

# 有向传感器网络覆盖增强问题的 贪婪迭代算法

陆克中,冯禹洪,毛 睿,罗秋明,刘 刚

(1. 深圳大学计算机与软件学院,广东深圳 518060; 2. 国家高性能计算中心深圳分中心,广东深圳 518060)

**摘 要:** 在有向传感器网络中,可以通过调整节点的感知方向来增强目标区域的覆盖率.提出了有向传感器网络覆盖增强问题的一种贪婪迭代算法,在每次迭代中,调整那些使得覆盖率增加最大的节点的感知方向,重复此迭代过程直至通过调整任一节点的感知方向已不能再增加覆盖率.此外,还提出了一种通过计算几何求解该算法中区域计算问题的方法,以提高计算精度和减少计算时间.大量的仿真实验结果表明,该算法的执行时间较短,收敛速度较快,能够获得比现有算法更高的目标区域覆盖率.

**关键词:** 无线传感器网络;有向传感器节点;覆盖增强;贪婪算法;迭代算法

**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2012) 04-0688-07

**电子学报 URL:** <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2012.04.011

## A Greedy Iterative Algorithm of Coverage Enhancing Problem in Directional Sensor Networks

LU Ke-zhong, FENG Yu-hong, MAO Rui, LUO Qiu-ming, LIU Gang

(1. College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518060, China;

2. National High Performance Computing Center at Shenzhen, Shenzhen, Guangdong 518060, China)

**Abstract:** In directional sensor networks, the coverage of the target region can be enhanced by adjusting the sensing directions of nodes. A greedy iterative algorithm of coverage enhancing problem in directional sensor network is proposed. In each iteration, the sensing directions of those nodes that can most increase the coverage are adjusted. This iteration process is repeated until the coverage cannot be increased by adjusting the sensing direction of any node. Moreover, a computational geometry method for region calculations in this algorithm is proposed to improve the computational accuracy and reduce the computing time. Extensive simulation results have shown that this algorithm has fast convergence speed and short execution time. And it outperforms existing algorithms with respect to the coverage of the target region.

**Key words:** wireless sensor network; directional sensor node; coverage enhancing; greedy algorithm; iterative algorithm

### 1 引言

无线传感器网络是近年来兴起的一种新型监控网络,在野外考察、环境监测、目标跟踪、抢险救灾、国防军事等领域具有广泛的应用前景<sup>[1~3]</sup>.存在某些传感器节点,如视频传感器<sup>[4,5]</sup>、超声传感器<sup>[6]</sup>、红外传感器<sup>[7]</sup>等,由于受到感知视角的限制,其感知区域是一个扇形区域,这样的传感器节点称为有向传感器节点,由有向传感器节点组成的无线传感器网络称为有向传感器网络.

覆盖控制<sup>[8,9]</sup>是无线传感器网络中的一个基本问题,直接影响着其监控性能:目标区域被传感器节点的

感知区域覆盖的比例越大,则在目标区域内发生的事件被传感器节点监控到的概率就越大.传感器节点通常由飞机或其他工具撒落在目标区域内,其初始位置和感知方向都具有随机性,导致目标区域的初始覆盖率并不高<sup>[10]</sup>.可以通过移动节点或者调整节点的感知方向来增强目标区域的覆盖率,但可移动节点的成本较高,因此本文的主要研究内容是如何通过调整节点的感知方向来最大化目标区域的覆盖率.

有向传感器网络覆盖增强问题的计算复杂性较高,现有的大部分算法<sup>[10~13]</sup>都是近似求解.本文提出了覆盖增强问题的一种贪婪迭代算法,通过不断调整节点的

感知方向来增强覆盖率,直至每个节点的感知方向都处于局部最优状态.仿真实验结果表明在目标区域的覆盖率方面,该算法要优于现有算法.

## 2 相关工作

Ma 等<sup>[14]</sup>率先进行了有向传感器网络的研究,提出了有向传感器节点的连续方向感知模型:节点可将其感知方向转动至任意角度.Tao 等<sup>[10]</sup>提出一个基于虚拟势场的有向传感器网络覆盖增强算法.在此算法中,每个节点的扇形覆盖区域都被看作一个物体,在它们之间虚拟斥力的作用下,每个扇形都以节点为轴心进行旋转,直到最终受力达到平衡.本文所要研究的问题与之相似,但仿真实验结果表明在目标区域覆盖率方面,我们提出的算法要优于该算法.

Ai 等<sup>[11]</sup>提出了有向传感器节点的离散方向感知模型:节点只能将其感知方向转动至若干个固定角度.对于离散方向感知模型下的覆盖增强问题,Cheng 等<sup>[12]</sup>提出一种分布式概率增强的贪婪算法,按照一定的优先级顺序来设置各个节点的感知方向.但在该算法中,每个节点的感知方向最多被调整一次,因此仍然可以继续通过调整节点的感知方向来增强覆盖率.

此外,传感器网络所要覆盖的也可以是以点表示的若干个目标.对于此情况下的覆盖增强问题,Ai 等<sup>[11]</sup>提出了一种贪婪算法,每次选出一个能够覆盖最多目标的节点,并设置该节点的感知方向,直至每个节点的感知方向都被设置.Cai 等<sup>[13]</sup>提出了另一种贪婪算法,每次先选出一个被最少节点覆盖的目标,再设置能够覆盖该目标的某个节点的感知方向.在这两个算法中,同样每个节点的感知方向也最多被调整一次.

## 3 问题描述

假设有向传感器网络由  $n$  个同构的有向传感器节点组成,分别以  $r$  和  $\theta$  表示每个节点的感知距离和感知视角,分别以  $p_i$  和  $\alpha_i$  分别表示节点  $i$  的位置坐标和感知方向,  $1 \leq i \leq n$ ,以  $R$  表示所要监视的多边形目标区域.定义目标区域的覆盖率  $\mu$  为目标区域被所有节点覆盖的比例,即

$$\mu = \frac{\text{Area}\left(R \cap \bigcup_{1 \leq i \leq n} \phi(r, \theta, p_i, \alpha_i)\right)}{\text{Area}(R)} \quad (1)$$

式(1)中,  $\text{Area}$  是个计算区域面积的函数,  $\phi(r, \theta, p_i, \alpha_i)$  表示节点  $i$  的感知区域,如图 1 所示,它是一个以  $p_i$  为圆心、半径为  $r$ 、角度为  $\theta$ 、且等分线与  $X$  轴夹角为  $\alpha_i$  的扇形区域.

假定各个节点在布置后便不再移动,可通过 GPS 接收器或定位协议<sup>[15]</sup>得到自身的位置坐标,且可将自

身的感知方向转动至任意角度,即  $R, r, \theta, p_1, p_2, \dots, p_n$  是已知的常量,而所有节点的方向序列  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  是变量,  $0 \leq \alpha_i < 2\pi, 1 \leq i \leq n$ . 因此,可以通过调整各个节点的感知方向来增强目标区域的覆盖率.

**定义 1** 对于某个给定  $R, r, \theta, p_1, p_2, \dots, p_n$  的有向传感器网络,若函数  $\mu(\alpha_1, \alpha_2, \dots, \alpha_n), 0 \leq \alpha_i < 2\pi, 1 \leq i \leq n$  在  $(\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)$  处取得最大值,则称  $(\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)$  为一个最优方向序列.有向传感器网络的覆盖增强问题即是对于任一给定  $R, r, \theta, p_1, p_2, \dots, p_n$  的有向传感器网络,求解它的一个最优方向序列.

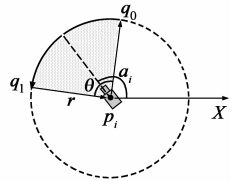


图1 有向传感器节点的感知区域

有向传感器网络覆盖增强问题的计算复杂性较高,在节点数量较大时,目前只能通过近似算法求解.

## 4 贪婪迭代算法

### 4.1 基本原理

定义节点  $i$  的私有覆盖区域为目标区域被节点  $i$  覆盖且不被其他任何节点覆盖的部分,以  $\varphi_i$  表示其面积,则有

$$\begin{aligned} \varphi_i(\alpha_1, \alpha_2, \dots, \alpha_n) &= \text{Area}(R \cap \phi(r, \theta, p_i, \alpha_i) - \bigcup_{1 \leq j \leq n, j \neq i} \phi(r, \theta, p_j, \alpha_j)) \end{aligned} \quad (2)$$

对于任意两个节点  $i$  和  $j$ ,当且仅当它们之间的距离  $\|p_i, p_j\| \leq 2r$  时,节点  $i$  和  $j$  的感知区域才有可能重叠,定义这样的两个节点互为邻居.以  $N_i$  表示节点  $i$  的邻居集合,即  $N_i = \{j | 1 \leq j \leq n, j \neq i, \|p_i, p_j\| \leq 2r\}$ . 式(2)可简化为:

$$\begin{aligned} \varphi_i(\alpha_1, \alpha_2, \dots, \alpha_n) &= \text{Area}(R \cap \phi(r, \theta, p_i, \alpha_i) - \bigcup_{j \in N_i} \phi(r, \theta, p_j, \alpha_j)) \end{aligned} \quad (3)$$

目标区域被覆盖的部分可分为两个不相交的区域:节点  $i$  的私有覆盖区域和目标区域被除  $i$  之外的其他节点覆盖的部分,  $1 \leq i \leq n$ . 因此,覆盖率也可表示为

$$\begin{aligned} \mu &= \frac{\text{Area}(R \cap \bigcup_{1 \leq j \leq n, j \neq i} \phi(r, \theta, p_j, \alpha_j)) + \varphi_i(\alpha_1, \alpha_2, \dots, \alpha_n)}{\text{Area}(R)} \quad 1 \leq i \leq n \end{aligned} \quad (4)$$

从式(4)可知,在其他节点的感知方向保持不变的前提下,可以通过调整某一节点  $i$  的感知方向  $\alpha_i$ ,使得  $\varphi_i(\alpha_1, \alpha_2, \dots, \alpha_n)$  的值增大,从而增大  $\mu(\alpha_1, \alpha_2, \dots, \alpha_n)$  的值.

**定义 2** 对于某个给定  $R, r, \theta, p_1, p_2, \dots, p_n$  的有向传感器网络,若函数  $\varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha_i, \dots, \alpha'_n), 0 \leq \alpha_i$

$< 2\pi$  在  $\alpha'_i$  处取得最大值, 则称节点  $i$  在  $(\alpha_1, \alpha_2, \dots, \alpha_n) = (\alpha'_1, \alpha'_2, \dots, \alpha'_n)$  时处于局部最优状态. 若所有节点在  $(\alpha_1, \alpha_2, \dots, \alpha_n) = (\alpha'_1, \alpha'_2, \dots, \alpha'_n)$  时都处于局部最优状态, 则称  $(\alpha'_1, \alpha'_2, \dots, \alpha'_n)$  为一个局部最优方向序列.

易知, 任何一个最优方向序列必定是一个局部最优方向序列, 反之则不然. 若  $(\alpha'_1, \alpha'_2, \dots, \alpha'_n)$  是一个局部最优方向序列, 则通过调整任一节点的感知方向已不能再增加覆盖率, 我们期望此时的覆盖率会接近于最优覆盖率.

## 4.2 算法描述

我们提出的贪婪迭代算法 (Greedy Iterative Algorithm, 简称 GIA 算法) 通过求解一个局部最优方向序列来得到覆盖增强问题的一个近似解. 在该算法的每次迭代中, 若某些节点处于非局部最优状态, 则调整这些节点的感知方向, 使它们进入到局部最优状态. 但是, 若在某次迭代中, 调整了两个互为邻居的节点的感知方向, 由于它们之间的覆盖区域可能有重叠, 反而有可能会降低覆盖率. 为了避免此情况的发生, 同时为了加快收敛速度, 采用以下贪婪策略选择每次迭代中需要调整感知方向的节点集合: 记节点的优先权为调整其感知方向至局部最优状态所增加的私有覆盖面积, 若在某次迭代中, 某节点在它的所有邻居中具有最大的优先权, 则在此次迭代中调整该节点的感知方向. 若两个节点的优先权相同, 则可以通过比较它们的标识来打破对称. 这样, 在每次迭代中, 需要调整感知方向的那些节点的感知区域是互不影响的, 在它们各自调整好其感知方向后, 目标区域被覆盖的面积增量实际上等于这些节点的私有覆盖面积增量之和.

在算法的迭代过程中, 由于邻居的感知方向发生了变化, 某个处于局部最优状态的节点很可能又进入了非局部最优状态, 因此可能需要多次调整某个节点的感知方向. 若在当前迭代中, 调整了节点  $i$  的感知方向, 令调整前的感知方向为  $\alpha'_i$ , 调整后的感知方向为  $\alpha_i^*$ . 易知  $\forall j \in N_i$ , 若函数  $\phi(r, \theta, p_j, \alpha_j)$ ,  $0 \leq \alpha_j < 2\pi$  的值域与  $\phi(r, \theta, p_i, \alpha'_i)$  或  $\phi(r, \theta, p_i, \alpha_i^*)$  有重叠部分, 则  $\varphi_j$  会发生改变, 否则  $\varphi_j$  将保持不变. 因此, 在每次迭代后, 只需重新计算与  $\phi(r, \theta, p_i, \alpha'_i)$  或  $\phi(r, \theta, p_i, \alpha_i^*)$  的最短距离小于  $r$  的那些节点的状态和优先权. GIA 算法可描述如下.

### 算法: GIA 算法

输入: 目标区域  $R$ , 感知距离  $r$ , 感知视角  $\theta$ , 每个节点的位置坐标  $p_1, p_2, \dots, p_n$ , 每个节点的初始感知方向  $\alpha_1^0, \alpha_2^0, \dots, \alpha_n^0$ .

输出: 每个节点的最终感知方向  $\alpha_1^1, \alpha_2^1, \dots, \alpha_n^1$ .

变量:  $\alpha'_i$  表示节点  $i$  当前的感知方向,  $\alpha_i^*$  表示节点  $i$  调整到局部最优状态时的感知方向,  $\varphi_i$  表示节点  $i$  当前的私有覆盖区域面积,  $\varphi_i^*$  表示节点  $i$  调整到局部最优状态时的私有覆盖区域面积, 1

$\leq i \leq n$ .  $S$  表示需要更新状态的节点集合.

- (1)  $(\alpha'_1, \alpha'_2, \dots, \alpha'_n) = (\alpha_1^0, \alpha_2^0, \dots, \alpha_n^0)$ ;
- (2)  $S = \{1, 2, \dots, n\}$ ;
- (3)  $changed = true$ ;
- (4) while ( $changed$ )
- (5) for each node  $i$  in  $S$
- (6)  $\varphi_i = \varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha'_n)$ ;
- (7)  $\alpha_i^* = \varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha_i, \dots, \alpha'_n)$  的最大值点;
- (8)  $\varphi_i^* = \varphi_i(\alpha_1, \alpha_2, \dots, \alpha_i^*, \dots, \alpha_n)$ ;
- (9) end for
- (10)  $S = \Phi$ ;
- (11)  $changed = false$ ;
- (12) for each node  $i$
- (13) if  $(\varphi_i^* - \varphi_i > d$  and  $(\varphi_i^* - \varphi_i, i) > \max_{j \in N_i} (\varphi_j^* - \varphi_j, j))$
- (14) for each node  $j$  in  $N_i$
- (15) if  $(\|p_j, \phi(r, \theta, p_i, \alpha'_i)\| < r$  or  $\|p_j, \phi(r, \theta, p_i, \alpha_i^*)\| < r)$
- (16)  $S = S \cup \{j\}$ ;
- (17) end if
- (18) end for
- (19)  $\alpha'_k = \alpha_k^*$ ;
- (20)  $\varphi'_k = \varphi_k^*$ ;
- (21)  $changed = true$ ;
- (22) end if
- (23) end for
- (24) end while
- (25)  $(\alpha_1^1, \alpha_2^1, \dots, \alpha_n^1) = (\alpha'_1, \alpha'_2, \dots, \alpha'_n)$ ;

在 GIA 算法返回一个局部最优方向序列后, 各个节点根据这个方向序列对其传感器进行实际的物理转动. 在 GIA 算法中,  $d \geq 0$  表示允许的私有覆盖面积的最小增量, 在实际中可根据需要的精度进行设定. 若  $d > 0$ , 则每次迭代后覆盖率最少增加了  $d/Area(R)$ , 而覆盖率不会超过 1, 因此 GIA 算法最多迭代  $Area(R)/d$  次.

**定理 1** 在 GIA 算法中, 若  $d = 0$ , 则当迭代次数趋向于无穷大时, 节点的方向序列会收敛于一个局部最优方向序列.

**证明** 令第  $t$  次迭代时节点的方向序列为  $(\alpha'_1(t), \alpha'_2(t), \dots, \alpha'_n(t))$ , 节点  $i$  的私有覆盖区域面积为  $\varphi'_i(t)$ , 节点  $i$  处于局部最优状态时的私有覆盖区域面积为  $\varphi_i^*(t)$ , 覆盖率为  $\mu(t)$ . 由于函数  $\varphi_i(\alpha_1, \alpha_2, \dots, \alpha_n)$ ,  $0 \leq \alpha_i < 2\pi$  的连续性, 因此, 只需证明当  $t \rightarrow \infty$  时,  $\forall i \in \{1, 2, \dots, n\}$ ,  $\varphi'_i(t) \rightarrow \varphi_i^*(t)$  即可.

令  $\Delta\varphi_{\max}(t) = \max_{1 \leq i \leq n} (\varphi_i^*(t) - \varphi'_i(t))$ . 在每次迭代中, 具有最大私有覆盖面积增量的节点都会执行 GIA 算法中的第 (14) ~ (21) 步, 可得到  $\forall t \geq 1, \mu(t+1) - \mu(t) \geq \Delta\varphi_{\max}(t)/Area(R)$ . 因此,  $\forall \varepsilon > 0$ , 数列  $\Delta\varphi_{\max}(t), t \geq 1$  中最多只有  $[1 - \mu(1)]Area(R)/\varepsilon$  项的值大于等于  $\varepsilon$ , 即当  $t \rightarrow \infty$  时,  $\Delta\varphi_{\max}(t) \rightarrow 0$ , 可得  $\forall i \in \{1, 2, \dots, n\}, \varphi'_i(t) \rightarrow \varphi_i^*(t)$ .

在 GIA 算法中的每次迭代中, 对于任一节点  $i$ ,  $\varphi'_i, \alpha_i^*, \varphi_i^*$  的计算以及判断它的优先权是否大于任一邻居的优先权都只需知道其所有邻居的位置坐标、感知

方向和优先权即可,也就是 GIA 算法是个局部式算法,可采用与文献[10,12]中相似的方法在有向传感器网络中分布式实现.在每次迭代后,调整了感知方向的那些节点都需要向它们各自的邻居通知其最新的感知方向,以  $w$  表示这些节点的数量,若采用广播通信机制,则共需发送  $O(w)$  个消息,若这些邻居的优先权发生改变,则这些邻居还需要向它们各自的邻居通知其最新的优先权,共需发送  $O(w\delta)$  个消息,其中  $\delta$  表示任一节点的最大邻居数目.也可采用集中式方法来实现:某个主节点先收集所有节点的位置坐标,然后完成 GIA 算法的计算过程,最后将计算结果通知给每个节点.由于无线传感器网络的生存时间一般很长,在通过覆盖增强算法设定各个节点的感知方向后,可以工作很长一段时间,因此实际上覆盖增强算法的相对代价较小.

### 4.3 区域计算

GIA 算法实现的难点是第(6)~(8)这三步,即计算  $\varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha'_n)$  的值以及计算函数  $\varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha_i, \dots, \alpha'_n), 0 \leq \alpha_i < 2\pi$  在何处取最大值.若采用基于网格的样本点法<sup>[13]</sup>,利用网格的交叉点来近似表示节点的覆盖区域,则计算时间较长且结果不精确.为此,我们提出了一种通过计算几何求解的方法.

先看如何计算  $\varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha'_n)$  的值,即求区域  $R \cap \phi(r, \theta, p_i, \alpha'_i) - \bigcup_{j \in N_i} \phi(r, \theta, p_j, \alpha'_j)$  的面积.我们把区域  $R$  和  $\phi(r, \theta, p_k, \alpha'_k)$  都表示为一个边序列,  $1 \leq k \leq n$ .如图 1 所示,节点  $i$  的感知区域表示为  $\widehat{p_i q_0, q_0 q_1, q_1 p_i}$ ,其中  $\widehat{q_0 q_1}$  表示从点  $q_0$  到点  $q_1$  的圆弧.接下来,就可采用与文献[16]中相似的方法,通过计算几何求解这些区域的交集、并集、差集和面积.区域  $R \cap \phi(r, \theta, p_i, \alpha'_i) - \bigcup_{j \in N_i} \phi(r, \theta, p_j, \alpha'_j)$  的边数为  $O(\delta^2)$ ,因此计算  $\varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha'_n)$  的时间复杂性为  $O(\delta^3)$ .

再看如何计算函数  $\varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha_i, \dots, \alpha'_n), 0 \leq \alpha_i < 2\pi$  在何处取最大值,求解过程分为以下几步.

(1) 节点  $i$  的私有覆盖区域也可表示为  $[R - \bigcup_{j \in N_i} \phi(r, \theta, p_j, \alpha'_j)] \cap \phi(r, \theta, p_i, \alpha'_i)$ ,因此

先求出区域  $R - \bigcup_{j \in N_i} \phi(r, \theta, p_j, \alpha'_j)$ ,以  $E_i$  表示,易知  $E_i$  中的边必是圆弧或线段,如图 2 所示.

(2) 以函数  $v_i(\omega), 0 \leq \omega < 2\pi$  表示以  $p_i$  为起点、长度为  $r$ 、与  $X$  轴夹角为

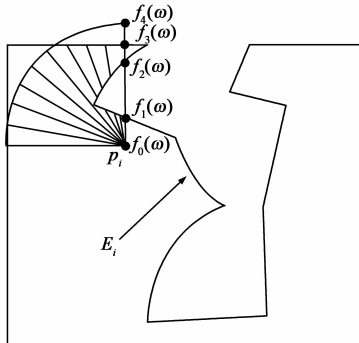


图2 计算节点的私有覆盖区域面积

$\omega$  的向量.  $\forall e \in E_i$ , 令函数  $f_e(\omega) = \|\mathbf{v}_i(\omega)$  和  $e$  的交点,  $p_i\|$ , 其定义域为  $\mathbf{v}_i(\omega)$  和  $e$  有交点时  $\omega$  的取值范围.若  $e$  为圆弧且  $\mathbf{v}_i(\omega)$  和  $e$  有两个交点,则需对应地定义两个函数  $f_e(\omega)$  和  $f'_e(\omega)$ .以  $F_i$  表示所有这样的函数的集合,即  $F_i = \{f_e(\omega), f'_e(\omega) \mid e \in E_i\}$ .

(3) 令函数  $g_i(\omega) = d\varphi_i(\alpha'_1, \alpha'_2, \dots, \omega, \dots, \alpha'_n)/d\theta, 0 \leq \omega < 2\pi$ , 表示以  $p_i$  为中心、半径为  $r$ 、与  $X$  轴夹角为  $\omega$  的单位角度的扇形在区域  $E_i$  内的面积,则

$$\varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha_i, \dots, \alpha'_n) = \int_{\alpha_i - \theta/2}^{\alpha_i + \theta/2} g_i(\omega) d\omega$$

可采用如下的方法求出  $g_i(\omega)$ :对  $F_i$  中所有在  $\omega$  处有定义的函数,按照递增顺序排序,不妨设为  $f_1(\omega), f_2(\omega), \dots, f_m(\omega)$ , 并令  $f_0(\omega) = 0, f_{m+1}(\omega) = r$ , 如图 2 所示.易知,若  $p_i$  位于区域  $E_i$  之内,则向量  $\mathbf{v}_i(\omega)$  上的区间  $[f_0(\omega), f_1(\omega)], [f_2(\omega), f_3(\omega)], \dots, [f_{\lfloor m/2 \rfloor}(\omega), f_{\lfloor m/2 \rfloor + 1}(\omega)]$  位于区域  $E_i$  之内,得到  $g_i(\omega) = \sum_{i=0}^{\lfloor m/2 \rfloor} [f_{2i+1}^2(\omega) - f_{2i}^2(\omega)]/2$ ;反之,若  $p_i$  位于区域  $E_i$  之外,则向量  $\mathbf{v}_i(\omega)$  上的区间  $[f_1(\omega), f_2(\omega)], [f_3(\omega), f_4(\omega)], \dots, [f_{\lfloor (m+1)/2 \rfloor - 1}(\omega), f_{\lfloor (m+1)/2 \rfloor}(\omega)]$  位于区域  $E_i$  之内,得到  $g_i(\omega) = \sum_{i=0}^{\lfloor (m+1)/2 \rfloor} [f_{2i}^2(\omega) - f_{2i-1}^2(\omega)]/2$ .

(4) 再通过数值分析的方法近似得到函数  $\varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha_i, \dots, \alpha'_n), 0 \leq \alpha_i < 2\pi$  的最大值:先将以  $p_i$  为中心、半径为  $r$  的圆盘等分为  $b$  个扇形,第  $k$  个扇形的右侧边与  $X$  轴的夹角为  $\omega_k = 2\pi k/b, 0 \leq k < b$ ;然后求出每个扇形在区域  $E_i$  内的面积,当  $b$  较大时,可近似表示为  $2\pi g_i(\omega_k)/b$ .令  $c = \lfloor b\theta/(4\pi) \rfloor$ ,即  $2c$  为节点  $i$  的感知区域所拥有的扇形个数,则有

$$\begin{aligned} & \varphi_i(\alpha'_1, \alpha'_2, \dots, \omega_0, \dots, \alpha'_n) \\ & \approx 2\pi \left[ \sum_{j=0}^{c-1} g_i(\omega_j) + \sum_{j=b-c}^{b-1} g_i(\omega_j) \right] / b, \\ & \varphi_i(\alpha'_1, \alpha'_2, \dots, \omega_k, \dots, \alpha'_n) \approx \varphi_i(\alpha'_1, \alpha'_2, \dots, \omega_{k-1}, \dots, \alpha'_n) \\ & + 2\pi \left[ g_i(\omega_{(k+c-1)\%b}) - g_i(\omega_{(k-c-1)\%b}) \right], \\ & 1 \leq k < b \end{aligned} \quad (5)$$

通过式(5),可递归求出函数  $\varphi_i(\alpha'_1, \alpha'_2, \dots, \alpha_i, \dots, \alpha'_n), 0 \leq \alpha_i < 2\pi$  在  $2\pi k/b, 0 \leq k < b$  处的值.当  $b$  较大时,即可近似得到其最大值.区域  $E_i$  的边数为  $O(\delta^2)$ ,计算  $E_i$  的时间复杂性为  $O(\delta^3)$ ,计算  $g_i(\omega)$  的时间复杂性为  $O(\delta^2 \log \delta)$ ,计算  $\varphi_i(\alpha'_1, \alpha'_2, \dots, \omega_k, \dots, \alpha'_n), 0 \leq k < b$  的时间复杂性为  $O(b\delta^2 \log \delta)$ ,因此计算最大值点的时间复杂性为  $O(b\delta^2 \log \delta + \delta^3)$ .

虽然上述方法也采用了数值分析的方法求近似值,但只对角度进行了离散化处理,而基于网格的样本

点法则同时对横坐标和纵坐标进行了离散化处理.在对角度进行离散化处理后,每个节点最多只能将其感知方向转动到  $b$  个角度,因此每个节点的最小私有覆盖面积增量都必定会大于某个正数,这样即使取  $d = 0$ , GIA 算法也会在有限迭代次数内结束.

### 5 性能评估

在本节我们通过仿真实验评估 GIA 算法的收敛性、覆盖率和执行时间等性能指标.我们用 C++ 语言设计了仿真程序 DSNCover 1.0,实现了 GIA 算法和文献[10]中提出的 PFCEA 算法.此外,由于对 GIA 算法做一些简单的改动就可使其运行在离散方向感知模型下,该程序也实现了离散方向模型下的 GIA 算法(简称 DGIA 算法)和文献[12]中提出的 PGreedy 算法.为了充分评估 GIA 算法的性能,GIA 算法中的参数设定为: $d = 0, b = 500$ .在仿真实验中,每个节点均匀随机地分布在正方形的目标区域内,且每个节点的初始感知方向也是均匀随机分布的.以  $l$  表示目标区域的边长, $\rho$  表示目标区域内的平均节点密度.

#### 5.1 收敛性

先测试 GIA 算法的收敛性.在实验中,固定  $r = 10m, \theta = \pi/2, \rho = 0.016m^{-2}$ ,  $l$  分别取 50m, 100m, 150m, 200m,相应地  $n$  分别取 40, 160, 360, 640,观察 GIA 算法在运行过程中不同迭代次数后覆盖率的变化,图3显示了100次实验结果的平均值.可以看出,随着迭代次数的增多,覆盖率不断得到增强.当问题规模扩大时,GIA 算法的总迭代次数也相应地增加,在  $l = 200m$  时总迭代次数为 100 左右.在迭代次数较小时,覆盖率随着迭代次数的增多而迅速增加,当完成约 20 次迭代后,覆盖率就可达到最终覆盖率的 99% 以上.

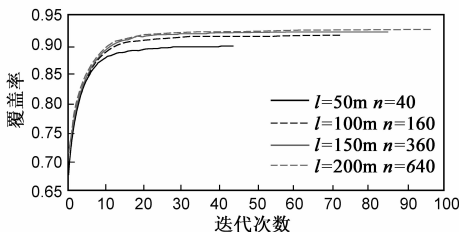


图3 GIA算法在不同迭代次数后的覆盖率

再测试 GIA 算法收敛结果的有效性.GIA 算法得到的是一个局部最优方向序列,相当于一个局部最优解,其结果除了取决于参数  $R, r, \theta, p_1, p_2, \dots, p_n$  外,也与  $\alpha_1, \alpha_2, \dots, \alpha_n$  的初始值相关.对于相同的  $R, r, \theta, p_1, p_2, \dots, p_n$ ,可测试 GIA 算法在  $\alpha_1, \alpha_2, \dots, \alpha_n$  取不同初始值时得到的覆盖率,若测试的次数足够多,则得到的最大覆盖率很可能接近于最优覆盖率.分别进行以下4组实验: $r = 10m, \theta = \pi/2, \rho = 0.016m^{-2}$ ,  $l$  分别取 50m, 100m,

150m, 200m,相应地  $n$  分别取 40, 160, 360, 640.对于每组实验,随机生成  $10^5$  次  $\alpha_1, \alpha_2, \dots, \alpha_n$  的初始值,GIA 算法得到的相对覆盖率的分布如图4所示,这里的相对覆盖率是指覆盖率与  $10^5$  次测试结果中的最大覆盖率之比.可以看出,对于相同的  $R, r, \theta, p_1, p_2, \dots, p_n$ ,在  $\alpha_1, \alpha_2, \dots, \alpha_n$  取不同初始值时,GIA 算法得到的最小覆盖率与最大覆盖率只相差在 7% 以内,在大多数情况下 GIA 算法得到的覆盖率与最大覆盖率只相差在 3% 以内.

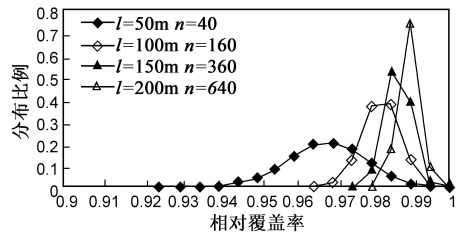


图4 GIA算法得到的覆盖率的分布

#### 5.2 覆盖率

图5显示了  $l = 50m, r = 10m, \theta = \pi/2, n = 40$  时 GIA 算法和 PFCEA 算法的某次运行结果.图5(a)显示了初始时各个节点的感知方向,此时覆盖率仅为 64.4%.图5(b)显示了运行 GIA 算法后各个节点的感知方向,可以看出,各个节点都将自身的感知方向调整到了局部最优角度,大部分目标区域已被节点覆盖,覆盖率达到了 91.3%.图5(c)显示了运行 PFCEA 算法后各个节点的感知方向,可以看出,此时仍能通过转动某些节点的感知方向来增强覆盖率,此外 PFCEA 算法没考虑到目标区域边界的影响,边界附近的某些节点的大部分感知区域已在目标区域之外,因此得到的覆盖率较低,为 72.7%.GIA 算法得到的覆盖率约为 PFCEA 算法的 1.26 倍,相对初始时的覆盖率增量约为 PFCEA 算法的 3.24 倍,优势较为明显.

影响覆盖增强算法的覆盖率的主要参数为  $R, r, \theta, p_1, p_2, \dots, p_n$ ,实验中分别改变  $R, r, \theta$  和  $n$  的值,并分别采用连续方向和离散方向这两种感知模型,观察 GIA 算法和 DGIA 算法在不同场景下得到的覆盖率,同时也运行了 PFCEA 算法和 PGreedy 算法以进行对比分析.分别进行以下4组实验:(1)  $l \in [50m, 200m], r = 10m, \theta = \pi/2, n = 160$ ;(2)  $l = 100m, r \in [5m, 15m], \theta = \pi/$

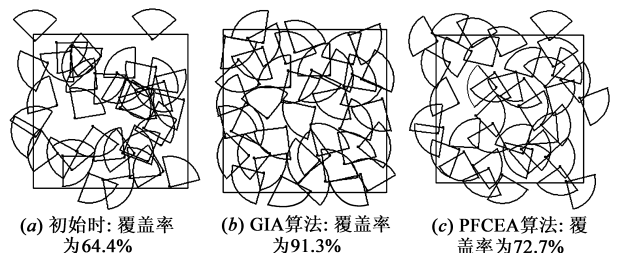
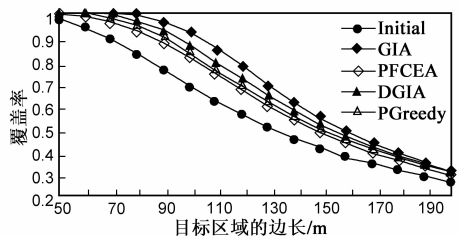
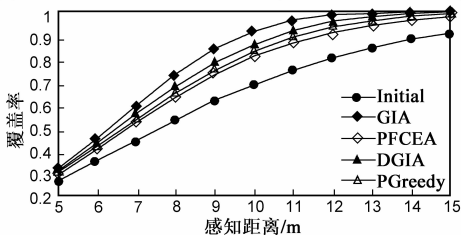


图5 GIA算法和PFCEA算法的某次运行结果 ( $l=50m, r=10m, \theta=\pi/2, n=40$ )

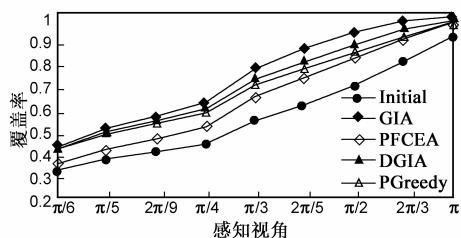
2,  $n = 160$ ; (3)  $l = 100\text{m}$ ,  $r = 10\text{m}$ ,  $\theta \in [\pi/6, \pi]$ ,  $n = 160$ ; (4)  $l = 100\text{m}$ ,  $r = 10\text{m}$ ,  $\theta = \pi/2$ ,  $n \in [50, 200]$ . 图 6(a) ~ (d) 显示了这 4 组实验的 100 次实验结果的平均值. 因为目标区域的面积为  $l^2$ , 单个节点的感知区域面积为  $\theta r^2/2$ , 所以基本上  $n\theta r^2/l^2$  越大, 覆盖率就越大, 实验结果证实了这点. 为了达到所需的覆盖率, 实际应用中可根据仿真实验结果来确定所需布置的节点数. 可以看出, 各个覆盖增强算法都在节点密度适中时覆盖率增量最大, GIA 算法得到的覆盖率增量最高可达到 0.241, 这是因为若节点密度太低, 则初始时感知的冗余区域较小, 反之若节点密度太高, 则初始时感知的盲区较小, 而覆盖增强是通过消灭冗余区域和盲区实现的. 对于连续方向感知模型, GIA 算法得到的覆盖率要优于 PFCEA 算法, 两者覆盖率之比最高可达到 1.21 倍, 两者覆盖率增量之比最高可达到 3.62 倍. 对于离散方向感知模型, DGIA 算法得到的覆盖率要优于 PGreedy 算法, 两者覆盖率之比最高可达到 1.05 倍, 两者覆盖率增量



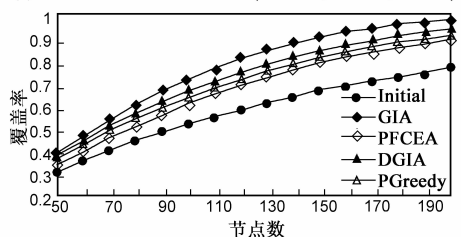
(a) 目标区域边长和覆盖率的关系 ( $r=10\text{m}$ ,  $\theta=\pi/2$ ,  $n=160$ )



(b) 感知距离和覆盖率的关系 ( $l=100\text{m}$ ,  $\theta=\pi/2$ ,  $n=160$ )



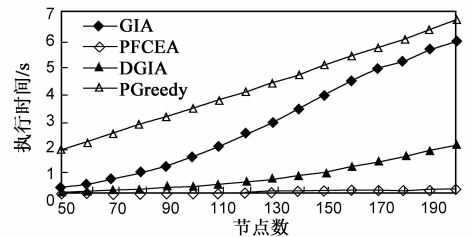
(c) 感知视角和覆盖率的关系 ( $l=100\text{m}$ ,  $r=10\text{m}$ ,  $n=160$ )



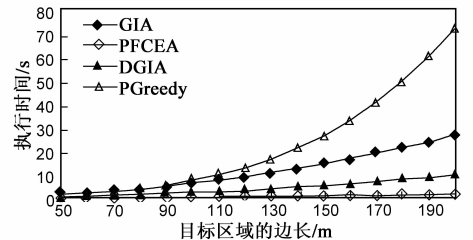
(d) 节点数和覆盖率的关系 ( $l=100\text{m}$ ,  $r=10\text{m}$ ,  $\theta=\pi/2$ )

图6 GIA算法、PFCEA算法、DGIA算法和PGreedy算法在不同实验场景下得到的覆盖率

之比最高可达到 1.32 倍.



(a) 节点数和执行时间的关系 ( $l=100\text{m}$ ,  $r=10\text{m}$ ,  $\theta=\pi/2$ )



(b) 整体规模和执行时间的关系 ( $r=10\text{m}$ ,  $\theta=\pi/2$ ,  $\rho=0.02\text{m}^{-2}$ )

图7 GIA算法、PFCEA算法、DGIA算法和PGreedy算法在不同实验场景下的执行时间

### 5.3 执行时间

最后来测试各个算法的执行时间, 实验平台的性能参数为: 2.66GHz CPU、2GB 内存. 分别进行以下 2 组实验: (1)  $l = 100\text{m}$ ,  $r = 10\text{m}$ ,  $\theta = \pi/2$ ,  $n \in [50, 200]$ ; (2)  $r = 10\text{m}$ ,  $\theta = \pi/2$ ,  $\rho = 0.02\text{m}^{-2}$ ,  $l \in [50\text{m}, 200\text{m}]$ ,  $n \in [50, 800]$ . 图 7(a)、(b) 显示了这 2 组实验的 100 次实验结果的平均值. 可以看出, 随着节点数的增多或整体规模的增大, 各个算法的执行时间会增长. 由于 GIA 算法的执行时间与迭代次数以及每次迭代所花的时间都有关系, 因此很难找到节点数与执行时间之间的确切关系. 但在大部分情况下, GIA 算法和 DGIA 算法都可在 30 秒内执行完. 因为 PFCEA 算法的计算过程比较简单, 其执行时间也是最短的. 而 PGreedy 算法中使用基于网格的样本点法计算区域, 其执行时间相对较长.

## 6 结论

在有向传感器网络中, 可以通过调整节点的感知方向来增强目标区域的覆盖率, 最大化目标区域的覆盖率是覆盖增强问题的目标. 提出了有向传感器网络覆盖增强的一种贪婪迭代算法, 该算法虽然针对连续方向感知模型设计, 但通过简单改动也可使其运行在离散方向感知模型下. 仿真实验结果表明该算法的执行时间较短, 收敛速度较快, 且绝大多数情况下都会收敛于一个较优的解, 在目标区域覆盖率方面, 该算法要优于现有的适用于连续方向感知模型的 PFCEA 算法和适用于离散方向感知模型的 PGreedy 算法.

### 参考文献

[1] Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey

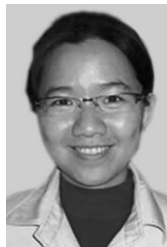
- [J]. Computer Networks, 2008, 52(12): 2292 – 2330.
- [2] Zhang J, Varadharajan V. Wireless sensor network key management survey and taxonomy[J]. Journal of Network and Computer Applications, 2010, 33(2): 63 – 75.
- [3] Akyildiz IF, Melodia T, Chowdhury KR. A survey on wireless multimedia sensor networks[J]. Computer Networks, 2007, 51(4): 921 – 960.
- [4] 陶丹, 马华东, 刘亮. 视频传感器网络中路径覆盖增强算法研究[J]. 电子学报, 2008, 36(7): 1291 – 1296.  
Tao Dan, Ma Hua-dong, Liu Liang. Study on path coverage enhancement algorithm for video sensor networks[J]. Acta Electronica Sinica, 2008, 36(7): 1291 – 1296. (in Chinese)
- [5] 陶丹, 孙岩, 陈后金. 视频传感器网络中最坏情况覆盖检测与修补算法[J]. 电子学报, 2009, 37(10): 2284 – 2290.  
Tao Dan, Sun Yan, Chen Hou-jin. Worst-case coverage detection and repair algorithm for video sensor networks[J]. Acta Electronica Sinica, 2009, 37(10): 2284 – 2290. (in Chinese)
- [6] Djugash J, Singh S, Kantor G, et al. Range-only slam for robots operating cooperatively with sensor networks[A]. Proceedings of the IEEE International Conference on Robotics and Automation[C]. Los Alamitos: IEEE Computer Society Press, 2006. 2078 – 2084.
- [7] Szwedczyk R, Mainwaring A, Polastre J, et al. An analysis of a large scale habitat monitoring application[A]. Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems[C]. New York: ACM Press, 2004. 214 – 226.
- [8] 凡高娟, 王汝传, 黄海平, 等. 基于容忍覆盖区域的无线传感器网络节点调度算法[J]. 电子学报, 2011, 39(1): 89 – 94.  
Fan Gao-juan, Wang Ru-chuan, Huang Hai-ping, et al. Tolerable coverage area based node scheduling algorithm in wireless sensor networks[J]. Acta Electronica Sinica, 2011, 39(1): 89 – 94. (in Chinese)
- [9] Ghosh A, Das SK. Coverage and connectivity issues in wireless sensor networks: a survey[J]. Pervasive and Mobile Computing, 2008, 4(3): 303 – 334.
- [10] 陶丹, 马华东, 刘亮. 基于虚拟势场的有向传感器网络覆盖增强算法[J]. 软件学报, 2007, 18(5): 1152 – 1163.  
Tao Dan, Ma Hua-dong, Liu Liang. A virtual potential field based coverage-enhancing algorithm for directional sensor networks[J]. Journal of Software, 2007, 18(5): 1152 – 1163. (in Chinese)
- [11] Ai J, Abouzeid AA. Coverage by directional sensors in randomly deployed wireless sensor networks[J]. Journal of Combinatorial Optimization, 2006, 11(1): 21 – 41.
- [12] 程卫芳, 廖湘科, 沈昌祥. 有向传感器网络最大覆盖调度算法[J]. 软件学报, 2009, 20(4): 975 – 984.  
Cheng Wei-fang, Liao Xiang-ke, Shen Chang-xiang. Maximal coverage scheduling in wireless directional sensor networks[J]. Journal of Software, 2009, 20(4): 975 – 984. (in Chinese)
- [13] Cai Y, Lou W, Li M, et al. Energy efficient target-oriented scheduling in directional sensor networks[J]. IEEE Transactions on Computers, 2009, 58(9): 1259 – 1274.
- [14] Ma HD, Liu YH. On coverage problems of directional sensor networks[A]. Proceedings of the 1st International Conference on Mobile Ad-Hoc and Sensor Networks[C]. Los Alamitos: IEEE Computer Society Press, 2005. 721 – 731.
- [15] Khan UA, Kar S, Moura JMF. Distributed sensor localization in random environments using minimal number of anchor nodes[J]. IEEE Transactions on Signal Processing, 2009, 57(5): 2000 – 2016.
- [16] 周培德. 计算几何—算法分析与设计(2版)[M]. 北京: 清华大学出版社, 2005.  
Zhou Pei-de. Computational Geometry – Algorithm Analysis and Design (2nd edition)[M]. Beijing: Tsinghua University Press, 2005. (in Chinese)

#### 作者简介



**陆克中** 男, 1982年2月出生, 安徽枞阳人. 2006年在中国科学技术大学获得博士学位. 现为深圳大学计算机与软件学院副教授, 主要从事无线传感器网络与并行分布式计算等方面的研究工作.

E-mail: kzlu@szu.edu.cn



**冯禹洪** 女, 1975年6月出生, 广东茂名. 2008年在中国科学技术大学获得博士学位. 现为深圳大学计算机与软件学院讲师, 主要从事物联网与并行分布式计算等方面的研究工作.

E-mail: yuhongf@szu.edu.cn

**毛睿** 男, 1975年5月出生, 安徽蚌埠人. 2007年在德克萨斯大学奥斯丁分校获得博士学位. 现为深圳大学计算机与软件学院副教授, 主要从事数据库与并行分布式计算等方面的研究工作.

E-mail: mao@szu.edu.cn

**罗秋明** 男, 1974年10月出生, 广东惠州人. 2003年在华中科技大学获得博士学位. 现为深圳大学计算机与软件学院副教授, 主要从事无线传感器网络与并行体系结构等方面的研究工作.

E-mail: lqm@szu.edu.cn

**刘刚** 男, 1978年3月出生, 甘肃兰州人. 2006年在中国科学技术大学获得博士学位. 现为深圳大学计算机与软件学院讲师, 主要从事并行分布式计算等方面的研究工作.

E-mail: gliu@szu.edu.cn