

# 控制与决策

Control and Decision

## 基于超级节点的分布式传感器节点定位算法

蒋俊正, 赵海兵

引用本文:

蒋俊正, 赵海兵. 基于超级节点的分布式传感器节点定位算法[J]. *控制与决策*, 2020, 35(12): 2898–2906.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0219>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 一种基于双编码遗传算法的机动微波接力网组网方法

Mobile microwave relay network construction method based on double coding genetic algorithm

*控制与决策*. 2020, 35(12): 2915–2922 <https://doi.org/10.13195/j.kzyjc.2019.0347>

### 基于操作风险的双模式传感器管理方法

A dual-mode sensor management method based on operational risk

*控制与决策*. 2020, 35(12): 2993–2998 <https://doi.org/10.13195/j.kzyjc.2018.1541>

### 双层相依网络化指挥信息系统级联失效研究

Cascading failure of double layer networked command information system

*控制与决策*. 2020, 35(12): 3017–3025 <https://doi.org/10.13195/j.kzyjc.2019.0696>

### 一种新的基于标签传播的复杂网络重叠社区识别算法

A novel algorithm for overlapping community detection based on label propagation in complex networks

*控制与决策*. 2020, 35(11): 2733–2742 <https://doi.org/10.13195/j.kzyjc.2019.0176>

### 微型无人机集群低时延组网规划方法

A low delay networking planning method for micro UAV swarm

*控制与决策*. 2020, 35(11): 2696–2706 <https://doi.org/10.13195/j.kzyjc.2018.1549>

# 基于超级节点的分布式传感器节点定位算法

蒋俊正<sup>†</sup>, 赵海兵

(桂林电子科技大学 信息与通信学院, 广西 桂林 541004)

**摘要:** 针对无线传感器网络中数目庞大的传感器节点难以进行有效定位的问题, 提出一种分布式的传感器节点迭代定位算法. 基于整个网络中相互重叠的子图, 该算法的每一步迭代涉及两个步骤: 一是每个子图内的高效定位, 二是相邻子图之间的局部一致. 对于每个子图, 采用共轭梯度法对节点进行局部定位; 之后, 对相邻子图重叠区域内节点的局部位置进行融合平均. 这两个步骤持续进行, 直至满足迭代终止条件. 仿真实验表明, 与现有分布式算法相比, 所提出算法的定位误差降低了一个数量级, 能够对大规模的无线传感器网络进行高效定位.

**关键词:** 无线传感器网络; 定位; 分布式; 迭代; 超级节点; 共轭梯度法; 子图间融合

中图分类号: TN911.7

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0219

引用格式: 蒋俊正, 赵海兵. 基于超级节点的分布式传感器节点定位算法 [J]. 控制与决策, 2020, 35(12): 2898-2906.

## A distributed sensor nodes localization algorithm based on super nodes

JIANG Jun-zheng<sup>†</sup>, ZHAO Hai-bing

(School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China)

**Abstract:** A distributed algorithm based on super nodes is proposed to iteratively locate the large number of sensor nodes in wireless sensor networks. Based on the overlapped decomposition of the entire network, two steps are involved at each iteration of the algorithm, one is the node localization method within each subgraph and the other is the local consensus among neighboring subgraphs. In specific, for each subgraph, the conjugate gradient method is employed to determine the position of nodes. And in subsequent, the position of each node will be adjusted by using a neighboring consensus strategy. These two steps proceed until the iterative termination condition is satisfied. Simulation results show that the proposed algorithm has an order of magnitude lower positioning error than the existing distributed algorithms and can efficiently locate nodes for large size wireless sensor networks.

**Keywords:** wireless sensor network; localization; distributed; iteration; super nodes; conjugate gradient method; fusion between subgraphs

## 0 引言

无线传感器网络 (wireless sensor network, WSN) 是由大量具有特定功能和简单计算能力的传感器构成的自组织网络, 可以对指定环境区域进行实时监测且以无线的方式进行通信, 已在军事、环境、医学、工业、农业等领域得到了广泛的应用<sup>[1-3]</sup>. 在众多应用场景中, 传感器感知参数的可靠性和传感器本身位置的精度是衡量 WSN 性能的两个重要指标. 尤其是在 WSN 执行监测任务时, 获知了传感器的位置, 采集回来的信息将会更有价值<sup>[3]</sup>. 因此, 对于 WSN 中传感器节点定位技术的研究很有意义.

在传感器中添加中国北斗卫星导航系统 (Beidou navigation satellite system, BDS) 或美国全球定位系统 (global positioning system, GPS) 接收模块, 便可直接

借助于导航定位系统获取节点位置. 然而, 添加定位模块不仅增加了传感器本身的功耗, 而且还会缩短 WSN 的使用寿命. 特别是对于大规模的 WSN, 需要考虑投入成本的问题<sup>[4]</sup>. 由于传感器很可能被部署在环境复杂多变的高山密林地帯, 当 BDS 和 GPS 信号受到山体、密林等阻碍时, 传感器节点将无法借助于导航系统进行定位<sup>[5]</sup>. 因此, 通常只在少数传感器中添加定位模块, 并将其尽可能地部署在能够接收 BDS 或 GPS 信号的位置, 这部分传感器节点称为已知位置 (location-aware, LA) 节点, 其他节点则称为未知位置 (location-unaware, LU) 节点<sup>[4]</sup>. 要想实现对未知位置节点的定位, 需要通过到达时间 (time-of-arrival, TOA)、到达时间差 (time-difference-of-arrival, TDOA)、到达角 (angle-of-arrival, AOA) 和接收信号强

收稿日期: 2019-02-28; 修回日期: 2019-08-15.

基金项目: 国家自然科学基金项目 (61761011); 广西省自然科学基金项目 (2017GXNSFAA198173).

<sup>†</sup>通讯作者. E-mail: jzjiang@guet.edu.cn.

度 (received-signal-strength, RSS) 等方法对节点进行测距, 再结合 LA 节点位置和节点间距离信息, 采用相应的算法对 LU 节点进行定位. 具体算法有三点定位算法、三角测量法、最大似然估计法、极小极大定位算法等<sup>[6-7]</sup>.

在现有的众多定位算法中, 依据定位时信息处理的实现方式, 可划分为集中式的定位算法和分布式的定位算法<sup>[8]</sup>. 例如文献[9-10]均为集中式的定位算法, 其特点是利用中央服务器存储和处理数据, 虽然定位精度较高, 但通信量较大, 对服务器的计算能力要求较高. 文献[9]采用半正定规划 (semi-definite programming, SDP) 松弛, 同时引入了一个正则项, 可有效降低优化变量的个数, 并采用梯度下降法细化节点位置, 最终提高了定位的准确性. 但是, 正则项系数的选取比较繁琐, 计算的复杂度较高. 文献[10]在不引入正则项的情况下采用了两步法, 首先利用三点定位法进行粗略的初步定位; 然后基于三点定位得出的初始值, 结合二阶泰勒近似给出的修正海森矩阵, 采用修正牛顿法对定位问题进行优化求解, 达到了良好的定位效果. 但是, 优化过程中需要矩阵求逆的运算, 算法复杂度较高. 文献[11-12]均为分布式的定位算法, 其特点是依据传感器自身的计算能力执行定位算法, 不需要强大计算能力的中央处理器, 而且降低了网络的通信量, 可对大规模 WSN 进行定位, 但是, 定位精度往往不高. 例如文献[11]是一种基于二阶锥规划 (second-order cone programming, SOCP) 的分布式定位算法, 每个节点收集来自邻居节点的信息, 采用二阶锥规划方法进行自身的定位, 然后不断与邻居节点交换信息, 实现分布式的定位. 文献[12]则是在每个节点上通过松弛非凸的最大似然估计的条件来求解定位问题, 且利用梯度法实现快速收敛, 具有良好的优化特性. 但是, 为了保证良好的定位效果, 要求 LU 节点需处于 LA 节点形成的凸包之内.

针对大规模 WSN 中的节点定位问题, 本文从减少具有计算能力的节点数目出发, 同时保证定位的准确度和效率, 提出一种基于超级节点的分布式迭代定位算法. 在该算法中, 假设只有 LA 节点具有一定的计算能力, 并作为超级节点. 在若干个相互重叠的子图基础上, 该算法的每步迭代分为子图内的高效定位和子图间融合两个步骤. 首先在每个独立的子图内, 将采用三点定位法的定位结果作为初始值, 利用共轭梯度法对局部定位问题进行优化求解; 然后, 对子图间重叠区域内传感器节点的局部位置进行融合取平均. 这两个步骤将持续进行, 直到满足迭代终止条件. 仿真实验表明: 与现有的集中式定位算法相比, 该

算法不仅定位准确, 而且能对规模更大的 WSN 进行定位; 与现有的分布式定位算法相比, 该算法不但定位准确度更高, 而且更高效.

## 1 定位问题的描述

鉴于 WSN 本身是一种多跳的自组织网络, 采用无向图  $G = (V, E)$  描述一个传感器节点分布于  $[0, 1] \times [0, 1]$  平面内的 WSN 全局信息,  $V(G)$  表示传感器节点的位置,  $E(G)$  表示节点间的连通性<sup>[12]</sup>.

在该平面区域内,  $N$  个 LU 节点和  $M$  个 LA 节点都是随机分布的, 记  $V(G) = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N, \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M\}$ . 其中:  $\mathbf{x}_i = [x_{2i-1}, x_{2i}]$ ,  $i \in \{1, 2, \dots, N\}$ ,  $x_{2i-1}$  和  $x_{2i}$  分别表示第  $i$  个 LU 节点的横坐标和纵坐标;  $\mathbf{a}_k = [a_{2k-1}, a_{2k}]$ ,  $k \in \{1, 2, \dots, M\}$ ,  $a_{2k-1}$  和  $a_{2k}$  分别表示第  $k$  个 LA 节点的横坐标和纵坐标. 则  $N$  个 LU 节点和  $M$  个 LA 节点的坐标位置分别表示为  $\mathbf{x}$  和  $\mathbf{a}$ , 即

$$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T, \quad (1)$$

$$\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M]^T. \quad (2)$$

令  $\mathbf{e}_{2i-1}$  和  $\mathbf{e}_{2i}$  分别表示单位矩阵  $\mathbf{I}_{2N}$  的第  $2i-1$  列和第  $2i$  列,  $\hat{\mathbf{e}}_{2k-1}$  和  $\hat{\mathbf{e}}_{2k}$  分别表示单位矩阵  $\mathbf{I}_{2M}$  的第  $2k-1$  列和第  $2k$  列, 于是  $x_{2i-1} = \mathbf{e}_{2i-1}^T \mathbf{x}$ ,  $x_{2i} = \mathbf{e}_{2i}^T \mathbf{x}$ ,  $a_{2k-1} = \hat{\mathbf{e}}_{2k-1}^T \mathbf{a}$ ,  $a_{2k} = \hat{\mathbf{e}}_{2k}^T \mathbf{a}$ , 从而节点间的欧氏距离为

$$\hat{d}_{ij}^2 = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = \mathbf{x}^T \mathbf{A}_{ij} \mathbf{x}, \quad (3)$$

$$\hat{d}_{ik}^2 = \|\mathbf{x}_i - \mathbf{a}_k\|_2^2 = \mathbf{x}^T \mathbf{B}_{ii} \mathbf{x} - 2\mathbf{a}^T \mathbf{C}_{ki} \mathbf{x} + \mathbf{a}^T \mathbf{D}_{kk} \mathbf{a}. \quad (4)$$

其中

$$\mathbf{A}_{ij} = (\mathbf{e}_{2i-1} - \mathbf{e}_{2j-1}) - (\mathbf{e}_{2i-1} - \mathbf{e}_{2j-1})^T + (\mathbf{e}_{2i} - \mathbf{e}_{2j})(\mathbf{e}_{2i} - \mathbf{e}_{2j})^T, \quad (5)$$

$$\mathbf{B}_{ii} = \mathbf{e}_{2i-1} \mathbf{e}_{2i-1}^T + \mathbf{e}_{2i} \mathbf{e}_{2i}^T, \quad (6)$$

$$\mathbf{C}_{ki} = \hat{\mathbf{e}}_{2k-1} \mathbf{e}_{2i-1}^T + \hat{\mathbf{e}}_{2k} \mathbf{e}_{2i}^T, \quad (7)$$

$$\mathbf{D}_{kk} = \hat{\mathbf{e}}_{2k-1} \hat{\mathbf{e}}_{2k-1}^T + \hat{\mathbf{e}}_{2k} \hat{\mathbf{e}}_{2k}^T. \quad (8)$$

由于受无线通信方式本身的限制, 并不是所有的节点都可以互相通信<sup>[13]</sup>, 只有距离在通信半径  $d_{\max}$  范围内的两节点才可以正常通信. WSN 节点间的连通性可记为  $E(G) = \{\rho_{ij}, \rho_{ik}\}$ . 其中:  $\rho_{ij}$  表示第  $i$  个 LU 节点与第  $j$  个 LU 节点可以直接相互通信,  $\rho_{ik}$  表示第  $i$  个 LU 节点与第  $k$  个 LA 节点可以直接相互通信. 把直接相互通信的节点视为邻居节点, 可以得到如下邻居节点的集合:

$$\mathbf{N}_1(i) = \{j : (i, j) \in \rho_{ij}\}, \quad (9)$$

$$\mathbf{N}_2(i) = \{k_1 : (i, k) \in \rho_{ik}\}. \quad (10)$$

其中:  $i \in \{1, 2, \dots, N\}$ ,  $\mathbf{N}_1(i)$  包含了可以与第  $i$  个 LU 节点直接相互通信的所有 LU 节点,  $\mathbf{N}_2(i)$  包含了可以与第  $i$  个 LU 节点直接相互通信的所有 LA 节点.

当 WSN 以无线的方式进行通信时, 节点发射信号的功率和接收到信号的功率都可以测得, 因此, 在无需额外添加设备的条件下, 便可通过 RSS 测距技术获取节点间的距离<sup>[4]</sup>. 但是, 无线电信号在传输过程中易受多径衰落和阴影衰落等影响, 根据信号衰减特性得到的测距数据并不准确. 因此, 节点间的测得距离  $d_{ij}$  和  $d_{ik}$ <sup>[5,9]</sup> 可表示为

$$d_{ij} = \hat{d}_{ij} |1 + \tau \varepsilon_{ij}|, \quad j \in \mathbf{N}_1(i), \quad (11)$$

$$d_{ik} = \hat{d}_{ik} |1 + \tau \varepsilon_{ik}|, \quad k \in \mathbf{N}_2(i), \quad (12)$$

$$|\varepsilon_{ij}| \text{ or } |\varepsilon_{ik}| < 1. \quad (13)$$

其中:  $\hat{d}_{ij}$  和  $\hat{d}_{ik}$  是节点之间的真实距离,  $\varepsilon_{ij}$  和  $\varepsilon_{ik}$  是产生的随机均匀噪声,  $\tau \in [0, 1]$  用于控制噪声强度.

考虑到成本和功耗的问题, 在 WSN 中通常仅部署少量的 LA 节点, 其余的大多数传感器节点是待定位的 LU 节点<sup>[4-5]</sup>. 如图 1 所示, 圆圈表示 LU 节点, 三角形表示 LA 节点, 连线表示可以相互通信. 在文献[11-12]的分布式定位算法中, 每个 LU 节点根据其邻居节点信息, 在自身上执行相应的定位算法. 而本文假设具有较强计算能力的少数 LA 节点为超级节点, 将定位任务分摊到每个 LA 节点上, 减少了具有计算能力的节点数目, 然后基于节点间的连通性, 结合 RSS 技术测得的节点间距离, 求解出 LU 节点的坐标位置.

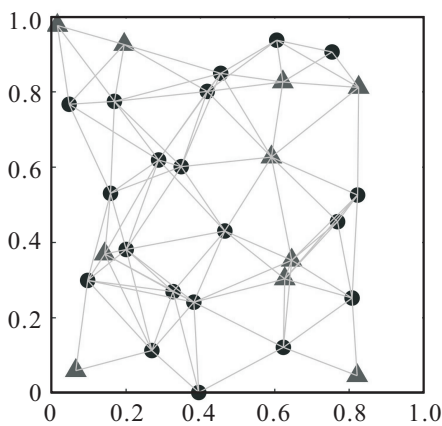


图 1 无向图  $G = (V, E)$

## 2 分布式定位算法

根据 WSN 中的节点间连通性  $E(G)$ , 将 LA 节点及其邻居节点所在的区域作为一个子图, 从而可以把无向图  $G = (V, E)$  分成  $M$  个子图  $G_m = (V_m, E_m)$ ,  $m \in \{1, 2, \dots, M\}$ . 其中:  $V_m(G_m)$  表示第  $m$  个子图  $G_m$  内传感器节点的位置信息,  $E_m(G_m)$  表

示第  $m$  个子图  $G_m$  内传感器节点间的连通性. 而且, 通过控制 LA 节点数目和节点间的通信半径  $d_{\max}$ , 使得每个 LU 节点都至少可以与两个 LA 节点通信, 即每个 LU 节点至少位于两个不同的子图中, 从而保证子图间有相互重叠的部分. 因此, 只需利用超级节点来对子图内的节点进行局部定位, 不但所需的具有计算能力的节点数目更少, 而且子图间融合更好地利用了区域间节点合作定位的优势, 使得定位更准确.

在  $M$  个相互重叠子图的基础上, 该分布式定位算法的每次迭代可分为子图内的高效定位和子图间融合两个步骤, 将分别在 2.1 节和 2.2 节中详细介绍.

### 2.1 子图内定位

基于子图  $G_m = (V_m, E_m)$ ,  $m \in \{1, 2, \dots, M\}$  包含的信息, 通过最小化邻居节点间距离误差的加权和<sup>[9]</sup>, 本文把子图内的定位问题归结为如下的一个无约束优化问题:

$$\min_{\mathbf{x}_m \in V_m} \sum_{(i,j) \in E_m} \omega_{ij}^2 \|\mathbf{x}_{m,i} - \mathbf{x}_{m,j}\|_2^2 - d_{ij}^2 \|^2 + \sum_{(i,k) \in E_m} \omega_{ik}^2 \|\mathbf{x}_{m,i} - \mathbf{a}_{m,k}\|_2^2 - d_{ik}^2 \|^2. \quad (14)$$

其中:  $\mathbf{x}_m$  表示第  $m$  个子图  $G_m$  内所有 LU 节点的位置坐标,  $\mathbf{x}_{m,i}$  表示第  $i$  个 LU 节点位于第  $m$  个子图  $G_m$  内的局部位置,  $\mathbf{x}_{m,j}$  表示第  $j$  个 LU 节点位于第  $m$  个子图  $G_m$  内的局部位置,  $\mathbf{a}_{m,k}$  表示第  $k$  个 LA 节点位于第  $m$  个子图  $G_m$  内的位置,  $\omega_{ij}$  和  $\omega_{ik}$  是对子图中节点间基于节点间距离反比的归一化权重, 有

$$\omega_{ij} = \frac{d_{ij}^{-1}}{\sum_{(i,j) \in E_m} d_{ij}^{-1} + \sum_{(i,k) \in E_m} d_{ik}^{-1}}, \quad (15)$$

$$\omega_{ik} = \frac{d_{ik}^{-1}}{\sum_{(i,j) \in E_m} d_{ij}^{-1} + \sum_{(i,k) \in E_m} d_{ik}^{-1}}. \quad (16)$$

这是从 RSS 测距技术的特点考虑的, 相距越远的两个节点, 测距时无线电信号越容易受到噪声干扰, 且受干扰的影响程度可能越大. 因此, 测距数据的可信度越低, 赋予的权重值也越小; 反之则越大.

结合节点间欧氏距离的表达式, 即式(3)~(8), 将目标函数(14)改写为

$$f(\mathbf{x}_m) = \min_{\mathbf{x}_m \in V_m} \sum_{(i,j) \in E_m} \omega_{ij}^2 (\mathbf{x}_m^T \mathbf{A}_{ij} \mathbf{x}_m - d_{ij}^2)^2 + \sum_{(i,k) \in E_m} \omega_{ik}^2 (\mathbf{x}_m^T \mathbf{B}_{ii} \mathbf{x}_m - 2\mathbf{a}_m^T \mathbf{C}_{ki} \mathbf{x}_m + \mathbf{a}_m^T \mathbf{D}_{kk} \mathbf{a}_m - d_{ik}^2)^2. \quad (17)$$

显然,  $f(\mathbf{x}_m)$  是关于 LU 节点坐标  $\mathbf{x}_m$  的高度非线性

非凸4次函数,直接求解比较困难. 对此,将式(17)作二阶泰勒近似,在超级节点上,将三点定位法对LU节点的定位结果作为初始值,采用共轭梯度法对定位问题进行优化求解.

### 2.1.1 三点定位

LU节点收集来自邻居节点的距离和位置信息,并传递给相邻的超级节点. 当LU节点的通信半径 $d_{\max}$ 内至少有3个LA节点时,超级节点选取距其最近的3个LA节点,采用三点定位法对该LU节点进行定位. 三点定位方法<sup>[13]</sup>如图2所示,A、B和C表示3个LA节点,坐标位置依次是 $(x_1, y_1)$ 、 $(x_2, y_2)$ 、 $(x_3, y_3)$ ;P表示1个LU节点;A、B和C三点到待求节点P的距离分别是 $d_1$ 、 $d_2$ 、 $d_3$ . 假设待求节点P的坐标位置是 $(x, y)$ ,则可得到3个关于圆的方程组

$$\begin{cases} \sqrt{(x-x_1)^2+(y-y_1)^2}=d_1, \\ \sqrt{(x-x_2)^2+(y-y_2)^2}=d_2, \\ \sqrt{(x-x_3)^2+(y-y_3)^2}=d_3. \end{cases} \quad (18)$$

对式(18)求解,可以得到LU节点P的具体坐标位置

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2(x_3-x_1) & 2(y_3-y_1) \\ 2(x_3-x_2) & 2(y_3-y_2) \end{bmatrix}^{-1} \times \begin{bmatrix} x_3^2-x_1^2+y_3^2-y_1^2+d_1^2-d_3^2 \\ x_3^2-x_2^2+y_3^2-y_2^2+d_2^2-d_3^2 \end{bmatrix}. \quad (19)$$

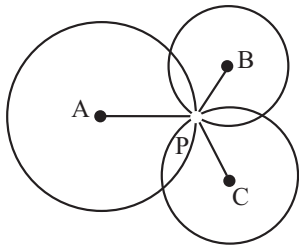


图2 三点定位法示意

由于测距所得的节点间距离存在误差,3个圆不可能恰好相交于一点,即这里求得的LU节点位置坐标并不太准确.

作为补充,当待求LU节点的通信半径内只有2个LA节点时,将2个LA节点的中心位置作为该LU节点的初步估计位置.

在完成对所有LU节点粗略的初步定位后,为了达到更高的定位精度,需要将本次定位结果作为初始值 $\mathbf{x}_{m,0}$ 进行下一步的迭代运算.

### 2.1.2 优化求解

在进行迭代优化之前,将 $f(\mathbf{x}_m)$ 在 $\mathbf{x}_{m,0}$ 处展开成Taylor级数,并取二阶近似<sup>[15]</sup>,有

$$f(\mathbf{x}_m) \approx f(\mathbf{x}_{m,0}) + \nabla^T f(\mathbf{x}_{m,0})(\mathbf{x}_m - \mathbf{x}_{m,0}) +$$

$$\frac{1}{2}(\mathbf{x}_m - \mathbf{x}_{m,0})^T \nabla^2 f(\mathbf{x}_{m,0})(\mathbf{x}_m - \mathbf{x}_{m,0}). \quad (20)$$

对式(17)分别进行一阶求导和二阶求导,可得

$$\begin{aligned} \nabla f(\mathbf{x}_m) = & \sum_{(i,j) \in \mathbf{E}_m} 4\omega_{ij}^2 (\mathbf{x}_m^T \mathbf{A}_{ij} \mathbf{x}_m - d_{ij}^2) (\mathbf{A}_{ij} \mathbf{x}_m) + \\ & \sum_{(i,k) \in \mathbf{E}_m} 2\omega_{ik}^2 (\mathbf{x}_m^T \mathbf{B}_{ii} \mathbf{x}_m - 2\mathbf{a}_m^T \mathbf{C}_{ki} \mathbf{x}_m + \\ & \mathbf{a}_m^T \mathbf{D}_{kk} \mathbf{a}_m - d_{ik}^2) (2\mathbf{B}_{ii} \mathbf{x}_m - 2\mathbf{C}_{ki}^T \mathbf{a}_m), \end{aligned} \quad (21)$$

$$\begin{aligned} \nabla^2 f(\mathbf{x}_m) = & \sum_{(i,j) \in \mathbf{E}_m} \omega_{ij}^2 [8\mathbf{A}_{ij} \mathbf{x}_m \mathbf{x}_m^T \mathbf{A}_{ij}^T + \\ & 4(\mathbf{x}_m^T \mathbf{A}_{ij} \mathbf{x}_m - d_{ij}^2) \mathbf{A}_{ij}^T] + \\ & \sum_{(i,k) \in \mathbf{E}_m} 2\omega_{ik}^2 [(2\mathbf{B}_{ii} \mathbf{x}_m - 2\mathbf{C}_{ki}^T \mathbf{a}_m) (2\mathbf{B}_{ii} \mathbf{x}_m - \\ & 2\mathbf{C}_{ki}^T \mathbf{a}_m)^T + 2(\mathbf{x}_m^T \mathbf{B}_{ii} \mathbf{x}_m - 2\mathbf{a}_m^T \mathbf{C}_{ki} \mathbf{x}_m + \\ & \mathbf{a}_m^T \mathbf{D}_{kk} \mathbf{a}_m - d_{ik}^2) \mathbf{B}_{ii}^T]. \end{aligned} \quad (22)$$

当二阶导数 $\nabla^2 f(\mathbf{x}_{m,0})$ 为半正定阵时,新的目标函数(20)是二次可微的凸函数. 可以观察到,式(22)等号右边的表达式中 $\mathbf{A}_{ij} \mathbf{x}_m \mathbf{x}_m^T \mathbf{A}_{ij}^T$ 和 $(2\mathbf{B}_{ii} \mathbf{x}_m - 2\mathbf{C}_{ki}^T \mathbf{a}_m) (2\mathbf{B}_{ii} \mathbf{x}_m - 2\mathbf{C}_{ki}^T \mathbf{a}_m)^T$ 均是半正定的,而 $(\mathbf{x}_m^T \mathbf{A}_{ij} \mathbf{x}_m - d_{ij}^2) \mathbf{A}_{ij}^T$ 和 $(\mathbf{x}_m^T \mathbf{B}_{ii} \mathbf{x}_m - 2\mathbf{a}_m^T \mathbf{C}_{ki} \mathbf{x}_m + \mathbf{a}_m^T \mathbf{D}_{kk} \mathbf{a}_m - d_{ik}^2) \mathbf{B}_{ii}^T$ 并不能保证其半正定性. 因此,对式(22)作如下修改<sup>[16]</sup>:

$$\begin{aligned} \nabla^2 f(\mathbf{x}_m) = & \sum_{(i,j) \in \mathbf{E}_m} 8\omega_{ij}^2 \mathbf{A}_{ij} \mathbf{x}_m \mathbf{x}_m^T \mathbf{A}_{ij}^T + \sum_{(i,k) \in \mathbf{E}_m} 2\omega_{ik}^2 (2\mathbf{B}_{ii} \mathbf{x}_m - \\ & 2\mathbf{C}_{ki}^T \mathbf{a}_m) (2\mathbf{B}_{ii} \mathbf{x}_m - 2\mathbf{C}_{ki}^T \mathbf{a}_m)^T, \end{aligned} \quad (23)$$

从而,可以采用凸优化的方法进行迭代优化求解. 这里,共轭梯度法的算法<sup>[15]</sup>流程如下.

step 1: 以三点定位得到的粗略定位结果作为初始值 $\mathbf{x}_{m,0}, p = 0$ .

step 2: 计算 $\mathbf{d}_{m,p} = -\nabla f(\mathbf{x}_{m,p})$ , 如果 $\|\nabla f(\mathbf{x}_{m,p})\|_2 < \varepsilon$  ( $\varepsilon = 10^{-8}$ ), 则停止计算; 否则, 进行一维搜索, 步长 $\lambda_p$ 满足

$$f(\mathbf{x}_{m,p} + \lambda_p \mathbf{d}_{m,p}) = \min_{\lambda > 0} f(\mathbf{x}_{m,p} + \lambda \mathbf{d}_{m,p}). \quad (24)$$

令 $\varphi(\lambda) = f(\mathbf{x}_{m,p} + \lambda \mathbf{d}_{m,p})$ ,  $\mathbf{x}_{m,p+1} = \mathbf{x}_{m,p} + \lambda_p \mathbf{d}_{m,p}$ , 则求 $\varphi(\lambda)$ 的极小点, 即

$$\nabla \varphi(\lambda) = \nabla f(\mathbf{x}_{m,p+1})^T \mathbf{d}_{m,p} = 0. \quad (25)$$

将式(19)在 $\mathbf{x}_{m,p}$ 处作二阶泰勒近似, 并求在 $\mathbf{x}_{m,p+1}$ 处的梯度, 式(25)可表示为

$$[\nabla^2 f(\mathbf{x}_{m,p})(\mathbf{x}_{m,p+1} - \mathbf{x}_{m,p}) + \nabla f(\mathbf{x}_{m,p})]^T \mathbf{d}_{m,p} = 0,$$

$$[\lambda_p \nabla^2 f(\mathbf{x}_{m,p}) \mathbf{d}_{m,p} + \nabla f(\mathbf{x}_{m,p})]^T \mathbf{d}_{m,p} = 0, \quad (26)$$

可得到

$$\lambda_p = -\frac{\nabla f(\mathbf{x}_{m,p})^T \mathbf{d}_{m,p}}{\mathbf{d}_{m,p}^T \nabla^2 f(\mathbf{x}_{m,p}) \mathbf{d}_{m,p}}. \quad (27)$$

step 3: 如果  $p < n$  ( $n$  是指  $\mathbf{x}_m$  的维度), 则转 step 4; 否则, 转 step 5.

step 4: 令  $\mathbf{d}_{m,p+1} = -\nabla f(\mathbf{x}_{m,p+1}) + \beta_p \mathbf{d}_{m,p}$ , 其中  $\beta_p = \|\nabla f(\mathbf{x}_{m,p+1})\|^2 / \|\nabla f(\mathbf{x}_{m,p})\|^2$ , 置  $p := p + 1$ , 转 step 2.

step 5: 令  $\mathbf{x}_{m,0} = \mathbf{x}_{m,p+1}$ ,  $\mathbf{d}_{m,0} = -\nabla f(\mathbf{x}_{p+1})$ , 置  $p = 0$ , 转 step 2.

## 2.2 子图间融合

该分布式定位算法中, 无向图  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$  被划分为  $M$  个相互重叠的子图  $\mathbf{G}_m = (\mathbf{V}_m, \mathbf{E}_m)$ ,  $m \in \{1, 2, \dots, M\}$ .

如图3所示, 三角形表示超级节点, 圆圈表示LU节点, 虚线所围区域表示一个子图, 实心小圆圈表示位于不同子图相互重叠部分的LU节点. 各个子图在独立地完成定位任务之后, 每个LU节点在不同子图中被估计出来的节点位置可能并不相同. 因此, 对于同一个LU节点, 存在多个来自不同子图内的局部估计值. 通过有效地利用子图间重叠区域内节点局部位置的融合, 可以更有利于促进子图之间的合作定位, 进一步提高分布式定位的准确度. 在这里, 对该LU节点所在各个子图中的节点位置取平均, 即

$$\bar{\mathbf{x}}_i = \frac{1}{|\mathbf{N}_2(i)|} \sum_{m \in \mathbf{N}_2(i)} \mathbf{x}_{m,j}, \quad i \in \{1, 2, \dots, N\}. \quad (28)$$

其中:  $|\mathbf{N}_2(i)|$  表示第  $i$  个LU节点的邻居中超级节点的数目, 即包含有第  $i$  个LU节点的子图数目;  $\mathbf{x}_{m,i}$  表示第  $i$  个LU节点在第  $m$  个子图中的节点位置.

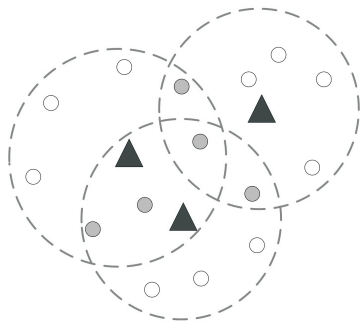


图3 部分相互重叠的子图示意

## 2.3 分布式算法流程总结

在该分布式的传感器节点迭代定位算法中, 以具有较强计算能力的LA节点作为超级节点, 以超级节点及其邻居节点所在区域作为子图, 把全局的定位问题归结为各个子图内的优化问题. 一方面, 对于各个

子图内无约束的优化问题, 将子图内节点间距离误差的加权和作为目标函数. 对于该目标函数的求解, 采用了共轭梯度法, 且没有盲目地随机选取优化的初始值, 而是将对LU节点粗略定位的初步估计位置作为初始值, 并确保搜索方向是沿着目标函数下降的方向, 进而保证节点的估计位置在不断地接近局部最优解, 使目标函数能尽快地收敛. 另一方面, LU节点在不同子图内被估计出来的节点位置可能并不相同, 因此, 对LU节点所在各个子图中的节点位置取平均, 能对在子图内原本定位误差较大的节点位置进行修正, 使其得到一个较好的初始值并重新参与下一步的迭代. 不断重复子图内的高效定位和子图间融合这两个步骤, 有效利用子图内节点和各个子图间协作定位的特点, 可使LU节点的位置估计误差不断减小, 能够准确高效地完成定位任务.

本文提出的基于超级节点的分布式算法步骤如下.

输入: 随机均匀生成节点分布图  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , 且以LA节点及其邻居节点所在区域作为子图, 可将其划分为  $M$  个相互重叠的子图  $\mathbf{G}_m = (\mathbf{V}_m, \mathbf{E}_m)$ ,  $m \in \{1, 2, \dots, M\}$ .

step 1: 子图内高效定位. 对于每个子图  $\mathbf{G}_m = (\mathbf{V}_m, \mathbf{E}_m)$ ,  $m \in \{1, 2, \dots, M\}$ , 首先, 对LU节点的位置进行粗略估计, 即在2.1.1节中采用三点定位法得到初步的定位结果  $\mathbf{x}_0$ ,  $t = 0$ ; 然后, 将其作为2.1.2节中共轭梯度法的初始值, 继续对LU节点的位置进行优化.

step 2: 子图间融合. 在子图间重叠区域内, 由于不同子图内的LU节点被估计出来的位置可能并不相同, 因此, 在2.2节中对子图间重叠区域内的节点位置进行融合并取平均, 得到本次迭代的定位结果  $\mathbf{x}_{t+1}$ .

step 3: 迭代终止判断. 迭代终止条件为  $\|\mathbf{x}_{t+1,i} - \mathbf{x}_{t,i}\|_2 \leq 10^{-2}$ , 用来判断优化变量  $\mathbf{x}_{t+1}$  是否已收敛. 其中:  $\mathbf{x}_{t+1,i}$  是本次迭代后第  $i$  个LU节点的位置坐标,  $\mathbf{x}_{t,i}$  是本次迭代前第  $i$  个LU节点的位置坐标. 若满足迭代终止条件, 则输出最终定位结果  $\mathbf{x}_{t+1}$ ; 否则, 重新将  $\mathbf{x}_{t+1}$  作为新的初始值, 返回step 2中共轭梯度法重新进行优化, 即令  $\mathbf{x}_t = \mathbf{x}_{t+1}$ ,  $t = t + 1$ .

输出: 最终定位结果  $\mathbf{x}_{i+1}$ .

## 3 仿真结果与分析

为了客观地评价该分布式定位算法的定位准确度, 这一节给出4个仿真实例. 采用文献[9-10]中提到的根均方距离(RMSD)作为评价指标, 即

$$\text{RMSD}^{(n)} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i^{(n)} - \hat{\mathbf{x}}_i\|_2^2}. \quad (29)$$

其中:  $\hat{\mathbf{x}}_i$  表示LU节点*i*的真实坐标,  $\mathbf{x}_i^{(n)}$  表示LU节点*i*在第*n*次实验的定位结果,  $\text{RMSD}^{(n)}$  是第*n*次实验的根均方距离. 仿真实验使用的软件是Matlab 2017b, 并运行于Intel i7-7700主频3.6 GHz的PC, 表1是仿真参数的设置及说明.

本文算法的仿真过程如下:

- 1) 采用GSPBOX工具箱生成随机均匀分布的*M*个LA节点和*N*个LU节点.
- 2) 设置通信半径 $d_{\max}$ 和噪声因子 $\tau$ , 获取每一个

节点的邻居节点, 并模拟节点间的测距数据 $d_{ij}$ 和 $d_{ik}$ .

3) 以LA节点及其邻居节点所在区域作为子图, 将整个WSN区域划分为*M*个相互重叠的子图.

4) 在每个子图内, 以2.1.1节中初步定位的结果作为初始值 $\mathbf{x}_0, t = 0$ ; 以2.1.2节中共轭梯度法对其进行优化.

5) 在子图间重叠区域内, 对LU节点在不同子图中的估计位置进行融合并取平均, 得到定位结果 $\mathbf{x}_{t+1}$ .

6) 若满足迭代终止条件, 则输出定位结果 $\mathbf{x}_{t+1}$ ; 否则 $\mathbf{x}_t = \mathbf{x}_{t+1}, t = t + 1$ , 返回4)中共轭梯度法重新进行优化.

表1 仿真参数的设置及说明

参数名称	符号或数值	备注
平面区域	$[0, 1] \times [0, 1]$	采用与参考文献[9-12]相同的单位正方形区域模拟WSN
节点规模	$M + N$	采用GSPBOX工具箱生成随机均匀分布的 <i>M</i> 个LA节点和 <i>N</i> 个LU节点
LA节点所占比例	$p = M/(M + N)$	在WSN中, 考虑到成本和功耗的问题, 一般LA节点所占比例较小
噪声因子	$\tau = 20$	仿真时用于控制随机均匀噪声的强度
通信半径 $d_{\max}$	$d_{\max}$	一方面, 保证每个LU节点至少可以与两个LA节点通信; 另一方面, 考虑到LA节点的计算能力, 用于控制每个子图内的节点数目不要太多
仿真实验次数	$T$	多次仿真增加实验结果的可信度

**例1** 为了验证LA节点数目和通信半径 $d_{\max}$ 的变化对于定位性能的影响, 将本文算法与文献[11-12]中分布式算法在如表1所示的仿真参数设置下进行对比实验. 在本例中, 节点数目 $M + N = 200$ , 实验次数 $T = 5$ . 在固定通信半径 $d_{\max} = 0.20$ 的情况下, 改变WSN中LA节点所占比例*p*, 本文算法与文献[11-12]算法5次仿真实验得到的平均RMSD和平均定位时间分别如图4和图5所示.

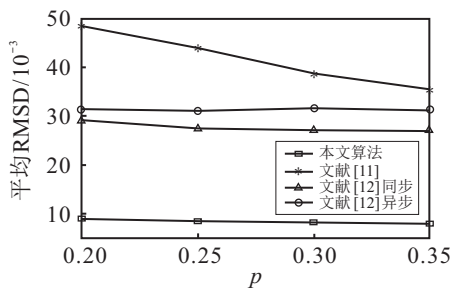


图4 例1中不同LA节点数目下各算法的平均RMSD

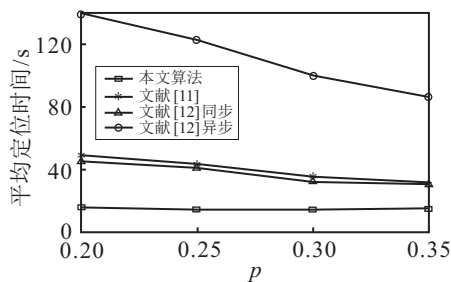


图5 例1中不同LA节点数目下各算法的平均定位时间

从图4可以看出, 随着LA节点所占比例*p*增大, 文献[11]算法的平均RMSD一直在减小, 而本文算法和文献[12]算法的平均RMSD变化不大, 且本文算法所得平均RMSD明显最小. 从图5可以看出, 随着LA节点所占比例*p*增大, 文献[11-12]算法的平均定位时间在减少, 而本文算法的平均定位时间变化很小且始终最少. 在固定WSN中LA节点所占比例*p* = 20的情况下, 改变通信半径的大小, 本文算法与文献[11-12]算法5次仿真实验得到的平均RMSD和平均定位时间分别如图6和图7所示.

从图6可以看出, 随着通信半径 $d_{\max}$ 的增大, 文献[11-12]算法的平均RMSD有所增大, 而本文算法的平均RMSD变化不大且始终最小. 从图7可以看出, 本文算法和文献[12]算法的平均定位时间随着通信半径的增大而增多, 而不同的通信半径 $d_{\max}$ 对文

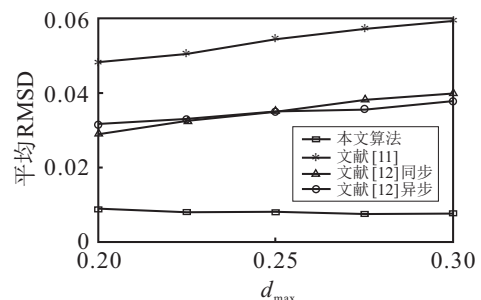


图6 例1中不同通信半径 $d_{\max}$ 下各算法的平均RMSD

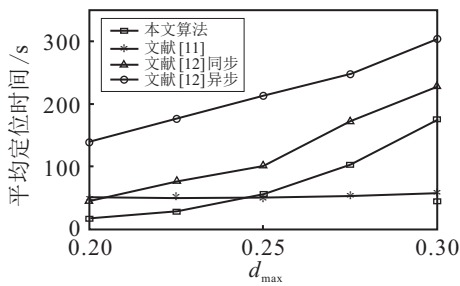


图7 例1中不同通信半径 $d_{max}$ 下各算法的平均定位时间

文献[11]算法基本无影响,但当通信半径 $d_{max}$ 较小时,还是本文算法的平均定位时间最少.总之,增大通信半径 $d_{max}$ ,节点的邻居节点数目增多,随之而增大的

通信量和计算量会影响定位的性能;增加LA节点的数目,虽然可以提高定位的性能,但却增加了节点本身的成本和功耗.综合考虑,本文算法在较少的LA节点数目和较小的通信半径下都能高效准确地完成定位任务.

**例2** 为了验证本文分布式算法与现有集中式算法在不同节点规模下的定位性能,将本文算法与文献[9-10]中两种集中式算法在如表1所示的仿真参数设置下进行对比实验.在固定WSN中LA节点所占比例 $p = 0.20$ 的情况下,改变节点规模,本文算法与文献[9-10]中算法10次仿真实验得到的平均RMSD和平均定位时间如表2所示.

表2 例2中各定位算法10次仿真所得的定位结果 (“\*”表示无法完成定位)

节点规模	$d_{max}$	文献[9]		文献[10]		本文算法	
		平均时间/s	平均RMSD	平均时间/s	平均RMSD	平均时间/s	平均RMSD
100	0.30	1.47	0.0149	11.86	0.0134	7.09	0.0141
200	0.20	8.21	0.0135	116.16	0.0092	15.64	0.0087
400	0.17	88.97	0.0057	*	*	54.44	0.0060
800	0.14	*	*	*	*	304.41	0.0037

从表2的仿真结果可以看出:当节点数目为100和200时,文献[9]算法定位最快,文献[10]算法定位最慢,而本文算法的定位速度仅次于文献[9],且与文献[9-10]中定位算法得到的平均RMSD相接近;但当节点数目达到400时,由于集中式定位算法本身通常需要庞大的数据计算量和通信量,文献[10]算法导致Matlab数据溢出而无法完成定位任务,本文算法不仅与文献[9]算法的平均RMSD很接近,而且所需定位时间更少;当节点数目继续增加到800时,本PC平台无法满足文献[9]算法对于计算能力的需求,也无法

再正常定位,而本文定位算法依然能在有限的时间内得到准确的定位结果.仿真结果表明,本文分布式算法的定位准确度达到了现有集中式算法的水平,而且能够对较大节点规模的WSN进行高效定位.

**例3** 为了验证本文分布式算法与现有分布式算法在不同节点规模下的定位性能,将本文算法与文献[11-12]中分布式算法在如表1所示的仿真参数设置下进行对比实验.在固定WSN中LA节点所占比例的情况下,不断增大节点规模,本文算法与文献[11-12]中算法5次仿真实验得到的平均RMSD和平均定位时间如表3所示.

表3 例3中各定位算法5次仿真所得的平均RMSD (“—”表示由于定位耗时太长,未记录数据)

节点数目	$d_{max}$	文献[11]定位算法		文献[12]同步算法		文献[12]异步算法		本文算法	
		平均时间/s	平均RMSD	平均时间/s	平均RMSD	平均时间/s	平均RMSD	平均时间/s	平均RMSD
500	0.16	157.95	0.0374	1042.15	0.0236	510.06	0.0260	96.93	0.0049
1000	0.12	349.86	0.0318	4514.65	0.0130	1252.82	0.0114	197.27	0.0035
2000	0.09	848.43	0.0259	—	—	11226.64	0.0084	384.02	0.0045
4000	0.07	1980.40	0.0263	—	—	—	—	1261.67	0.0084
8000	0.05	4999.70	0.0238	—	—	—	—	2624.62	0.0095

从表3的仿真结果可以看出:当节点数目为500、1000和2000时,文献[12]算法定位耗时最长,文献[11]算法的平均RMSD最大,本文算法不但平均定位时间最短,而且平均RMSD也最小;当节点数目达到4000和8000时,文献[12]中两种算法定位耗时更长,因此未在表格中记录相关实验数据,而本文算法依然

比文献[11]算法的平均RMSD降低了一个数量级,定位耗时也更短.仿真结果表明,与文献[11-12]中分布式算法相比,本文算法在大规模的WSN中定位更准确、更高效.

**例4** 为了以更直观的形式呈现上述仿真实例中各算法的定位结果,将本文算法与文献[9-12]中分

布式算法在如表1所示的仿真参数设置下进行对比实验. 在本例中, 节点规模  $M + N = 200$ , LA节点所占比例  $p = 0.20$ , 通信半径  $d_{\max} = 0.20$ , 实验次数  $T = 5$ , 本文算法与文献[11-12]算法5次仿真实验得

到的平均节点位置如图8所示. 其中: 空心圆表示LU节点的真实位置, 实心五角星表示LU节点5次仿真实验的平均估计位置, 实心三角形表示LA节点的位置, 连线表示定位的误差.

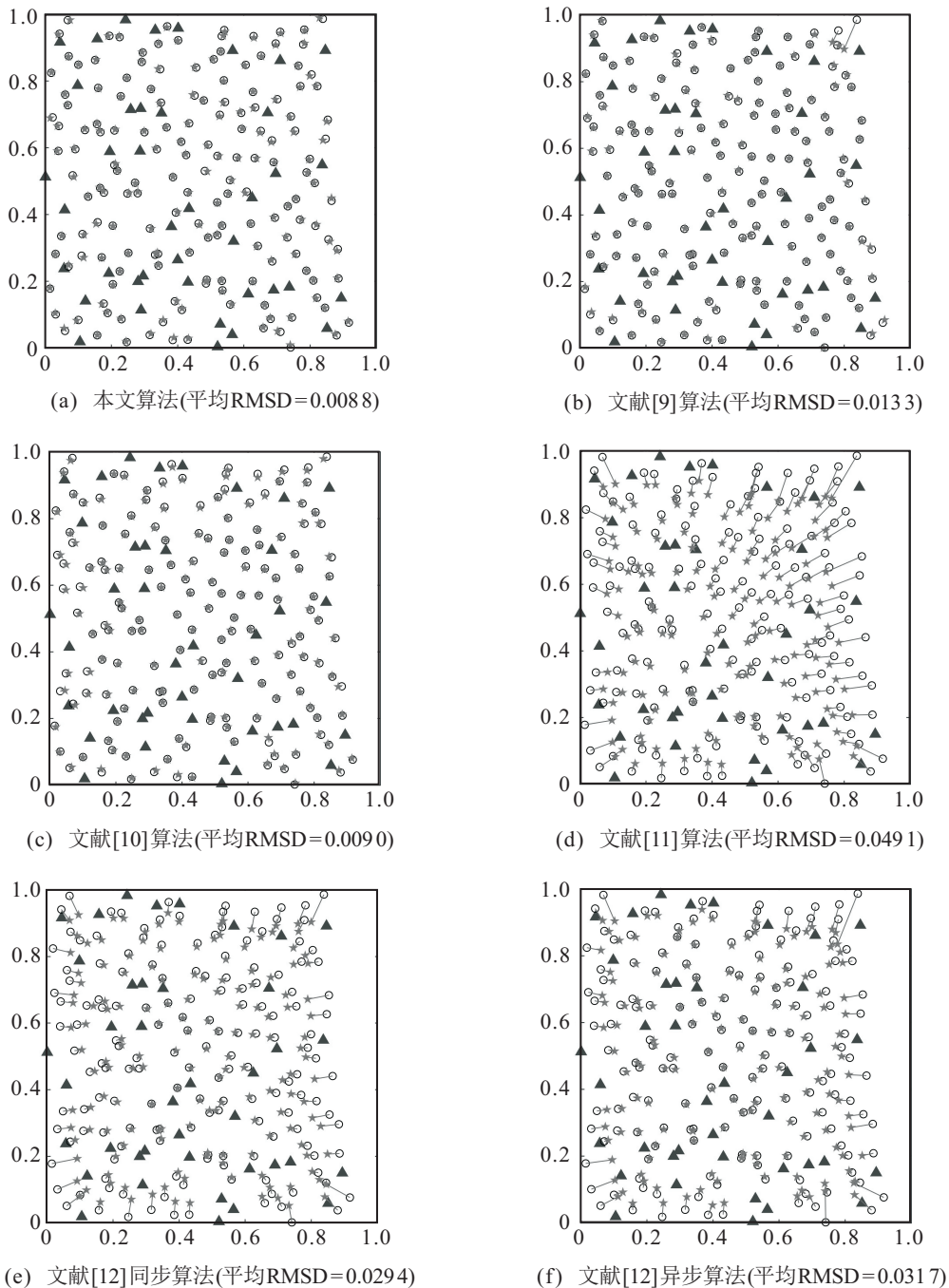


图8 例4中各算法5次仿真实验的平均定位结果

从图8可以看出: 文献[9-10]中集中式算法要比文献[11-12]中分布式算法定位更准确, 而且文献[11-12]中算法明显对区域边缘处LU节点存在较大的定位误差; 本文分布式算法得到了与文献[9-10]中集中式算法相近的定位结果, 能很好地完成对区域内节点的定位任务. 仿真结果表明, 与文献[9-12]中算法相比, 本文分布式算法定位能对整个WSN区域内的节

点进行准确有效地定位.

## 4 结论

本文提出了一种基于超级节点的分布式传感器节点迭代定位算法, 减少了所需具有计算能力的节点数目, 以超级节点及其邻居节点所在区域作为子图. 每一步的迭代中, 首先在子图内进行高效定位, 然后对子图重叠区域的节点进行子图间融合. 这两个

步骤不断重复进行,直至满足迭代的终止条件. 仿真实验结果表明,与现有算法相比,本文算法定位更准确,能够对大规模的无线传感器网络进行更加高效的定位. 后续工作将进一步研究更加合理的子图划分和子图融合方法,并探索大规模 WSN 中异步的分布式定位算法.

#### 参考文献(References)

- [1] 于晓升, 王莹, 孟亚男, 等. 基于 IMM-IKF 的无线传感器网络非视距节点定位方法[J]. 控制与决策, 2018, 33(6): 1069-1074.  
(Yu X S, Wang Y, Meng Y N, et al. Non-line of sight node localization method based on IMM-IKF for wireless sensor networks[J]. Control and Decision, 2018, 33(6): 1069-1074.)
- [2] 余修武, 周利兴, 余齐豪, 等. 基于刚分簇与鸡群优化的深井无线传感网络定位算法[J]. 西南交通大学学报, 2019, 54(4): 870-878.  
(Yu X W, Zhou L X, Yu Q H, et al. Localization algorithm for mine wireless sensor network based on rigid cluster and chicken swarm optimization[J]. Journal of Southwest Jiaotong University, 2019, 54(4): 870-878.)
- [3] Mofarreh-Bonab M, Ghorashi S A. A low complexity and high speed gradient descent based secure localization in wireless sensor networks[C]. International Conference on Computer and Knowledge Engineering. Mashhad, 2013: 300-303.
- [4] Keller Y, Gur Y. A diffusion approach to network localization[J]. IEEE Transactions on Signal Processing, 2011, 59(6): 2642-2654.
- [5] Patwari N, Ash J N, Kyperountas S, et al. Locating the nodes: Cooperative localization in wireless sensor networks[J]. IEEE Signal Processing Magazine, 2005, 22(4): 54-69.
- [6] 何杰, 吴得决, 李希飞, 等. 利用测距值优化的室内三维定位算法[J]. 北京邮电大学学报, 2017, 40(3): 37-42.  
(He J, Wu D Y, Li X F, et al. A 3-D indoor localization algorithm using distance optimization[J]. Journal of Beijing University of Posts and Telecommunications, 2017, 40(3): 37-42.)
- [7] 刘宏, 韩亚波, 张时斌, 等. 基于自适应罚函数优化粒子群的 WSN 定位算法[J]. 传感技术学报, 2018, 31(8): 1253-1257.  
(Liu H, Han Y B, Zhang S B, et al. WSNs localization algorithm based on self-adaptive penalty function optimization particle swarm optimization[J]. Chinese Journal of Sensors and Actuators, 2018, 31(8): 1253-1257.)
- [8] Tomic S, Beko M, Rui D. RSS-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes[J]. IEEE Transactions on Vehicular Technology, 2014, 64(5): 2037-2050.
- [9] Biswa P, Liang T C, Toh K C, et al. Semidefinite programming approaches for sensor network localization with noisy distance measurements[J]. IEEE Transactions on Automation Science & Engineering, 2006, 3(4): 360-371.
- [10] 蒋俊正, 赵海兵, 李杨剑, 等. 基于二阶泰勒近似的传感器节点定位方法[P]. 中国: 201810438409.X, 2018-10-12.  
(Jiang J Z, Zhao H B, Li Y J, et al. A method for sensor nodes localization based on second-order Taylor approximation[P]. China: 201810438409.X, 2018-10-12.)
- [11] Srirangarajan S, Tewfik A, Luo Z Q. Distributed sensor network localization using SOCP relaxation[J]. IEEE Transactions on Wireless Communications, 2008, 7(12): 4886-4895.
- [12] Soares C, Xavier J, Gomes J. Simple and fast convex relaxation method for cooperative localization in sensor networks using range measurements[J]. IEEE Transactions on Signal Processing, 2015, 63(17): 4532-4543.
- [13] 罗阳倩子, 廖威. 基于粒子滤波的 RSSI 测距优化的牛顿定位算法[J]. 仪表技术与传感器, 2017(6): 116-119.  
(Luo Y Q Z, Liao W. RSSI range optimization newton localization algorithm based on particle filter[J]. Instrument Technique and Sensor, 2017(6): 116-119.)
- [14] 王芳. 射频 RSS 聚类与多传感器融合的室内定位算法[J]. 计算机工程与设计, 2018, 39(6): 1553-1558.  
(Wang F. RF RSS-based clustering and multi-sensor integration location algorithm[J]. Computer Engineering and Design, 2018, 39(6): 1553-1558.)
- [15] 陈宝林. 凸优化理论与算法[M]. 北京: 清华大学出版社, 2005: 291-306.  
(Chen B L. Convex optimization theory and algorithm[M]. Beijing: Tsinghua University Press, 2005: 291-306.)
- [16] 蒋俊正. DFT 调制滤波器组的设计算法研究[D]. 西安: 西安电子科技大学雷达信号处理重点实验室, 2011: 21-28.  
(Jiang J Z. Design algorithms of DFT modulated filter banks[D]. Xi'an: National Lab of Radar Signal Processing, Xidian University, 2011: 21-28.)

#### 作者简介

蒋俊正(1983—), 男, 教授, 博士生导师, 从事图信号处理、分布式信号、大规模传感器网络数据处理、大数据的半监督学习等研究, E-mail: jzjiang@guet.edu.cn;

赵海兵(1990—), 男, 硕士生, 从事 WSN 中节点定位算法的研究, E-mail: 1552125802@qq.com.

(责任编辑: 李君玲)