The second phase of the I-AFL algorithm uses an adaptive mass-spring optimization algorithm to optimize the node position and prevent the position from falling into local optimum value during the iterative optimization process. Both the theoretical analysis and simulation results show that compared with the traditional algorithm, the proposed algorithm is more precise, robust and converges faster.

**Key words:** wireless sensor networks; anchor-free localization algorithm; mass-spring model

## 1 引言

如何快速准确地对无线传感器网络 (Wireless Sensor Networks, WSN) 中的无线节点进行定位是近年来一个热点问题<sup>[1]</sup>. 目前大部分无线节点定位算法属于基于锚点的定位算法, 对无锚点定位算法的研究较少<sup>[2]</sup>. 在某些特殊场合, WSN 中所有节点的位置信息都无法事先获取, 在此情况下, 无锚点定位算法就显得十分重要.

无锚点定位算法分为集中式和分布式算法<sup>[3]</sup>. 集中式算法是将所需信息传送到指定的中心节点进行计算, 典型算法有 MDS-MAP<sup>[4]</sup>. 分布式算法是仅需要节点间的信息交换和协调, 由节点自行计算自身位置. 此类算法容错能力强, 健壮性和拓展性好, 典型算法有 AFL<sup>[5]</sup>. 无锚点定位算法常基于测距<sup>[6]</sup>技术实现定位. 最早分布式无锚点定位算法是由 Priyantha 等人提出的 AFL 算法. AFL 算法第一阶段采用多跳技术, 生成一个全局网络坐标系统, 第二阶段采用质点弹簧优化 (Mass-Spring Optimization, MSO) 算法优化所有节点的位置. 在 AFL 算

法基础上, 一些研究人员利用测量距离信息代替跳数的方法估计节点的位置对 AFL 算法进行了改进, 得到 R-AFL (Refined Anchor Free Localization)<sup>[7]</sup> 算法. 另有一些研究者提出了基于跳数量化的定位算法<sup>[8~10]</sup>, 部分文献给出了与 MSO 算法相关的其它的定位算法<sup>[11~16]</sup>. 本文基于跳数量化和质点弹簧模型, 在原有的 AFL 算法基础上, 提出一种包含两个阶段的定位精度更高的无锚点定位算法 (Improved AFL, I-AFL).

## 2 I-AFL 算法

本文所研究的是所有节点均未知自身位置信息的无锚点 WSN 中的节点定位问题. 不失一般性地, 假设节点之间可以通过测距技术获得距离信息, 每个节点都有自己唯一的 ID, 所有节点具有相同的通信范围, 其最大通信半径为  $R$ . 节点随机布局在二维平面, 节点根据通信信息形成二维网络. 在 WSN 中, 每个节点通信半径以内的其它节点, 称为邻居节点. 网络中所有单个节点在其通信范围  $R$  内的平均节点个数, 称为网络连通度. 假

设全局网络有  $N$  个节点,记第  $i$  个节点为  $n_i$ . 建立邻居节点图  $G=(V, W)$ , 这里节点  $n_i \in V$  代表节点,  $(n_i, n_j) \in W$  代表节点  $n_i$  和节点  $n_j$  互为邻居节点.

## 2.1 选择参考节点

AFL 第一阶段采用一种多跳的方法估计无锚点 WSN 中的节点坐标, 建立初始坐标系. 首先节点使用最短路径算法计算 WSN 中任意两个节点之间的最短路径, 记节点  $n_i$  与节点  $n_j$  之间的最短路径跳数为  $h_{i,j}$ , 其中  $h_{i,j} = h_{j,i}$ . 建立坐标系时, AFL 算法需要在全局网络中选择 5 个节点作为临时参考节点, 步骤如下:

(1) 选择任意节点  $n_0$ , 通常选最小 ID 号的节点, 然后选取满足  $h_{0,1}$  最大的节点  $n_1$ , 并广播节点  $n_1$  的 ID 号;

(2) 选取满足  $h_{1,2}$  最大的节点  $n_2$ , 并广播节点  $n_2$  的 ID 号;

(3) 选取满足  $|h_{1,3} - h_{2,3}|$  最小的节点  $n_3$ , 若有几个节点同时满足最小值的条件, 则在这些节点当中选取满足  $|h_{1,3} + h_{2,3}|$  最大的节点作为  $n_3$ ;

(4) 选取满足  $|h_{1,4} - h_{2,4}|$  最小的节点  $n_4$ , 若有多个同时满足最小值条件的节点, 选择满足  $h_{3,4}$  最大的节点.

(5) 选取满足  $|h_{1,5} - h_{2,5}|$  最小和  $|h_{3,5} - h_{4,5}|$  最小的节点  $n_5$ .

用广播选定的 5 个节点的位置信息, 求出其它节点  $n_i$  到 5 个参考节点的最短路径跳数  $h_{1,i}, h_{2,i}, h_{3,i}, h_{4,i}, h_{5,i}$ , 设节点  $n_i$  到节点  $n_5$  的极坐标为  $(\rho_i, \theta_i)$ , 则节点  $n_i$  的极坐标可由如下公式计算:

$$\rho_i = h_{5,i} \times R \quad (1)$$

$$\theta_i = \tan^{-1} \left( \frac{h_{1,i} - h_{2,i}}{h_{3,i} - h_{4,i}} \right) \quad (2)$$

## 2.2 初始坐标系估计

式(1)和(2)是一种非测距的估计方法, 存在如下问题: 若有两个相近的节点  $n_1$  和  $n_2$ , 由于半径  $R$  范围内的节点都为一跳, 存在  $\rho_1 = \rho_2$  和  $\theta_1 = \theta_2$  的情况, 且  $\rho$  值大于真实距离, 造成节点累积的距离误差较大. 本文利用测量距离信息, 采用类似文献[8~10]中的方法对式(1)和式(2)进行改进.

首先将通信半径  $R$  平均划分为  $\beta$  段, 每段长度为  $R/\beta$ , 即一跳分辨率为  $R/\beta$ . 设  $L_{ij}$  为节点  $n_i$  到节点  $n_j$  间的测量距离, 此时  $h_{i,j} = 1$ . 那么划分半径  $R$  后一跳对应的距离范围是  $0 < L_{ij} < R/\beta$ , 两跳对应的距离范围是  $R/\beta \leq L_{ij} \leq 2R/\beta$ . 同理,  $a$  跳对应的距离范围是  $(a-1)R/\beta \leq L_{ij} \leq aR/\beta$ . 可见, 改进后的跳数可反映相邻节点间距离的大小, 且能容忍一定范围内的测量误差. 在此方法下, 相邻节点根据测量距离分别计算跳数, 记节点  $n_i$  到节点  $n_j$  的最短路径跳数为  $H_{i,j}$ , 最短路径

距离为  $D_{i,j}$ , 则有:

$$H_{i,j} = \sum_{i:i \rightarrow j, i \neq j} \left[ L_{ij} \times \frac{R}{\beta} \right] \quad (3)$$

$$D_{i,j} = \sum_{i:i \rightarrow j, i \neq j} L_{ij} \quad (4)$$

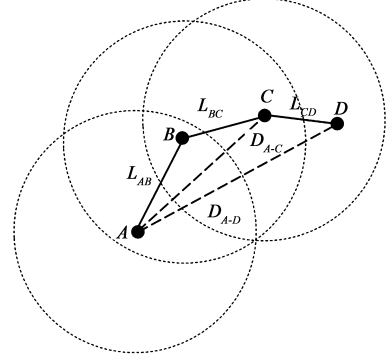


图1 4个节点的最短路径示例

如图1所示, 圆为节点的通信范围, 实线代表邻居节点间最短距离, 虚直线代表非邻居节点间最短距离,  $L_{AB}$  和  $L_{BC}$  满足  $R < L_{AB} + L_{BC} \leq 2R$ , 其中必有一段距离大于  $R/2$ , 即  $L_{AB} > R/2$  或  $L_{BC} > R/2$ . 若  $L_{AB}$  对应的跳数为  $H_{A,B}$ ,  $L_{BC}$  对应的跳数为  $H_{B,C}$ , 则有  $\beta < H_{A,B} + H_{B,C} \leq 2\beta$ , 其中  $H_{A,B} > \beta/2$  或  $H_{B,C} > \beta/2$ . 图1中两条虚线  $D_{A-C}$  和  $D_{A-D}$  代表两点的实际距离, 显然  $D_{A-C} \leq H_{A,C} \cdot R/\beta \leq 2R$  和  $D_{A-D} \leq H_{A,D} \cdot R/\beta \leq 3R$ . 若  $L_{ij}$  不存在测量误差, 则有  $D_{A-C} \leq D_{A,C} \leq H_{A,C} \cdot R/\beta \leq 2R$  和  $D_{A-D} \leq D_{A,D} \leq H_{A,D} \cdot R/\beta \leq 3R$ . 若从节点  $n_i$  到节点  $n_j$  的路径上有  $b$  个节点, 则有:

$$D_{i-j} \leq D_{i,j} \leq H_{i,j} \cdot R/\beta \leq (b-1)R \quad (5)$$

从式(5)可知, 采用量化的方法估计距离可减少节点间估计距离累积的误差, 比直接计算半径  $R$  的方法更准确. 因此, 式(1)可修正为:

$$\rho'_i = \lambda (H_{5,i} \times \frac{R}{\beta}) \quad (6)$$

其中参数  $\lambda$  是缩放因子, 定义为:

$$\lambda = \frac{1}{2} \left( \frac{D_{1,2}}{H_{1,2} \times (R/\beta)} + \frac{D_{3,4}}{H_{3,4} \times (R/\beta)} \right) \quad (7)$$

其中,  $D_{1,2}$  和  $D_{3,4}$  表示两参考节点间的最短路径距离, 显然  $\lambda \leq 1$ . 在测距存在误差的情况下, 对于较小的跳数, 存在  $\lambda > 1$ . 参数  $\lambda$  的存在能够保证估计的坐标系相对于真实的坐标系是误差较小的放大模型, 在优化算法迭代过程中节点移动时不容易出现翻转问题. 参数  $\lambda$  也对坐标系上 5 个参考节点的坐标进行缩放调整. 结合式(7), 式(6)可化为:

$$\rho'_i = \frac{1}{2} \left( \frac{D_{1,2}}{H_{1,2}} + \frac{D_{3,4}}{H_{3,4}} \right) H_{5,i} \quad (8)$$

此时式(2)变为:

$$\theta'_i = \tan^{-1}\left(\frac{H_{1,i} - H_{2,i}}{H_{3,i} - H_{4,i}}\right) \quad (9)$$

因此,未知节点  $n_i$  可以通过新的极坐标公式计算出坐标估计值为  $(\rho'_i \cdot \cos\theta'_i, \rho'_i \cdot \sin\theta'_i)$ .

### 2.3 AMSO 算法

在 I-AFL 算法第一阶段完成后,网络中所有节点仍然存在较大位置误差,需要对所有节点的位置进行优化.如图 2 所示,任选一个节点  $n_i, i \in \{1, 2, \dots, N\}$ , 节点  $n_i$  的邻居节点为  $n_j, j \in \{1, 2, \dots, m_i\}$ , 其中  $m_i$  为节点  $n_i$  的邻居节点个数,  $(n_i, n_j) \in W$ . 记节点  $n_i$  为邻居节点  $n_j$  的中心节点. AMSO 算法步骤如下:正向传播阶段,中心节点将位置信息发送到邻居节点,邻居节点根据接收到的信息计算节点间的距离误差.若存在距离误差,则调整邻居节点的位置;反向传播阶段,若调整后的邻居节点与中心节点的距离误差不为零,则将邻居节点位置信息发送到中心节点,此时中心节点将根据距离误差矢量和来调整自身位置. AMSO 算法的正向和反向传播阶段构成闭环调整系统,中心节点能够根据邻居节点的反馈信息自适应地调整至稳定状态.

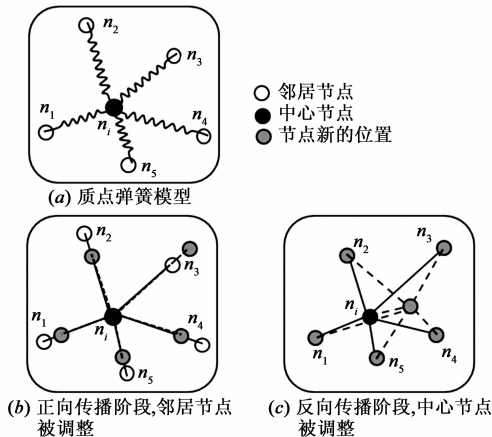


图2 AMSO算法执行过程

AMSO 算法的执行过程如图 2 所示,图 2(a)为任意节点  $n_i$  与它的邻居节点的质点弹簧模型.  $n_i$  与其邻居节点间的测量距离与估计距离的误差大小视为弹簧力,若估计距离大于测量距离,则弹簧力为拉力,若估计距离小于测量距离,则弹簧力为推力.令节点  $n_i$  的坐标为  $(x_i, y_i)$ , 节点  $n_j$  的坐标为  $(x_j, y_j)$ . 设  $d_{ij}$  为节点  $n_i$  和节点  $n_j$  的估计距离, 其中,  $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ . 如前面所述,  $L_{ij}$  为节点  $n_i$  和节点  $n_j$  的测量距离. 设  $\mathbf{v}_{ij}$  是从节点  $n_i$  到节点  $n_j$  的单位矢量, 则  $\mathbf{v}_{ij} = ((x_j - x_i)/d_{ij}, (y_j - y_i)/d_{ij})$ , 其中  $\mathbf{v}_{ij} = -\mathbf{v}_{ji}$ . 此时中心节点  $n_i$  到邻居节点  $n_j$  的弹簧力定义为:

$$\mathbf{F}_{ji} = \mathbf{v}_{ji}(d_{ij} - L_{ij}) \quad (10)$$

中心节点  $n_i$  受到的弹簧合力定义为:

$$\mathbf{F}_i = \sum_{j=1}^{m_i} \mathbf{F}_{ij} \quad (11)$$

中心节点  $n_i$  与所有邻居节点的能量定义为:

$$E_i = \sum_{j=1}^{m_i} (d_{ij} - L_{ij})^2 \quad (12)$$

那么全局定位问题转化为求全局能量的最小, 即

$$\arg \min \frac{1}{2} \sum_{i=1}^N E_i.$$

对于分布式定位,上述定位问题可以转化为寻求各个节点能量  $E_i$  最小值. 为了使能量  $E_i$  快速减少,如图 2(b)所示,可将中心节点  $n_i$  的位置信息传播给它的邻居节点,通过同时调整邻居节点的位置来减少误差.

从邻居节点  $n_j$  的角度来看,节点  $n_j$  到节点  $n_i$  的弹簧力为:

$$\mathbf{F}_{ji} = \mathbf{v}_{ji}(d_{ij} - L_{ij}) \quad (13)$$

根据文献[5]的描述,节点移动的步长是弹簧力的  $1/(2m_i)$ . 设节点  $n_j$  的邻居节点数目为  $m_j$ , 则取邻居节点  $n_j$  的更新步长为:

$$\mathbf{S}_{ji} = \mathbf{F}_{ji}/(2m_j), j \in \{1, 2, \dots, m_i\} \quad (14)$$

当所有邻居节点都更新了  $\mathbf{S}_{ji}$  之后,尽管中心节点  $n_i$  与邻居节点的距离误差减小,但是中心节点  $n_i$  可能处于不平衡状态. 为解决此问题,在式(14)调整这些邻居节点位置之后,将已更新的邻居节点的位置信息反向传播到中心节点  $n_i$  中,然后中心节点  $n_i$  根据此时的弹簧合力调整自身位置,如图 2(c)所示,此时节点  $n_i$  更新步长为:

$$\mathbf{S}_i = \mathbf{F}_i/(2m_i) \quad (15)$$

迭代执行式(14)和式(15)调整全局网络的节点位置.迭代的终止条件采用与文献[5]类似的方法.为保证节点的更新速度一致,节点在第一次执行 AMSO 算法之后,下一次的 AMSO 算法执行时间应该是在所有邻居节点都执行了一次 AMSO 算法之后. AMSO 算法的伪代码如下所示:

#### 算法 1 节点 $n_i$ 的 AMSO 算法

令中心节点  $n_i$  和邻居节点  $n_j$  的位置分别为  $P_i, P_j$

令 AMSO 算法迭代中止次数为  $T$

while  $k < T$  do

/\* 正向传播阶段 \*/

发送  $P_i$  到所有邻居节点

for  $j \in \{1, 2, \dots, m_i\}$  do

$\mathbf{F}_{ji} = \mathbf{v}_{ji}(d_{ij} - L_{ij})$

if  $\mathbf{F}_{ji} \neq \mathbf{0}$  then

更新节点  $n_j$  的位置  $P_j = P_j + \frac{\mathbf{F}_{ji}}{2m_j}$

end if

```

end for
/* 反向传播阶段 */
for  $j \in \{1, 2, \dots, m_i\}$  do
    节点  $n_i$  收集邻居节点的位置信息  $\{p_j, j \in \{1, 2, \dots, m_i\}\}$ 
end

$$\mathbf{F}_i = \sum_{j=1}^{m_i} \mathbf{v}_{ij}(d_{ij} - L_{ij})$$

更新节点  $n_i$  的位置  $P_i = P_i + \frac{\mathbf{F}}{2m_i}$ 
/* 检测阶段 */
while  $j \in \{1, 2, \dots, m_i\}$  do
    等待节点  $n_j$  执行一次 AMSO 算法
end while
 $k = k + 1$ 
end while

```

若所有节点没有测量距离功能,仅有连通信息.此时可采用通信范围信息  $R$  代替测量距离,只需将式(10)、(11)分别修改为:

$$\mathbf{F}'_{ij} = \mathbf{v}_{ij}(d_{ij} - R) \quad (16)$$

$$\mathbf{F}'_i = \sum_{j=1: d_{ij} > R}^{m_i} \mathbf{F}'_{ij} \quad (17)$$

### 3 AMSO 算法性能分析

#### 3.1 迭代法的收敛性

AMSO 算法步骤分为两个阶段,正向传播阶段主要是调整两个节点的相对距离,中心节点  $n_i$  与邻居节点  $n_j$  之间估计距离的迭代方程为:

$$d_{ij}(k+1) = \left(1 - \frac{1}{2m_j}\right) d_{ij}(k) + \frac{1}{2m_j} L_{ij} \quad (18)$$

其中,  $k$  为算法迭代次数. 设式(18)的解为  $d_{ij}^*$ , 则有:

$$d_{ij}^* = \left(1 - \frac{1}{2m_j}\right) d_{ij}^* + \frac{1}{2m_j} L_{ij} \quad (19)$$

由式(18)和式(19)得到距离误差估计式为:

$$\epsilon(k+1) = \left(1 - \frac{1}{2m_j}\right)^{k+1} [d_{ij}(0) - d_{ij}^*] \quad (20)$$

对任意初始值  $d_{ij}(0)$  有:

$$\lim_{k \rightarrow \infty} \epsilon(k+1) = \left(1 - \frac{1}{2m_j}\right)^{k+1} [d_{ij}(0) - d_{ij}^*] = 0 \quad (21)$$

可见,两个节点间估计距离的迭代方程收敛,其充要条件是  $|1 - 1/(2m_j)| < 1$ . 反向传播阶段主要是调整节点  $n_i$  的平衡位置,中心节点  $n_i$  位置迭代方程为:

$$p_i(k+1) = p_i(k) + \frac{1}{2m_i} \sum_{j=1}^{m_i} \mathbf{v}_{ij} [d_{ij}(k) - L_{ij}] \quad (22)$$

设式(22)的解为  $p_i^*$ , 由式(19)可得  $d_{ij}^* = L_{ij}$ , 则

$\lim_{k \rightarrow \infty} d_{ij}(k) = L_{ij}$ . 那么

$$\lim_{k \rightarrow \infty} p_i(k+1) = \lim_{k \rightarrow \infty} p_i(k) = p_i^* \quad (23)$$

此时位置迭代方程也是收敛的,因此 AMSO 算法是

收敛的.

#### 3.2 收敛速度

假设网络中节点  $n_i \in V$  的邻居节点个数为  $m$  (此时  $m_i = m_j = m$ ), 通常取  $m \geq 3$ . 设中心节点  $n_i$  的弹簧合力为  $\mathbf{F}$ . AMSO 算法先以  $\mathbf{S}_j$  的步长更新所有邻居节点, 再以  $\mathbf{S}_i$  的步长更新中心节点. 那么, 节点  $n_i$  执行 AMSO 算法一次后, 共有  $(m+1)$  个节点被更新, 从单个节点  $n_i$  的角度来考虑, 误差下降为:

$$e_{\text{AMSO}} = |\mathbf{F}/m| - |\mathbf{F}/(4m^2)| \quad (24)$$

而 MSO 算法执行一次时, 只有中心节点  $n_i$  被更新, 误差下降为:

$$e_{\text{MSO}} = |\mathbf{F}/(2m)| \quad (25)$$

则两误差下降比率为:

$$\eta = \frac{e_{\text{AMSO}}}{e_{\text{MSO}}} = 2 - \frac{1}{2m} \quad (26)$$

则在相同的迭代次数下, AMSO 算法收敛速度是 MSO 算法的  $2 - 1/(2m)$  倍. 显然, 全局迭代更新时, AMSO 算法会比 MSO 算法收敛得快.

#### 3.3 算法复杂度

由于 I-AFL 算法第一阶段和原来 AFL 算法的通信复杂度和计算复杂度相同. 因此算法复杂度的分析主要针对优化算法部分.

**通信复杂度:** AMSO 算法正向传播阶段是以广播方式传播信息, 通信复杂度为  $O(1)$ , 反向传播阶段需要将  $m$  个邻居节点的位置信息发送给中心节点, 复杂度为  $O(m)$ . 因此 AMSO 通信复杂度为  $O(m+1)$ . 而 MSO 算法仅执行 AMSO 算法的反向传播阶段, 通信复杂度  $O(m)$ . 因此 AMSO 算法相对于 MSO 算法增加了少量的通信量.

**计算复杂度:** 假设 AMSO 算法和 MSO 算法都迭代一次. AMSO 算法正向传播阶段, 所有邻居节点都进行调整, 每个邻居节点计算方式相同, 此时计算复杂度为  $O(m)$ . 反向传播阶段, 中心节点的弹簧合力是通过计算每对节点的弹簧力而得到, 计算复杂度为  $O(m)$ , 则 AMSO 算法的计算复杂度为  $O(2m)$ . MSO 算法中每个节点仅通过弹簧合力进行调整, 显然 MSO 算法的计算复杂度为  $O(m)$ . 因此 AMSO 算法计算复杂是 MSO 算法的 2 倍.

#### 3.4 误差分析

由式(24)和式(25)可知, 在相同的初始网络下, 当两种优化算法执行相同的迭代次数时, AMSO 算法比 MSO 算法得到的网络的误差更小.

### 4 仿真分析

假设节点随机部署在  $100\text{m} \times 100\text{m}$  区域, 节点数分

别取 50, 75, 100, 125, 150, 175, 200, 225, 250<sup>[5]</sup>, 网络连接度大于 3, 节点的通信半径  $R$  为 20m. 根据文献[5]中 MSO 算法的实验, 取 AMSO 算法和 MSO 算法迭代中止次数为 50. 为了评估算法性能, 本文采用两种性能指标: 平均定位误差 (mean localization error) 和全局能量比率 (Global Energy Ratio, GER) 误差<sup>[5]</sup>.

平均定位误差是描述节点当前位置与真实位置之间的误差, 定义如下:

$$\text{Error}(\%R) = \sum_{i=1}^N \sqrt{(x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2} / (NR) \quad (27)$$

其中  $N$  是网络节点总数,  $(\bar{x}_i, \bar{y}_i)$  是节点  $n_i$  的真实坐标.

GER 误差定义如下:

$$\text{GER} = 2\sqrt{\sum_{i,j;i < j} \hat{e}_{ij}^2} / N(N-1) \quad (28)$$

其中  $\hat{e}_{ij} = (d_{ij} - L_{ij}) / L_{ij}$ .

#### 4.1 I-AFL 算法第一阶段中参数 $\beta$ 的影响

图 3 显示了 I-AFL 算法中参数  $\beta$  在不同节点密度下的误差. 本文  $\beta = 1, 2, \dots, 8$ , 通过式(3)计算新的跳数. AFL 算法第一阶段记为 AFL-1. 由图可见 AFL-1 的平均定位误差和 GER 误差都相当大, 这是由于 AFL-1 直

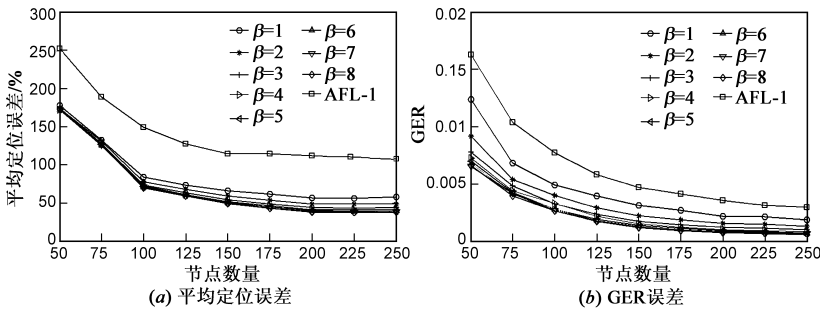


图3 I-AFL算法中参数 $\beta$ 在不同节点密度下的误差

图 5 是 AMSO 和 MSO 算法在不同的网络连通度下的性能对比, (a) 图和 (b) 分别加入 10% $R$  和 50% $R$  的位置误差. 显然两种优化算法的 GER 误差随着网络连通度的增大而减小, 但是 AMSO 算法比 MSO 算法的 GER 误差要小, 因此在网络连通度的较低的情况下, AMSO 算法比 MSO 算法更有优势.

#### 4.3 I-AFL 算法的性能

图 6 为不同节点密度下 I-AFL, AFL, R-AFL 和 MDS-MAP 四种不同定位算法的 GER 误差. 此时不同的节点密度对应的网络连通度分别是 6.44, 8.24, 10.55, 13.13, 15.73, 18.32, 20.94, 23.55 和 26.2. 由图 6 可以知, I-AFL 算法的定位误差随着节点密度或网络连通度的增

接采用半径  $R$  作为一跳所累积的距离误差较大.  $\beta = 1$  曲线表示 AFL-1 中所有节点经过缩放因子  $\lambda$  调整后的误差, 明显比原来 AFL-1 的误差要小.  $\beta > 1$  的曲线表示 I-AFL 算法的第一阶段的定位误差. I-AFL 第一阶段是跳数量化方法和缩放因子  $\lambda$  调整共同作用的结果, 网络的平均定位误差和 GER 误差都小. 图中  $\beta = 6, 7, 8$  的三条曲线几乎重合, 说明参数  $\beta = 6, 7, 8$  对 I-AFL 算法的定位效果几乎一样. 因此, 本文选取  $\beta = 8$  为本文 I-AFL 算法第一阶段的参数. 当  $\beta = 8$  时, 在不同的节点密度下, I-AFL 第一阶段相对于 AFL-1 平均定位误差下降了 54.08%, GER 误差下降了 71.68%. 因此, I-AFL 比 AFL-1 的定位方法更有效.

#### 4.2 I-AFL 算法第二阶段中优化算法的性能

图 4 为 AMSO 和 MSO 算法在两种不同节点密度 (将 50 和 250 个节点分别随机布局在 100m  $\times$  100m 区域), 下迭代速度的比较, 网络中加入 50% $R$  的位置误差. 显然, 在两种不同节点密度的情况下, AMSO 算法均比 MSO 算法收敛要快, 但是付出的代价是增加了计算复杂度和通信复杂度. 因此, 为了减少 AMSO 算法的复杂度, 在  $N$  个节点布局下, AMSO 算法的迭代次数可取  $\lceil 22.5 - 0.05N \rceil$ ; 在网络连通度为  $C$  时, AMSO 算法的迭代次数可取  $\lceil 23.28 - 0.51C \rceil$ .

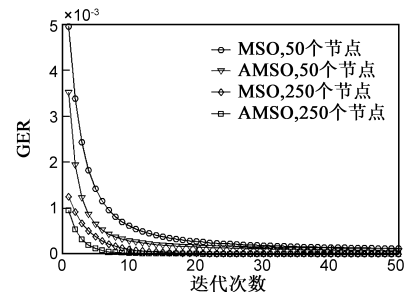
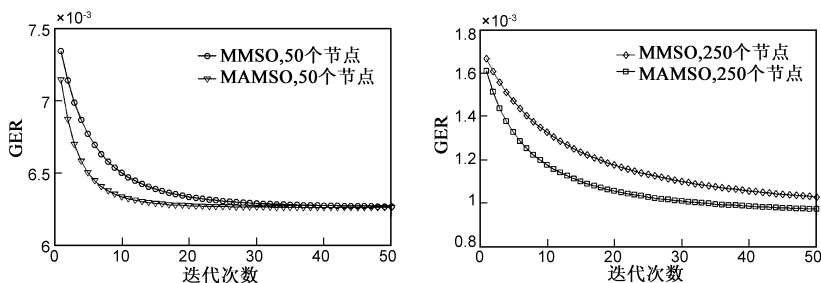


图4 AMSO和MSO算法在两种不同节点密度下迭代速度的比较

加而下降, 相比经典的 AFL, MDS-MAP 和改进的 R-AFL 算法定位误差更小. 因此本文提出的 I-AFL 算法定位效果优于其它三种算法.

## 5 结论

本文提出的 I-AFL 算法第一阶段采用基于跳数量化的方法将节点的通信半径长度量化为多跳, 充分利用测量距离信息建立更小误差的坐标系统, 第二阶段采用分布式 AMSO 算法优化网络中所有节点的位置, 提高了算法收敛速度, 并使收敛后的定位误差更小. 仿真结果表明, I-AFL 算法在不同的节点密度和不同的网络连通度下具有良好的性能.



(a) 50个节点时AMSO算法与MSO算法的GER误差 (b) 250个节点时AMSO算法与MSO算法的GER误差

图5 AMSO和MSO算法在不同的网络连通度下的性能

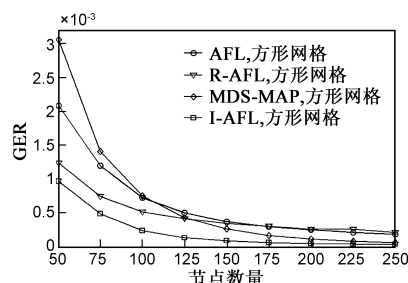


图6 不同节点密度下四种不同定位算法的GER误差

## 参考文献

- [1] Akyildiz I F, Su W, et al. Wireless sensor networks: a survey [J]. *Computer Networks*, 2002, 38(4): 393 – 422.
- [2] Franceschini F, Galetto M, et al. A review of localization algorithms for distributed wireless sensor networks in manufacturing [J]. *International Journal of Computer Integrated Manufacturing*, 2009, 22(7): 698 – 716.
- [3] Pal A. Localization algorithms in wireless sensor networks: Current approaches and future challenges [J]. *Network Protocols and Algorithms*, 2010, 2(1): 45 – 73.
- [4] Shang Y, Ruml W, Zhang Y, et al. Localization from mere connectivity [A]. *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing* [C]. New York: ACM, 2003. 201 – 212.
- [5] Priyantha N B, Balakrishnan H, et al. Anchor-free distributed localization in sensor networks [A]. *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems* [C]. New York: ACM, 2003. 340-341.
- [6] Mao G, Fidan B, Anderson B. Wireless sensor network localization techniques [J]. *Computer Networks*, 2007, 51(10): 2529 – 2553.
- [7] Stephen M J, Reddy P, Chittineni S P. Refined anchor free localization for wireless sensor monitoring in wireless sensor networks [J]. *Indian Journal of Computer Science and Engineering*, 2011, 2(2): 196 – 201.
- [8] Li X, Shi H, Shang Y. A partial-range-aware localization algorithm for ad-hoc wireless sensor networks [A]. 29th Annual IEEE International Conference on Local Computer Networks [C]. Florida: IEEE, 2004. 77 – 83.
- [9] Yang S, Yi J, Cha H. Hcrl. A hop-count-ratio based localization in wireless sensor networks [A]. SECON'07. 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks [C]. San Diego: IEEE, 2007. 31 – 40.
- [10] Gu X, Jiang Y, Shi S. A novel distance-assisted localization algorithm in wireless sensor networks [A]. 2010 International Conference on Computer Design and Applications (ICDDA) [C]. Qinhuangdao: IEEE, 2010, 2. V2-575 – V2-579.
- [11] Xunxue C, Zhiguan S, Jianjun L. Distributed localization for anchor-free sensor networks [J]. *Journal of Systems Engineering and Electronics*, 2008, 19(3): 405 – 418.
- [12] Tran D A, Nguyen T. Localization in wireless sensor networks based on support vector machines [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2008, 19(7): 981 – 994.
- [13] Seet B C, Zhang Q, Foh C H, et al. Hybrid RF mapping and Kalman filtered spring relaxation for sensor network localization [J]. *IEEE Sensors Journal*, 2012, 12(5): 1427 – 1435.
- [14] Yu J, Kulkarni S R, Poor H V. A distributed spring model algorithm for sensor localization using dimension expansion and hyperbolic tangential force [A]. 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) [C]. San Juan: IEEE, 2011. 381 – 384.
- [15] 张会清, 石晓伟, 等. 基于 BP 神经网络和泰勒级数的室内定位算法研究 [J]. *电子学报*, 2012, 40(9): 1876 – 1879. ZHANG Hui qing, SHI Xiao-wei, et al. Research on indoor location technology based on back propagation neural network and Taylor series [J]. *Acta Electronic Sinica*, 2012, 40(9): 1876 – 1879. (in Chinese)
- [16] 孟颖辉, 陈剑, 等. 基于贪婪思想的二阶段 WSN 定位算法 [J]. *电子学报*, 2014, 42(2): 328 – 334. MENG Ying-hui, CHEN Jian, et al. Two-stage localization algorithm based on greedy idea for wireless sensor networks [J]. *Acta Electronic Sinica*, 2014, 42(2): 328 – 334. (in Chinese)

## 作者简介

黄以华 男, 1973 年 4 月出生于福建省福清市. 现为中山大学电子与通信工程系教授. 从事近场通信、芯片与系统设计及固态电子器件等方面的研究工作. 在国内外发表学术论文 20 余篇.

E-mail: huangyih@mail.sysu.edu.cn

赵汝威 男, 1991 年 1 月出生于广东省江门市. 2012 年毕业于五邑大学电子系. 现为中山大学电子与通信工程系硕士研究生.

E-mail: 308979062@qq.com