

高密度无线传感器网络分簇定位算法

刘 瑜, 衣 晓, 何 友

(海军航空工程学院信息融合技术研究所, 山东 烟台 264001)

摘 要: 节点自身定位是无线传感器网络应用的支撑技术之一。提出了一种适用于大规模高密度无线传感器网络的分簇定位算法。首先定义了节点的势作为簇首选举依据, 网络中节点间的距离由接收信号强度和通信半径的关系间接计算得到, 各簇内的拓扑信息由簇首保存, 簇首利用线性规划法实现簇内相对定位; 随后从 sink 节点开始逐步进行簇间位置融合, 最终实现全网的绝对定位。相比集中式的凸规划定位算法, 所提算法计算复杂度低、通信量小、定位精度高, 且不需要预先知道环境中的信号衰减因子, 有一定的抗噪声干扰能力。仿真结果显示, 在节点按均匀网格分布和均匀随机分布两种情况下, 所提算法能取得较好的定位效果。

关键词: 无线传感器网络; 节点自身定位; 线性规划; 分簇

中图分类号: TP 393

文献标志码: A

DOI: 10.3969/j.issn.1001-506X.2012.08.11

Cluster localization scheme for high-density wireless sensor networks

LIU Yu, YI Xiao, HE You

(Research Institute of Information Fusion, Naval Aeronautical and Astronautical University, Yantai 264001, China)

Abstract: The node self-localization is one of the supporting technologies in wireless sensor networks. A distributed cluster localization scheme (linear programming-cluster localization scheme, LP-CLS) is introduced, which is based on linear programming. First, the authority of node is defined as the criterion for voting cluster and the distance between neighbor nodes is estimated by the relation between received signal strength indicator and communication range. Then, according to linear programming, the relative coordinates of nodes in a cluster are calculated out by the relevant cluster node. Finally, starting from sink node, coordinates between neighbor clusters are combined and the whole relative coordinates are converted to absolute coordinate. The simulation results indicate that LP-CLS outperforms Convex significantly in the aspect of localization accuracy and algorithm complication whether nodes are placed regularly or randomly.

Keywords: wireless sensor networks (WSN); node self-localization; linear programming; cluster

0 引 言

无线传感器网络(wireless sensor networks, WSN)^[1]是由部署在监测区域内大量的廉价微型传感器节点组成, 能够协作地感知、采集和传输网络覆盖区域中感知对象的信息。WSN 广泛应用于军事侦察与监控、紧急救灾抢险等领域。

在传感器所采集的数据中, 与传感器的位置相结合的信息才有意义, 节点的位置信息对传感器网络的监测活动至关重要^[2]。节点自身定位就是网络中的未知节点根据少数已知位置的锚节点, 按照某种定位机制确定自身的位置。现有大部分定位算法按照定位过程中信息的处理方式可分为两类^[3-16]: 集中式定位算法和分布式定位算法。集中式

计算是指把所需信息传送到某个中心节点, 并在那里进行节点定位计算, 优点在于从全局角度统筹规划, 计算量和存储量几乎没有限制, 可以获得相对精确的位置估算。它的缺点包括与中心节点位置较近的节点会因为通信开销大而过早地消耗完电能, 导致整个网络与中心节点信息交流中断而无法实时定位等, 不适用于节点数目庞大的 WSN。集中式的代表算法主要有多维定标(multi-dimensional scaling-map, MDS-MAP)算法和凸规划算法(Convex)等。分布式计算是指依赖节点间的信息交换和协调, 由节点自行计算的定位方式。

本文旨在满足定位精度的前提下, 设计低开销(较少通信量)、低成本(较少信标)的定位算法, 以降低传感器网络定位过程中节点的能耗, 延长网络生命周期。为此, 提出了

收稿日期: 2011-08-30; 修回日期: 2012-05-08。

基金项目: 山东省自然科学基金(ZR2011FQ002)资助课题

作者简介: 刘瑜(1986-), 男, 博士研究生, 主要研究方向为多传感器信息融合。E-mail: liuyuy77360132@126.com

一种基于线性规划的分簇定位算法 (linear programming-cluster localization scheme, LP-CLS), 该算法属于混合式的信息处理方式。该算法无需预先知道环境中的信号衰减因子, 节点仅根据其邻居的连通性信息即可实现分簇和间接测距, 实现简单, 各个簇内进行一次位置计算, 二维监测区域只需 3 个信标节点即能实现全网定位, 定位精度能够满足大多数应用要求, 计算复杂度和通信开销较小, 且能有效抵抗噪声干扰, 适用于大规模高密度部署的 WSN 节点自身定位。

1 适用于定位的分簇算法

1.1 网络模型及预备知识

首先将 WSN 表示为一个无向图 $G=(V, E)$, 其中, 顶点为 V , 对应着网络中的传感器节点, 边为 E , 对应节点之间的距离或者跳数等相关信息。网络中每个节点具有一个唯一的编号, 称为 ID。每条边连通的两个顶点 (即能够直接通信的两个节点) 互称为邻居节点, 没有邻居的节点称为孤立节点, 同时属于两个或多个簇的节点称为中间节点。

对于大规模 WSN 定位问题, 分簇的目的是为了使网络具有可扩展性, 把整个网络划分为多个子图实现分布式定位, 降低网络通信量和计算复杂度。分簇的要求是适合簇内定位算法的实现, 同时在保证相邻簇间连通的基础上, 尽可能减少簇的数量。簇的数量减少可使簇融合过程中的通信量降低, 计算量减小。同时分簇还要使各相邻簇之间存在着 2 个或更多的中间节点, 使相邻簇构成的图不存在割集。

由于分簇定位算法属于全局分布式、簇内集中式的计算方式, 簇首不仅需要存储大量簇内节点的相关信息, 还需要对节点簇进行管理, 其通信量和计算量无疑要大于簇内节点, 能耗也高于簇内节点, 故簇首的拓扑位置与剩余能量对于整个簇的性能 (特别是寿命) 有着巨大影响。本文为表征节点在网络中拓扑位置的重要性以及当选簇首的能力, 综合考虑节点平均连通度和剩余能量等信息, 引入节点平均连通度和节点势两个参数辅助分簇。

定义 1 节点连通度 (node connectivity, NC)

若网络中所有传感器节点构成的集合为 $U(x)$, 其基数为 $f_c(U(x))=N$, 表示集合 $U(x)$ 内元素的个数为 N , 对于某一节点 $x_i \in U(x) (1 \leq i \leq n)$, 其邻居节点集合为 $NB_{x_i}(x)$, 其基数为 $f_c(NB_{x_i}(x))=cn$, 则节点 x_i 的平均连通度 NC_{x_i} 为

$$NC_{x_i} = \frac{\sum_{k=1}^m f_c(NB_{x_k}(x))}{cn} \quad (1)$$

式中, $x_k \in NB_{x_i}(x)$ 为节点 x_i 的邻居节点; $NB_{x_k}(x)$ 即为节点 x_k 的邻居节点集合。

定义 2 节点势 C

网络中节点 x_i 的平均连通度为 NC_{x_i} , 剩余能量为 RP_{x_i} , 则节点 x_i 的势 C_{x_i} 为

$$C_{x_i} = \frac{NC_{x_i}}{NC_{\max}} + \frac{RP_{x_i}}{P} \quad (2)$$

式中, NC_{\max} 为节点 x_i 的邻居节点中平均连通度的最大值; P 为初始能量。

1.2 基于节点势的分簇算法

分簇过程从位于网络中心的 sink 节点开始, 首先以 sink 节点作为其中一个簇首, 簇内节点依次按规则寻找邻近簇首, 直至网络中除孤立节点外的任一节点都能找到对应的簇首。如果相邻簇的公共节点 (即中间节点) 只有一个, 则将此中间节点升级为簇首, 再次搜索邻近簇首。按照以上方法, 簇从 sink 节点逐渐向网络各个方向扩展, 直到整个网络被覆盖。簇首为 x' 的簇内节点寻找相邻簇首的具体规则如图 1 所示。

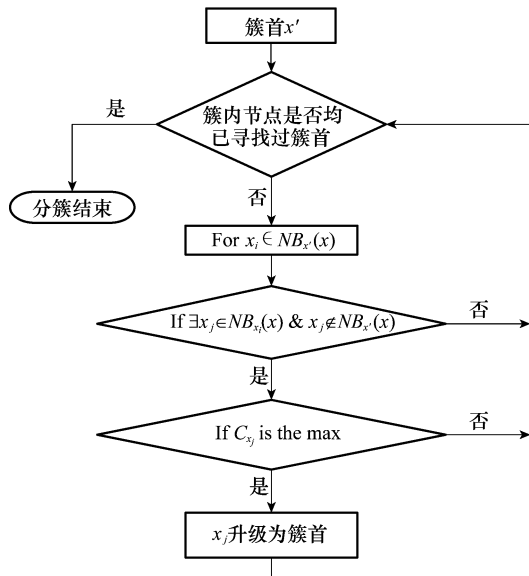


图 1 节点寻找簇首流程图

随后进行重叠簇的检测, 若某簇的簇内节点全部属于另一个簇, 则取消该簇, 簇首降为普通节点。

对于整个网络, 为使任何两个簇之间可以通过单跳或者多跳的方式进行融合, 相邻簇之间必须是相互连通的。图 2 即为仿真所得的簇首在网络中的相对位置, 是一种星形与树形的混合结构, 从网络中心的 sink 节点开始, 把各个方向的邻居簇首作其子节点, 逐步向外扩展, 最终覆盖整个网络。从网络分簇所需时间方面分析, 本文分簇算法的实现过程是中心向四周全方位的并发式寻找簇首, 相比单一的树形递进式拓扑更节省时间。

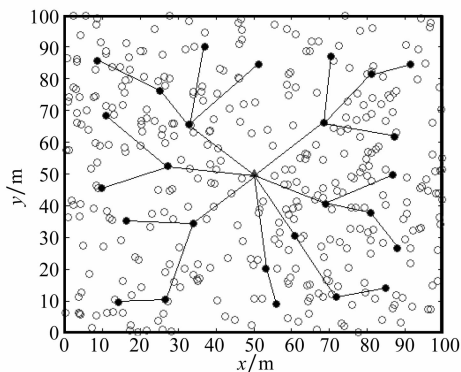


图 2 簇首拓扑结构

1.3 分簇算法分析

对于单个簇来说,簇内节点到簇首的跳数直接关系到簇内通信量的大小,而簇内节点数目对簇首执行定位算法的计算量有直接影响。考虑到本文定位算法是基于静态的节点密集分布的传感器网络,为防止单个簇内节点过多而过早消耗完簇首的能量,本文设计的分簇算法限定簇首的管理范围仅为一跳通信半径内,即簇首管理的节点是自身邻居节点。若网络中所有簇首构成的集合为 $C(x') \subset U(x)$,以下证明采用基于节点势的分簇方法,除孤立节点之外,网络中所有节点都能被簇覆盖。

定理 1 节点 $x_i \in U(x)$,若 $f_c(NB_{x_i}(x)) \neq 0$,其中, $f_c(NB_{x_i}(x))$ 为集合 $NB_{x_i}(x)$ 的基数,则 $\exists x'_k \in C(x')$,使得 $x_i \in NB_{x'_k}(x)$ 。

证明 根据基于节点势的分簇算法,网络中除了 sink 节点外每个节点都需要寻找相邻的簇首,相邻簇之间至少存在一个中间节点,即 $f_c(NB_{x'_i}(x) \cap NB_{x'_j}(x)) \geq 1$, $x'_i, x'_j \in C(x')$ 为相邻的两个簇首。若 $NB_{x'_i}(x) \cap NB_{x'_j}(x) = \{x_k\}$,则根据分簇算法所述, x_k 升级为簇首 x'_k ,可得 $x'_i, x'_j \in NB_{x'_k}(x)$ 。

故 $\{x'_i, x'_k\} \subset NB_{x'_i}(x) \cap NB_{x'_k}(x), \{x'_j, x'_k\} \subset NB_{x'_j}(x) \cap NB_{x'_k}(x)$,满足相邻簇之间至少存在两个中间节点的要求。也就是说,只要网络中的节点存在邻居节点,则此节点如果不属于以邻居节点为簇首的簇内,那就是邻居节点的簇首。即除孤立节点之外,网络中不存在不被簇覆盖的节点。证毕

若网络运行过程中节点的拓扑结构不改变,则全网进行一次分簇定位即可。若分簇定位后有新节点加入网络,则新加入的节点首先寻找通信范围内是否存在簇首。若有,则加入该簇首所在的簇;否则,新加入节点以自身为簇首构建新簇,根据邻居节点的位置信息即可实现定位。

2 簇内定位算法

2.1 算法思想

将 WSN 看成一个无向图 $G=(V, E)$,若节点 $u, v \in V$ 间可以通信,对应的接收信号强度(received signal strength, RSS)为 $RSS_{u,v}$,对于顶点 u 和 v 构成的边 $e(u, v) \in E$ 都存在一个与其相关联的标量 $s(u, v) \in S$,在本文中令

$$s(u, v) = RSS_{u,v} \quad (3)$$

设 $U \subset V$ 为未知节点的集合,则节点自身定位可以描述为未知节点 $u \in U$ 的坐标求解问题。

设各个簇内任意两个可通信节点 u 和 v 之间的距离为 $d(u, v)$,则可以利用 $d(u, v)$ 与 $s(u, v)$ 的关系,找出簇内所有可通信节点之间的相对距离,进而利用线性规划方法计算出各节点相对于簇首的坐标,最后各簇进行融合,形成全网相对坐标,再根据 3 个或者 3 个以上信标信息将相对坐标转化为绝对坐标,实现全网定位。

2.2 簇内节点间距离计算方法

假设传感器节点在监测区域内均匀随机分布,节点拥

有相同的通信半径 r ,通信范围是一个规则的圆形区域。如图 3 所示。

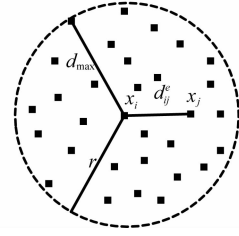


图 3 节点间距离计算方法示意图

考虑某一节点 x_i 的通信范围内的网络拓扑结构,邻居节点 x_j 到 x_i 的 RSS 为 RSS_{ij} ,估计距离为 d_{ij}^e ,由对数-常态分布模型可知

$$RSS_{ij} = RSS(d_0) - 10n \lg \left(\frac{d_{ij}^e}{d_0} \right) - X_e \quad (4)$$

式中, $d > d_0$, d_0 是参考距离, $RSS(d_0)$ 为对应的参考能量; X_e 是均值为 0 的高斯变量,表示系统误差等因素; n 为信号衰减因子。令

$$A = RSS(d_0) - X_e \quad (5)$$

则有

$$RSS_{ij} = A - 10n \lg d_{ij}^e \quad (6)$$

$$10n \lg d_{ij}^e = A - RSS_{ij} \quad (7)$$

若圆内均匀随机分布足够多的节点,则可以根据 RSS 与距离成反比的关系,认为最小的接收信号强度 RSS_{\min} 对应最大的距离 d_{\max} ,则有

$$10n \lg d_{\max} = A - RSS_{\min} \quad (8)$$

$$\frac{10n \lg d_{ij}^e}{10n \lg d_{\max}} = \frac{A - RSS_{ij}}{A - RSS_{\min}} \quad (9)$$

$$d_{ij}^e = d_{\max}^{\frac{A - RSS_{ij}}{A - RSS_{\min}}} \approx r^{\frac{A - RSS_{ij}}{A - RSS_{\min}}} \quad (10)$$

可见,由式(10)可以计算出圆心节点与其邻居节点间的估计距离。同理,按此方法可以估算出单个簇内以至于整个网络中所有能够直接通信的两个节点之间的距离。

2.3 线性规划定位算法

各个簇内以簇首 x'_i 为原点建立坐标系, $x'_i \in C(x')$,因为是一跳簇,故簇内共有节点数为 $m(m = f_c(NB_{x'_i}(x)) + 1)$,节点 $x_i(x_{x_i}, y_{x_i})$ 和节点 $x_j(x_{x_j}, y_{x_j})$ 之间的估算距离 d_{ij}^e 可由式(10)计算得到,实际距离为 d'_{ij} , $d'_{ij} \geq d_{ij}^e$,且有

$$d'_{ij} = \sqrt{(x_{x_i} - x_{x_j})^2 + (y_{x_i} - y_{x_j})^2} \quad (11)$$

再考虑单个簇内的网络拓扑结构,若簇内随机均匀分布足够多的节点,由于簇首的通信范围近似为圆形区域,则可认为簇内所有节点的质心近似为簇首的位置。为进一步提高估计精度,引入节点到簇首的距离的倒数作为权值,可以将簇内所有节点的加权质心和簇首位置之间距离向量的范数作为目标函数,以式(11)作为约束条件,将单个簇内的节点定位可以描述为二次约束的规划问题。

$$\begin{cases} \min \|L_x\|_2 \\ \text{s. t. } \sqrt{(x_{x_i} - x_{x_j})^2 + (y_{x_i} - y_{x_j})^2} \leq d'_{ij} \\ 1 \leq i, j \leq m, i \neq j \end{cases} \quad (12)$$

$$L_x = \left[\sum_{i=1}^{m-1} \frac{x_i}{\sum_{i=1}^{m-1} \frac{1}{d_{ik}^e}}, \sum_{i=1}^{m-1} \frac{y_i}{\sum_{i=1}^{m-1} \frac{1}{d_{ik}^e}} \right] \quad (13)$$

式中, d_{ik}^e 为簇内节点 x_i 到簇首 x_k' 的估计距离。考虑到二次约束的规划求解问题具有较大计算量, 不适用于单个普通传感器节点。受文献[15]启发, 本文对二次约束的规划问题进行简化。

由式(11)可知, 理论上节点 x_j 的实际位置分布在以节点 x_i 为圆心, 以 d_{ij}^e 为半径的圆内, 如图 4 中圆形区域部分所示, 若用此圆的外接正方形来代替圆形约束区域, 则式(12)可以近似描述为

$$\begin{cases} \min \|L_x\|_2 \\ \text{s. t.} \begin{cases} |x_{x_i} - x_{x_j}| \leq d_{ij}^e \\ |y_{x_i} - y_{x_j}| \leq d_{ij}^e \end{cases} \\ 1 \leq i, j \leq m, i \neq j \end{cases} \quad (14)$$

从式(14)可以看出, 二次约束的规划问题转化成为线性规划问题。若簇内能直接通信的节点对个数(即可连通的边数)为 k , 则求解二次约束的规划问题的计算复杂度优于 $O(k^3)$, 而求解上述线性规划问题的计算复杂度优于 $O(k^2)$, 故约束条件简化以后大大降低了计算复杂度, 减小了计算量。

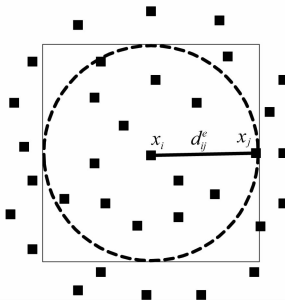


图 4 约束条件简化示意图

各簇首计算出簇内相对坐标后, 以处于网络中间的 sink 节点所在的簇作为初始全局坐标系, 逐步向外实现局部集中式、整体分布式的坐标统一。其中, 相邻簇通过中间节点向全局坐标系转换, 随后中间节点发送新坐标给相邻未合并的簇的簇首。这些簇首因获得中间节点在全局坐标系的坐标, 也可以向全局坐标系转换, 依此方法逐步往外合并, 直至全局融合为一个簇。

两局部坐标系合并时, 需要对其中一个坐标系进行旋转、平移、缩放等变换, 实质是通过矩阵变换来实现, 转换公式为

$$Z_a = S \cdot Z_r \cdot R + T \quad (15)$$

式中, Z_r 是转化前的相对坐标; S 是一个标量, 表示缩放; R 为旋转和反射; T 表示坐标平移。

3 LP-CLS 算法描述

LP-CLS 算法的具体步骤如下:

步骤 1 节点部署后, 网络初始化。每个节点需要发送

两次数据包。首先各节点广播包含自身 ID 的数据包, 则任一节点都能够感知到通信范围内的邻居节点, 并记录邻居节点的 ID 和对应的 RSS; 随后各节点再将自身邻居节点的 ID 广播出去, 同样, 各节点能够获知通信范围内所有节点的邻居节点(称为两跳邻居节点)信息, 并再次记录此信息的 RSS。至此, 网络中任一节点 x_i 存储的信息可由表 1 来表示。

表 1 网络中节点 x_i 存储的信息列表

邻居节点	两跳邻居节点信息	第一次 RSS	第二次 RSS
x_l	$NB_{x_l}(x)$	RSS_{il}^1	RSS_{il}^2
x_j	$NB_{x_j}(x)$	RSS_{ij}^1	RSS_{ij}^2
\vdots	\vdots	\vdots	\vdots
x_k	$NB_{x_k}(x)$	RSS_{ik}^1	RSS_{ik}^2

表中, $x_l, x_j, \dots, x_k \in U(x)$ 。

步骤 2 各节点计算自身到邻居节点的距离。各节点将信息列表中对对应邻居节点的两跳 RSS 求均值, 取其最小值 RSS_{\min} , 对应距自身最远的邻居节点。取 $d_{\max} = r$, 则可利用式(10)估计出自身到邻居节点的距离。

步骤 3 采用基于节点势的分簇算法对网络进行划分。通过步骤 1 的两跳信息收发, 节点获知两跳通信半径内节点的个数和对应的 ID, 则可以利用式(2)计算出节点势。分簇首先从处于网络中央的 sink 节点开始, 采用 1.2 节所述的基于节点势的分簇算法对网络进行分簇。

步骤 4 簇内相对定位。簇内节点广播邻居节点到自身的距离, 簇首接收后记录本簇内节点对的距离信息。这样簇首就能够获知本簇内所有能通信的节点对之间的距离, 则可利用 2.3 节所述的线性规划法求解簇内节点的相对坐标。

步骤 5 簇间坐标融合。以 sink 节点所在的簇为基准, 逐步向外实现局部集中式、整体分布式的坐标统一。相邻簇通过中间节点进行坐标转换, 直至全网融合为一个簇, 实现全网定位。

4 仿真及性能分析

4.1 仿真参数设置

仿真实验的实验参数和条件为(为使仿真结果更为直观及不失一般性, 本文中的距离参数均以米(m)为单位)

(1) 实验环境: 100 m × 100 m 的标准矩形平地区域分布 400 个节点, 所有节点的最大通信距离为 r 。

(2) 节点势: 考虑到实际环境中传感器节点在能量消耗上的差异, 本文仿真时令节点的剩余能量在 $(0.5 \sim 0.8)P$ 之间随机分布, 其中 P 表示节点预装的初始能量。

(3) 平均连通度(average node connectivity, ANC), 表示网络中节点连通度的平均值, 为

$$ANC = \frac{\sum_{i=1}^N NC_{x_i}}{N} \quad (16)$$

(4) 信号衰减因子 n , 表示信号能量随距离衰减的速率。本文仿真为模拟环境真实性, 引入高斯噪声 X_n , 取 $\sigma = 4$ 。

(5) 定义平均定位精度 (average localization accuracy, ALA) 为

$$ALA = \frac{1}{Nr} \sum_{i=1}^N \sqrt{(x_{x_i}^e - x_{x_i}^a)^2 + (y_{x_i}^e - y_{x_i}^a)^2} \quad (17)$$

式中, $(x_{x_i}^a, y_{x_i}^a)$ 为未知节点 x_i 的实际坐标, $(x_{x_i}^e, y_{x_i}^e)$ 为 x_i 的估计坐标。

为体现本文方案的优越性, 仿真选取经典的凸规划定位算法进行比较。

4.2 定位误差随 ANC 变化情况

在 $n=2$ 的网络环境下, 图 5 显示了节点均匀网格分布和均匀随机分布时算法的定位精度随 ANC 的变化情况。

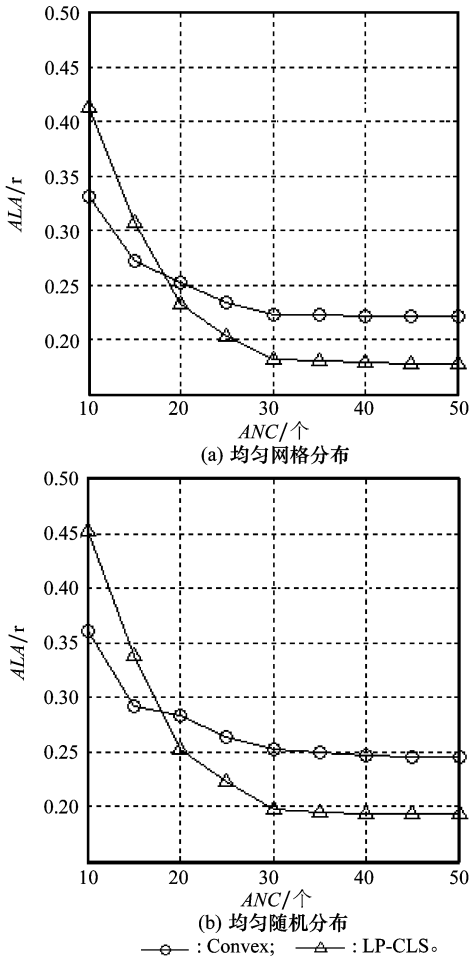


图 5 定位精度随 ANC 变化情况

可见, Convex 和 LP-CLS 的定位误差均随 ANC 的增加而逐渐减小, 这是因为传感器节点的邻居节点越多, 节点之间的约束信息也就越多, 而这两种算法的主要思想就是利用节点间的约束信息来实现定位, 所以 ANC 的增加为算法求解更精确的定位结果提供了条件。从图 5 可以看出, 当 ANC 较小时, LP-CLS 的定位精度略低于 Convex, 这是因为 LP-CLS 设定簇首到簇内节点的最远距离 $d_{max} = r$, 而低节点密度网络中单个簇内的节点数量过少, 使得 d_{max} 与 r 之间有较大差距, 故节点间存在一定的测距误差, 影响之后

的线性规划定位算法。而随着 ANC 逐渐增加, LP-CLS 的定位精度相对 Convex 有了大幅度的提升, 这是因为节点间的测距精度也是随着 ANC 增加而提高, 且单个簇内节点间的约束信息增多, 有利于线性规划算法实现精确定位, 加之 LP-CLS 将网络划分为多个小区域分别定位, 相比 Convex 更能取得局部最优解。特别地, 当 $ANC=30$ 时, 对于均匀网格部署的情况, LP-CLS 的定位精度比 Convex 提高 5%, 达到 17%, 而对于均匀随机部署的情况, LP-CLS 的定位精度为 19%, 比 Convex 降低 6%, 这充分表现出了 LP-CLS 在节点高密度均匀分布的网络中具有明显优势。

4.3 定位误差随信号衰减因子变化情况

通过改变节点最大通信半径 r 以保持 $ANC = 20$, 图 6 展现了节点均匀网格分布和均匀随机分布时算法定位精度随信号衰减因子的变化情况。

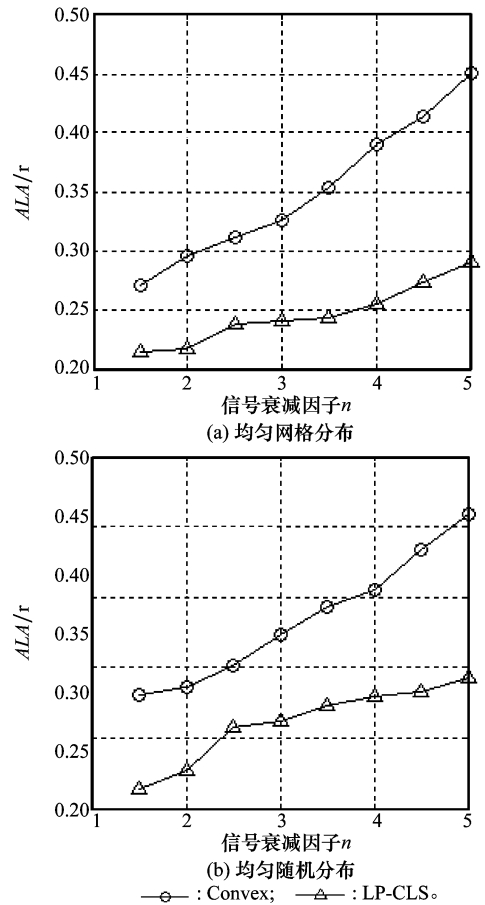


图 6 定位精度随信号衰减因子变化情况

可以看出, 信号干扰对 Convex 定位带来了严重的负面影响。而相对 Convex 来说, LP-CLS 的定位精度随 n 的变化较小, 这是因为 LP-CLS 不是直接利用测距模型来测距, 而是根据网络中节点间的连通性实现间接测距, 故 n 的变化对 LP-CLS 没有太大的影响。但是, n 的增大影响了网络中节点间的连通性, 间接降低了网络平均连通度, 从而对 LP-CLS 的测距带来一定的负面影响。

4.4 能耗分析

WSN 中, 节点的能量主要消耗在通信上, 而发送数据所需的能量占通信能耗的主要部分, 故本文通过计算定位过程中所有节点发送数据包的数量来衡量通信开销。对于集中式的凸规划定位算法, 各节点需要以多跳的方式将自身信息传送到 sink 节点, 而本文算法 LP-CLS 在一个小的区域内进行, 簇内节点仅需与簇首通信, 故在通信开销方面集中式的 Convex 明显大于分簇式的 LP-CLS。

相对于典型的凸规划集中式相对定位算法, LP-CLS 用分布式的分簇算法, 不需要收集全网拓扑信息, 有效减少了后期相对定位阶段和坐标转换阶段所需的通信量, 降低了全网的通信能耗。Convex 中求解全局二次规划问题的计算复杂度为 $O(L^3)$, 其中, L 为整个网络中可连通的边数。而 LP-CLS 的簇内求解线性规划问题的计算复杂度优于 $O(k^3)$, 则整个网络的计算复杂度为 $O(k^3) \times (L/k) = O(Lk^2)$, 其中, k 为簇内可连通的边数。因此, LP-CLS 的计算复杂度要远小于 Convex。在簇间融合方面, LP-CLS 使用递进式坐标合并方式, 从 sink 节点开始, 逐步向外合并, 避免冗余转换, 能有效减低通信量, 提高算法实用性。

5 结 论

本文首先提出一种基于节点势的分簇算法, 原理简单, 容易实现且分簇速度快, 能够以较低的能耗实现对整个网络覆盖, 同时保证相邻簇之间有足够的公共节点。其次, 本文采用的节点间距离计算方法消除了接收信号强度测距模型中信号衰减因子的影响, 具有普遍意义, 是一种适用于未知环境下高密度 WSN 节点间的距离估算方法。再者, 在分簇的基础上, 提出了基于线性规划的分簇定位算法, 相对于 Convex 具有明显的优势, 且降低了计算复杂度和通信开销, 适用于能量受限的高密度 WSN。下一步将研究不规则场景下分簇和定位算法。

参考文献:

- [1] Ren F Y, Huang H N, Lin C. Wireless sensor networks[J]. *Journal of Software*, 2003, 14(2): 1148-1157.
- [2] Chris S, Jan M R, Jan B. Location in distributed ad-hoc wireless sensor network[C]// *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal*, 2001: 2037-2040.
- [3] 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 F B, Shi L, Ren F Y. Self-localization systems and algorithms for wireless sensor networks[J]. *Journal of Software*, 2005, 16(5): 857-868.)
- [4] He T, Huang C D. Range-free localization schemes for large scale sensor networks[C]// *Proc. of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2003: 81-95.
- [5] 衣晓, 刘瑜, 黄越平. 无线传感器网络 Range-free 自身定位算法仿真分析[J]. *海军航空工程学院学报*, 2009, 24(4): 369-375. (Yi X, Liu Y, Huang Y P. Simulation and analysis of range-free self-location algorithm for wireless sensor network[J]. *Journal of Naval Aeronautical and Astronautical University*, 2009, 24(4): 369-375.)
- [6] Niculescu D, Nath B. Ad hoc positioning system (APS) using AOA[C]// *Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2003: 1734-1743.
- [7] 肖玲, 李仁发, 罗娟. 基于非度量多维标度的无线传感器网络节点定位算法[J]. *计算机研究与发展*, 2007, 44(3): 399-405. (Xiao L, Li R F, Luo J. A sensor localization algorithm in wireless sensor networks based on nonmetric multidimensional scaling[J]. *Journal of Computer Research and Development*, 2007, 44(3): 399-405.)
- [8] Shi Q J, He C, Chen H Y. Distributed wireless sensor network localization via sequential greedy optimization algorithm[J]. *IEEE Trans. on Signal Processing*, 2010, 58(6): 3328-3340.
- [9] Wang Y, Wang X. Range-free localization using expected hop-progress in wireless sensor networks[J]. *IEEE Trans. on Parallel and Distributed Systems*, 2009, 20(10): 1540-1552.
- [10] Yi X, Liu Y, Deng L, et al. An improved DV-Hop positioning algorithm with modified distance error for wireless sensor network[C]// *Proc. of the 2nd International Symposium on Knowledge Acquisition and Modeling*, 2009: 216-218.
- [11] Amitangshu P. Localization algorithms in wireless sensor networks: current approaches and future challenges[J]. *Network Protocols and Algorithms*, 2010, 2(1): 45-74.
- [12] King C, King S L, Vincent T. Localization in sensor networks with limited number of anchors and clustered placement[C]// *Proc. of the IEEE Wireless Communications and Networking Conference*, 2007: 4425-4429.
- [13] Jaime L, Jesus T, Miguel G, et al. A hybrid stochastic approach for self-location of wireless sensors in indoor environments[J]. *Sensors*, 2009, 5(9): 3695-3712.
- [14] 苗勇, 崔莉. 一种低计算复杂度的无线传感器网络分簇定位算法[J]. *高技术通讯*, 2009, 19(4): 348-355. (Miao Y, Cui L. A low complexity cluster localization method for wireless sensor network[J]. *Chinese High Technology Letters*, 2009, 19(4): 348-355.)
- [15] 王珊珊, 殷建平, 张国敏, 等. 求解无线传感器网络定位问题的线性规划算法[J]. *计算机研究与发展*, 2009, 46(5): 705-712. (Wang S S, Ying J P, Zhang G M, et al. A linear programming algorithm for wireless sensor networks localization[J]. *Journal of Computer Research and Development*, 2009, 46(5): 705-712.)
- [16] Sangwoo L, Hyunjae W, Chaewoo L. Wireless sensor network localization with connectivity-based refinement using mass spring and Kalman filtering[J]. *The European Association for Signal Processing Journal on Wireless Communications and Networking*, 2012, 152(1): 1-21.