

一种无线传感器网络定位问题中的分簇算法^{*})

王珊珊 殷建平 张国敏 蔡志平

(国防科技大学计算机学院 长沙 410073)

摘要 本文提出一种用于平面无线传感器网络定位的分簇算法。首先,锚节点根据地理位置划分感知区域;然后,通过拓扑发现过程和锚节点间的信息交换,实现锚节点对周围网络拓扑的感知;最后,根据就近原则将所有未知节点分配到以锚节点为边缘的各个子区域中,而分配到各个子区域中的未知节点和边缘的锚节点,则构成网络中的各个簇,并由每个簇的主节点保存本簇内全部拓扑信息。该算法可以实现多跳节点的分簇,并且具有较小的通信量,可用于多种基于分簇的分布式定位算法中,有助于解决大规模无线传感器网络的定位问题。仿真实验结果显示,在锚节点按网格分布和随机分布两种情况下,该算法都可以得到良好的分簇结果。

关键词 无线传感器网络,定位,分簇

Clustering Algorithm for Wireless Sensor Networks Localization

WANG Shan-shan YIN Jian-ping ZHANG Guo-min CAI Zhi-ping

(School of Computer Science, National University of Defense Technology, Changsha 410073, China)

Abstract This paper presents a clustering algorithm for wireless sensor networks localization. First, sensing region of a planar network is divided into several sub-areas by the anchor nodes according to their geographical positions, which are all known. Then, the anchor nodes implement the perception to their surrounding network topology through discovering topologies and communicating among themselves. At last, each unknown node is assigned to a sub-area according to the principle of proximity. The assigned unknown nodes in a sub-area and the anchor nodes at the fringe of the sub-area altogether form one cluster in the network. And the main node of each cluster saves all the topology information in the cluster. This algorithm can implement n -hop nodes clustering and has lesser communication traffic. It can be used in many clustering-based distributed localization algorithms and will be helpful in large scale wireless sensor networks localization problems. Simulation results show that the algorithm performs well when the anchor nodes placed on the grid or randomly.

Keywords Wireless sensor networks, Localization, Clustering

1 引言

无线传感器网络(WSN, Wireless Sensor Networks)是一种具有感知能力、计算能力和无线通信能力的全新的信息处理平台,被广泛应用于国防军事、环境监测、交通管理、医疗卫生等诸多领域,实现复杂的大范围监测和追踪任务,而网络节点自身定位是大多数应用的基础和前提。近些年来,国内外在无线传感器网络节点自身定位方面的研究已经取得了较为丰硕的成果,但是由于节点能量有限和 WSN 本身的特性决定了没有一种通用的 WSN 节点自身定位算法,不同的算法可能适合于不同的应用。

对于一些分布式的定位算法,可以通过分簇的方法来实现。本文针对一些定位问题的需求,提出一种可应用于 WSN 定位问题中的分簇算法。假设平面网络中的锚节点已经根据它们的地理位置将感知区域划分为多个子区域,锚节点位于各个子区域的边缘,该算法通过拓扑发现过程和锚节点间的信息交换实现锚节点对其周围网络拓扑的感知,然后根据这些拓扑信息将所有未知节点分配到各个子区域中,从而将各个子区域中的未知节点和边缘的锚节点一起构成网络中的各个簇。该分簇算法可以将距离锚节点多跳的未知节点划分到

簇中,并且利用合理的拓扑发现算法使得分簇过程的通信量较小。该算法可用于多种基于分簇的分布式定位算法中,实现大规模 WSN 的节点定位,尤其是在锚节点数量较少的情况下,更能体现出该分簇算法的优势,加快网络的覆盖速度。

本文组织如下:第 2 节介绍相关的研究工作;第 3 节提出一种无线传感器定位问题中的分簇算法;第 4 节对实验结果进行分析;最后简要总结全文。

2 相关研究工作

对于大规模的 WSN,网络中有成千上万的传感器节点,采用集中式的定位算法对节点进行定位会产生很高的计算量,并且网络信息收集的过程也会产生极高的通信量。因此,对于大规模的 WSN 的定位问题,一般采用分布式的定位算法。

典型的分布式定位算法,如 Bounding box^[1], DV-hop^[2,3], Euclidean^[2,3]和 Robust Position^[4]等,都是通过节点自身与周围节点间的信息交换和协调,由节点自行计算进行定位的,这类算法需要较高的锚节点密度。还有一些分布式的定位算法是通过将网络分簇或分层来实现的,文献[5]提出了一种基于分簇的相对定位算法,先将网络按照一定规则进

^{*}) 基金项目:国家自然科学基金资助项目(60603062)(60373023);国家重点基础研究发展计划(973)(2007CB310901);湖南省自然科学基金项目(06JJ3035)。王珊珊 硕士生,研究方向为无线传感器网络定位算法;殷建平 教授,博士生导师,研究方向为人工智能、网络算法与网络应用、信息安全等;张国敏 博士生,研究方向为图像处理;蔡志平 博士,研究方向为计算机网络测量、优化算法、信息安全。

行分簇,以每个簇的主节点为原点在各个簇内建立局部坐标系,然后通过局部坐标系间的转换建立全局坐标系。分布式的半定规划定位(DSDP)算法^[6]先将锚节点按照地理位置分簇,将那些与锚节点直接通信的未知节点划分到锚节点所在的簇,在簇内使用半定规划定位算法计算未知节点的坐标,然后选择一些误差较小的点将其升级为锚节点,继续计算其余未知节点,DSDP算法中的分簇仅将距离锚节点一跳的未知节点分配到簇中。凸规划定位算法是一种集中式的定位算法,文献^[7]中已经提出通过将一个大网络分成小的子网络的方法来实现分布式凸规划定位算法,从而解决大规模 WSN 的定位问题,但并没有给出具体的算法和仿真实验。

3 一种 WSN 定位中的分簇算法

假设网络中的锚节点已经根据它们的地理位置将感知区域划分成了多个子区域,当锚节点按网格分布时,可以根据网格来划分;当锚节点随机分布时,可以根据锚节点的已知位置,利用平面三角剖分等算法对感知区域进行划分。如图 1 所示,图 1(a)为锚节点按网格分布,按照每四个锚节点划分为一个子区域;图 1(b)为锚节点随机均匀分布,同样根据锚节点的位置用每四个或三个锚节点划分一个子区域。每个子区域中选出一个锚节点作为主节点。算法将判断其余的每个未知节点应归属于哪个子区域中,而每个子区域中的所有未知节点和边缘的锚节点构成一个簇,从而将所有节点划分为多个簇。分簇之后,在簇内可以根据实际情况采用某种定位算法对未知节点进行定位,例如凸规划定位算法、半定规划定位算法、MDS-MAP 算法等,进而实现对整个网络节点的定位。

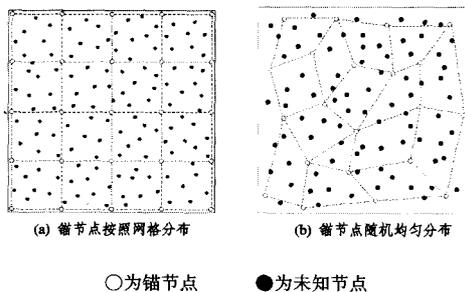


图 1 锚节点将网络划分为多个区域

分簇过程大致可分为两个步骤。

(1) 拓扑感知:包含拓扑发现和拓扑信息交换两方面的内容,即每个锚节点探测其周围的未知节点及网络拓扑结构,并与其周围的锚节点交换发现的拓扑信息,从而实现该锚节点对其周围拓扑的感知。

(2) 分簇:根据锚节点探测得到的网络拓扑信息,采用“就近原则”将所有的未知节点划分到以锚节点为边界的各个子区域中。

3.1 拓扑感知

拓扑感知包括拓扑发现和锚节点间拓扑信息交换两方面的内容。首先,将 WSN 看作一个无向图 $G=(V,E)$, V 为网络中传感器节点的集合, E 为相连接的节点组成的边的集合。 G 中共有 n 个节点,其中 m 个锚节点(位置已知的节点)。若点 $u,v \in V$ 间可以通信,则存在一条边 $e(u,v) \in E$ 。拓扑发现的过程就是为了探测网络的拓扑结构,让每个锚节点了解其周围的点和边。

拓扑发现的过程由锚节点来完成,每个锚节点通过广播探测包来获取周围网络的拓扑结构,这里称广播探测包的锚

节点为源节点。每个源节点广播一个探测包,探测包中的内容包含源节点 ID(PathHead)、包经过的路径 Path 和路径的跳数 hop。广播前 Path 为空, hop 的值为 0。每个源节点广播探测包 $(PathHead, Path, hop)$, 收到包的邻居节点将自己的 ID 加入 Path, 将 $hop+1$, 然后继续广播包。每个节点只广播最先收到的那个包, 将以后收到的包中的内容保存, 不广播。广播过的节点就不再进行广播。具体的广播过程中, 原节点运行 Algorithm 1, 未知节点运行 Algorithm 2。所有的广播结束后, 能确保网络中的每条边都能够被发现, 并且每条边上只进行一次通信, 每个节点在这个过程中只进行一次广播, 因此广播过程总的通信量为 $O(n)$, n 为传感器节点个数。

Algorithm 1 ANCHORNODE_BROADCAST

1. Path ← \square
2. hop ← 0
3. PathHead ← AnchorNodeID
4. Broadcast packet $(PathHead, Path, hop)$

Algorithm 2 UNKNOWNNODE_BROADCAST

1. if receive a packet $(PathHead, Path, hop)$
2. add UnknownNodeID to the Path's end
3. hop ← hop + 1
4. if haven't broadcast
5. broadcast the packet
6. else
7. save the Path
8. end if
9. end if

广播结束后,每个未知节点保存了一些路径,这些路径可能是从不同的源节点传来的,每条路径的头节点 PathHead 就是发出这条路径的源节点。然后,每个未知节点根据保存的所有路径信息建立可达源节点集合 SourceSet, SourceSet 中每个元素包含两项内容:一是可达的源节点,一是该节点到达这个源节点的跳数。然后每个未知节点将其保存的全部内容,如图 2 所示,包括它的 SourceSet 和保存的 Path 一起按照 Pathhead 返回给源节点,无需再次路由。每个源节点收到周围未知节点发来的内容之后,收集每条 Path 中的连通信息(即每条边)和 Path 中所有节点 ID 以及每个节点的 SourceSet, 此时也完成了全部拓扑发现的过程。未知节点发送过程的通信量为 $O(l(n-m))$, l 为最长的 Path 的长度, n 为传感器节点个数, m 为锚节点个数。

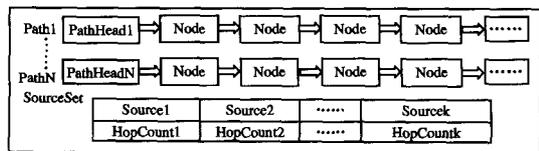


图 2 每个未知节点发送给源节点的内容

拓扑发现结束后,每个锚节点收集了它周围的拓扑结构和其中未知节点的可达源节点集合 SourceSet。为了让拓扑信息交换的内容更加完整,为后面的分簇提供更充分的信息,每个锚节点还要做的一项工作就是对其收集到的所有节点的 SourceSet 进行数据整合。所谓数据整合就是将收集到的每条边两端的两个节点的 SourceSet 置成相同,根据实际情况可重复整合多次。

锚节点之间拓扑信息交换的目的是让已划分好的每个子区域的主节点了解本区域周围的全部拓扑结构信息。主节点是从本子区域的锚节点中选出的一个节点,不仅负责收集本子区域的全部拓扑信息,而且是下一步进行分簇运算的节点。算法保证每个子区域只有一个主节点,并且各个子区域之间没有公共的主节点。信息交换就是让每个子区域中的非主节

点的锚节点, 将其在拓扑发现过程中收集到的拓扑结构信息(包括节点和节点间的连通信息)以及未知节点的 SourceSet 一并发送给本子区域的主节点, 如图 3。图中每个子区域左上角的锚节点为主节点, 其余锚节点分别向各个子区域的主节点发送保存的拓扑信息。

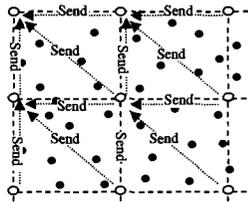


图 3 锚节点向本子区域内的主节点发送拓扑结构信息

整个拓扑感知的过程如图 4 所示。

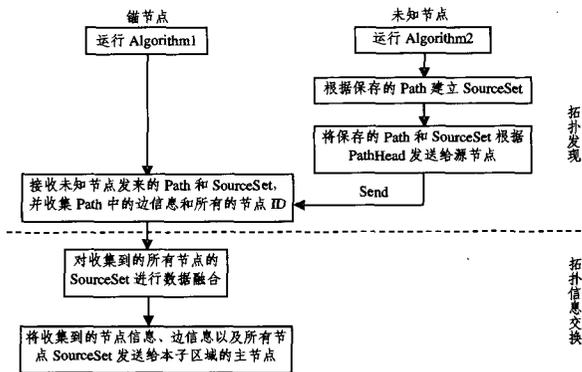


图 4 拓扑感知全过程

3.2 分簇

通过以上过程, 每个子区域的主节点了解了本子区域边缘全部锚节点周围网络的拓扑结构信息。而我们希望的是让主节点只留下本子区域边界以内节点的拓扑信息, 尽量将本子区域边界以外的节点排除出去, 如图 5 所示, 这也就是分簇的过程。

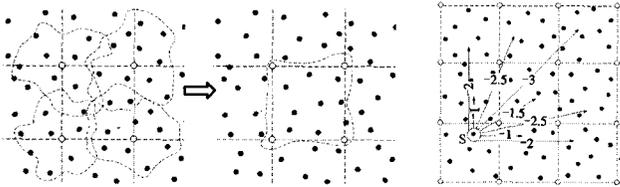


图 5 分簇示意图(将不属于本子区域的节点排除)

图 6 节点 S 被错分时, 分配到不同子区域内的得分情况

分簇工作由每个子区域内完成了拓扑感知全过程的主节点来完成, 分簇遵循一种“就近原则”。所谓“就近原则”是指如果一个未知节点到达某一个子区域边缘锚节点的跳数之和最小, 那么就认为该未知节点在这个子区域内。根据“就近原则”, 如果一个子区域的主节点判断出它感知到的某一个未知节点不在它所在的子区域内, 就将这个未知节点及其相关的连通信息从该主节点保存的拓扑信息中删除, 逐个判断后, 最后剩下的那些没有被删除的节点就被认为是本子区域内的节点, 最后将这些节点以及本子区域边缘的锚节点合起来作为一簇, 从而完成分簇的过程。

下面以图 1(a)中的网络为例, 具体说明分簇过程。在图 1(a)所示的网络中, 每 4 个锚节点划定一个子区域, 虚线表示每个子区域被划定的边界。拓扑感知过程结束后, 每个子区域的主节点收集到的拓扑信息包括周围网络的节点集合、边的集合以及每个节点的可达源节点集合 SourceSet。对于某子区域 C 的主节点 A 来说, 它要对其感知到的每一个未知节点进行判断, 决定其是否属于本子区域。具体分簇过程遵循以下原则:

- (1) 对于主节点 A 收集到的未知节点 a, 在节点 a 的 SourceSet 中找到所有属于同一个子区域的 4 个源节点;
- (2) 根据节点 a 的 SourceSet 中保存的到各个源节点的跳数, 分别对每 4 个同一子区域的源节点求出节点 a 到达该 4 个源节点的跳数和;
- (3) 比较求出的所有跳数和, 如果最小的跳数和所对应的那 4 个源节点不是子区域 C 的锚节点, 则将节点 a 及其相连的边从主节点 A 保存的拓扑信息中删除;
- (4) 如果没有从节点 a 的 SourceSet 中找到同一个子区域的 4 个源节点, 则寻找 3 个同一子区域的源节点, 之后过程类似(2)-(3)。

4 实验结果

首先给出分簇结果的评价标准。在理想情况下, 每个子区域边缘锚节点连接形成的多边形所包围的那些未知节点应该被分配到该子区域内。而实际情况下, 可能有一些未知节点被分配到该子区域外部, 即归属到别的子区域内, 此时称这种分配为不好的分配。对于这些不好的分配, 给出一个评分机制来衡量分配的结果。如果未知节点被分配到理想情况下所期望分配到的子区域内, 则该未知节点得分为 0; 如果被分配到有两个公共锚节点的相邻子区域内, 则该未知节点得分为 -1; 如果被分配到一个公共锚节点的相邻子区域内, 则得分为 -1.5。以此类推。如图 6, 节点 S 被错分到其他子区域时的得分如图中箭头上的数字所示。设网络中共有 n 个未知节点, 未知节点 i 的分簇得分为 $mark_i$, 则将所有节点的平均分簇得分定义为

$$meanMark = \frac{1}{n} \sum_{i=1}^n mark_i$$

我们就以这个所有节点的平均分簇得分的值作为评价整个网络分簇结果好坏的标准。

本实验在 $12R \times 12R$ (R 为传感器节点的通信半径) 的感知区域内随机产生均匀分布的 1000 个传感器节点, 锚节点数为 16。实验 1 中, 将 16 个锚节点按网格均匀的安置在整个感知区域内, 共将感知区域划分成 9 个子区域, 将其分别标示为 1, 2, ..., 9 号子区域, 也作为分簇后的簇号。仿真实验用各个簇的簇号来标记分簇后被分配到该簇的每个节点, 一次实验的分簇结果如图 7 所示。随机生成 10 个这样的网络, 分别对每个网络进行分簇, 10 次实验得到的平均 $meanMark$ 为 -0.268。

实验 2 中, 将 16 个锚节点随机均匀分布在整个网络, 根据其位置信息用这 16 个锚节点将网络划分成 9 个子区域, 然后对未知节点进行分簇。一次实验的分簇结果如图 8 所示。同样随机生成 10 个这样的网络, 分别对每个网络进行分簇, 10 次实验得到的平均 $meanMark$ 为 -0.352。

从实验 1, 2 中可以看出该分簇算法可以达到较好的分簇

(下转第 59 页)

除 IPTD 外均兼顾了出入测量时钟不精确同步的情形,可供不同代价的 NSLA 监测参考。下一步工作是以计算简便快速、冲突低、节约空间和重要字段可恢复(可选)为目标,研究应选择 IPv6 分组的哪些不变部分及使用何种算法来生成分组识别标记;研究各种应用特征实现 Web, FTP, BT 等常用应用流量的快速识别。

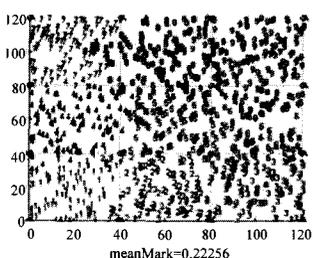
参考文献

- [1] ITU-T Study Group 2. Framework of a service level agreement (ITU-T E. 860) [EB/OL]. <http://www.itu.org>. 2002-06-29
- [2] GB 917. SLA management handbook (public evaluation/version 1. 5) [EB/OL]. <http://www.tnforum.org>. 2001-06-01
- [3] TMF 701. Performance reporting concepts & definitions (public version 2. 0) [EB/OL]. <http://www.tnforum.org>. 2001-11-01
- [4] 成卫青, 龚俭, 葛立青, 等. 端到端 SLA 性能测量与分析[J]. 东南大学学报(自然科学版), 2004, 34(5): 589-593
- [5] ITU-T Y. 1540. Internet protocol data communication service - IP packet transfer and availability performance parameters[EB/OL]. <http://www.itu.org>. 2002-12-14
- [6] ITU-T Y. 1541. Network performance objectives for IP-based services[EB/OL]. <http://www.itu.org>. May 2002
- [7] IETF RFC2330. Framework for IP performance metrics[EB/OL]. <http://www.ietf.org/>. 1998-05-05
- [8] IETF RFC2679. A One-way Delay Metric for IPPM[S]. Sep. 1999
- [9] IETF RFC2680. A One-way Packet Loss Metric for IPPM[S]. Sep. 1999

- [10] IETF RFC2681. A Round-trip Delay Metric for IPPM[S]. Sep. 1999
- [11] IETF RFC3148. A Framework for Defining Empirical Bulk Transfer Capacity Metrics [EB/OL]. <http://www.ietf.org/>. 2001-07-01
- [12] IETF RFC3393. IP Packet Delay Variation Metric for IP Performance Metrics[S]. Nov. 2002
- [13] 何宝宏. IP 网络性能参数与指标要求[J]. 电信工程技术与标准化, 2001 (5): 26-29
- [14] IETF RFC2460. IPv6 Specification[S]. Dec. 1998
- [15] 葛立青. 分布式 SLA 监测系统后处理器的研究与实现[D]. 硕士学位论文. 南京: 东南大学计算机科学与技术系, 2004
- [16] IETF RFC2401. Security Architecture for the Internet Protocol [S]. Nov. 1998
- [17] IETF RFC 3729. Application Performance Measurement MIB [S]. March 2004
- [18] ITU-T F. 700. Framework Recommendation for multimedia services[EB/OL]. <http://www.itu.org>. Nov. 2000
- [19] ITU-T G. 1010. End-user multimedia QoS categories[EB/OL]. <http://www.itu.org>. Nov. 2001
- [20] Cisco M A R. Network-based Application Recognition and Distributed Network-Based Application Recognition[EB/OL]. http://www.cisco.com/en/US/products/ps6616/products_ios_protocol_group_home.html. 2005-8-26
- [21] Choi T S, Kim C H, Yoon S H, et al. Content-aware Internet application traffic measurement and analysis[C]// 2004 IEEE/IFIP Network Operations and Management Symposium (NOMS 2004). 2004:511-524

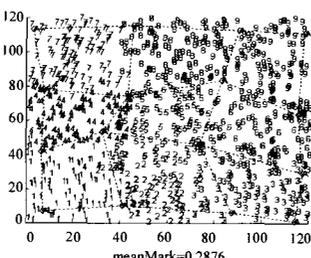
(上接第 31 页)

效果,将未知节点划分到以锚节点为边界的各个子区域中。从图 7,8 中可以看出,子区域边界(图中虚线所示)附近的一些节点可能没有完全准确地被划定到理想情况下应该所在的子区域,而被划分到了边界另一侧的子区域,这是由于节点周围的连通性质导致的。因为分簇后每个簇的主节点已经了解了本簇内节点的全部拓扑信息,因此这样微小的偏移并不会影响簇内运用定位算法进行节点定位。



$R=10$, 节点数为 1000,
16 个锚节点按网格分布,
位于图中虚线的交点处,
 $meanMark=0.22256$ 。

图 7 实验 1 分簇结果图



$R=10$, 节点数为 1000,
16 个锚节点随机均匀分布,
位于图中虚线的交点处,
 $meanMark=0.2876$ 。

图 8 实验 2 分簇结果图

结束语 本文针对一些 WSN 定位问题的需求,提出一种有助于解决大规模无线传感器网络定位问题的分簇算法。算法通过拓扑发现过程和锚节点间的信息交换实现锚节点对其周围网络拓扑的感知,根据就近原则将所有未知节点分配到以锚节点为边缘的各个子区域中,从而将分配到各个子区域中的未知节点和边缘的锚节点一起构成网络中的各个簇。同时,每个簇的主节点保存了本簇内的全部拓扑信息,为下一步的定位做好了充分的准备。该算法可以实现多跳节点的分

簇,并且具有较小的通信量,可用于多种基于分簇的分布式定位算法中,有助于解决大规模无线传感器网络的定位问题。例如,可以用来实现分布式的凸规划定位算法。特别是在锚节点数量较少的情况下,更能体现出该算法的优势,能够有效地完成对网络的分簇。同时,该分簇算法也可以用于无线传感器网络的其他领域,例如分簇路由等。本文在锚节点按网格分布和随机分布两种情况下分别进行了仿真实验,实验结果证明了该算法可以得到较好的分簇结果。

参考文献

- [1] Simic S N, Sastry S. UCB/ ERL M02 / 26, Distributed localization in wireless ad hoc networks [R]. UC Berkeley, 2002
- [2] Nicolescu D, Nath B. Ad -Hoc Positioning Systems (APS) [A] // Proceedings of 2001 IEEE Global Telecommunications Conference (IEEE GLOBECOM'01) [C]. San Antonio, TX, USA IEEE Communications Society, 2001, 5: 2926-2931
- [3] Nicolescu D, Nath B. DV based positioning in ad hoc networks [J]. Journal of Telecommunication Systems, 2003, 22 (1/4): 267-280
- [4] Savarese C, Rabay J, Langendoen K. Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks[A]. Ellis C S, ed // Proceedings of the USENIX Technical Annual Conference [C]. Monterey, CA, USA USENIX Press, 2002: 317 -327
- [5] Iyengar R, Sikdar B. Scalable and distributed GPS free positioning for sensor networks // Proc. of IEEE Int'l Conf. on Communications 2003. Vol. 1, Anchorage: IEEE Communications Society, 2003: 338-342
- [6] Biswas P, Ye Y. A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization [R]. Technical report. Dept of Management Science and Engineering, Stanford University, October 2003
- [7] Doherty L, Pister K S, Ghaoui L E. Convex position estimation in wireless sensor networks [A] // Proc. of the IEEE INFOCOM 2001 [C]. Vol 3, Anchorage: IEEE Computer and Communications Societies, 2001: 1655-1663