

文章编号: 1001-0920(2012)12-1810-06

传感器网络中陷阱空洞的分布式检测及修复

王力立, 吴晓蓓

(南京理工大学 自动化学院, 南京 210094)

摘要: 针对传感器网络难以实现完全覆盖的情况, 研究陷阱覆盖方式下陷阱空洞的检测和修复, 提出了分布式的检测和修复方法. 陷阱空洞检测方法可以让节点分布式自主地确定空洞边界曲线, 从而得到精确的空洞信息, 判断出该空洞是否是陷阱空洞. 陷阱空洞修复方法基于边权图的概念, 通过多轮计算确定新增节点的位置. 所提出算法充分考虑了监测区域的边界效应, 并且比现有算法在需要新增的节点数目和节能方面更有优势, 仿真结果表明了它的有效性.

关键词: 传感器网络; 陷阱覆盖; 陷阱空洞

中图分类号: TP393

文献标志码: A

Decentralized detection and patching of trap coverage holes for sensor networks

WANG Li-li, WU Xiao-bei

(School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China. Correspondent: WANG Li-li, E-mail: mail_wll@126.com)

Abstract: For the condition that it is difficult to satisfy the requirement of full coverage in sensor networks, the problem of detecting and patching trap coverage holes is considered, and a trap hole-detection method and a trap hole-patching method are proposed. The trap hole-detection method exactly finds out the borderline of all coverage holes in a decentralized way. Therefore, the network can get the information of these holes, and determine if they are trap holes. The trap hole-patching method is based on the conception of edge-weighted graph. The locations of new nodes for patching the trap holes are estimated by using the two algorithms for several rounds. The algorithms consider the impact of network boundary compared with traditional methods, the proposed algorithms efficiently save the energy and need fewer nodes for patching trap holes. Simulation results show the effectiveness of the proposed method.

Key words: sensor networks; trap coverage; trap hole

1 引言

无线传感器网络的覆盖性能反映了网络节点对监测区域的监控程度, 是衡量无线传感器网络服务质量的一项基本指标^[1], 也是近年来研究的主要课题之一^[2-4]. 区域覆盖问题是最为常见的一类传感器网络覆盖问题, 它要求监测区域中的每个点至少被一个传感器节点的感知范围覆盖. 监测区域中未被任何传感器覆盖的点的集合即为覆盖空洞. 覆盖空洞会导致监测信息感知不完整, 还有可能造成网络通信不连通.

目前, 针对区域覆盖的覆盖空洞问题已经有了一些解决方案, 其中最具有代表性的是虚拟势场法. 该方法将具有移动能力的传感器节点虚拟成带电粒子,

节点再根据某种运动规则扩散到受力平衡位置. 文献[5-8]便是基于这种方法, 有效地提高了网络覆盖率; 但在大规模部署中, 要实现区域完全覆盖是相当困难的. 对此, 文献[9]提出了“陷阱覆盖”的概念, 称传感器网络可以对某监测区域提供某一阈值的陷阱覆盖, 是指该监测区域中的任一覆盖空洞的直径不超过该阈值. 空洞直径是该空洞区域中两点间欧氏距离中的最大值, 直径大于阈值的覆盖空洞为陷阱空洞. 陷阱覆盖放宽了区域覆盖的要求, 它允许监测区域中存在直径不大于阈值的覆盖空洞, 并且阈值比区域覆盖中的性能指标“覆盖率”^[7]更好地反映了网络覆盖质量. 因为, 即使覆盖率很高, 最大空洞直径也可能很

收稿日期: 2011-08-02; 修回日期: 2011-12-01.

基金项目: 南京理工大学自主科研专项计划项目(2011YBXM113, 2011YBXM114).

作者简介: 王力立(1987-), 女, 博士生, 从事无线传感器网络的研究; 吴晓蓓(1958-), 女, 教授, 从事网络控制、无线传感器网络等研究.

大;而在陷阱覆盖中只要动态目标的位移大于阈值,则一定会被传感器网络发现,保证了最坏的覆盖情况.文献[10]将陷阱覆盖的特点描述为观测事件可能发生在监测区域中的任何一处,或者入侵者可以在监测区域中任意运动.文献[11]提出了可配置覆盖粒度的概念,利用图论技术组成合适的覆盖集,从而调整空洞直径;但此方法会引起网络拓扑非常大的变化,而且更适合节点分布十分稠密的网络.文献[12]在中继机器人网络中,进一步运用陷阱覆盖跟踪和捕获移动目标.

陷阱覆盖可以广泛应用于入侵监测、火灾报警、有毒气体监测等场合.陷阱覆盖与区域覆盖相比,问题目标不同也更具灵活性.但现有覆盖空洞的研究多是基于区域覆盖的情况,其成果无法直接应用到陷阱覆盖下的空洞检测和修复中.对此,本文针对陷阱覆盖的特点,并考虑监测区域的边界效应,提出了可以在节点上并行进行的陷阱空洞检测算法(THDA)和陷阱空洞修复算法(THPA).

2 陷阱空洞的检测算法

本文无特殊申明时均基于如下假设:

1) 网络中的传感器节点是同构的,即它们的感知范围和通信范围是相同的.

2) 任意节点*i*的感知范围是以节点*i*为圆心, r_s 为半径的圆.通信范围是以节点*i*为圆心, r_c 为半径的圆,并且 $r_c \geq 2r_s$,以确保邻居节点之间可以直接通信.

3) 节点可以通过某种定位算法或者定位装置获得自身的位置信息.

4) 节点以随机的方式部署在正方形的监测区域内,考虑边界因素.

2.1 确定覆盖空洞边界弧段

某一节点*i*的邻居节点集定义为

$$N(i) = \{j \in N \mid d(i, j) \leq 2r_s, j \neq i\}. \quad (1)$$

其中: N 表示网络传感器节点集合, $d(i, j)$ 表示节点*i*与节点*j*之间的欧氏距离, r_s 表示节点的感知半径.邻居节点的位置信息可通过各节点广播自己的位置信息来获得.

本文的陷阱空洞检测算法是基于“有效覆盖范围”^[13]和“有效弧段”的思想,分别定义如下:

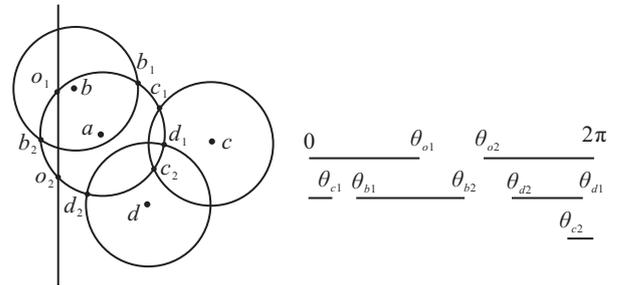
定义 1 节点*i*的有效覆盖范围是节点覆盖范围*C*(*i*)与被监测区域*A*的交集,表示为*C_e*(*i*),即

$$C_e(i) = [C(i) \cap A]. \quad (2)$$

定义 2 节点*i*的有效弧段是其有效覆盖范围*C_e*(*i*)的感知圆周弧段,表示为*S_e*(*i*),即

$$S_e(i) = \partial(C_e(i)) - \partial(A) \cap \partial(C_e(i)). \quad (3)$$

如图 1(a)所示,监测区域边界以及邻居节点*b*,*c*,*d*与节点*a*的感知圆周相交于点 $o_1, o_2, b_1, b_2, c_1, c_2, d_1, d_2$,它们将节点*a*的有效覆盖边界分割为若干小弧段,节点*a*的有效弧段是 $\widehat{o_1c_1o_2}$.其中,弧段 $\widehat{c_1b_1}$ 和 $\widehat{o_2d_2}$ 是未被节点*a*的邻居节点覆盖的有效弧段.



(a) 节点*a*位置情况 (b) 节点*a*的覆盖情况

图 1 节点*a*的部署情况

由覆盖空洞的概念可以得知,覆盖空洞的边界曲线由节点感知圆周上未被其他节点覆盖的有效弧段和未被覆盖的监测区域边界线构成.只要找出这些空洞边界弧段和监测区域边界线,再将它们按照一定的规则首尾相连即可准确地得到覆盖空洞.下文给出寻找未被覆盖的有效弧段的具体过程.

以图 2 为例,节点*i*(x_i, y_i)与其邻居节点*j*(x_j, y_j)的感知圆周的交点在节点*i*的感知圆周上的弧度为

$$\theta_{j1}, \theta_{j2} = \begin{cases} \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) \pm \arccos\left(\frac{d(i, j)}{2r}\right), & y_j - y_i \geq 0, x_j - x_i \geq 0; \\ \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) \pm \arccos\left(\frac{d(i, j)}{2r}\right) + \pi, & x_j - x_i < 0; \\ \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) \pm \arccos\left(\frac{d(i, j)}{2r}\right) + 2\pi, & y_j - y_i < 0, x_j - x_i \geq 0. \end{cases} \quad (4)$$

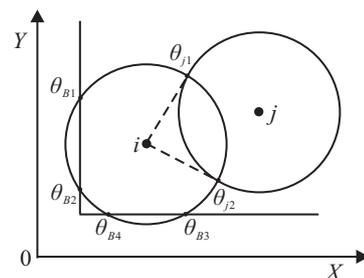


图 2 交点计算模型

节点*i*的感知圆周与边界 $x = x_B$ 的交点在节点*i*的感知圆周上的弧度

$$\begin{cases} \theta_{B1} = \arccos\left(\frac{x_B - x_i}{r}\right), \\ \theta_{B2} = 2\pi - \arccos\left(\frac{x_B - x_i}{r}\right). \end{cases} \quad (5)$$

节点 i 的感知圆周与边界 $y = y_B$ 的交点在节点 i 的感知圆周上的弧度, 当 $y_B - y_i \geq 0$ 时, 有

$$\begin{cases} \theta_{B3} = \arcsin\left(\frac{y_B - y_i}{r}\right), \\ \theta_{B4} = \pi - \arcsin\left(\frac{y_B - y_i}{r}\right); \end{cases} \quad (6)$$

当 $y_B - y_i < 0$ 时, 有

$$\begin{cases} \theta_{B3} = \arcsin\left(\frac{y_B - y_i}{r}\right) + 2\pi, \\ \theta_{B4} = \pi - \arcsin\left(\frac{y_B - y_i}{r}\right). \end{cases} \quad (7)$$

利用交点在节点感知圆周上的弧度可以求得各交点的坐标

$$\begin{cases} x_k = x_i + r \times \cos \theta_k, \\ y_k = y_i + r \times \sin \theta_k. \end{cases} \quad (8)$$

图 1(b) 刻画了节点 a 的感知圆周的被覆盖情况, 图中第 1 行表示有效弧段的弧度范围 $\Phi_e = [0, \theta_{o1}] \cup [\theta_{o2}, 2\pi]$, 第 2 和第 3 行表示被覆盖的感知弧段的弧度范围 $\Phi_c = [0, \theta_{c1}] \cup [\theta_{b1}, \theta_{b2}] \cup [\theta_{d2}, \theta_{d1}] \cup [\theta_{c2}, 2\pi]$. 弧段 $\Phi_u = \Phi_e - \Phi_c \cup \Phi_e = [\theta_{c1}, \theta_{b1}] \cup [\theta_{o2}, \theta_{d2}]$, 即为其未被覆盖的有效弧段. 覆盖空洞边界弧段是由各个节点分布式地自行判定, 其算法程序流程如图 3 所示.

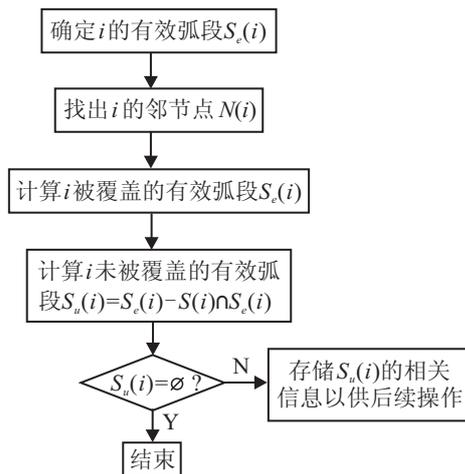


图 3 空洞边界弧段判定流程

2.2 陷阱空洞的判定

为找到陷阱空洞, 各个节点可以建立一张信息元素表, 用于记录其未被覆盖的有效弧段的相关信息, 包括弧段范围和弧段端点位置, 以及与弧段端点相关联的邻居节点 ID 等. 信息元素表中关于某个未被覆盖的有效弧段 s_g 的数据如表 1 所示. 以图 1 中节点 a 未被覆盖的有效弧段 $\widehat{c_1 b_1}$ 为例, 其弧度范围 $\Phi_{c_1, b_1} = [\theta_{c1}, \theta_{b1}]$ 可通过式 (4) 求得, $\widehat{c_1 b_1}$ 的两个端点坐标可通过式 (8) 求得, 与 $\widehat{c_1 b_1}$ 的端点 c_1 和 b_1 相关联的节点分别为节点 c 和节点 b . 五元组 $(\Phi_g, (x_g^1, y_g^1), (x_g^2, y_g^2), id_1,$

$id_2)$ 可以充分表征弧段 s_g 以及它与空洞的关系. $flag$ 是状态标志, 表示在空洞判定过程中 s_g 的被访问情况.

表 1 信息元素表

符号	说明
Φ_g	弧段 s_g 的弧度范围
x_g^1, y_g^1	弧段 s_g 的端点 1 的坐标
x_g^2, y_g^2	弧段 s_g 的端点 2 的坐标
id_1	与 s_g 的端点 1 关联的邻居节点 ID
id_2	与 s_g 的端点 2 关联的邻居节点 ID
$flag$	状态标志 (s_g 被访问过为 0, 否则为 1)

陷阱空洞的判定是基于各节点的信息元素表中的信息进行的, 判定过程通过举例描述如下.

1) 节点 i 取其信息元素表中某一弧段 $\widehat{p_k p_{k+1}}$, 将 $\widehat{p_k p_{k+1}}$ 的有关信息发送给与端点 p_k 关联的节点 j .

2) 节点 j 收到来自 i 的数据包后, 寻找它的信息元素表中以 p_k 为端点的弧段 $\widehat{p_k p_l}$, 节点 j 再将 $\widehat{p_k p_l}$ 的有关信息以及从节点 i 接收到的数据一起打包发送给与端点 p_l 相关联的节点 h .

3) 节点 h 重复节点 j 的上述操作, 此数据传输过程一直在节点间继续, 直至起始节点 i 收到与弧段 $\widehat{p_k p_{k+1}}$ 端点 p_{k+1} 关联的邻节点 q 发来的数据包.

4) 此时, 起始节点 i 已经掌握了弧 $\widehat{p_k p_{k+1}}$ 所属空洞的全部边界组成信息. 需要注意, 如果节点的信息元素表中某条弧段的一端或两端并不与任何节点关联, 则意味着该覆盖空洞处在监测区域的边界上, 默认覆盖空洞的边界由该节点所收集到的弧段以及一部分监测区域边界线组成, 结束操作. 称这样的覆盖空洞为外部空洞, 区别于位于网络内部的空洞.

5) 起始节点 i 收到空洞边界信息数据包后, 即可进行陷阱空洞判定. 节点 i 计算数据包中所有弧段端点以及相关边界顶点两两间的欧氏距离, 如果有大于阈值 d_{\max} 的值, 则该空洞即是陷阱空洞, 否则只是一般的覆盖空洞.

上述步骤 5) 中空洞判定规则的有效性可由下列定理证明.

定理 1 覆盖空洞 H 的边界顶点集为 V , 如果对于 $\forall v_i, v_j \in V$, 都有 $d(v_i, v_j) \leq d_{\max}$, 则 H 不是陷阱空洞.

证明 H 的顶点集为 V , 通过选取 V 中的顶点构成集合 V_1 , 即 $V_1 \subset V$, 适当连接 V_1 中的元素, 可以生成 V 的凸壳 Ω . 因为平面点集的凸壳是凸多边形, 所以 Ω 内任意两点的距离最大值 D_{\max} 必定在顶点之间取得, 假设 $d(v_k, v_l) = D_{\max}$, $v_k, v_l \in V_1$. 因为凹区域 $H \subseteq \Omega$, 所以 H 的最大直径 $\text{diam}(H) = d(v_k, v_l)$. 由陷阱覆盖的概念可知, 当 $D_{\max} \leq d_{\max}$ 时, H 不是陷阱空洞, 换言之, 定理 1 成立. \square

3 陷阱空洞的修复算法

针对陷阱覆盖的特点,本文提出陷阱空洞修复算法(THPA).当利用THDA算法检测出了网络中的某个陷阱覆盖空洞后,相应的起始节点根据其接收到的数据包中弧段端点信息建立边权图 $G(V, E)$.顶点集 V 包含所有弧段端点,边集 E 中的每一条边都是集合 V 的二元子集 $\{v_i, v_j\}$,如果 $d(v_i, v_j) \geq d_{\max}$,则 v_i, v_j 之间存在一条边 e_{ij} ,边 e_{ij} 的权重 $w_{ij} = d(v_i, v_j)$.THPA算法将利用边权图 G 在相邻两边的角平分线上确定新节点的部署位置,具体的算法描述如下:

- 1) 运行THDA,检测出陷阱覆盖空洞的数目num和边界弧段;
- 2) If num > 0,则选择一个空洞及其相应的起始节点,建立该节点的边权图 $G(V, E)$;
- 3) 找出权重最大的边 e_{ij} , $v_l = \operatorname{argmax}\{\deg(v_i), \deg(v_j)\}$; //选择该边的两个顶点中度最大的顶点;
- 4) 找到 $\angle v_{l-1}v_l v_{l+1}$ 的平分线上与 v_l 相距 r_s 的位置点 P , v_{l-1} 和 v_{l+1} 是空洞边界线上直接与 v_l 相邻的弧段端点;
- 5) 在点 P 处添加新的节点;
- 6) 从 G 中删除边 e_{ij} 、顶点 v_l ,以及与顶点 v_l 相关的边;
- 7) If $E \neq \emptyset$,回到3);
- 8) End If //对应7);
- 9) End If //对应2);
- 10) 回到1);
- 11) End

运行过程中,遇到边的权重相等或者顶点的度相等的情况时,程序将任选其中一个.

本文提出的陷阱空洞检测算法是分布式算法,每个节点自主独立地决定是否成为起始节点,并主动发送数据包,所以可能会出现同一个空洞边缘上的多个节点同时成为起始节点的情况,这可能会造成空洞边界确定失败.为避免这一现象,为每个节点设定一个随机时延,节点在随机时延之后才决定是否成为起始节点.由于起始节点搜集了有关空洞的所有信息,最后的陷阱空洞判定是在该节点上分布式地进行,或者是将数据传输到基站由基站完成判定.因此,起始节点的计算量和通信量较大,耗能较多.为了均衡网络中节点能耗,防止节点因能量耗尽而失效,时延的长短主要依赖于节点剩余能量以及它的未被标志的未被覆盖有效弧段数,即 $\text{flag} = 1$ 的弧段数.随机时延如下式所示:

$$\tau_i = T_d \times \left(\alpha_1 \times \frac{1}{E_i} + \alpha_2 \times N_i + \xi_i \right), \quad (9)$$

剩余能量越多,未被覆盖的有效弧段越少,从而等待

时延越短.其中: T_d 是时间常数,其值远大于节点传输、发送单位数据所需的通信时间; E_i 表示节点 i 当前的剩余能量; N_i 表示节点 i 信息元素表中 $\text{flag} = 1$ 的弧段数; ξ_i 是取值范围在 $[0,1]$ 之间的随机因子,以防多个节点时延相同; α_1 和 α_2 是可调权重参数.在本文中,令 $T_d = 2\text{ s}$, $\alpha_1 = 2$, $\alpha_2 = 1$.

时延结束后,如果该节点的信息元素表中仍有未被标志的弧段,则它成为起始节点.

4 仿真分析

本文对上述算法进行了仿真实验,并与空洞修复算法CHPA^[4]进行性能对比. CHPA算法通过在空洞边界中垂线上某个特定位置上添加节点来逐步缩小空洞大小.传感器网络初始部署方式为模拟飞机随机抛撒,节点位置随机产生.设节点感知半径为5m,通信半径为10m.本文仿真均基于Matlab7.0的平台,并在主频2.9GHz、内存500M的计算机环境下运行.

4.1 仿真示例

图4的仿真中,网络监测区域面积为 $(50 \times 50)\text{m}^2$.图4(a)反映了网络的初始部署情况;图4(b)为陷阱空洞阈值 $d_{\max} = 10\text{ m}$ 时,运行THDA算法的结果:网络中存在的3个陷阱空洞都被正确地检测出来,其中1个为外部空洞,2个为内部空洞;图4(c)为 $d_{\max} = 15\text{ m}$ 时,运行THDA算法的结果:网络中只有2个陷阱空洞,其中1个为外部空洞,1个为内部空洞;图4(d)~图4(e)为 $d_{\max} = 10\text{ m}$ 时,利用THPA算法对陷阱空洞进行修复的结果,可见THPA算法运行了二轮之后网络中的陷阱空洞即可全被修复.

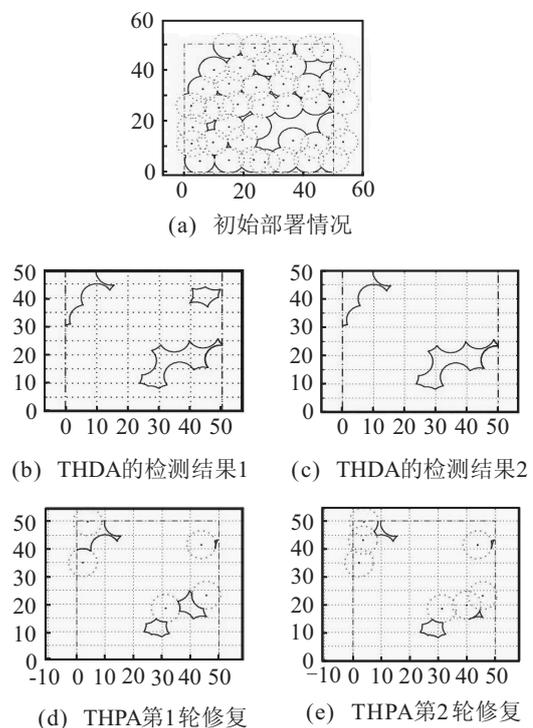


图4 陷阱空洞的检测与修复示例

由图4可见,阈值的取值直接影响了空洞的检测结果,所以在工程应用中,需要根据具体背景情况选取合适的阈值大小,即本文的阈值确定具有面向需求的特性.

4.2 收敛性分析

为了分析本文陷阱空洞检测和修复算法的收敛性,即算法终止运行的轮数,在保证网络连通性的前提下,针对5种不同的网络节点规模进行实验,并与CHPA算法进行了对比. CHPA算法与THPA算法的终止条件都是网络中不再存在陷阱空洞,换言之,本节分析的是这两种算法在相同修复效果下的收敛性. 在各个节点规模下随机生成10个拓扑结构,分别计算算法的收敛次数,取其平均值. 网络监测区域面积为 $(100 \times 100)\text{m}^2$,陷阱空洞直径阈值 $d_{\max} = 10\text{m}$.

表2 算法收敛性

节点数	100	150	200	250	300
THPA 轮数	12	13	9	11	9
CHPA 轮数	16	15	13	13	11

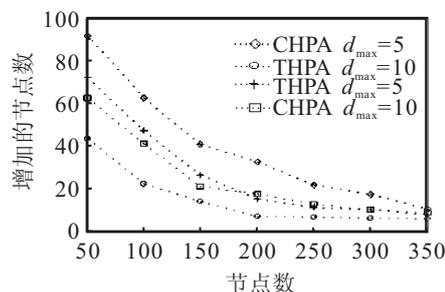
从表2可以看出,本文THPA算法的收敛性并不随着网络中节点规模的变化而发生显著变化. 这主要是当节点规模较小时,空洞面积较大,虽然所需要的新增节点变多,但每一轮的修复算法能够修复的空洞范围却比节点规模大时有所增加,所以算法轮数受到了限制. 与此同时,可以看出CHPA算法的收敛性却随着节点数的降低而变差,这是因为CHPA算法的修复思想更适用于区域覆盖,而并未充分利用陷阱覆盖的特点来修复陷阱空洞,当节点数变少、空洞增大时所需要的修复轮数自然随之增加. 另外还可以发现,在相同节点数的情况下,CHPA算法的修复轮数要多于THPA算法. 这一方面是因为CHPA算法没有充分利用陷阱覆盖的特点,修复策略不具备针对性;另一方面是因为并不是每一轮CHPA算法都能起到修复陷阱空洞的作用. 但是,THPA算法每一轮运行却都是有效的. 总之,本文的算法具有较好的收敛性,可以在较短的时间步长内完成网络中陷阱空洞的检测及修复.

4.3 新增节点数的比较

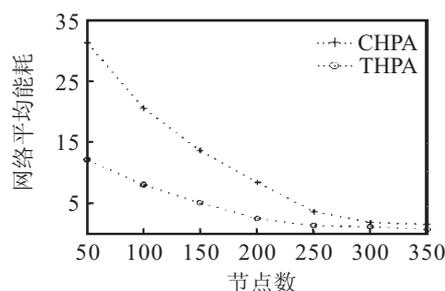
仿真中,网络监测区域面积为 $(100 \times 100)\text{m}^2$,网络中初始部署的节点数以间隔50递增到350.

图5(a)统计了在 $d_{\max} = 5\text{m}$, $d_{\max} = 10\text{m}$ 的情况下,分别利用THPA和CHPA修复陷阱空洞平均所需新增的节点数与网络中原有节点数的关系. 由图5(a)可见,随着节点数逐渐变大所需新增节点逐渐减少,阈值大时所需新增节点数少于 d_{\max} 小时所需新增节点数. 这是因为节点多、阈值大,网络中陷阱空洞就少,所需新增节点相应减少. 此外,THPA方法

比CHPA方法所需新增的节点数要少,这主要是由于CHPA方法选择的新增节点位置有一定的随机性,缺乏对空洞形状的整体考虑,而THPA算法充分利用了局部信息.



(a) 所需增加节点数对比



(b) 网络能耗对比

图5 算法性能对比图

4.4 网络能耗的比较

仿真中,网络监测区域面积为 $(100 \times 100)\text{m}^2$,网络中初始部署的节点数以间隔50递增到350,阈值 $d_{\max} = 10\text{m}$;每个节点发送和接收一个数据包的耗能分别为

$$E_{Tx}(k, d) = E_{\text{elec}} \times k + \varepsilon_{\text{amp}} \times k \times d^2,$$

$$E_{Rx}(k) = E_{\text{elec}} \times k.$$

其中: ε_{amp} 为信号放大器倍数, E_{elec} 为无线通讯电路的能耗, d 为信号传输距离, k 为传输数据包的长度. 在仿真实验中,取 $\varepsilon_{\text{amp}} = 10\text{pJ/bit/m}^2$, $E_{\text{elec}} = 50\text{nJ/bit}$.

图5(b)考察了利用THPA和CHPA方法修复空洞网络所需的平均能耗随着原有节点数增加的变化. 从图中可以看出,运行THPA的能耗要小于CHPA,在网络节点数少时差距更加明显. 这是因为,每当决定向网络中增加一个节点时,CHPA算法都会调用一次THDA算法用于判断所要添加的新节点是否位于空洞内部,这将导致网络额外消耗能量. 当空洞规模较大,需要增加的节点数较多时,这种额外消耗会显著增大. 相比之下,THPA方法有效地节约了网络能量,从而延长了网络寿命.

5 结 论

针对陷阱覆盖的特点,本文提出了陷阱空洞的检测和修复方法——THDA算法和THPA算法. 仿真实

验表明,这两个算法可以有效检测出网络中的陷阱空洞并修复它们,而且算法充分考虑了边界效应.在相同的网络环境下与现有修复算法相比,本文的算法更有优越性.本文算法可以有机地与“基于圆周覆盖的冗余节点判定算法”结合,形成一个集成了冗余节点休眠判定、覆盖空洞检测、覆盖空洞修复的综合性的传感器网络节点调度协议,这将是下一步的研究方向.

参考文献(References)

- [1] Karl H, Willing A. 无线传感器网络协议与体系结构[M]. 北京: 电子工业出版社, 2007: 314-316.
(Karl H, Willing A. Protocols and architectures for wireless sensor networks[M]. Beijing: Publishing House of Electronics Industry, 2007: 314-316.)
- [2] Jia J, Chen J, Chang G. Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm[J]. Computers and Mathematics with Applications, 2009, 57(11/12): 1756-1766.
- [3] Younis M, Akaya K. Strategies and techniques for node Placement in wireless sensor networks: A survey[J]. Ad Hoc Networks, 2008, 6(4): 621-655.
- [4] Yao J X, Zhang J Y, Kanno J, et al. Decentralized detection and patching of coverage holes in wireless sensor networks[C]. Proc of SPIE. Orlando: SPIE Press, 2009: 1-10.
- [5] Tan G, Jarv I S S A, Kermarrec A. Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks[J]. IEEE Trans on Mobile Computing, 2009, 8(6): 836-848.
- [6] Salah A, Rabie A R. Efficient deployment algorithms for mobile sensor networks[C]. Int Conf on AIS. Povia de Varzim: IEEE Press, 2010: 1-6.
- [7] 李明, 石为人. 虚拟力导向差分算法的异构移动传感器网络覆盖策略[J]. 仪器仪表学报, 2011, 32(5): 1043-1050.
(Li M, Shi W R. Virtual force-directed differential evolution algorithm based coverage-enhancing algorithm for heterogeneous mobile sensor networks[J]. Chinese J of Scientific Instrument, 2011, 32(5): 1043-1050.)
- [8] Wang G, Gao G, Porta T. Movement-assisted sensor deployment[C]. IEEE Trans on Mobile Computing. Barcelona: IEEE Press, 2006, 5(6): 640-652.
- [9] Paul B, Zizhan Z. Trap coverage: Allowing coverage holes of bounded diameter in wireless sensor networks[C]. Proc of IEEE INFOCOM. Rio de Janeiro: IEEE Press, 2009: 136-144.
- [10] Bai X L, Ding L, Teng J, et al. Directed coverage in wireless sensor networks: Concept and quality[C]. Proc of IEEE Int Conf on MASS. Macau: IEEE Press, 2009: 476-485.
- [11] Dogruel M, Ozgunzer U. Distributed coverage in wireless ad hoc and sensor networks by topological graph approaches[C]. Proc of IEEE Int Conf on ICDCS. Genova: IEEE Press, 2010: 106-115.
- [12] Miao H W, Chia C O, Wu X W, et al. Coverage-hole trap model in target tracking using distributed relay-robot network[C]. Proc of ACM Symposium on Applied Computing. Sierre: ACM Press, 2010: 1299-1304.
- [13] 傅质馨, 朱寅寅, 吴晓蓓. 能量有效的无线传感器网络节点调度算法[J]. 信息与控制, 2010, 39(6): 720-731.
(Fu Z X, Zhu Y Y, Wu X B. Energy-efficient sensor scheduling algorithm for wireless sensor network[J]. Information and Control, 2010, 39(6): 720-731.)

(上接第1809页)

- [11] Titterton D H, Weston J L. Strapdown inertial navigation technology[M]. London: Peter Peregrinus Ltd, 1997.
- [12] 太松月. 光纤陀螺捷联惯性导航系统标定测试技术研究[D]. 哈尔滨: 哈尔滨工程大学自动化学院, 2007.
(Tai S Y. Research on calibration and test technique of the FOG-strapdown inertial navigation system[D]. Harbin: College of Automation, Harbin Engineering University, 2007.)
- [13] 张志鑫, 夏金桥, 蔡春龙. 光纤陀螺标度因数分段标定的工程实现[J]. 中国惯性技术学报, 2008, 16(1): 99-103.
(Zhang Z X, Xia J Q, Cai C L. Engineering realization of calibrating FOG's scale factor in segments[J]. J of Chinese Inertial Technology, 2008, 16(1): 99-103.)