ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.3, March 2009, pp.644−659
doi: 10.3724/SP.J.1001.2009.03207

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

公平的有向传感器网络方向优化和节点调度算法*

温 俊+, 蒋 杰, 窦文华

(国防科学技术大学 计算机学院, 湖南 长沙 410073)

Equitable Direction Optimizing and Node Scheduling for Coverage in Directional Sensor Networks

WEN Jun+, JIANG Jie, DOU Wen-Hua

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

+ Corresponding author: E-mail: wenjun@nudt.edu.cn

Wen J, Jiang J, Dou WH. Equitable direction optimizing and node scheduling for coverage in directional sensor networks. *Journal of Software*, 2009,20(3):644−659. http://www.jos.org.cn/1000-9825/3207.htm

Abstract: To meet the coverage challenges arising in directional wireless sensor networks, this paper presents two distributed direction optimizing algorithms and a node scheduling: enhanced greedy algorithm (EGA), equitable direction optimization (EDO) and neighbors sensing scheduling (NSS) protocol. EGA algorithm optimizes direction merely according to the amount of uncovered targets. It is used as the baseline for comparison. EDO adjusts the directions of nodes to cover the critical targets superiorly and allocates sensing resource among nodes fairly to minimize the coverage differences between nodes. The utility function is introduced in EDO to assess the value of a direction contributed to overall networks sensing. The factors which affecting the utility are composed of the targets in per direction, the coverage of targets and the neighbor's decision of direction. EDO always selects the direction with the maximum utility as the working direction. NSS arranges all sensors into multiple cover sets and allows a node to join several cover sets. Through employing local cover set, NSS identifies a redundant node and decides whether it can sleep while taking residual energy to account. Nodes are activated in turn and the energy is consumed evenly to prolong the network life. The simulation shows that EDO outperforms EGA up to 30% in terms of critical coverage, and the combination of EDO and NSS prolongs the lifetime distinctly.
Key words: directional sensor networks; multiple cover sets; utility function; node schedule

摘 要: 为了解决有向传感器网络中点目标覆盖控制问题,分别提出了两种方向优化算法和一个节点调度协议:改进的贪婪(enhanced greedy algorithm,简称 EGA)、公平的方向优化(equitable direction optimization,简称 EDO)算法和邻居节点调度协议(neighbors sensing scheduling,NSS).EGA 基于覆盖最多未覆盖的目标数选取工作方向,其不足是可能忽略临界目标.EDO 优化算法调节节点的工作方向,优先覆盖临界目标,公平分配感知资源,减小目标覆盖度的差异,EDO 算法使用效用值评价每个方向对网络覆盖质量的贡献大小,影响效用值的因素包括每个方向上的目标数、目标的覆盖度和邻居节点的方向决策,EDO 总是选择效用值最大的方向作为工作方向.NSS 协议引入局部覆盖集的概念,通过局部覆盖集判断当前节点是否为冗余节点,并在考虑节点剩余能量时决定节点是否可以转为睡眠,调

* Supported by the National Natural Science Foundation of China under Grant No.60603061 (国家自然科学基金)
Received 2007-07-04; Accepted 2007-11-05

度协议允许一个节点加入多个覆盖集,覆盖集轮流工作,使网络生存期最大化.仿真实验结果表明,分布式的 EDO 算法比 EGA 算法具有更好的方向优化性能,临界目标的覆盖质量提高了 30%,同时明显地提高了网络生存期.

关键词: 有向传感器网络;多覆盖集;效用函数;节点调度

中图法分类号: TP393 **文献标识码**: A

无线传感器网络被广泛地应用于多个领域[1,2],如环境监测[3],生物观察[4],智能农业[5],工业生产[6],医疗保健[7],军事国防等诸多方面,由此而激发了大量的研究工作.由于不同的应用环境需要网络提供相应的覆盖质量保证,从而提出了多种覆盖控制问题.总的来说,覆盖控制问题可以分为 3 类:1) 点目标覆盖,即网络覆盖若干个离散的点目标.2) 区域覆盖,即网络覆盖整个或部分的目标区域.3) 栅栏覆盖,当目标穿越网络时,目标被发现和不被发现的最大概率.上述覆盖问题的求解有赖于合适的节点感知模型,其中最常用的感知模型有 2 种:1) 二元感知模型,即当目标点到节点的距离小于节点感知半径时,目标点被节点覆盖.2) 概率感知模型,目标点被节点感知的概率与目标点到节点的距离相关,距离越大,被感知的概率越小,距离越小,感知概率越大.但是上述两种感知模型有一个共同的假设:节点的感知能力与方向无关.

但是现实中却存在着一类传感器节点,它们在不同方向上具有不同的感知能力,如视频传感器、麦克风和红外传感器等,其感知能力限制在视角范围(field-of-view)内,即只有在工作方向上才可以有效感知,其他方向上则不能,这类传感器被称为有向的(directional).假设大量的有向传感器节点通过随机部署,分布在一组离散目标的周围,由于节点分布的随机性和感知能力的方向性导致某些目标没有被覆盖,或者其覆盖度很低,因而限制了网络的生存期,降低了感知资源的利用率.所以,研究人员希望在满足节点硬件成本受限的情况下,通过简单的装置允许节点在若干个方向上调节使其工作在最优方向上,提高目标覆盖质量,以延长网络生存期.如视频传感器中的镜头和成像部件安装在可以转动的云台上,但是基于成本和能量受限的考虑,不要求节点动态、实时地调节其工作方向.

本文研究的目标是如何优化节点的工作方向,公平分配感知资源,改善临界目标的覆盖度,提高网络覆盖资源的利用率,通过节点调度算法以延长网络的生存期,即有向传感器网络的多覆盖集问题.本文主要研究工作包括以下 3 个方面:

1) 首先提出了贪婪的方向优化算法.此算法选择覆盖最多数量的未被覆盖目标的方向作为工作方向,贪婪算法具有思想直观、计算复杂度低的优点,本文将用贪婪的方向优化算法与 EDO 算法作性能比较.

2) 提出了 EDO(equitable direction optimization)方向优化算法.此算法调节节点的工作方向,优先覆盖临界目标,公平分配感知资源,减小目标覆盖度的差异,EDO 算法使用效用值评价每个方向对网络覆盖质量的贡献大小.影响效用值的因素有:每个方向上的目标数、目标的覆盖度和邻居节点的决策信息,EDO 总是选择效用值最大的方向作为工作方向.

3) NSS(neighbors sensing scheduling)协议引入局部覆盖集的概念,通过局部覆盖集判断当前节点是否为冗余节点,并在考虑节点剩余能量时决定节点是否可以转为睡眠,调度协议允许一个节点加入多个覆盖集,覆盖集轮流工作,节点能量均匀消耗,使网络生存期最大化.

分布式、局部化的方向优化算法和调度协议具有较好的可扩展性.根据局部信息,EDO 算法最小化目标覆盖度差异,更公平地为目标分配覆盖资源,NSS 调度协议将网络节点划分出多个覆盖集,调度节点轮流工作,使能量均匀消耗,达到最大化网络生存期的目标.

本文第 1 节介绍当前主要的覆盖控制研究工作.第 2 节描述有向传感器网络的多覆盖集问题.第 3 节简要介绍贪婪的方向优化算法.第 4 节详细讨论 EDO 算法和算法特性.第 5 节叙述 NSS 协议及其实现.第 6 节使用实验仿真结果评估 EDO 和 NSS 协议性能.第 7 节是总结.

1 相关工作

覆盖控制是无线传感器网络研究领域中的一个基本问题,覆盖反映了网络对目标的感知质量,目前已经提出了许多有关覆盖控制的问题和解决方法[8].

Slijepcevic[9]首先提出了 SET-K COVER 问题,利用网络点的冗余特性,将所有节点划分为多个不相交的覆盖集,通过覆盖集轮流工作延长网络的生存期,互不相交的覆盖集越多,网络生存期越长.Berman[10]对上述思想提出了改进,不再要求每个节点只能属于一个集合.Ye[11]提出了 PEAS,是一种基于探测的分布式节点密度控制算法,节点睡眠一段时间后将被唤醒,在设定距离内探测是否存在活跃节点,若存在,则继续睡眠,若没有,则进入活跃状态,PEAS 的计算开销小.Tian[12]基于邻居节点的位置关系,判断某个节点的感知区域是否被其邻居节点覆盖而成为冗余节点,冗余节点可以进入低能耗的睡眠状态.Zhang[13]提出一种覆盖质量可调的覆盖协议(CCP),CCP 协议不依赖于节点的位置信息,而是通过估算节点间的距离计算覆盖质量.文献[13]认为,CCP 与 OGDC 的性能相当,但是更加灵活,当覆盖比例为 90%时,活跃节点数量可以减少 22%.Yan[14]提出了一种可以为目标区域不同点提供不同覆盖质量的节点调度算法,该算法使用离散网格点作为网络覆盖区域的近似,并使用网格点来判断目标区域是否被充分覆盖.Wang[15]设计了一种覆盖质量与生存期折衷的 pCover 协议,通过降低网络的覆盖质量换取更长的网络生存时间.与全覆盖的算法相比,pCover 在提供部分覆盖(~90%)时能够显著地(2.3~7 倍)延长网络的生存期.Xu[16]提出了分簇结构下解决簇内观测区域可靠覆盖问题的高效节能方案,通过 1-损毁最小覆盖间的轮流工作延长网络可用时间,通过增加一个冗余最小覆盖,屏蔽来自单个最小覆盖的节点失效,体现容错效果.Liu[17]提出了一个数学模型,只要已知监测范围和节点的感知半径的比值就可以计算出达到服务质量期望所需要的节点数,与大部分研究覆盖的文献不同,该模型不基于节点的位置信息.Gupta[18]从能量高效查询执行的角度讨论了无线传感器网络的连通覆盖问题(MCSC),构造的最小连通覆盖集是一个 NP 难的问题,Gupta 分别设计了集中式和分布式的贪婪算法.Jiang[19]为解决连通覆盖集问题,设计了一种基于目标区域 Voronoi 划分的集中式近似算法(CVT),当节点通信半径大于等于 2 倍的感知半径时,CVT 算法构造的节点集是连通的,当节点通信半径小于 2 倍感知半径时,设计了一种基于最小生成树的连通算法确保连通.

Cardei[20]考虑了离散目标点的能量高效覆盖问题,如何构建最大数量的不交节点集是一个 NP 完全问题,为此,他提出了一种基于混合整数规划的启发式算法,Cardei[21]对前一个问题进行了扩展,不再限制每个节点只能加入一个节点集合,而是允许一个节点同时加入多个节点集合.Sen[22]提出了节点最小距离受限情况下的覆盖控制问题,即假设存在一组感知点,一组可行的节点放置点和节点间最小距离的阈值,如何在可行放置点上部署最少数量的节点,使得所有感知点都被覆盖而任意两个节点间的距离不小于最小距离阈值.Sen 给出了该问题难解性的分析,并提出了两种启发式的算法.Wang[23]基于事件的漏报概率和误报概率定义节点的感知区域,通过多节点协调,在保证漏报率和误报率时扩大节点的感知区域.Gu[24]提出了一种分层的、非对称的一致化感知覆盖架构(uSense),uSense 架构中普通节点支持状态切换(generic switching),而一个独立的计算实体提供全局调度(global scheduling),它可以支持多种覆盖控制算法.Meguerdichian[25]首次提出了无线传感器网络的栅栏覆盖模型,栅栏覆盖模型考虑的是当移动目标沿着任意路径穿越无线传感器网络的部署区域时,网络检测到该移动目标的概率问题,该文提出了 MSP 和 MBP 路径,其中 MSP 路径上的每个点与最近节点的距离最小,而 MBP 路径上每一点都与最近节点距离最大.该文提出了分别基于 Voronoi 划分和 Delaunay 三角剖分的集中式算法.

关于有向传感器网络的覆盖问题,Ma[26]首次提出了有向传感器网络的概念,并且研究了有向传感器网络的部署策略和连通性问题.Tao[27]基于可旋转的有向感知模型提出了在给定覆盖度下节点数量估计方法,使用感知连通子图(SCSG)的方法将网络划分为多个部分,降低了算法的时间复杂度.Alhussein 等人[28]提出离散目标的最少节点最大覆盖(MCMS)问题,即寻找最少的节点覆盖最多的目标,该文给出了使用整数线性规划、集中式贪婪算法和分布式贪婪算法求解 MCMS 问题.Adriaens[29]研究了有向传感器网络中的栅栏覆盖问题,并且基于 FOV-Voronoi 图提出了寻找 MBP 的集中式算法.

与本文的研究比较相关的是 Cai 等人[30]提出的有向多覆盖集问题(MDCS).Cai 证明了 MDCS 是 NP 完全的,并应用混合整数规划方法提出了 Progressive,Prog-Resd 和 Feedback 集中化的启发式算法.本文的研究区别

于文献[30]的工作:1) 基于分布式和局部化的 EDO 方向优化算法只需要直接邻居的目标信息就可以优化节点方向,文献[30]中的 Progressive,Prog-Resd 和 Feedback 是全局化的、集中式的算法.2) NSS 调度协议不需要为每个覆盖集预先分配工作时间,而是通过局部覆盖集和节点剩余能量自发决定是否工作或睡眠,文献[30]为每个覆盖集预先计算出工作时间.3) 节点的工作方向优化后不再改变,由于 NSS 协议通过覆盖集多次轮流工作方式为每个覆盖集分配合适的工作时间,因而动态调整节点方向将增加计算和方向调整的能量开销.文献[30]由于使用集中式算法可以为每个覆盖集预先分配工作时间,因而允许同一节点在不同的覆盖集中工作在不同的方向上,而方向调整的能量开销增加得不多.

2 问题描述

有向节点把感知圆盘划分为 P 个感知区域,每个感知区域是一个以节点 u 为原点引出的两条射线和半径为 r_s 圆弧所围成的扇形区域,r_s 称为 u 的感知半径,扇形区域的夹角 θ 称为视角(FOV),向量 $\overrightarrow{o_j}$ 平分 θ 角,$\overrightarrow{o_j}$ 的指向称为 u 的感知方向,其值记为 α,大小满足 $0\le\alpha<2\pi$,其中 r_s 与 θ 是常量,如图 1 所示.有向感知模型可以使用形如 $\left(ID_u,r_s,\theta,\overrightarrow{o_{u,j}}\right)$ 的四元组来描述.

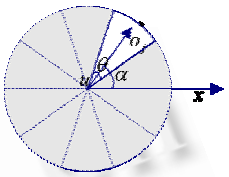

Fig.1 Directional sensor node
图 1 有向传感器节点

若目标 a 被节点 u 覆盖,则意味着 a 到 u 的距离小于 r_s 且向量 $\overrightarrow{a_u}$ 的方向位于 $\left(\alpha-\frac{\theta}{2},\alpha+\frac{\theta}{2}\right]$ 上.下文中若未特别指出,将不再区分节点、有向节点和传感器节点.下面简要介绍本文出现的常用记号:

S:	有向节点集合.	r_s:	节点的感知半径.
N:	总节点数.	P:	节点的方向数.
A:	目标集合.	$A_{u,p}$:	节点 u 在方向 p 上的目标集合.
q:	节点的感知视角.	$a_{u,p,i}$:	节点 u 在方向 p 上的第 i 个目标.
$o_{u,p}$:	节点 u 的第 p 个感知方向.	k:	目标的覆盖度.
M:	总目标数.	L:	节点的固有生存期.

定义 1. 若节点 u,v 间的距离小于 $2r_s$,则称 u,v 互为邻居节点.

定义 2. 若节点 u,v 分别存在方向 $o_{u,p}$ 和 $o_{v,l}$ 满足 $A_{u,p}\cap A_{v,l}\neq\varnothing$,即 u,v 分别在第 p 方向和第 l 方向上存在公共目标,则 u 的第 p 方向与 v 的第 l 方向关联.

定义 3. 当每个目标至少被一个活跃节点感知时,网络持续工作的时间称为网络生存期.

定义 4. 网络中覆盖度最小的目标称为临界目标,其覆盖度称为临界覆盖度.

定义 5. 有向覆盖集.

设 $A=\{a_1,\ldots,a_M\}$ 为有限个目标的集合,2^A 为 A 的子集的集合,$D_{ip}\in 2^A(0<i\le n,0<p\le P)$,$S=\{s_1,\ldots,s_P\}$ 为有向传感器节点集合且 $s_i\in\{D_{i1},\ldots,D_{iP}\}$.若 C 是 A 的一个有向覆盖集,对于 A 中的任意元素 a 至少存在一个 $D_{ip}\in C$ 满足 $a\in D_{ip}$ 且 $|s_i\cap C|\le 1$.

定义 6. 有向多覆盖集问题.

在给定 A 和 S 的情况下,找出一组 $C_1,\ldots,C_H$ 和对应的非负数 $t_1,\ldots,t_H$,使得 $\sum_{h=1}^{H}t_h$ 最大.其形式化的描述如下:

Maximize:
$$\sum_{h=1}^{H}t_h \tag{1}$$

subject to
$$\sum_{h=1}^{H}\sum_{p=1}^{P}x_{i,p,h}\cdot t_h\le L,\forall u_i\in S \tag{2}$$

$$\sum_{h=1}^{H}\sum_{p=1}^{P} x_{i,p,h} \le 1, \forall u_i \in S \tag{3}$$

$$\sum_{o_i \in C_h}^{a_m \in o_{i,p}} x_{i,p,h} \ge a_m \in A, h = 1,...,H, x_{i,p,h} \in \{0,1\} \tag{4}$$

目标函数(1)表示网络生存期最大化,约束(2)表明每个节点在各个覆盖集中的工作时间总和不超过节点的固有寿命,约束(3)表明每个节点只能工作在一个方向上,约束(4)要求每个目标至少被一个节点所覆盖.文献[30]已证明了有向传感器网络的多覆盖集问题是 NP 完全的.

本文分两步求解有向多覆盖集问题:1) 方向优化.调节节点工作在对网络覆盖贡献最大的方向上,优先考虑临界目标,当方向优化后,节点将不再动态地调整其工作方向.2) 节点调度.将节点划分为多个覆盖集,每个覆盖集轮流工作一段时间,使节点能量均匀消耗,最大化网络生存期.下面首先讨论贪婪的方向优化算法.

3 贪婪算法

贪婪算法(greedy algorithm)在优化有向传感器节点的工作方向时具有计算复杂度低的优点,因而我们首先简述贪婪优化算法的基本思想.贪婪算法中节点以覆盖目标数最多的方向作为工作方向,此时,节点不需要与邻居节点交换信息,但是由于没有考虑目标的覆盖状况,容易遗漏孤立的目标.

改进的贪婪算法(enhanced greedy algorithm,简称 EGA)对基本的贪婪算法作了改进,其基本思想是:称覆盖未被覆盖的目标为有效覆盖,那么节点每次找出最多有效覆盖的方向作为工作方向.算法过程是:1) 节点检测每个方向上未被覆盖的目标;2) 选择最多有效覆盖的方向作为工作方向并根据有效覆盖的数量确定节点优先级;3) 高优先级的节点向邻居节点广播选择的方向和覆盖的目标;4) 低优先级的节点记录目标状态并重新选择工作方向;5) 当所有节点都已优化了工作方向时,算法终止.当网络冗余部署而存在多个不交覆盖集时,贪婪算法每得到一个覆盖集后需要将目标重置为未覆盖状态后才能继续执行,直到所有节点都已优化.

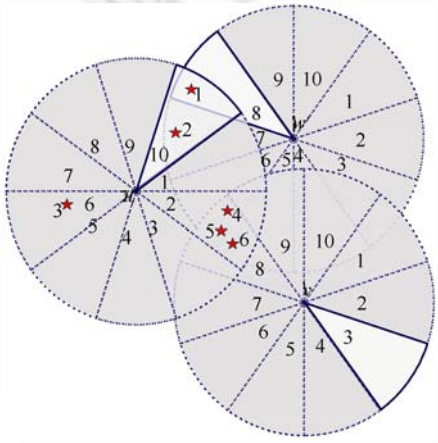

Fig.2 An instance of directional coverage

图 2 有向节点的目标覆盖

基本的贪婪算法总是基于节点自身的覆盖信息独立地选取工作方向,只需比较自己在各个方向上覆盖的目标数,因此计算简单.由于不与邻居节点交换消息,消息开销小.但是缺点也较为明显:不能与邻居节点协调地为目标合理分配覆盖资源.例如,对图 2 的情形,基本的贪婪算法使得节点 u 选取 $o_{u,2}$,节点 v 选取 $o_{v,8}$,节点 w 选取 $o_{w,6}$,但是目标 a_1,a_2,a_3 却不被任何节点所覆盖.改进的贪婪算法需要向邻居节点通告自己的方向决策,增加了消息开销,优点是改善了覆盖资源的分配.例如,对于图 2 的情形,改进的贪婪算法在选取方向时其中一个最优情况是节点 u 选取 $o_{u,10}$,节点 v 选取 $o_{v,8}$,节点 w 选取 $o_{w,7}$,只有目标 a_3 不被覆盖,最差的情况是节点 u 选取 $o_{u,8}$,节点 v 选取 $o_{v,8}$,节点 w 选取 $o_{w,7}$ 或 $o_{w,8}$,则目标 a_3 和 a_1 或 a_3 和 a_8 两个目标不被覆盖,因而改进的贪婪算法在适当增加消息的开销下却改善了目标的覆盖质量.总之,两种贪婪的优化算法具有简单、计算复杂度低的优点,但还存在不足:1) 结果通常局部最优.2) 贪婪算法没有考虑目标的覆盖度,尤其对于孤立的临界目标没有优先分配感知资源,从而容易成为限制网络生存期的瓶颈.3) 算法需要重置目标的覆盖状态.为了改善贪婪算法的不足,EDO 算法优先考虑了临界目标,不必重置目标的覆盖状态,避免了多次迭代.

4 EDO 算法

假设节点的生存期为 1,那么网络生存期总是小于或等于临界目标的覆盖度,即临界目标的覆盖度为网络生存期的上界,临界目标的覆盖度越小,网络生存期越短.EDO 方向优化算法可以提高覆盖资源的利用率,公平

地分配感知资源,改善临界目标的覆盖度,下面详细讨论 EDO 算法中的主要思想和技术.

4.1 节点状态

EDO 算法的工作过程是:网络部署后,所有节点扫描各个方向上的目标信息,与邻居节点交换目标信息后,估算每个方向的效用值,选取效用值最大的方向作为工作方向,最后将自己的方向决策通知邻居节点.因而在 EDO 算法中,节点存在 4 种可能的状态:1) 目标检测(target testing),节点获取各个方向上的目标信息;2) 消息交换(message exchanging),节点广播目标信息和记录邻居消息;3) 方向决策(decision making),计算各个方向上的效用值,选择效用值最大的方向作为工作方向;4) 决策公告(decision informing),向邻居节点广播自己的方向决策.

4.2 目标-方向映射表

当网络部署后,节点逐一检测每个工作方向,获取每个工作方向上的目标信息,建立目标-方向映射表.目标信息应包括可以唯一标识目标的标识符或其坐标.节点向其邻居节点广播目标-方向映射表,同时接收邻居节点的广播消息.通过合并,每个节点建立了一份局部的目标-方向映射表,它反映了在自己与邻居节点范围内目标的覆盖情况.为了减少计算量,在合并映射表时只考虑本地映射表中已存在的目标就足够了.表 1 描述了在图 2 所示情况下,各个节点建立的目标-方向映射表.

Table 1 Target-Direction mapping in individual node

表 1 节点建立的目标-方向映射表

u	a_1	$o_{u,10}$ $o_{w,8}$	v	a_1	$\varnothing$	w	a_1	$o_{w,8}$ $o_{u,10}$
	a_2	$o_{u,10}$ $o_{w,7}$		a_2	$\varnothing$		a_2	$o_{w,7}$ $o_{u,10}$
	a_3	$o_{u,6}$		a_3	$\varnothing$		a_3	$\varnothing$
	a_4	$o_{u,2}$ $o_{v,8}$ $o_{w,6}$		a_4	$o_{v,8}$ $o_{u,2}$ $o_{w,6}$		a_4	$o_{w,6}$ $o_{u,2}$ $o_{v,8}$
	a_5	$o_{u,2}$ $o_{v,8}$ $o_{w,6}$		a_5	$o_{v,8}$ $o_{u,2}$ $o_{w,6}$		a_5	$o_{w,6}$ $o_{u,2}$ $o_{v,8}$
	a_6	$o_{u,2}$ $o_{v,8}$ $o_{w,5}$		a_6	$o_{v,8}$ $o_{u,2}$ $o_{w,5}$		a_6	$o_{w,5}$ $o_{u,2}$ $o_{v,8}$

4.3 效用计算

在获取了目标-方向映射表后,每个节点使用效用函数分别计算每个方向上的效用值,效用值表达了节点覆盖某个目标或目标集合时,节点对网络覆盖质量的贡献.式(5)给出了计算单个目标 a 的效用值表达式,式(6)是节点计算方向 p 的效用值表达式.

$$U(a)=M^{-(k+k'-1)} \tag{5}$$

$$U(u,O_p)=\begin{cases}\Sigma^U(A_{u,o_p}), A_{u,o_p}\neq\varnothing\\ 0, \quad A_{u,o_p}\neq\varnothing\end{cases} \tag{6}$$

式(5)左边表示目标 a 的效用值.式(5)右边中的$(k+k')$称为目标的覆盖度,其中 k 是已决策节点,为目标 a 提供的覆盖度,即被 k 个已决策的节点覆盖,k'为未决策节点提供的可能覆盖度,即被 k'个尚未决策节点的某个方向所覆盖.式(6)左边是 u 在方向 p 上的效用值,右边是该方向的目标集合 $A_{u,p}$ 上所有的目标效用之和,即 u 在 p 方向上的效用值是在此方向上所有目标的效用值之和,当某个方向上没有目标时,其效用值为 0.

影响效用值的因素有:1) 目标覆盖度.为了公平地分配感知资源,节点应优先工作在含有临界目标的方向上,反映在效用函数上就是当节点覆盖临界目标时的效用要大于覆盖其他目标时获得的效用.2) 目标数量.对于一个节点,若在两个方向存在覆盖度相同而数量不同的目标,显然节点工作在目标数较多的方向上的效用大于目标数少的方向上的效用.3) 邻居节点的方向决策.当邻居节点没有做出方向决策时,邻居节点被认为具有相同的概率工作在任何方向上,反映在效用函数中 k'的取值.当邻居节点决策后,若决策方向包含此目标或目标集合时,那么目标效用函数中的 $k'-1,k$ 增加 1,而$(k+k')$保持不变;若决策方向不包含此目标或目标集合时,那么 $k'-1,k$ 不变,而$(k+k')$变小,那么目标的效用值变大.例如,假设图 2 中的节点都未决策,那么 w 节点在 $o_{w,6}$ 上有 a_4 和 a_5 两个目标,而且 $k=0,k'=3$,则此时 $U(w,o_6)=2M^{-2}$.若 u 现在决定工作在 $o_{u,10}$ 上,那么 a_4 和 a_5 的 $k=0,k'=2$,则此时 $U(w,o_6)=2M^{-1}$.

4.4 覆盖冲突

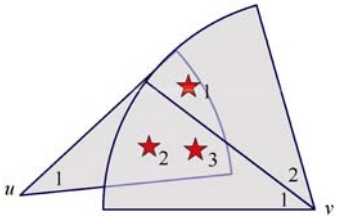

Fig.3 An instance of coverage collision

图 3 覆盖冲突

现在计算图 3 中节点在各个方向上的效用值,其中 u,v 都未作方向决策.节点 v 的 $o_{v,1}$ 方向上存在两个目标,其覆盖度为 2,因而 v 在 $o_{v,1}$ 方向上的效用为 $2M^{-1}$.v 的 $o_{v,2}$ 方向上有一个目标,其覆盖度为 2,因而 v 的 $o_{v,2}$ 方向上的效用为 M^{-1}.节点 u 的 $o_{u,1}$ 方向上有 3 个目标,其覆盖度都是 2,但是 v 在任意时刻只能工作在一个方向上,即$\{a_1\}$与$\{a_2,a_3\}$的覆盖度不能同时达到 2,即节点某个方向上的目标集合被另一个节点的多个方向划分(与多个方向关联)时出现了覆盖冲突.为了计算覆盖冲突时的效用,可以使用分时覆盖(one time one choice)的方法:若 v 选取在 $o_{v,1}$ 上工作,那么 v 必定无法覆盖$\{a_1\}$,此时 $U(u,o_1)=1+2M^{-1}$,当 v 选取在 $o_{v,2}$ 上工作,那么 v 必定无法覆盖$\{a_2,a_3\}$,此时 $U(u,o_1)=2+M^{-1}$.假设 v 以相同概率工作在各个关联方向上,那么对于图 3 中的节点 v,以概率 0.5 工作在 $o_{v,1}$ 上,以概率(1–0.5)工作在 $o_{v,2}$ 上,因而 u 在 $o_{u,1}$ 方向上的值取为$(1.5+1.5M^{-1})$.分时覆盖的方法可以计算冲突覆盖时的效用值,但是当节点某个方向上的目标存在多个冲突覆盖时(目标集合与多个邻居节点存在冲突覆盖,如节点和目标都密集分布时),分时覆盖方法搜索的空间将呈指数增长,计算的复杂度快速增加.

等效覆盖(equivalent coverage)的方法可以降低计算冲突覆盖时效用值的复杂度.等效覆盖方法的主要思想是为涉及覆盖冲突的目标计算出一个等效覆盖度,再根据式(5)和式(6)计算此方向的效用值.通常,等效覆盖度的值小于 1,这是由于引起覆盖冲突的节点不能为涉及覆盖冲突的目标的覆盖度同时增加 1,若假设此节点为这些目标提供的等效覆盖度为 k_e,那么式(5)和式(6)在使用 k_e 计算的效用值应等于分时覆盖方法计算的效用值,因而等效覆盖的方法相对于分时覆盖的方法减小了计算量.等效覆盖度则反映了引起覆盖冲突的节点对目标集合提供的一致的(uniform)感知质量,它体现了一种平均的思想,即从目标集合角度来看,导致冲突覆盖的节点给目标集合提供了一个折衷的覆盖度而不考虑节点此时是否存在冲突覆盖.等效覆盖方法计算的效用值与分时覆盖方法计算的保持相等.对于等效覆盖度有下列定理成立.

定理 1. 若节点 u 某一方向与节点 v 的 $f(f\le P)$个方向关联,则 v 提供 u 在此方向上的等效覆盖度 k_e 满足:

$$k_e=\log_M\frac{fM}{fM-M+1} \tag{7}$$

当目标数远大于 1 时,

$$k_e\approx\log_M\frac{f}{f-1} \tag{8}$$

证明:设 u 在某个方向上的目标集合为 A,A 中的目标落在 v 的 f 个方向上,那么位于每个方向上的目标满足 $A=A_1\cup A_2\cup\ldots\cup A_f$.若 $|A_j|$表示 A_j 的目标数,$k_{j,i}$ 表示不考虑 v 时 A_j 上的第 i 个目标的覆盖度,则 u 在此方向上的效用满足下式:

$$\begin{aligned}&\sum_{j=1}^{f}\sum_{i=1}^{|A_j|}M^{-(k_{j,i}+k_e-1)}=\frac{1}{f}\left(\sum_{i=1}^{|A_j|}M^{-(k_{1,i}+1-1)}+\sum_{j=1}^{f}\sum_{i=1}^{|A_j|}M^{-(k_{j,i}-1)}\right)+\\&\frac{1}{f}\left(\sum_{i=1}^{|A_2|}M^{-(k_{2,i}+1-1)}+\sum_{i=1}^{|A_1|}M^{-(k_{1,i}-1)}+\sum_{j=3}^{f}\sum_{i=1}^{|A_j|}M^{-(k_{j,i}-1)}\right)+\ldots+\frac{1}{f}\left(\sum_{i=1}^{|A_f|}M^{-(k_{f,i}+1-1)}+\sum_{j=1}^{f}\sum_{i}^{|A_j|}M^{-(k_{j,i}-1)}\right)\end{aligned} \tag{9}$$

式(9)的左边是使用等效覆盖度计算的效用.式(9)右边共有 f 项,每一项表示 v 工作在其中一个关联方向时的效用分量,通过合并和化简上式得到下面的形式,

$$M^{-k_e}\sum_{j=1}^{f}\sum_{i=1}^{a_j}M^{-(k_{j,i}-1)}=\frac{1}{f}\left(\sum_{j=1}^{f}\sum_{i=1}^{a_j}M^{-(k_{j,i}+1-1)}+f\sum_{j=1}^{f}\sum_{i=1}^{a_j}M^{-(k_{j,i}-1)}-\sum_{j=1}^{f}\sum_{i=1}^{a_j}M^{-(k_{j,i}-1)}\right) \tag{10}$$

$$M^{-k_e}=\frac{1}{f}\left(M^{-1}+f-1\right) \tag{11}$$

因而可以求得

$$k_e = \log_M \frac{fM}{fM - M + 1} \tag{12}$$

当 $M>>1$ 时,可以忽略式(12)右边项中的 1,因而得到

$$k_e \approx \log_M \frac{f}{f - 1} \tag{13}$$

□

假设如图 2 所示的是节点处于完成目标检测和消息交换,还未作方向决策的时刻,则表 2 给出了 u,v 和 w 在各个方向上的效用值.

Table 2 Utilities calculated by nodes before making decision

表 2 节点未决策时计算的效用值

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
u	0	$3M^{-2}$	0	0	0	1	0	0	0	$1+M^{-1}$
v	0	0	0	0	0	0	0	$3M^{-2}$	0	0
w	0	0	0	0	M^{-2}	$2M^{-2}$	M^{-1}	M^{-1}	0	0

由于邻居节点的决策会影响效用值.假设图 2 中的 u 工作在 $o_{u,6}$ 上,那么 v 和 w 在计算效用值时不再计入 u 对目标覆盖度的贡献,此时,v 和 w 在各个方向山的效用值见表 3.表 4 则是伪码描述的效用计算过程.

Table 3 Utilities calculated by v and w when u works at $o_{u,6}$

表 3 u 工作在 $o_{u,6}$ 上时,v 和 w 计算的效用值

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
v	0	0	0	0	0	0	0	$3M^{-1}$	0	0
w	0	0	0	0	M^{-1}	$2M^{-1}$	1	1	0	0

4.5 回退决策

效用值越大,表明此方向上目标的覆盖度越小或者目标越多,因而节点选择效用值最大的方向作为工作方向并将此决策和效用值向邻居节点广播.邻居节点也是采用上述原则确定工作方向,然而,不同的节点其最大效用值可能不同,而拥有较大效用值的节点具有比其他节点更高的优先级,因而若每个节点都立即广播其决策,将导致大量低优先级的决策失效,从而增加了通信开销.节点可以通过决策回退机制避免低优先级的节点率先广播其决策.决策回退机制设置一个决策计时器 t_d,初始值设为 t_1e^{-U},U 是节点的最大效用值,如果效用值越大,则回退的时间越短,节点可以优先广播其决策.回退决策机制的基本工作过程为:1) 启动决策计时器;2) 在决策计时器归零之前,若收到相同或高优先级的消息,则记录此消息,检查和重新计算自己在各个方向上的效用值并重新选择工作方向,重置决策计时器.3) 决策计时器归零时,向邻居节点广播自己的决策和效用值并调整工作方向.4) 继续接收和记录邻居节点的决策信息,直到所有邻居都已决策.

回退决策解决了节点决策的优先级问题,但是在应用中可能出现反序决策现象,假设节点 u,v,w 的效用值存在如下关系:$U(u)>U(v)>U(w)$;而且 u,v 互为邻居;v,w 互为邻居但是 u,w 不相邻.由于 u 的决策计时器最先归零而广播决策,v 更新其效用值,若其新效用值仍大于 w 的效用值,但是由于 v 重置了计时器而 w 却没有,因而 w 可能先于 v 广播决策,从而导致了反序决策.

为了解决反序决策问题,需要改进决策回退机制:1) 决策计时器归零的节点首先向邻居发送决策请求消息,决策请求消息包含着决策方向和效用值信息;2) 收到决策请求消息的邻居节点,暂停决策计时器,检查效用值,若效用值比本地效用值大,则不响应,若比本地效用值小,则回复否决消息;3) 若发送决策请求消息的节点在等待 t_r 内没有收到否决消息,则在 t_v 时间间隔内发送生效消息;若在 t_r 内收到任何一条否决消息,则取消决策,重置决策计时器;4) 收到决策请求后的邻居节点等待(t_r+t_v)时间,若在此等待时间内未收到决策生效消息,则继续倒计时,若收到决策生效消息,则更新效用值,重置决策计时器.

如图 4 所示,u,v,w 和 q 四个节点其效用值满足 $U(u)>U(v)>U(w)>U(q)$,其中 u,v 互为邻居;v,w 和 q 互为邻居,但是 u 与 w,q 不相邻.在 t_1 时刻,u 的决策计时器率先倒计时结束,广播决策请求消息,由于其效用值最大因而没

有收到否决消息,在 t_2 时刻,广播决策生效消息并调整工作方向,v 节点收到 u 的决策生效消息后,使用更新后的效用值重置决策计时器.在 t_3 时刻,w 结束决策等待,广播决策请求消息,v 和 q 暂停倒计时.w 在(t_4-t_3)内收到了 v 的否决消息,在 t_4 时刻,决定取消当前决策.由于直到 t_5 时刻,没有收到 w 的决策生效消息,因而 v 和 q 继续倒计时,而 w 的决策计时器被重置.在 t_6 时刻,v 倒计时结束,广播确认请求,由于其效用值最大且没有收到否决消息,因而在 t_7 时刻,发送决策生效消息,w 与 q 更新效用值并重置决策计时器.

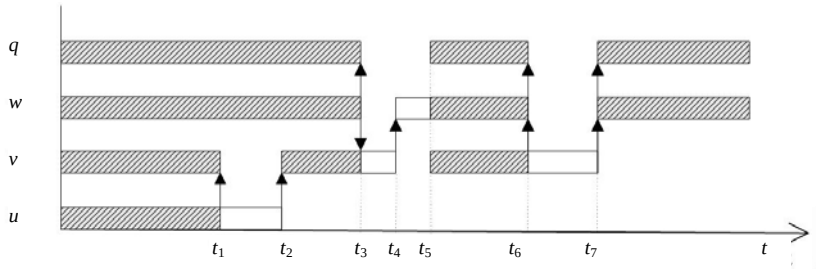

Fig.4 Improved backoff decision

图 4 改进后的回退决策过程

表 5 是回退决策过程的伪码描述,其中语句 4~18 是判断收到的消息类型,其中语句 6、11、13 分别表示接收到了决策请求消息、否决消息和决策生效消息.语句 19~34 根据定时器的状态实现决策回退.由于语句 15 和语句 29 改变了决策列表的状态,当语句 16 和语句 30 判断决策列表中已包含了自身节点和所有邻居节点,则 MakeDecision 过程终止.

4.6 算法复杂性

下面通过时间复杂性的分析,比较 3 种方向优化算法的性能.基本的贪婪优化算法其计算复杂性最低,首先是检测每个方向覆盖的目标,其时间开销为 $O(P)$,接着在 P 个方向上找出覆盖目标数最大的方向,则时间开销为 $O(\lg P)$,最终,算法的时间复杂性为 $O(P)$.改进的贪婪算法中的第 1 步是检测每个方向覆盖的目标,则时间开销为 $O(P)$,第 2 步是查找最大有效覆盖的方向和确定优先级,若以最大有效覆盖数为优先级,则其时间复杂度为 $O(\lg P)$.理想决策过程是按照最大有效覆盖数从大到小的顺序依次广播其方向决策,由于每当一个节点作出决策其他节点需要重新检查和选取最大有效覆盖数,那么整个算法的时间复杂度为 $O(NP\lg P)$.事实上,改进的贪婪算法的时间复杂性与其采用的决策机制有关,上面给出的是在理想情况下的时间开销.现在讨论由表 4 和表 5 描述的 EDO 算法的时间复杂性.表 5 中只有语句 3 一个循环,循环的规模与接收到的消息相关.循环内的语句 6~12 和语句 19~34 都是常数时间的操作,语句 14 的时间开销为 $O(P\lg P)$,由于任意节点的最多邻居节点数不超过$(N-1)$,在最差情况下,即由于偶然因素导致低优先级的节点总是先于高优先级的节点倒计时结束,节点收到的消息数不超过 $N(N-1)$,其中包括$(N-1)$个决策生效消息则进入语句 14,因而算法的时间复杂性为 $O((N(N-1)-(N-1))+(N-1)P\lg P)$,由于 $P<<N$,最终 EDO 的时间开销为 $O(N^2)$.因而最差情况下的 EDO 算法仍是多项式时间的,在理想情况下,即回退决策机制可以保证每个节点按照最大效用值由大到小的顺序依次通告其方向决策,则其时间开销为 $O(NP\lg P)$.

Table 4 Target testing and utility evaluating procedure

表 4 目标检测和效用计算过程

Procedure	EvaluateUtility
1	Initialize all the target set for each direction to *null*
2	Traverse all directions and add the targets which locate in the direction p to the set A_p
3	Send a coverage message containing the targets, direction information, initialize the message timer by t_0
4	**while**(t_0>0)
5	**if**(coverage messages arrive from neighbors)
6	Evaluate the utility values for all incident directions
7	Reset the message timer by t_0
8	**end if**
9	**end while**

Table 5 Backoff decision procedure

表 5 决策回退过程

Procedure	MakeDecision
1	Initialize the DECISIONLIST by *null*
2	Set and activate t_d with t_1e^{-U}
3	**while**(1)
4	**if**(a message arrives from a neighbor)
5	**switch**(message)
6	**case**(a decision request):
7	Pause t_d, set and activate t_w with (t_r+t_v)
8	**if**(the utility in message is less than its self's)
9	Send a denial message
10	**end if**
11	**case**(a decision denial for self):
12	Reset t_d with t_1e^{-U}
13	**case**(a decision startup):
14	Update the utility values for each direction and reset the t_d with t_1e^{-U}
15	Add the decision to the tail of DECISIONLIST
16	if(DECISIONLIST contains all neighbors and itself) return DECISIONLIST
17	**end if**
18	**end switch**
19	**if**(t_d pauses) **and** (t_w= =0)
20	Restart t_d
21	**else**
22	**if**(t_d= =0)
23	Send a decision request message
24	Set and activate t_w with (t_r+t_v),Set and activate t_r
25	**if**(t_r= =0) **and** (t_v is not active)
26	Set and activate t_v
27	**else**
28	**if**(t_v= =0)
29	Send a decision startup message and add itself to DECISIONLIST
30	**if**(DECISIONLIST contains all neighbors and itself) return DECISIONLIST
31	**end if**
32	**end if**
33	**end if**
34	**end if**
35	**end while**

4.7 消息开销

EDO 算法总的消息开销包括节点交换覆盖信息时的和节点决策时的消息开销.若分别使用 lg$N$,lg$P$ 和 lg$M$ 长度的比特信息表示节点、方向和目标,那么节点在交换覆盖信息时总的消息开销不能够超过(NlgNP+Mlg$M$)比特.节点决策时,在理想情况下,即节点按照效用值大小确定的优先级依次决策,那么节点决策的消息开销为(NlgNP)比特.在最差情况下,即由于偶然因素,导致低优先级的节点总是先于高优先级的节点倒计时结束,那么,节点决策的消息开销为 $\left(\frac{N(N-1)\lg NP}{2}+\frac{(N-1)(N-2)b}{2}\right)$ 比特,其中 b 为决策否决消息的长度.

4.8 EDO算法性质

定理 2. EDO 在有限时间内终止.

证明:初始时刻,节点按照效用值从大到小的顺序排列为 $U_1>U_2>\ldots>U_n$.在节点 1 决策延迟结束之前,任何一个邻居节点的决策请求都被其否决,因而只有节点 1 才能首先做出决策而且不再改变自己的决策.当节点 1 决策生效后,剩余节点更新效用值,再按照从大到小的顺序又被重新排列,新的最大效用值节点接着做出决策并保持其决策不受邻居决策影响,依此类推,直到所有节点做出决策. □

定理 3. 若网络中存在唯一的临界目标,则覆盖临界目标的方向效用值最大.

证明:若 a^*是临界目标,则对于任一目标 a 存在 $(k_{a^*}+k'_{a^*})<(k_a+k'_a)$,其效用值分别为

$$U(a^*)=M^{-(k_{a^*}+k'_{a^*}-1)} \tag{14}$$

$$U(a)=M^{-(k+k'-1)} \tag{15}$$

对于任意节点 u 在不覆盖临界目标的方向 p 上的效用为 $U(u,o_p)=\Sigma U(A_{u,o_p})$ 且 $|A_{u,p}|<M$,则

$$\Sigma U\left(A_{u,p}\right)<\sum_{i=1}^{M}M^{-(k_i+k_i'-1)}\le M\cdot M^{-\left((k_{a^*}+k_{a^*}+1)-1\right)}=M^{-\left(k_{a^*}+k_{a^*}-1\right)} \tag{16}$$

□

定理 3 保证了即使是孤立的临界目标,覆盖它的节点也具有最高的方向决策优先级.

5 NSS 调度协议

调度协议把节点集合划分为多个覆盖集并使其轮流工作,从而延长网络的生存期.每个覆盖集满足 3 个约束条件:1) 每个节点在一个覆盖集中只出现一次;2) 每个节点的总工作时间不超过节点的固有生存周期;3) 覆盖所有目标.

NSS 协议基本思想是:1) 把整个网络生存期划分为多个时间间隔,工作一个时间间隔称为一轮(round),每轮时间长度远小于节点的生存期;2) 睡眠节点依据自己的能量状态决定在下一轮是否转为活跃状态.在前一轮末,活跃/睡眠节点交换剩余能量信息,从而决定是否在下一轮转为活跃;3) 活跃节点依据被唤醒节点的目标覆盖关系判断是否为冗余节点,冗余节点在下一轮转为睡眠.

剩余能量和节点对目标的覆盖关系决定了每轮工作周期的覆盖集,由于在每轮工作周期内覆盖集仅工作一小段时间,通过多次轮流工作达到合理分配覆盖集工作时间的目标.下面讨论初始覆盖集的建立、局部覆盖集和冗余节点的判定:1) 初始覆盖集的建立,EDO 算法终止时所有节点都处于活跃状态,节点执行贪婪算法得到初始覆盖集,即每次选择覆盖最多、未被覆盖的目标的节点作为活跃节点,当初始覆盖集建立后,其他节点则转入睡眠状态.2) 局部覆盖集的划分,EDO 算法终止时,每个节点覆盖了一组目标,这些目标可能被另一个或多个邻居节点所覆盖,那么节点根据邻居节点对这组目标的覆盖关系把邻居节点划分为多个局部覆盖集并为每个局部覆盖集分配一个 ID.3) 冗余节点的判定,被唤醒的邻居节点广播含有自身剩余能量的消息.通过比较,当前节点记录剩余能量比自身多的消息,并将对应的节点加入其所属的局部覆盖集列表中,若得到一个包含任意一个局部覆盖集所有成员的列表,则当前节点向邻居节点广播入选节点的消息,则此局部覆盖集开始工作,当前节点就变为冗余节点,可以进入睡眠状态.

下面简要描述 NSS 协议工作过程:1) 建立初始覆盖集并工作一个间隔时间;2) 活跃节点在每轮工作周期末向邻居节点广播包含有剩余能量的消息;3) 处于睡眠状态的邻居节点在每轮结束前醒来,并接收活跃节点的剩余能量消息,比较自己与活跃节点的剩余能量.若睡眠节点的剩余能量比活跃节点少,则决定在下一轮继续睡眠;若睡眠节点的剩余能量比活跃节点多,且超过设定的阈值,则发送包含有自身剩余能量的响应消息;4) 活跃节点把发送响应消息的邻居节点分配到相应的局部覆盖集中,若活跃节点收到了一个完整的局部覆盖集响应消息,则向邻居节点广播入选的局部覆盖集 ID 消息;若活跃节点未能收到一个完整的局部覆盖集节点响应消息,则活跃节点在下一轮仍将继续工作,未能“入选”的节点继续睡眠.

6 性能评估

为了评估算法性能,我们用 C++语言设计了一个仿真程序,实验环境是一台运行 Windows XP Sp2 系统,配置了 Pentium IV 3.0GHz 处理器,主存为 1G 的工作站.在仿真实验中,假设目标区域是一个 500×500 的矩形区域,节点和目标随机分布于目标区域内,通过改变节点数、方向数、感知半径和目标数评估 EDO 算法以及 NSS 调度协议在不同环境下的性能.主要的性能指标是目标的最大覆盖度、平均覆盖度、临界覆盖度和网络生存期.

6.1 EGA与EDO比较

前面讨论了 EGA 和 EDO 方向优化算法,EGA 依据当前覆盖最多未覆盖目标的数量选择工作方向,EDO 增加了目标覆盖度的考虑,因而临界目标被优先覆盖.假设节点和目标随机分布于目标区域,当节点感知半径为

100,方向数为 3,目标数为 50 时,比较不同节点数下目标最大、平均和临界覆盖度.表 6 分别列出了网络部署后以及两个方向优化算法关于目标覆盖度与节点数的数值关系,从数值上可以看出:1) 节点随机部署后,出现未被覆盖的目标概率很大,EGA 和 EDO 可以显著减少未被覆盖目标的数量.2) 临界目标覆盖度比最大的目标覆盖度和平均的目标覆盖度小很多.3) EGA 和 EDO 的临界覆盖度随节点数的增加而增大,临界覆盖度的增长速度快于最大和平均覆盖度的增长速度.4) EDO 的临界覆盖度比 EGA 算法的临界覆盖度改善了近 30%.

Table 6 Target coverage of after placement, EGA and EDO

表 6 网络部署后,EGA 及 EDO 的目标覆盖度

Sensors	Coverage after placed			Coverage of EGA			Coverage of EDO		
	Max	Mean	Critical	Max	Mean	Critical	Max	Mean	Critical
45	4.840	1.568	0.000	5.433	2.326	0.200	4.800	2.249	0.300
50	5.200	1.746	0.000	5.466	2.612	0.233	4.950	2.496	0.400
55	5.630	1.934	0.010	5.966	2.902	0.466	5.266	2.803	0.700
60	5.800	2.111	0.010	6.366	3.182	0.766	5.833	3.040	1.000
65	6.090	2.253	0.010	6.633	3.521	0.766	6.066	3.354	1.166
70	6.590	2.462	0.020	7.000	3.659	1.000	6.266	3.506	1.333
75	6.740	2.587	0.020	7.633	3.930	1.066	6.733	3.786	1.433
80	7.330	2.815	0.030	7.800	4.211	1.266	7.000	4.043	1.666
85	7.540	3.014	0.030	8.233	4.474	1.600	7.533	4.317	2.100
90	7.740	3.129	0.040	9.266	4.832	1.600	8.000	4.602	2.133

6.2 NSS与Greedy-MSC比较

Cardei 在文献[21]中提出了集中式的 Greedy-MSC 调度算法,此算法被应用于全向传感器网络的点目标覆盖控制,它包含 3 个输入参数:节点集合、目标集合和工作周期.Greedy-MSC 算法每次迭代时,找出覆盖当前临界目标的节点并将其加入到覆盖集,当所有目标都被覆盖时得到一个覆盖集,每个覆盖集轮流工作,每个工作周期为 w,直到出现目标不再被活跃节点覆盖时返回网络生存期.

在本文提出的分布式调度协议 NSS 中,节点依据剩余能量和节点对目标的覆盖关系决定了每轮工作周期的覆盖集,由于在每轮工作周期内覆盖集仅工作一小段时间,通过多次轮流工作达到合理分配覆盖集工作时间的目标.当 P=1 时,传感器变为全向的,比较此时的 NSS 和 Greedy-MSC 算法性能.假设目标区域内节点和目标随机分布,当节点感知半径为 250,目标数为 5 时,比较在不同节点数下的网络生存期.表 7 列出了两种调度算法关于网络生存期与节点数的数值关系,从数值上可以看出:1) NSS 网络生存期近似地与节点数成线性关系.2) NSS 比 Greedy-MSC 的性能略低,但差别不大.

Table 7 Networking lifetime under NSS and Greedy-MSC

表 7 NSS 和 Greedy-MSC 的网络生存期

Sensors	Lifetime of NSS	Lifetime of Greedy-MSC
25	9.300	10.900
30	11.500	13.900
35	12.500	14.900
40	13.800	16.900
45	16.300	19.900
50	17.300	20.900
55	18.000	24.900
60	20.500	27.800
65	24.500	29.700
70	26.000	33.400
75	29.500	36.300

6.3 实验结果

下面分别通过改变节点数、方向数、感知半径和目标数来评估 EDO 算法和 NSS 调度协议在不同环境下的性能.

(1) 节点数的影响.图 5 是当目标数为 50,感知半径为 120,方向数为 3 时,改变节点数量时计算的临界覆盖度和网络生存期.数据表明,临界覆盖度和网络生存期随着节点数量的增加呈线性增长的关系,当节点数在 30~40 时,网络生存期与临界目标的覆盖度曲线重合,原因是此时节点比较稀疏,节点之间由于覆盖相同的目标

所产生的关联比较少,因而覆盖集的数量与不相交覆盖集的数量近似,而临界目标的覆盖度则决定了不相交覆盖集数量,因而临界目标的覆盖度与网络生存期相同.

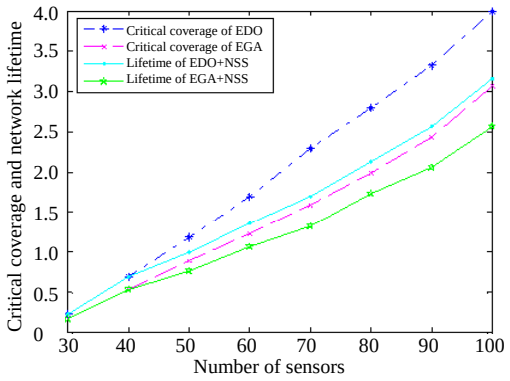

Fig.5 Critical coverage, networking lifetime vs. node number

图 5 临界覆盖度和网络生存期与节点数的关系

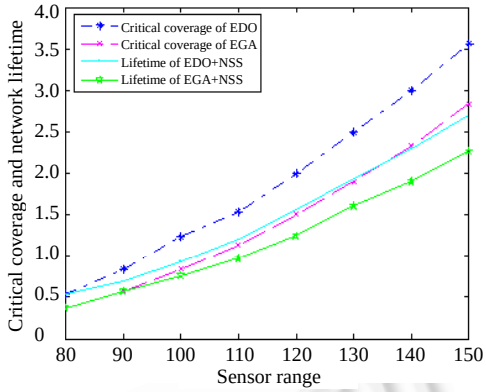

Fig.6 Critical coverage, networking lifetime vs. sensing range

图 6 临界覆盖度和网络生存期与感知半径的关系

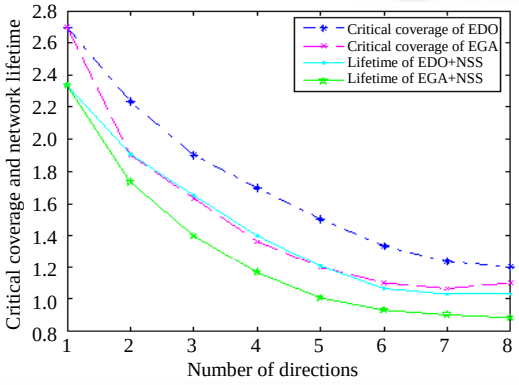

Fig.7 Critical coverage, networking lifetime vs. direction number

图 7 临界覆盖度和网络生存期与方向数的关系

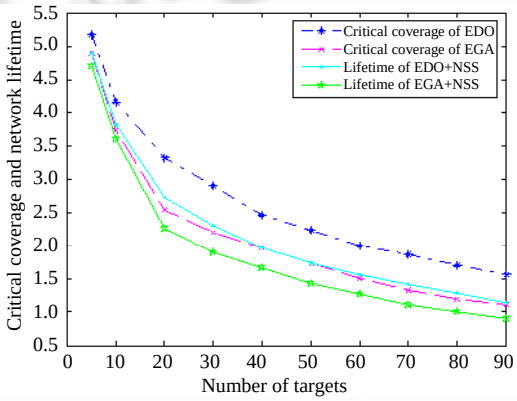

Fig.8 Critical coverage, networking lifetime vs. target number

图 8 临界覆盖度和网络生存期与目标数的关系

(2) 感知半径的影响.图 6 是当目标数为 50,节点数为 70,方向数为 3 时,感知半径与临界覆盖度和网络生存期的关系曲线.网络生存期和临界覆盖度与感知半径近似地保持线性关系,EDO 的临界目标覆盖度比 EGA 的临界目标覆盖度要大,EDO 的网络生存期也有所提高.当节点感知半径较小时,网络生存期与临界目标的覆盖度曲线重合,是由于感知半径小,节点之间由于覆盖相同的目标所产生的关联比较少的原因.

(3) 节点方向数的影响.图 7 是当目标数为 50,节点数为 70,感知半径为 120 时,方向数与临界目标覆盖度和网络生存期的关系曲线.临界目标的覆盖度和网络生存期随着方向数的增加而减小.当方向数为 1 时,有向节点就变成全向节点,两种优化算法的临界目标覆盖度重合,网络生存期的曲线也重合了.当方向数增加时,临界覆盖度都减小了,网络生存期也就降低了.

(4) 目标数的影响.图 8 是节点数为 70,感知半径为 120,方向数为 3 时,目标数与临界目标覆盖度和网络生存期的关系曲线.临界目标的覆盖度和网络生存期随着目标数的增加而减小.当目标数小于 5 时,4 条曲线基本重合.

7 结 论

本文针对无线传感器网络中的有向多覆盖集问题,提出了贪婪的方向优化算法和公平的方向优化算法.贪婪的方向优化算法选择覆盖最多数量的未被覆盖目标的方向作为工作方向,贪婪算法具有思想直观、计算复杂度低的优点. EDO 方向优化算法考虑了每个方向上的目标数、目标的覆盖度以及邻居节点的决策信息.相对于贪婪优化算法,EDO 更公平地分配感知资源,改善了临界目标的覆盖度,提高了网络覆盖资源的利用率,EDO 一次执行就可以优化所有节点的方向.此外,本文还讨论了邻居节点调度协议.NSS 不必找出所有覆盖集和预先分配工作时间,而是引入局部覆盖集的概念,通过局部覆盖集判断当前节点是否为冗余节点,并在考虑节点剩余的能量时决定节点是否可以转为睡眠,覆盖集轮流工作,使网络生存期最大化,最后,使用仿真实验验证了 EDO 算法和 NSS 的性能.此外,实验数据也表明,临界目标覆盖度比最大和平均的目标覆盖度小很多,从平均覆盖度与最大覆盖度比较接近可以看出,临界目标的数量很少但却严重地影响了网络的生存期,因而下一步的工作可以针对这个问题研究改善临界目标覆盖度的部署策略,如考虑加入少量的移动节点.

References:

[1] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks. IEEE Communications Magazine, 2002, 40:101−114.

[2] Ren FY, Huang HN, Lin C. Wireless sensor networks. Journal of Software, 2003,14(7):1282−1291 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/1282.htm

[3] Arici T, Altunbasak Y. Adaptive sensing for environment monitoring using wireless sensor networks. In: Tachikawa K, Taga T, *et al*., eds. Proc. of the IEEE Wireless Communications and Networking Conf. (WCNC). New Orleans: IEEE Communications Society, 2003. 2347−2352.

[4] Mainwaring A, Culler D, Polastre J. Wireless sensor networks for habitat monitoring. In: Raghavendra CS, Sivalingam KM, eds. Proc. of the 1st ACM Int'l Workshop on Wireless Sensor Networks and Application. Atlanta: ACM Press, 2002. 88–97.

[5] Delin KA, Jackson SP. The sensor Web: A new instrument concept. In: Proc. of the Int'l Society for Optical Engineering Symposium on Integrated Optics. 2001. 1−9.

[6] Li M, Liu YH. Underground structure monitoring with wireless sensor networks. In: Abdelzaher TF, Guibas LJ, eds. Proc. of the 6th Int'l Conf. on Information in Sensor Networks. Cambridge: ACM Press, 2007. 69−78.

[7] Schwiebert L, Gupta SKS, Weinmann J. Research challenges in wireless networks of biomedical sensors. In: Rose C, Palazzo S, eds. Proc. of the ACM/IEEE Conf. on Mobile Computing and Networking (MOBICOM). Rome: ACM Press, 2001. 151−165.

[8] Ren Y, Zhang SD, Zhang HK. Theories and algorithms of coverage control for wireless sensor networks. Journal of Software, 2006,17(3):422−433 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/422.htm

[9] Slijepcevic S, Potkonjak M. Power efficient organization of wireless sensor networks. IEEE Int'l Conf. on Communications, 2001.

[10] Berman P, Calinescu G, shah C, Zelikovsky A. Power efficient monitoring management in sensor networks. In: Proc. of the Conf. on IEEE Wireless Communication and Networking. Atlanta: IEEE Communications Society, 2004. 2329−2334.

[11] Ye F, Zhong G, Lu S, Zhang L. PEAS: A robust energy conserving protocol for long-lived sensor networks. In: Almeroth K, Calvert K, eds. Proc. of the IEEE Int'l Conf. on Network Protocols (ICNP). Atlanta: IEEE Computer Society, 2003. 28−37.

[12] Di T, Nicolas DG. A coverage-preserving node scheduling scheme for large wireless sensor networks. In: Raghavendra CS, Sivalingam KS, eds. Proc. of the 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications. Atlanta: ACM Press, 2002.

[13] Zhang M, Chan MC, Ananda AL. Coverage protocol for wireless sensor networks using distance estimates. In: Cruz R, Heinzelman W., eds. Proc. of the Mesh and Ad Hoc Communications and Networks (SECON). San Diego: IEEE Communications Society, 2007. 183−192.

[14] Ting Y, Tian H, John AS. Differentiated surveillance for sensor networks. In: Akyildiz IF, Estrin D, *et al*., eds. Proc. of the 1st Int'l Conf. on Embedded Networked Sensor Systems. Los Angeles: ACM Press, 2003. 51−62.

[15] Wang L, Kulkarni SS. Sacrificing a little coverage can substantially increase network lifetime. In: Sivalinga K, Yarvis M, eds. Proc. of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON). Reston: IEEE Communications Society, 2006. 326−335.

[16] Xu Q, Wang Y. Solving reliable coverage in fault tolerant energy efficient wireless sensor networks. Journal of Software, 2006,17:184−191 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/s184.htm

[17] Liu M, Cao JN, Zheng Y, Chen LJ, Xie L. Analysis for multi-coverage problem in wireless sensor networks. Journal of Software, 2007,18(3):127−136 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/127.htm

[18] Himanshu G, Samir RD, Quinyi G. Connected sensor cover: Self-Organization of sensor networks for efficient query execution. In: Gerla M, Ephremides A, Srivastava M, eds. Proc. of the 4th ACM Int'l Symp. on Mobile Ad Hoc Networking and Computing. Annapolis: ACM Press, 2003. 189−200.

[19] Jiang J, Fang L, Zhang HY, Dou WH. An algorithm for minimal connected cover set problem in wireless sensor networks. Journal of Software, 2006,17(2):175−184 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/175.htm

[20] Cardei M, Du DZ. Improving wireless sensor network lifetime through power aware organization. ACM Wireless Networks, 2005, 11:333−340.

[21] Cardei M, Thai MT, Li Y, Wu W. Energy-Efficient target coverage in wireless sensor networks. In: Znati T, Knightly E, Makki K, eds. Proc. of the IEEE Computer Communications (Infocom). Miami: IEEE Communications Society, 2005. 1976−1984.

[22] Sen A, Das N, Zhou L, Shen B, Murthy S, Bhattacharya P. Coverage problem for sensors embedded in temperature sensitive environments. Technical Report: TR-06-015, Department of Computer Science and Engineering, Arizona State University, 2006.

[23] Wang W, Srinivasan V, Chua KC, Wang B. Energy-Efficient coverage for target detection in wireless sensor networks. In: Abdelzaher T, Guibas L, Kaiser B, *et al*., eds. Proc. of the Int'l Conf. on Information Processing in Sensor Networks (IPSN). Cambridge: IEEE Signal Processing Society, 2007. 313−322.

[24] Gu Y, Hwang J, He T, Du DH. uSense: A unified asymmetric sensing coverage architecture for wireless sensor networks. In: Shatz SM, Abdelrahman TS, eds. Proc. of the Int'l Conf. on Distributed Computing Systems (ICDCS). Toronto: IEEE Computer Society, 2007. 8−18.

[25] Meguerdichian S, Koushanfar F, Potkonjak M, Srivastava M. Coverage problems in wireless ad-hoc sensor networks. In: Sengupta B, Kermani P, Lee D, eds. Proc. of the IEEE Computer Communications (Infocom). Anchorage: IEEE Communications Society, 2001. 1380−1387.

[26] Ma HD, Liu YH. On coverage problems of directional sensor networks. In: Znati T, He YX, eds. Proc. of the Int'l Conf. on Mobile Ad-Hoc and Sensor Networks. LNCS 3794, Springer-Verlag, 2005. 721−731.

[27] Tao D, Ma HD, Liu L. Coverage-Enhancing algorithm for directional sensor networks. In: Cao J, *et al*. eds. Proc. of the MSN 2006. LNCS 4325, Berlin: Springer-Verlag, 2006. 256−267.

[28] Alhussein JA, Abouzeid A. Coverage by directional sensors in randomly deployed wireless sensor networks. Journal of Combinatorial Optimization, 2006,11:21−41.

[29] Adriaens J, Megerian S, Potkonjak M. Optimal worst-case coverage of directional field-of-view sensor networks. In: Sivalingam K, Yarvis M, Zorzi M, eds. Proc. of the Sensor and Ad Hoc Communications and Networks. Reston: IEEE Communications Society, 2006. 336−345.

[30] Cai Y, Lou W, Li ML. Target-Oriented scheduling in directional sensor networks. In: Domingo-Pascual J, Smirnow M, eds. Proc. of the IEEE Infocom 2007. Anchorage: IEEE Communications Society, 2007. 1550−1558.

附中文参考文献:

[2] 任丰原,黄海宁,林闯.无线传感器网络.软件学报,2003,14(7):1282−1291. http://www.jos.org.cn/1000-9825/14/1282.htm

[8] 任彦,张思东,张宏科.无线传感器网络中覆盖控制理论与算法.软件学报,2006,17(3):422−433. http://www.jos.org.cn/1000-9825/17/422.htm

[16] 徐强,汪芸.容错节能无线传感器网络中可靠覆盖问题的解决方案.软件学报,2006,17:184−191. http://www.jos.org.cn/1000-9825/17/s184.htm

[17] 刘明,曹建农,郑源,陈力军,谢力.无线传感器网络多重覆盖问题分析.软件学报,2007,18(3):127−136. http://www.jos.org.cn/1000-9825/18/127.htm

[19] 蒋杰,方力,张鹤颖,窦文华.无线传感器网络最小连通覆盖集问题求解算法.软件学报,2006,17(2):175−184. http://www.jos.org.cn/1000-9825/17/175.htm

温俊(1979−),男,安徽合肥人,博士生,主要研究领域为无线传感器网络.

蒋杰(1976−),男,博士,助理研究员,主要研究领域为无线传感器网络,Ad-Hoc网络.

窦文华(1946−),男,教授,博士生导师,CCF 高级会员,主要研究领域为高级计算机网络.