

基于密度分簇的无线传感器网络定位算法

王 勇, 胡良梁, 袁巢燕

(重庆理工大学计算机科学与工程学院 重庆 巴南区 400054)

【摘要】针对MDS-MAP(P)算法存在节点间最短路径距离计算误差、合并误差及算法复杂度过高等问题,提出了一种基于密度分簇的算法MDS-MAP(DB)。该算法选择邻居节点数最多的节点作为分簇机制的开始节点,一跳邻居节点组成的簇域内利用三角不等式法则测距,两跳内节点组成的簇域内利用最短路径法测距,且每个簇域内只有簇头节点执行测距算法,降低了测距误差及算法计算复杂度,提高了算法的性能。仿真实验结果表明,该算法具有更小的定位误差。

关键词 分簇; 分布式定位; 多维标尺; 节点密度; 无线传感器网络

中图分类号 TP309

文献标志码 A

doi:10.3969/j.issn.1001-0548.2013.03.017

Density-Based Clustering Localization Algorithm for Wireless Sensor Networks

WANG Yong, HU Liang-liang, and YUAN Chao-yan

(College of Computer Science and Engineering, Chongqing University of Technology Banan Chongqing 400054)

Abstract The MDS-MAP(P) localization algorithm has shortage of calculation errors of the shortest path distance, combined error, and the higher complexity of the algorithm. In order to solve these problems, A density-based clustering localization algorithm called MDS-MAP(DB) is proposed. The algorithm selects a node which has most neighbor nodes as the start node in the clustering mechanism. The clustering which is built by 1 hop neighbor nodes uses the triangle inequality estimating missing distances. The clustering which is built by 2 hop nodes uses the shortest path distance estimating missing distances. In all clustering, only the cluster head executes the algorithm of estimating distance. In this way, the error of estimating distances and the computational complexity of this algorithm have been reduced. The results of simulation show that the proposed algorithm can reduce the average estimation errors.

Key words clustering; localization algorithm; multidimensional scaling; node density; wireless sensor networks

无线传感器网络是一种自组织的、分布式处理和快速展开的特殊的Ad-hoc网络,是信息技术领域里应用广泛的多学科交叉的新兴技术。它综合了微处理技术、嵌入式计算技术、通信技术和传感器技术等,一般由无线传感器节点、网关节点和传输网络和远程监控中心4个基本部分组成。由于无线传感器网络具有感知能力、计算能力、自适应能力等突出特点,目前在工业、军事及农业等领域有着广泛的应用。

当前,根据无线传感器网络的定位算法是否测距,可将现有的定位算法分为两类:基于测距的定位算法(Range-Based)和无需测距的定位算法(Range-Free)^[1]。基于测距的定位算法有RSSI^[2]、

TOA^[3]和TDOA^[4]。RSSI算法通过测量从发送节点到接收节点的信号强度估计两者之间的距离。TOA通过测量信号的传输时间计算两邻居节点的距离。TDOA利用两个信号的到达时间差计算未知节点与邻居节点的距离值。无需测距的定位算法有质心算法^[5]、DV-hop^[6]和APIT^[7]。质心算法是待定位节点以所有在其通信范围内的信标节点的几何质心作为自己的位置坐标估计值。DV-hop将待定位节点到信标节点之间的距离用网络中节点平均每跳的距离和到信标节点间的跳数的乘积来标记,再用三边测量法或最大似然估计法确认待定位节点的位置信息。APIT算法中的每个未知节点通过不断判别自己是否处在由锚节点所确定的三角形区域内来对自身进

行定位。

MDS-MAP^[8]是一种基于多维标尺的定位算法, 是一种集中式定位算法, 在Range-Based环境下能达到很高的精度。当网络覆盖区域不规则时, 会产生较大的误差。MDS-MAP(P)^[9]是一种分布式定位算法, 是对MDS-MAP的改进, 该算法需要各个节点进行分簇及执行最短路径算法, 计算复杂度高, 节点能耗大。基于密度的分簇算法MDS-MAP(DB)不同于MDS-MAP(P)算法为每个节点建立一个局部定位图, 而是根据节点密度情况快速将整个网络划分为若干个局部可定位区域, 简称LPA(local positioning area), 并且在每个LPA中只有簇头节点利用三边不等法计算不相邻节点间的距离, 降低了测距误差及计算复杂度, 提高了算法的定位精度, 并且能够实现快速定位。

1 MDS-MAP(P)定位算法

MDS-MAP(P)算法采用了分布式的思想, 把大型网络分成多个相对均匀、较规则的小型网络。它的基本步骤是:

1) 局部子图的构建。网络中每个节点构建自身的局部子图, 局部子图由每个节点及其 R 跳(R -hop)范围内的邻居节点组成。一般设 R 为2。

2) 用最短路径法计算每个局部子图中节点间的距离并构建距离矩阵。通过MDS方法计算每个子图的相对坐标, 并利用最小二乘法对计算结果进行优化处理。

3) 按照一定的顺序把所有局部子图合并成一个全局图, 既网络中所有网络节点的相对坐标。合并子图的方法有平行合并子图法和顺序合并子图法。

4) 和MDS-MAP定位算法一样, 根据网络中的锚节点坐标, 把全局相对坐标转化为全局绝对坐标。

MDS-MAP(P)合并的过程主要通过增加两个相邻子图的公共节点的数目来降低误差。显然, 两个相邻子图越大($R \geq 2$), 公共节点数就越多, 定位精度越高, 但由最短路径算法求得的不相邻两节点的距离误差变大。如果两个相邻子图选择1跳范围内的邻居节点($R = 1$), 虽然不相邻两节点的测距误差少, 但公共节点数偏少, 定位精度低, 并且MDS-MAP(P)算法需要每个节点进行分簇及执行最短路径算法, 计算复杂度高, 节点能耗大。针对MDS-MAP(P)定位算法存在的问题, 基于密度分簇的定位算法MDS-MAP(DB)按照邻居节点密度的高低对节点进

行排序, 并对排序后的节点进行分簇, 使相邻簇域间具有充分的公共节点, 且簇域是由1跳邻居节点组成, 不相邻两节点的距离用三边不等法计算。如果某一节点1跳内的簇域与前一簇域公共节点数小于某一阈值, 那么该簇域由其及2跳内的节点组成, 不相邻两节点的距离用最短路径算法求出。这样在降低了测距误差的同时保证了两簇域间有充分的公共节点。各簇域内只有簇头节点进行测距并指定一个簇头节点执行局部网络的融合算法, 与MDS-MAP(P)相比能有效地降低合并过程中的误差、测距误差及算法复杂度。

2 基于密度分簇的定位算法

首先建立网络模型。给定一个网络 $G = (V, E)$, 其中, $V = \{v_1, v_2, \dots, v_n\}$ 表示网络中传感器节点集合, $e(i, j) \in E$ 表示节点 i 与节点 j 的边, 定义函数 $d(i, j): E \rightarrow R$ 为节点 i 与节点 j 之间的距离。

基于密度分簇的定位算法MDS-MAP(DB)包括3个步骤: 1) 基于邻居节点密度的分簇算法; 2) 局部网络节点定位; 3) 局部坐标融合及全局绝对坐标建立。

2.1 基于邻居节点密度的分簇算法

网络稳定后, 网络中每个节点泛洪自己的节点信息并建立自身的2跳内节点列表。 $N(i) = \{j: (i, j) \in E\}$ 表示节点 i 在1跳内的邻居节点, $N'(i)$ 表示节点 i 两跳内的节点, $N(i) \subset N'(i)$ 。 N_i 与 N'_i 分别表示节点 i 在1跳邻居节点的个数及2跳内节点个数。则 $S = \{N_1, N_2, \dots, N_n\}$ 表示按节点序列 $I = \{1, 2, \dots, n\}$ 排序的1跳邻居节点个数集合。对 S 按递减排序, 排序后为 $S' = \{N'_1, N'_2, \dots, N'_n\}$, 对应的节点标号序列为 $I' = \{1', 2', \dots, n'\}$ 。

1) 在 I' 集中选取第一个元素 $1'$ 作为分簇算法的起始簇头节点, 其簇域由 $N(1') \cup 1'$ 组成, 记为 $LPA(1')$, 既 $LPA(1') = N(1') \cup 1'$ 。

2) 令 $I' = I' - \{1'\} \cup N(1')$ 。

3) 设置常量 $LPA(0') = LPA(1')$, $\varepsilon = 5$ 。

4) 在 I' 中选取第一个元素 i' 作为新簇的开始节点, 其簇域记为 $LPA(i')$ 。

5) 若 $LPA(j') \cap LPA(i') \leq \varepsilon$, $LPA(j')$ 表示前一个簇域, ε 表示两簇域的公共节点数。则 $LPA(i')$ 由节点 i' 与其两跳内的节点 $N'(i')$ 组成, 即 $LPA(i') = i' \cup N'(i')$, $I' = I' - \{i'\} \cup N'(i')$ 。

6) 若 $LPA(j') \cap LPA(i') > \varepsilon$, 则 $LPA(i')$ 由节点

i' 与其1跳内的邻居节点 $N(i')$ 组成, 即 $LPA(i') = i' \cup N(i')$, $I' = I' - \{i' \cup N(i')\}$ 。

7) 重复步骤4)~步骤6), 直到 $I' = \emptyset$ 。

分簇结束后, 簇域主要分为两种情况, 一种由簇头及1跳内邻居节点组成; 一种由簇头及2跳内节点组成。一般情况下, 第一种簇域的个数要大于第二种的数目。同时属于多个簇的公共节点起到簇域间的连接作用。簇域间的融合算法由簇头间指定一个簇头来执行, 其他簇头只需提供各自簇域内节点坐标。

2.2 局部网络节点定位

对于第一种情况的簇域, 簇域内不相邻节点的距离用三边不等式方法^[10]计算; 对于第二种情况, 不相邻节点的距离用Dijkstra算法产生。局部网络节点 a_i 的定位处理过程采用文献[11]给出的3个步骤计算:

1) 估算簇域内节点间距离。

$LPA(i)$ 由节点 i 及 $N(i)$ 组成, 且 $LPA(i)$ 内节点 i 到 $N(i)$ 的距离为已知值, 对于不相邻的两节点 i 与 j 的距离 d_{ij} , 可以通过三边不等式来求出, 定义 B_{ij} 和 b_{ij} 分别为 d_{ij} 的上界和下界, 有:

$$B_{ij} = \min_{h:(i,h) \in E, (h,j) \in E} \{d_{ih} + d_{hj}\} \quad (1)$$

$$b_{ij} = \max_h \{\max\{d_{ih}\}, \max\{d_{hj}\}\} \quad (2)$$

则可以通过下式估计:

$$d_{ij} = (B_{ij} + b_{ij})/2 \quad (3)$$

若 $LPA(i)$ 由节点 i 及 $N'(i)$ 组成, $LPA(i)$ 内不相邻节点间的距离 d_{ij} 利用Dijkstra算法求出。

2) 通过步骤1)后, 建立簇域 $LPA(i)$ 内节点间的距离矩阵 D , 运用MDS方法求出簇域 $LPA(i)$ 内各节点的相对坐标。

3) 协强系数(stress)优化。

通过式(4)和式(5)对步骤2)求出的 $LPA(i)$ 的相对坐标迭代优化:

$$a_i \leftarrow \frac{1}{N_i} \sum_{j \in N(i)} [a_j + d_{ij}(a_i - a_j) \text{inv}(\|a_i - a_j\|)] \quad (4)$$

$$\text{inv}(x) = \begin{cases} 1/x & x \neq 0 \\ 0 & x = 0 \end{cases} \quad (5)$$

2.3 局部坐标融合

局部簇域的合并按照排序后的簇头节点序列 I' 进行合并, 直到所有的节点都包含在全局地图中, 利用文献[12-13]提出的融合算法求出全局网络的绝对坐标。假设 $LPA(i)$ 与 $LPA(j)$ 为两个相邻局部网

络, 并且 $N(i) \geq N(j)$ 。通过式(6)把 $LPA(j)$ 节点坐标变化到 $LPA(i)$ 的坐标系里。

$$x_i = sR(x_j) + x_0 \quad (6)$$

式中, R 为旋转因子; s 为缩放因子; x_0 为平移因子。

3 算法分析与仿真

假设网络中簇域的个数为 k 个, 簇域内平均邻居节点数为 m 。簇头及1跳内邻居节点组成的簇域个数为 i , 簇头及2跳内节点组成的簇域个数为 $k-i$, 一般情况下 $i > k-i$, 网络节点数为 n 。根据第2节 MDS-MAP(DB)算法描述可得: 2.1节分簇算法时间复杂度为 $O(nm^2)$, 2.2节局部网络节点定位时间复杂度为 $O(km^3)$, 2.3节局部坐标融合算法时间复杂度为 $O(n)$, 与MDS-MAP(P)相比, 节点分簇算法时间复杂度由 $O(n^3)$ 降到 $O(nm^2)$, 局部网络节点定位时间复杂度由 $O(m^3n)$ 降到 $O(km^3)$, 局部坐标融合时间复杂度由 $O(k^3n)$ 降到 $O(n)$, 其中 m 远远小于 n 。

算法仿真在Matlab7.7.0平台下进行。分别在随机均匀分布的矩形网络及C型网络里与MDS-MAP(P)进行平均定位误差对比分析。图1为200个节点随机部署在一个 $10r \times 10r$ 的方形区域里的邻居关系图, r 表示单位长度, 此处 $r=10$ 。图2为200个节点随机部署在一个C形区域里的邻居关系图, 范围为 $10r \times 10r$ (此处 $r=10$)。两种部署情况含相同的锚节点个数, 均为6个。节点通信半径 $a=15$ 。假设未知节点经定位算法估计的坐标位置为 $X_{e,i}$, 其实际坐标位置为 $X_{t,i}$, 则平均定位误差定义为:

$$\text{error} = \sum_{i=1}^n \sqrt{(X_{e,i} - X_{t,i})^2} / (n \times r)$$

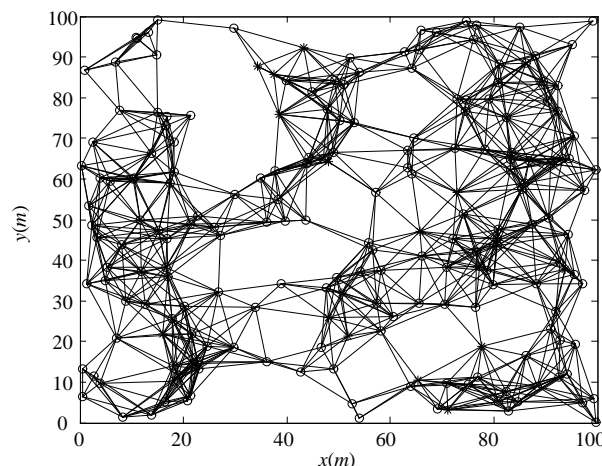


图1 矩形区域节点邻居关系图

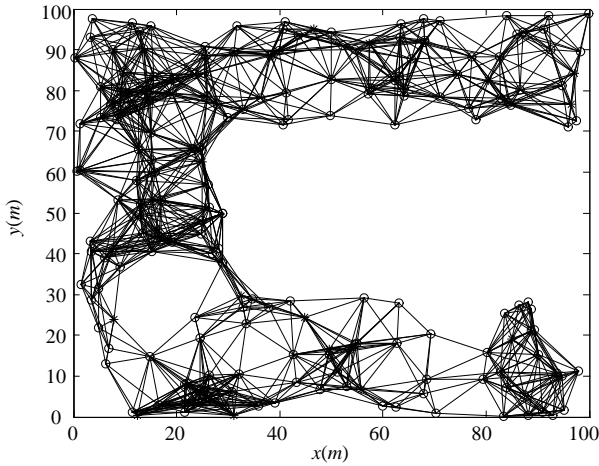


图2 C形区域节点邻居关系图

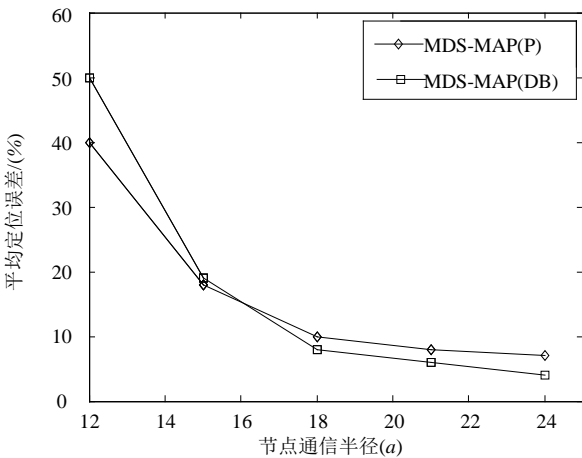


图3 矩形区域定位误差对比图

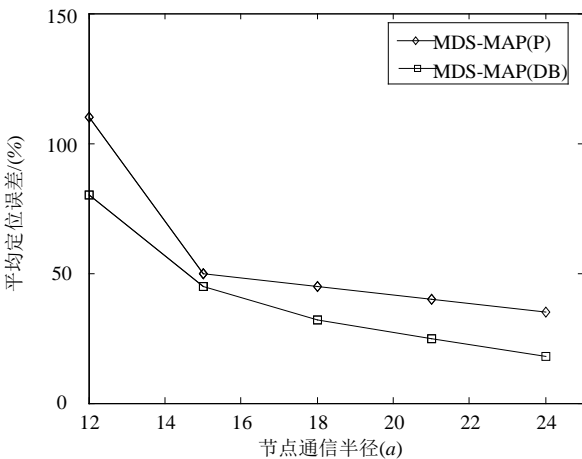


图4 C形区域定位误差对比图

图3及图4分别显示了在矩形网络和C形网络中两种算法随通信半径 a 的变化而导致的定位误差的变化。从图3中可以看出,当通信半径偏低时,本文算法包含的两跳簇域比较多,与MDS-MAP(P)相比,定位误差偏高主要在于簇域数量及公共节点数偏低。当通信半径达到一定值时,节点平均连通度增加,簇域主要由一跳簇域组成,簇域间公共节点数量增多,不相邻节点间距离用三边不等法则测量,

测距误差降低,定位精度提高。

4 结论

本文提出了一种基于密度分簇的定位算法MDS-MAP(DB)。算法分为3个阶段:根据邻居节点密度对网络分簇;局部网络节点定位;局部坐标融合及全局绝对坐标建立。该算法与MDS-MAP(P)相比,能有效地降低2跳内节点距离误差值及计算复杂度,减少网络能耗损失,提高网络的生存能力。通过仿真实验证明,与MDS-MAP(P)算法相比,MDS-MAP(DB)具有较高的定位精度。

参考文献

- [1] 王福豹, 史龙, 任丰原. 无线传感器网络中的自身定位系统和算法[J]. 软件学报, 2005, 16(5): 857-868.
WANG Fu-bao, SHI Long, REN Feng-yuan. Self-localization systems and algorithms for wireless sensor networks[J]. Journal of Software, 2005, 16(5): 857-868.
- [2] BAHL P, PADMANABHAN V N. RADAR, an inbuilding RF-based user location and tracking system[C]//Proceedings of the IEEE INFOCOM. Tel Aviv, Israel: IEEE, 2000.
- [3] HARTER A, HOPPER A, STEGGLES P, et al. The anatomy of a context aware application[J]. Wireless Networks, 2002, 8(2-3): 187-197.
- [4] GIROD L, ESTRIN D. Robust range estimation using acoustic and multimodal sensing[C]//IEEE International Conference on Intelligent Robusts and Systems. [S.l.]: IEEE, 2001.
- [5] BULUSU N, HEIDEMANN J, ESTRIN D. GPS-less low cost outdoor localization for very small devices[J]. IEEE Personal Communications Magazine, 2000, 7(5): 28-34.
- [6] NICOLESCU D, NATH B. Ad Hoc positioning systems (APS)[C]//Proceedings of IEEE GLOBECOM. [S.l.]: IEEE, 2001.
- [7] HE Tian, HUANG Cheng-du, BRIAN M B, et al. Range-free localization schemes for large scale sensor networks [C]//Proceedings of the 9th Annual International Conference on Mobile Computing and Networking. [S.l.]: [s.n.], 2003.
- [8] SHANG Y, RUML W, ZHANG Y. Localization from connectivity in sensor networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2004, 15(11): 961-974.
- [9] SHANG Y, RUML W. Improved MDS-based localization [C]//Proc of the IEEE Infocom. Hong Kong, China: IEEE, 2004.
- [10] ZHANG Lei, LIU Li-gang, GOTSMAN C, et al. An as-rigid-as-possible approach to sensor network localization[J]. ACM Transactions on Sensor Networks, 2010, 6(4): 1-21.
- [11] GOTSMAN C, KOREN Y. Distributed graph layout for sensor network[J]. Journal of Graph Algorithms and Application, 2005, 9(3): 327-346.
- [12] 马震, 刘云, 沈波. 分布式无线传感器网络定位算法MDS-MAP(D)[J]. 通信学报, 2008, 29(6): 57-62.
MA Zhen, LIU Yun, SHEN Bo. Distributed locating algorithm for wireless sensor networks-MDS-MAP(D)[J]. Journal on Communications, 2008, 29(6): 57-62.