基于贪婪思想的二阶段无线传感器网络定位算法

孟颖辉,陈 剑,闻英友,赵 宏
(东北大学信息科学与工程学院,辽宁沈阳 110819)

摘 要: 近些年来,将优化算法应用到节点定位问题当中成为了一个研究热点.本文假设下一次定位结果为准确坐标,对前后两次定位结果邻居节点之间距离关系进行深度分析和推导,得到一个邻域函数.在此基础上根据贪婪思想,提出了贪婪定位算法.为了达到更精确的定位结果,本文将贪婪定位算法分成两个阶段:第一阶段,根据贪婪迭代优化得到一组初始定位结果;第二阶段将满足一定条件的未知节点升级为锚节点,重新执行第一阶段的过程,重复第二阶段,直到没有未知节点可以升级为锚节点为止.实验结果表明,无论是定位精确度还是算法执行时间,本文所提算法都比当前的一些优化定位算法要好.

关键词: 节点定位;优化算法;邻域函数;贪婪思想;迭代优化
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2014)02-0328-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2014.02.018

Two-Stage Localization Algorithm Based on Greedy Idea for Wireless Sensor Networks

MENG Ying-hui, CHEN Jian, WEN Ying-you, ZHAO Hong
(*School of Information Science and Engineering, Northeastern University, Shenyang, Liaoning* 110819, *China*)

Abstract: Using optimization algorithm to solve the node localization problem has become a research focus. This paper makes deep analysis on the distance relationship between two successive localization results and designs a neighborhood function, and then proposes the greedy localization algorithm based on greedy idea. The proposed algorithm is divided into two phases. In the first phase, a set of estimated positions is generated based on the greedy iterative optimization. In the second phase, some unknown nodes will be elevated to anchor nodes, and the first phase is executed again. The second phase is repeated until there is no node that can be elevated to an anchor node. Finally, the experimental results show the proposed algorithm achieves more accurate result and take less time than the existing optimization localization algorithms.

Key words: node localization; optimization algorithm; neighborhood function; greedy idea; iterative optimization

1 引言

无线传感器网络(WSN)是传感器节点以自组网形式构建的无线网络,主要用来对某个区域进行监控[1].由于 WSN 一般部署在一个大的或者是环境比较恶劣的区域,节点的具体位置往往是不可控制的.对于大多数 WSN 的应用来说,如果节点采集和传送的数据不包含具体的位置,那么这个信息将变得毫无意义[2~4].另外传感器节点位置信息还可以提高路由效率,提供网络的覆盖质量,实现网络的负载均衡以及网络拓扑的自配置[5,6].节点通常有两种方法获得自身的位置:一种是装配 GPS 模块,这种方法代价和耗能高不适合 WSN;另外一种是人工配置,这种方法不适合大型 WSN,并且在某些环境下根本无法实现[7,8].综上所述,节点定位技术成为了 WSN 研究的关键技术之一.

到目前为止,学术界提出了很多节点定位算法,这些算法都有一个共同假设就是:假设一部分节点知道自己位置,这些节点称为锚节点(anchor node),剩余的节点称为未知节点(unknown node).根据定位技术可以将定位算法分成:测距相关(range-based)和测距无关(range-free)[9].测距相关的定位算法使用 AOA、TOA、TDOA、RSSI 等技术获得两个相邻节点之间的测量距离[10,11].

收稿日期:2012-12-19;修回日期:2013-03-19;责任编辑:孙瑶
基金项目:国家自然科学基金(No.60903159,No.61173153);中央高校基本科研业务费(No.110818001,No.100218001,No.110404014,No.110318001);沈阳市科技计划项目(No.1091176-1-00);中国博士后科学基金(No.20110491508,No.2012T50248)

而测距无关的定位算法是利用节点之间的连通度等信息进行节点位置的估计.根据算法执行的地点可以分成两类:分布式定位算法和集中式定位算法[12].分布式定位算法是指每个节点自身执行定位算法.集中式定位算法是指节点将定位所需要的信息传送到一个处理能力强大的计算机,在那里进行所有节点的定位计算.

本文用组合优化的方法对定位问题进行分析.假设下一组解为节点的准确坐标,对这组解所产生的邻居节点之间的距离关系和前一组解所产生的邻居节点之间的距离关系进行分析和推导,最终得到本文的邻域函数.由于邻域函数是最优解成立的必要条件,因此需要目标函数来判断是否接受下一次的解.根据设计的邻域函数和目标函数,基于贪婪思想提出了贪婪定位算法.为了达到更好的定位精度,本文将贪婪定位算法分成两个阶段,最后实验结果证明了算法的有效性.

2　相关工作

本文重点关注测距相关的集中式定位算法.文献[13,14]中将多维定标数据分析技术应用到定位算法当中,提出了 MDS-MAP 定位算法.它首先根据节点之间的最短距离和 MDS 技术生成无线传感器网络节点的相对坐标,再根据锚节点的真实坐标和相对坐标之间的转化将相对坐标系统转换成绝对坐标系统.MDS-MAP 在网络中节点密度减小时,定位误差增大.近些年来,将一些优化策略应用到定位算法当中成为了一个研究热点.文献[15,16]中将模拟退火优化算法应用到定位算法当中,提出了 SA 定位算法.首先随机产生一组初始解,然后在一次循环当中对某一个节点随机产生一次抖动(SA 定位算法的抖动是指节点在一个随机方向上移动一段距离),对于是否接受抖动后的节点新坐标,通过目标函数来判断新估计坐标的优劣性.如果新估计坐标与前一组相比没有优势,再通过概率计算判断是否接受当前新估计坐标,重复这一过程直到达到结束条件.模拟退火定位算法最大的缺点是计算时间较长,经过测试在循环次数不够的情况下,定位误差很大.文献[17]中对模拟退火定位算法进行了改进,在执行 SA 定位算法之前加入了三边测量计算,然而在锚节点比较少的时候,很可能不会出现未知节点周围有大于等于 3 个邻居锚节点的情况.文献[18]将多目标进化策略应用到了定位算法当中,设计了两个目标函数.用两种变异进化操作来对节点的位置进行不断的循环矫正,从而提出了 PAES 定位算法,这种算法的缺点同 SA 定位算法一样,具有计算复杂度高的缺点,而且文献中的测试结果也显示出定位结果不够准确.

从上面的描述可以看到,定位问题可以用优化算法进行解决.然而智能优化算法虽然在理论上能够得到全局最优解,可缺点就是搜索效率低.本文针对这一问题,对邻居节点之间的距离关系进行了深度分析,设计了一个合理的邻域函数,并在此基础上提出一种基于贪婪思想的二阶段定位算法,简称 GIL(Greedy Idea Localization).为了测试算法的有效性,在同等条件下,本文将贪婪定位算法和同样是测距相关的集中式定位算法进行了性能比较,测试结果显示,GIL 算法无论是在定位精度还是执行时间上都有优势.

3　贪婪定位算法

用组合优化方法来解决定位问题,需要对定位问题进行分析并设计目标函数和邻域函数.本文的 GIL 算法是一种测距相关的集中式定位算法,而且只考虑二维平面上的定位问题.

3.1　定位问题分析

假设无线传感器网络有 m 个锚节点,坐标为 (x_k, y_k),$(k=1,\cdots,m)$;有 n 个未知节点,坐标为 (x_i, y_i),$(i=1,\cdots,n)$;da_{ik} 表示未知节点 i 和锚节点 k 之间的实际测量距离;du_{ij} 表示未知节点 i 和 j 之间的测量距离.节点之间的距离是通过欧氏距离计算的.节点之间的邻居关系是容易得到的,不失一般性,本文用一个数学模型来计算节点之间的邻居关系:$d_{ij} \leqslant R$,d_{ij} 表示任何两个节点之间的距离,R 是节点的通信距离.如果两个节点之间的距离满足上述条件,互为邻居节点.na_{ik} 表示未知节点 i 和锚节点 k 之间的邻居关系;nu_{ij} 表示未知节点 i 和 j 之间的邻居关系.这里需要说明的是,如果两个节点是邻居关系,对应的邻居关系值是 1,否则是 0.

定位算法的目标就是在节点所部署的区域找到一组未知节点的估计坐标,这个估计坐标越接近实际坐标越好.首先需要设计一个目标函数,已知的信息只有节点之间的邻居关系和节点之间的实际距离,如果一个未知节点的估计坐标到它所有邻居节点之间估计距离和真实距离的差值的平方越小,我们认为这个节点定位越准确,设计的目标函数如下.

$$CF = \sum_{i=1}^{n}\sum_{k=1}^{m} na_{ik}\cdot(da_{ik} - \overline{da}_{ik})^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} nu_{ij}\cdot(du_{ij} - \overline{du}_{ij})^2 \tag{1}$$

$\overline{da}_{ik}$ 和 $\overline{du}_{ij}$ 分别表示未知节点的估计坐标之间的距离.一组估计坐标对应的 CF 值越小,我们就认为这组估计坐标越接近实际位置.

3.2　邻域函数设计

本节介绍邻域函数的推导过程.假设 $\overline{du}_{ij}^{t}$ 和 $\overline{du}_{ij}^{t+1}$ 分别表示第 t 和 $t+1$ 次优化后得到的未知节点之间的估计距离.已知两个节点之间的真实距离 du_{ij},假设第 $t+1$ 次优化后得到的是节点 i 和 j 的估计坐标是精确的,那么它们之间必定满足如下关系:

$$(\overline{du}_{ij}^{t+1})^2 = (\bar{x}_i^{t+1} - \bar{x}_j^{t+1})^2 + (\bar{y}_i^{t+1} - \bar{y}_j^{t+1})^2 = du_{ij}^2$$
$$= \left[\frac{du_{ij}}{\overline{du}_{ij}^t}\cdot(\bar{x}_i^t - \bar{x}_j^t)\right]^2 + \left[\frac{du_{ij}}{\overline{du}_{ij}^t}\cdot(\bar{y}_i^t - \bar{y}_j^t)\right]^2 \quad (2)$$

式(2)中的坐标分别表示第 t 和 $t+1$ 次优化后为未知节的估计坐标.对式(2)进行变化可以得到如下的坐标关系:

$$\bar{x}_i^{t+1} - \bar{x}_j^{t+1} = \frac{du_{ij}}{\overline{du}_{ij}^t}\cdot(\bar{x}_i^t - \bar{x}_j^t)$$
$$\bar{y}_i^{t+1} - \bar{y}_j^{t+1} = \frac{du_{ij}}{\overline{du}_{ij}^t}\cdot(\bar{y}_i^t - \bar{y}_j^t) \quad (3)$$

式(3)是式(2)成立的充分条件.如果未知节点 i 和它所有的邻居节点之间都满足式(2)对于本文来说,我们就认为这个位置是准确的估计坐标,这里所说的邻居节点包含邻居未知节点和锚节点.当未知节点 i 和它所有的邻居节点之间都满足式(3)时,一定满足如下坐标关系:

$$\sum_{j=1}^{n} nu_{ij}\cdot(\bar{x}_i^{t+1} - \bar{x}_j^{t+1}) + \sum_{k=1}^{m} na_{ik}\cdot(\bar{x}_i^{t+1} - x_k)$$
$$= \sum_{j=1}^{n} nu_{ij}\cdot\frac{du_{ij}}{\overline{du}_{ij}^t}\cdot(\bar{x}_i^t - \bar{x}_j^t) + \sum_{k=1}^{m} na_{ik}\cdot\frac{da_{ik}}{\overline{da}_{ik}^t}\cdot(\bar{x}_i^t - x_k);$$
$$\sum_{j=1}^{n} nu_{ij}\cdot(\bar{y}_i^{t+1} - \bar{y}_j^{t+1}) + \sum_{k=1}^{m} na_{ij}\cdot(\bar{y}_i^{t+1} - y_k)$$
$$= \sum_{j=1}^{n} nu_{ij}\cdot\frac{du_{ij}}{\overline{du}_{ij}^t}\cdot(\bar{y}_i^t - \bar{y}_j^t) + \sum_{k=1}^{m} na_{ik}\cdot\frac{da_{ik}}{\overline{da}_{ik}^t}\cdot(\bar{y}_i^t - y_k) \quad (4)$$

式(4)中 $\overline{da}_{ik}^t$ 表示未知节点 i 和它邻居锚节点 k 之间的第 t 次优化后估计坐标之间的估计距离.从式(3)和式(4)的描述可以看到,式(3)是式(4)的充分条件.满足式(4)却不一定满足式(3),所以需要用目标函数进行判断.式(4)对于所有的未知节点都成立,可以得到第 t 和 $t+1$ 次所有未知节点到所有邻居节点横纵坐标之间关系,用矩阵表示如下:

$$(\boldsymbol{N}_1 + \boldsymbol{N}_2)\cdot\bar{\boldsymbol{U}}^{t+1} - \boldsymbol{NA}\cdot\boldsymbol{A} = (\boldsymbol{D}_1 + \boldsymbol{D}_2)\cdot\bar{\boldsymbol{U}}^t - \boldsymbol{D}_3\cdot\boldsymbol{A} \quad (5)$$

$$n_{1_{ij}} = \begin{cases} \sum_{j=1}^{n} nu_{ij}, & i = j \\ -nu_{ij}, & i \neq j \end{cases}$$

$$n_{2_{ij}} = \begin{cases} \sum_{k=1}^{m} na_{ij}, & i = j \\ 0, & i \neq j \end{cases}$$

$$d_{1_{ij}} = \begin{cases} \sum_{j=1}^{n} nu_{ij}\cdot\frac{du_{ij}}{\overline{du}_{ij}^t}, & i = j, \overline{du}_{ij}^t \neq 0 \\ -nu_{ij}\cdot\frac{du_{ij}}{\overline{du}_{ij}^t}, & i \neq j, \overline{du}_{ij}^t \neq 0 \\ 0, & \overline{du}_{ij}^t = 0 \end{cases}$$

$$d_{2_{ij}} = \begin{cases} \sum_{k=1}^{m} na_{ik}\cdot\frac{da_{ik}}{\overline{da}_{ik}^t}, & i = j, \overline{da}_{ik}^t \neq 0 \\ 0, & i \neq j \text{ or } \overline{da}_{ik}^t = 0 \end{cases}$$

$$d_{3_{ik}} = \begin{cases} na_{ik}\cdot\frac{da_{ik}}{\overline{da}_{ik}^t}, & \overline{da}_{ik}^t \neq 0 \\ 0, & \overline{da}_{ik}^t = 0 \end{cases} \quad (6)$$

式(5)中的 $\bar{\boldsymbol{U}}^t(n\times 2)$ 和 $\bar{\boldsymbol{U}}^{t+1}(n\times 2)$ 表示第 t 和 $t+1$ 次优化后的估计坐标矩阵;$\boldsymbol{A}(n\times 2)$ 表示锚节点的坐标矩阵;$\boldsymbol{NA}(n\times m)$ 表示未知节点和锚节点之间的准确邻居关系矩阵;矩阵 $\boldsymbol{N}_1$、$\boldsymbol{N}_2$、$\boldsymbol{D}_1$、$\boldsymbol{D}_2$ 和 $\boldsymbol{D}_3$ 的计算方法如式(6)所示.对式(5)进行移动变化,最终可以得到本文的邻域函数如下:

$$\bar{\boldsymbol{U}}^{t+1} = inv(\boldsymbol{N}_1 + \boldsymbol{N}_2)\cdot[(\boldsymbol{D}_1 + \boldsymbol{D}_2)\cdot\bar{\boldsymbol{U}}^t + (\boldsymbol{NA} - \boldsymbol{D}_3)\cdot\boldsymbol{A}] \quad (7)$$

给一组未知节点的坐标,可以通过式(7)得到另外一组估计坐标.正是因为前面描述的满足式(4)并不一定满足式(3),由式(7)所得到的不一定是最优解,因此需要进行判断.对于本文来说我们判断两组解的优劣就是根据式(1)所表示的目标函数,目标函数值越小,我们就认为定位越准确.如果 $\bar{\boldsymbol{U}}^{t+1}$ 计算得到的目标函数值比 $\bar{\boldsymbol{U}}^t$ 得来的目标函数值小,我们就接受当前解,否则不接受.

3.3 贪婪定位算法描述

本节给出 GIL 算法的具体流程.算法分为两个阶段:第一个阶段称为初始定位;第二阶段称为精确定位,根据第一阶段未知节点的初始估计位置,将那些满足一定要求的未知节点升级为锚节点,重新进行第一阶段的计算,然后重复这一过程,直到没有未知节点能够升级为锚节点为止.图 1 是 GIL 算法的整体流程图.

3.3.1 初始定位阶段

初始定位阶段是根据前面所设计的目标函数和邻域函数,从一组随机产生的初始解出发,经过迭代计算出一组解作为未知节点的初始估计位置,实现步骤如算法 1 所示.

算法 1 GIL 算法初始定位阶段

①随即生成初始解:$\bar{\boldsymbol{U}}^0$;

②用 comCF(…)计算当前解 CF_0 的值.设置几个参数:$k=0$;$\bar{\boldsymbol{U}}^1 = \bar{\boldsymbol{U}}^0$;$CF_1 = CF_0$;

③判断 $k=0 \parallel (CF_0 - CF_1 > CF_{Error}$ && $k \leqslant num)$,如果满足转④;如果不满足,转⑥;

④设置一些参数:$k = k+1$;$\bar{\boldsymbol{U}}^0 = \bar{\boldsymbol{U}}^1$;$CF_0 = CF_1$;

⑤根据邻域函数:$\bar{\boldsymbol{U}}^1$ = optimalX(…)计算下一组解;comCF(…)计算 CF_1,转③;

输出 $\bar{\boldsymbol{U}}^1$.

其中函数 optimalX(…)是根据邻域函数由一组估计坐

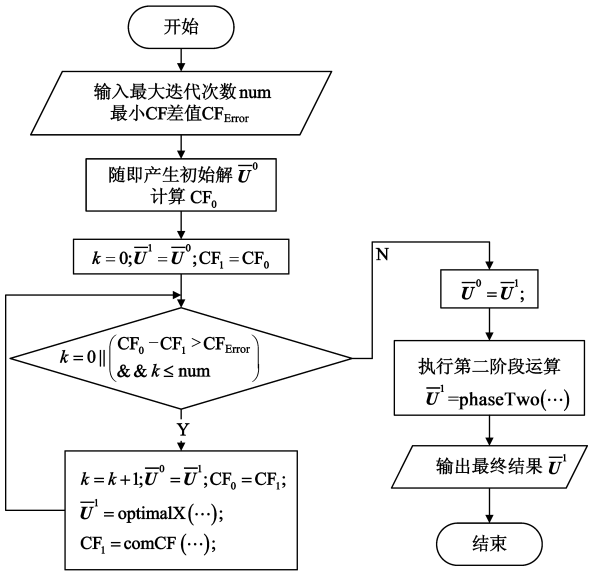

图1　贪婪算法流程图

标获得另外一组估计坐标;comCF(…)计算当前解的 CF 值;CF_{Error}是前后两次目标函数的最小差值;num 是循环的最大次数.在第三步中,我们可以看到算法只接受优于当前解的新解,设置的 CF_{Error}越小定位结果越准确.从上述算法描述可以看到,所有的解都是从初始解通过邻域函数计算得来的,由此可见,初始解很大程度上影响了最优解的精度.本文的初始解的产生分为两种情况:一种是所有未知节点从同一个随机坐标位置出发;另外一种是所有未知节点从不同的随机位置出发,两种初始解对 GIL 算法定位结果的影响会在测试阶段进行对比.

3.3.2　精确定位阶段

精确定位阶段利用初始阶段的定位结果进行未知节点定位精度的优化.依据邻域函数的设计过程可以看到,一个未知节点的估计位置本质上是根据它到所有邻居节点的距离计算得来的,这就容易出现翻转现象.

图 2 是翻转现象的图形表示.假设未知节点 A 是由邻居节点 B、C 和 D 进行定位的.从图中可以看到 B、C 和 D 近似的在一条直线上,这种现象可能是由于网络部署后就在一条直线上,也可能是由于邻居节点的定位误差造成的.但是未知节点 A 不知道,它只会根据它到 B、C 和 D 节点的距离进行自己的位置估计,这时就很可能定位到 A′,因为 A′到 B、C 和 D 的距离和 A 是一样的,所以 CF 的值也是不变的,从而导致错误的定位,这就是翻转现象.判断一个节点是否发生翻转现象对于 WSN 来说并不困难,因为

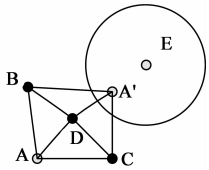

图2　翻转现象

WSN 的节点一般部署比较密集.如果一个未知节点的定位结果发生了翻转现象,那么就很可能会影响它的邻居关系.图 2 中,未知节点 A 由于翻转现象被误定位到 A′,而此时 A′就会成为节点 E 的邻居节点,而实际上 E 和 A 不是邻居节点.因此在初始定位阶段结束后,对定位结果进行分析.根据定位结果中未知节点的估计坐标求得节点之间的估计邻居关系,这里的邻居关系有可能是错误的.如果此时某个未知节点的邻居关系和真实邻居关系完全一样,就认为这个未知节点是定位准确的,否则认为这个未知节点还需要进行定位,具体步骤如算法 2 所示.

算法 2　GIL 算法精确定位阶段

①执行算法 1 得到一组估计坐标;

②用估计坐标求得一组新的邻居关系,如果某个未知节点的邻居关系和实际邻居关系完全相同,我们就认为这个未知节点被精确定位,升级它为锚节点,将它的估计坐标作为准确坐标;而那些邻居关系和实际邻居关系不相同的未知节点,则进行下一次的计算.如果此次没有一个未知节点能够升级为锚节点,转④,否则转③;

③步骤②所升级的锚节点和以前的锚节点组成新的锚节点矩阵,而那些认为没有被精确定位的未知节点组成新的未知节点矩阵,转①;

④输出最终未知节点的估计坐标,此时即使那些认为没有精确定位的未知节点,也将它们此次循环的估计坐标返回.

4　实验结果与分析

为了测试 GIL 算法的性能,用 MATLAB 进行了模拟测试.由于本文是为了测试 GIL 算法的有效性,在测试过程当中,假设节点之间的真实距离是准确的.本机的配置为:Intel Core 2 Duo CPU E7200 2.53 Ghz,2.0GB 内存.仿真环境是将 100 个未知节点随即部署在一个 100m × 100m 正方形区域内,形成一个 WSN,然后改变一些参数来测试这些参数对 GIL 算法的影响.最后通过和其余几种测距相关的集中式定位算法进行性能比对,来证明 GIL 算法的优势.WSN 定位算法的评价标准通常采用的计算方法如下:

$$\text{error}=\frac{1}{nR}\sum_{i=1}^{n}\sqrt{(x_i-\bar{x}_i)^2+(y_i-\bar{y}_i)^2}\times 100\% \quad (8)$$

其中 n 是未知节点个数,是 R 节点通信半径,(x_i,y_i)是未知节点真实坐标,$(\bar{x}_i,\bar{y}_i)$未知节点的估计坐标.

4.1　初始解对算法的影响

本节测试不同初始解对 GIL 算法的影响.初始解的生成分两种情况:一种是所有节点都从同一随机位置出发;另外一种是所有未知节点从不同的随机位置出发.随机生成一组 WSN,图 3 是两种初始解定位结果.每种类型的初始解测试了 5 组,从图 3 中可以看到从同一随机位置出发所产生的定位结果都要远远好于从不同位置出发所产生的定位结果.同时也可以看到从同一位置

出发的定位结果也不是固定不变的,这是因为算法本身是局部优化算法,最优解和初始解有直接关系.因此我们可以多测几组从同一位置出发的初始解,取目标函数值最小的那个定位结果作为最终的位置估计.

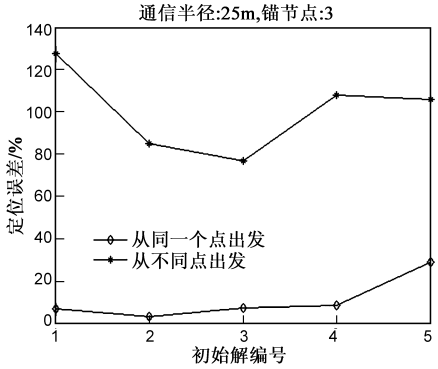

图3 不同初始解对定位误差的影响

4.2 锚节点位置对算法的影响

本节测试锚节点位置对 GIL 算法的影响.这里锚节点的数目选择 3 到 5 个,每一种情况都随机产生 5 组不同锚节点的位置.随即产生一组 WSN,定位结果如图 4 所示.

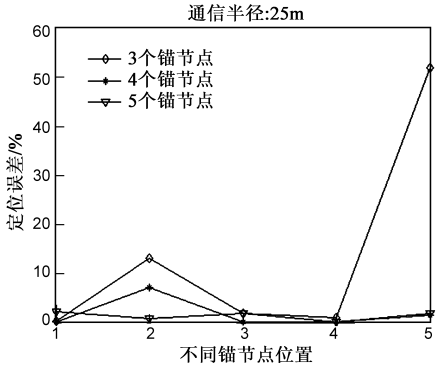

图4 不同锚节点位置对定位误差的影响

从图 4 中可以看到,锚节点为 3 个的时候,不同的位置对定位结果的影响比较大.这是因为 3 个锚节点分布在一条直线或者近似一条直线的概率比较大,所以定位误差会有较大起伏;4 个锚节点的时候起伏就明显减少,到 5 个的时候基本稳定.同时从图 3 中也可以看到,锚节点分布合理的时候,3 个锚节点就可以达到很好的定位效果,这说明锚节点的位置对 GIL 算法的影响主要体现在锚节点数量比较少的情况.对于锚节点较少的情况,如果能够将锚节点放置到边缘,比如将这 3 个锚节点放置到正方形区域的任意 3 个角落;如果是 4 个锚节点,就放置到 4 个角落的位置,定位结果如图 5 所示.这里采用了 10 组随即产生的 WSNs,从图 5 中可以看到,如果将锚节点放置到区域的角落位置,只需要 3 个或者 4 个锚节点,定位结果就非常理想.虽然定位结果还有一点起伏,但是定位精度非常高,都在 2.5% 以内.

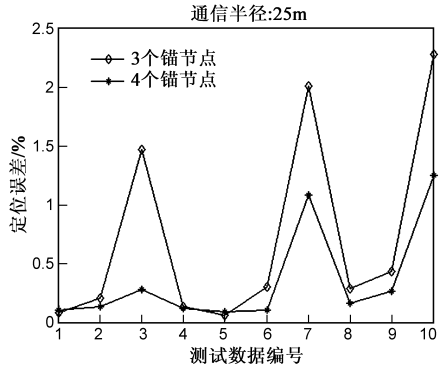

图5 锚节点放在角落对定位误差的影响

4.3 通信半径对算法的影响

本节测试通信半径对 GIL 算法的影响.随机产生一组 WSN,并用 3 到 5 个锚节点来测试通信半径对 GIL 算法的定位结果的影响,图 6 是测试结果图.

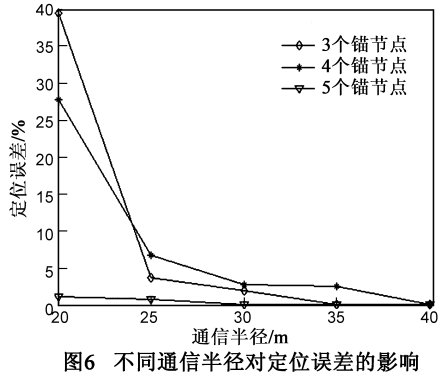

图6 不同通信半径对定位误差的影响

从图 6 中可以看到,不论锚节点为几个,定位精度都是随着通信半径的增大而减小.这是因为 GIL 算法利用的就是节点到邻居节点之间的距离约束关系进行计算的,而通信半径越大,节点到邻居节点之间的距离约束关系就越多,定位准度就越高.

4.4 算法对比

本节主要测试 GIL 在同等条件下同 SA、MDS-MAP 和 PAES 定位算法的性能对比.表 1 和表 2 分别为 SA 和 PAES 算法的参数值.本文的 GIL 算法的两个参数 num 和 CF_{Error}分别为 1000 和 0.001,算法运行 5 次,取 CF 值最小的解.

表 1 SA 算法参数值

T	$\min T$	Δd	num	a	b	p	q
20	$10e-10$	25	40.000	0.8	0.9	4	4

表 2 PAES 算法参数值

size	regions	number offitness	P_m	P_n
20	5	100.000	0.9	0.3

这里测试的锚节点个数从 3 到 10.为了测试算法的稳定性,图 7 给出 GIL、SA、MDS-MAP 和 PAES 定位算法在十组随机产生的 WSNs 当中,不同锚节点的性能对比,图 8 是算法的执行时间.这里的定位误差和执行时间取的是十组 WSNs 定位误差和执行时间的平均值.

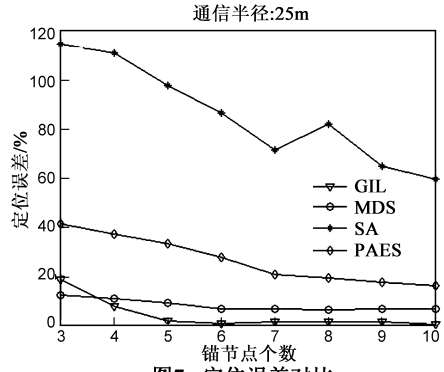

图7　定位误差对比

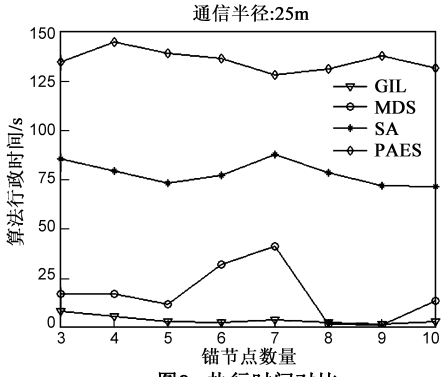

图8　执行时间对比

从图 7 中可以看到,在同等条件下,SA 定位算法的定位误差最大,这里很大的原因是参数的设置问题,参数的设置决定了循环的次数和搜索的空间.模拟退火属于智能优化算法,这种算法的特点是全局搜索最优解,理论上来说能够找到全局最优解,但缺点是计算时间太长.从图 8 中可以看到,即使达到目前的这种定位误差,时间已经很长.从图 7 和图 8 中可以看到,同为组合优化的贪婪思想定位算法,无论是定位误差还是定位执行时间都要大大减少.这是因为贪婪定位算法属于局部搜索,如果邻域函数和初始解设计合理就能快速收敛,虽然最后得到的不一定是最优解,但是对于无线传感器网络来说只要定位精度达到要求就可以.从图 7 中可以看到,GIL 定位算法和 MDS-MAP 定位算法相比,只有在 3 个锚节点的时候,GIL 定位算法会稍差于 MDS-MAP 算法,其余都要优于 MDS-MAP 算法,而执行时间,从图 8 中可以看到 GIL 算法基本都占有优势.在 3 个锚节点时 GIL 定位算法之所以不准是由于锚节点的分布集中或者近似在一条直线上造成的.

5　总结

本文对优化算法在无线传感器网络定位算法当中的应用进行了研究,用组合优化的方法对定位问题进行了分析.假设下一次定位计算得到的是未知节点准确坐标的前提下,依据邻居节点之间的距离关系,分析了前后两组估计坐标之间的关系,最终得到了一个邻域函数,这个邻域函数是产生最优解的必要条件,用目标函数来判断两组解的优劣.再根据贪婪思想只接受比前一组解优的新解,进行迭代循环,可以得到一组未知节点的初始估计坐标.之后根据节点之间的邻居关系,将一些满足条件的未知节点升级为锚节点,并将估计坐标作为最终的定位坐标.将新组成的锚节点和未知节点重新进行上述运算,直至没有未知节点可以升级为锚节点为止,输出最终的定位结果.最后通过实验分析 WSN 各种因素对 GIL 定位算法的影响.同时将 GIL 和 SA、MDS-MAP、PAES 定位算法在同等条件下进行了对比,测试结果显示,除了在 3 个锚节点的时候,定位误差会稍差于 MDS-MAP.其余情况无论是定位精度还是执行时间都要优于其他三种定位算法,从而证实了本文所提算法的有效性.在下一步的工作当中,我们将研究测距误差对算法的影响以及改进方法.

参考文献

[1] Akyildiz IF, Su WL, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks[J]. IEEE Communication Magazine, 2002,40(8):102 – 114.

[2] 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 FB, Shi L, Ren FY. Self-localization systems and algorithms for wireless sensornetworks[J]. Journal of Software, 2005,16(5):857 – 868.(in Chinese)

[3] 沙超,王汝传,孙力娟,黄海平.无线传感器网络中一种信标节点可迁移的协作定位方法[J].电子学报,2010,38(11):2625 – 2629.
Sha Chao, Wang Ru-chuan, Sun Li-juan, Huang Hai-jing. A coorperating localization method based on beacon transfer in wireless sensor networks[J]. Acta Electronica Sinica, 2010, 38(11):2625 – 2629.(in Chinese)

[4] Vempaty A, Ozdemir O, Agrawal K, Chen H, Varshney PK. Localization in wireless sensor networks: Byzantines and mitigation techniques[J]. IEEE Journal and Magazines, 2013, 61(6):1495 – 1508.

[5] Mao GQ, Fidan B, Anderson BDO. Wireless sensor network localization techniques[J]. Computer Networks, 2007, 51(10): 2529 – 2553.

[6] Redondi A, Chirico M, Borsani L. An integrated system based

on wireless sensor networks of patient monitoring, localization and tracking[J]. AD Hoc Networks, 2013, 11(1): 39 – 53.

[7] 李东岳,王英龙,魏诺,刘颖慧.信号强度和运动向量结合的无线传感器网络移动节点定位[J].电子学报,2010,38(2A):221 – 224.
Li Dong-yue, Wang Ying-long, Wei Nuo, Liu Ying-hui. Localization algorithm for mobile nodes in wireless sensor networks based on single strength and motion vector[J]. Acta Electronica Sinica, 2010, 38(2A): 221 – 224. (in Chinese)

[8] Mao GQ. Fidan B. Anderson BDO. Wireless sensor network localization techniques[J]. Computer Networks, 2007, 51(10): 2529 – 2553.

[9] Wu G, Wang S, Wang B, Dong Y. A novel range-free localization based on regulated neighborhood distance for wireless ad hoc and sensor networks[J]. Computer Networks, 2012, 56(16): 3581 – 3593.

[10] 戴桂兰,赵冲冲,邱岩.一种基于球面坐标的无线传感器网络三维定位机制[J].电子学报,2008,36(7):1297 – 1303.
Dai Gui-lan, Zhao Chong-chong, Qiu Yan. A localization-scheme based on sphere for wireless sensor networks in 3D [J]. Acta Electronica Sinica, 2008, 36(7): 1297 – 1303. (in Chinese)

[11] 刘志华,陈家兴,陈霄凯.无线传感器网络中序列定位新算法的研究[J].电子学报,2010,38(7):1552 – 1556.
Liu Zhi-Hua, Chen Jia-xing, Chen Xiao-kai. A new algorithm research of sequence-based localization technology in wireless sensor networks[J]. Acta Electronica Sinica, 2010, 38(7): 1552 – 1556. (in Chinese)

[12] Kim SW, Lee JM, Lee JH. Distributed network localization for wireless sensor networks[J]. Applied Mathematics & Information Sciences, 2012, 6(3): 1109 – 1115.

[13] Yi Shang, Wheeler Ruml, Ying Zhang, Markus P J Fromherz. Localization from mere connectivity[A]. Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing[C]. Annapolis: ACM, 2003. 201 – 212.

[14] Ji X, Zha HY. Sensor positioning in wireless ad hoc sensor network with multidimensional scaling[A]. Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies[C]. USA: IEEE, 2004. 2652 – 2661.

[15] Kannan AA, Mao GQ, Vucetic BS. Simulated annealing based localization in wireless sensor network[A]. Proceedings of the IEEE Conference on Local Computer Networks[C]. USA: IEEE, 2005. 512 – 514.

[16] Kannan A A, Mao G, Vucetic B. Simulated annealing based wireless sensor network localization with flip ambiguity mitigation[A]. Proceedings of the 63rd IEEE Vehicular Technology Conference[C]. USA: IEEE, 2006. 1022 – 1026.

[17] Niewiadomska-Szynkiewicz E, Marks M. Optimization schemes for wireless sensor network localization[J]. International Journal of Applied Mathematics and Computer Science, 2009, 19(2): 291 – 302.

[18] Vecchio M, Lopez VR, Marcelloni F. A two-objective evolutionary approach based on topological constraints for node localization in wireless sensor networks[J]. Applied Soft Computing, 2012, 12(7): 1891 – 1901.

作者简介

孟颖辉 男,1984 年 1 月出生,河南郑州人.现为东北大学博士研究生,主要从事无线传感器网络关键技术、网络安全等方面的研究工作.
E-mail: yinghuimeng@126.com

陈 剑 男,1980 年出生于湖南邵阳.博士,东北大学讲师,主要从事多媒体无线通信系统、网络管理及视频信号建模技术等方面的研究工作.