

一种分布式的1-栅栏覆盖构建算法

应可珍^{①②} 邬锦彬^① 夏明^① 雷艳静^① 肖致友^① 陈庆章^{*①}

^①(浙江工业大学计算机科学与技术学院 杭州 310014)

^②(浙江财经大学东方学院 海宁 314408)

摘要: 在构建栅栏覆盖的过程中, 需要考虑网络构建成本和生存时间这两个核心问题。对于前者主要考虑节点数和信息传输量, 而对于后者则应避免网络因个别节点的死亡而陷入瘫痪。该文针对上述两个问题, 提出一种基于1-栅栏覆盖的分布式栅栏覆盖算法(Distributed Barrier Coverage Algorithm, DBCA), 算法通过聚类分簇和簇内路径选择这两个过程来实现覆盖。理论分析和仿真结果表明, 该文所提出的算法能够有效减少网络部署的节点数以及传输的信息量。在部署节点达到700时, 该算法比最优节点选择算法(Optimal Node Selection Algorithm, ONSA)和本地栅栏覆盖协议(Localized Barrier Coverage Protocol, LBCP)算法传输的信息量分别减少25%和41.6%, 生存时间分别多44%和30%。

关键词: 无线传感器网络; 栅栏覆盖; k-HOP分簇; 最大流; 路径规划

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2016)03-0720-08

DOI: 10.11999/JEIT150664

A Distributed Algorithm for Constructing 1-barrier Coverage

YING Kezhen^{①②} WU Jinbin^① XIA Ming^① LEI Yanjing^①

XIAO Zhiyou^① CHEN Qingzhang^①

^①(Department of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310014, China)

^②(Dongfang College, Zhejiang University of Finance & Economics, Haining 314408, China)

Abstract: The constructing cost and lifetime are two core problems when constructing a barrier coverage of Wireless Sensor Network (WSN). For the former, the amount of nodes and information transferred are considered very much. And the WSN shutdown Caused by just for some specific nodes died should be avoided. This paper proposes an algorithm named Distributed Barrier Coverage Algorithm (DBCA) to construct a distributed 1-barrier coverage by using k-HOP clustering and path planning. Theory analysis and simulation results show that the algorithm can reduce the number of nodes and information transferred effectively. When the deployed nodes reach the number of 700, it performs better than Optimal Node Selection Algorithm (ONSA) and Localized Barrier Coverage Protocol (LBCP) algorithm in reducing transferring information with 25% and 41.6%, and in prolonging lifetime with 44% and 30%.

Key words: Wireless Sensor Network (WSN); Barrier coverage; k-HOP clustering; Maximum flow; Path planning

1 引言

无线传感器网络(Wireless Sensor Network, WSN)由一系列微小节点组成, 节点数量根据网络需要可以是几十个, 也可以是成百上千个。若组成WSN的节点、链路特性和网络协议不同, 则形成的是异构网络, 反之则是同构网络^[1]。根据不同场景下

网络覆盖粒度的不同, 可将网络分为地毯覆盖、栅栏覆盖和扫视覆盖3种^[2]。在战场或其他环境下, 将WSN的节点部署到带状的边境区域, 使之形成一个监控地带, 当人或物穿越区域时可被带状区域中的节点及时感知并示警即称为栅栏覆盖^[3]。当移动物体穿越目标时, 若能至少被 k 个节点感知, 则称之为 k -栅栏覆盖^[4], 这也可以描述为穿越侦测程度。同时, 还可以进一步将网络分为弱栅栏覆盖和强栅栏覆盖。前者仅能侦测以固定方向穿越区域的入侵者, 而后者可侦测以任何方向穿越区域的入侵者^[4,5]。

构建栅栏覆盖的算法可分为集中式和分布式两类。集中式算法往往简单有效, 是栅栏覆盖的典型算法。其中文献[6]提出的BICS(Barrier Information

收稿日期: 2015-06-02; 改回日期: 2015-12-04; 网络出版: 2016-01-22

*通信作者: 陈庆章 qzchen@zjut.edu.cn

基金项目: 浙江省自然科学基金(LY14F020020, LQ12F02015), 国家自然科学基金(61379023, 61401397)

Foundation Items: The Natural Science Foundation of Zhejiang Province (LY14F020020, LQ12F02015), The National Natural Science Foundation of China (61379023, 61401397)

Coverage Set)算法利用概率传感模型来构建栅栏覆盖,它假设感测信息可在两个传感器之间交换,通过使用贪婪算法尽可能地找出边界传感器集合,并致力于降低整个过程中的传感器使用数量。文献[7]提出的 ONSA (Optimal Node Selection Algorithm) 算法则以优化穿越侦测程度和减少侦测节点为目标,通过连通 Sink 节点、减少数据转传节点个数的方式来达到目标。而文献[8]则从调整每个节点的覆盖区域着手来研究如何实现最少的能量消耗来覆盖同样的区域,并给出了节点能量在离散和连续情况下调整的实现方案。集中式算法虽然简单有效,但算法运行期间信息传输量大。在节点众多的大规模网络情况下,集中式算法会导致感知节点因负载过大而不及处理众多的信息,因此不适用。

由于上述问题,许多学者开展分布式栅栏覆盖算法的研究。文献[9]针对入侵路径较短的特性,提出了 LBCP (Localized Barrier Coverage Protocol) 算法,该算法保证入侵路径宽度在 L 以内的入侵者都能被侦测出来,且穿越侦测程度 k 至少为 1。算法同时设计了相关协议,将传感器分为“开启”、“睡眠”及“开启中”3个状态。“开启”状态下传感器检查自身为中心的区域是否为 k -栅栏覆盖,若不是则传感器进入“睡眠”状态。此外,文献[10]提出的分治(divide-and-conquer)算法推导构建强 k -栅栏覆盖所需部署传感器的密度,并以此为条件来实现部署目标。而文献[11]则从数据融合技术着手来构建栅栏覆盖,以减少节点间的通信开销。

在上述算法中,集中式的 BICS 算法和分布式的 LBCP 算法的主要目标都是优化网络生存时间,但构建成本巨大。与之相比,集中式的 ONSA 算法通过引入 Sink 节点大大降低了网络的构建成本,但同时也由于对 Sink 节点的依赖,导致网络的生存时间降低。分治算法虽然是分布式算法,但由于采用两个方向的构建模式,导致成本非常高。总体而言,目前还没有一种有效的分布式算法能够同时优化栅栏覆盖网络的构建成本以及生存时间两个性能指标。针对这种情况,本文提出了一种新的基于 1-栅栏覆盖的分布式算法 DBCA (Distributed Barrier Coverage Algorithm) 算法,该算法兼顾了网络的构建成本和网络生存时间,同时也适用于大型的传感网络。

2 算法模型与问题描述

2.1 网络模型

如图 1 所示,假设监控区域是一个长为 l 宽为 w 的矩形区域。对于区域内的传感器做如下假设:

- (1)每个传感器节点均有独立 ID,节点的传输半径为 R_t ,感知半径为 R_s ,且 $R_t \geq 2R_s$;
- (2)相邻传感器可通过信息传输来测量两两之间的距离;
- (3)信息在传输过程中不会产生过多的延迟,MAC 层中的封包皆能在有限的延迟内抵达目的节点;
- (4)源节点和目的节点部署在区域的两端,源节点是网络构建的发起端,目的节点是结束端。

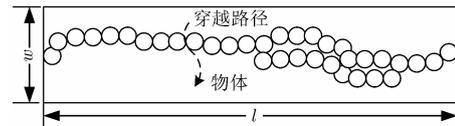


图1 栅栏覆盖示意图

2.2 侦测概率

在栅栏覆盖网络中,传感器利用传输能量的高低来判断是否侦测到入侵物体,由于传感器到目标事件发生地点的距离不同,传输能量也各有不同^[12],从而导致不同的侦测概率。概率模型是衡量侦测概率的重要方法,本研究主要采用文献[13]提出的概率模型来计算侦测概率。用 $p_i(d)$ 表示传感器 i 与目标物体之间的侦测概率,其中 d 表示两者间的距离。则 $p_i(d)$ 为

$$p_i(d) = \begin{cases} 0, & d \geq r + r_e \\ e^{-\lambda a^\beta}, & r - r_e < d < r + r_e \\ 1, & d \leq r - r_e \end{cases} \quad (1)$$

式中, $a = d - (r - r_e)$,其中 r 为感知半径, r_e 为传感器侦测的不确定性,且 $r_e < r$, d 为传感器与目标间的距离, λ 及 β 为衰减系数。此模型仿真了一定范围内侦测设备的侦测行为,其中的 r , λ 和 β 的值取决于具体的传感器种类及周围实际环境因素。其中若目标物体延两节点连线的中垂线穿越,则其被侦测到的概率相对最低^[14]。

3 分布式 1-栅栏覆盖算法 DBCA

3.1 算法处理流程

如图 2 所示,整个算法包括 k-HOP 分簇子算法和簇内路径选择子算法(Path Selection within Cluster Algorithm, PSCA),分别用于实现簇型网络的构建和簇内路径的选择。k-HOP 分簇分初始化、构建和回溯 3 个阶段来实现 k 跳簇的构建。之后执行 PSCA 子算法,该算法通过构建网络流图、寻找流图内的最短路径及移除生存时间少且相对花费大的节点从而最终形成构建栅栏覆盖网络的节点集。

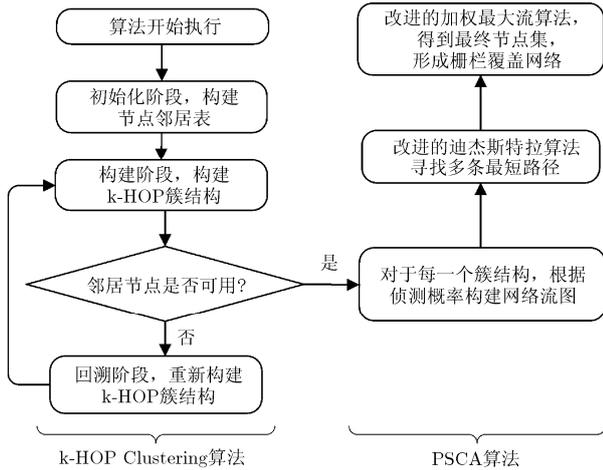


图2 DBCA分布式栅栏覆盖算法的处理流程

3.2 k-HOP 分簇子算法设计

3.2.1 节点的初始化 网络部署结束后，每个节点执行初始化操作。此处将节点分为普通节点和标量节点，邻居节点中度数最大的节点将作为标量节点，其他节点则作为普通节点。每个节点首先初始化自身的局部变量，具体如表 1 所示，源节点初始化时将 $Flag_n, Hop_n, Clst_n, Head_n$ 置为 1，其他信息置为 0，非源节点全部置为 0。而后广播 HELLO 信息，该信息包括了节点自身的 ID 号和标量节点标识。节点收到来自邻居节点的 HELLO 信息后获得它们的 ID、邻居节点度数以及距离信息并更新自身的局部变量，更新完成后便等待下一个收到的信息并处理它。信息处理结束后，非源节点等待源节点发来的信息。

表 1 节点局部变量表

变量名	说明
λ	侦测概率的阈值
N_n	与节点 n 相邻的传感器的侦测概率大于等于 λ 的邻居列表
N_{pre}	节点 n 所属簇的前一个节点的符合条件的邻居列表
$Flag_n$	判断节点 n 是否已被选为簇的标量节点(0: 否; 1: 是)
Pre_n	记录节点 n 所属簇的前一个节点
Hop_n	节点 n 为簇的第几跳节点
$Clst_n$	节点当前所属的簇编号
$Head_n$	节点 n 作为簇头所属的簇编号，值为 0 则表示非簇头节点
$Tail_n$	节点 n 作为簇尾所属的簇编号，值为 0 则表示非簇尾节点

3.2.2 k-HOP 分簇 初始化结束后源节点 s 通过向邻居节点发送 CONSTRUCT 信息从而发起簇的构建

操作，其中 CONSTRUCT 信息包含了源节点和目的节点的 ID 信息。此时， s 即为第 1 个簇的第 1 个标量节点。而后用该标量节点更新邻居节点表中所有节点的簇编号，并选出下 1 个标量节点，将 $Flag_n$ 置 1 且 Hop_n 加 1，邻居节点表中其他节点的 $Flag_n$ 置为 0，由于同一个簇的簇编号相同，因而它们具有相同的 $Clst_n$ 值。当标量节点的 Hop_n 增加到 k ，则该簇形成完毕。在邻居节点表中度数最大的点作为下一簇的起始节点，继续前面的过程，直到执行到网络汇点 t 为止。当 t 收到 CONSTRUCT 信息时，即代表此网络已构成。每个节点均通过 $Clst_n$ 来标志所从属的簇，并将此作为下一阶段簇内路径选择算法的依据。

图 3 展示了一个 k-HOP 的分簇过程。假设 k 为 3，每个圆表示一个节点，圆内上方的数字表示节点编号，下方的数字表示度数。其中编号 1 和 11 的点作为网络的源点 s 和汇点 t 。经过 k-HOP 算法分簇之后，编号为 1, 4, 6, 9, 11 的节点均为标量节点，同时所有节点被分为两个簇。

3.2.3 回溯设计 当簇的构建过程中出现无法从当前邻居节点表中选择合适的下一标量节点或无邻居节点可选的状况，则当前节点发送 BACKTRACKING 信息给上一节点进行回溯，该信息包含了源节点和目的节点的 ID 信息。当某节点收到 BACKTRACKING 信息，会从自身维护的邻居节点列表中删除此信息的来源节点，避免再次选择同一节点而造成多余的信息传递。随后从新的邻居列表中选择一个新的节点并对其发送 CONSTRUCT 信息。若新的邻居列表已无节点可选，则必须再将 BACKTRACKING 信息传递给前一个节点并把自身的 $Flag_n$ 设为 0，避免开启多余的节点。

3.3 PSCA 子算法设计

簇型网络构建完毕即执行 PSCA 子算法，这是一种改进的最小费用最大流 (Minimum Cost Maximum Flow, MCMF) 算法^[15]。在各个簇中，需要计算相邻节点 i 与 j 之间的距离，得到它们的侦测

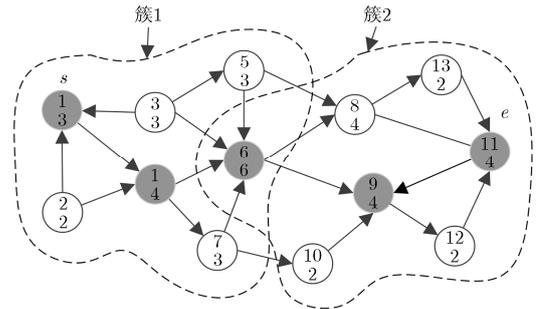


图3 k 为3时k-HOP分簇算法示意图

概率 $p(i, j)$ ，若其小于网络侦测概率门限值 λ ，则这两个节点视为不连通，反之则连通。在随后的路径选择阶段，需要同时关注每个节点的能量使用情况以及激活的节点数，使得选中的节点能量调度最大化，从而达到网络生存时间最大化且网络成本最小化的目的。

3.3.1 问题分析 网络流图 $G = (V, E)$ ，其中 V 表示当前簇的传感节点， E 表示有效感知范围内两个传感节点间的连接。 G 中所有顶点皆有权值，边皆有容量。边的容量即为两节点间的穿越侦测概率 $p(i, j)$ ，从一个网络节点到另一网络节点的流量不可超过容量设定。

算法的目标是选出一部分节点使得这些节点构成的网络权值最小且网络的最大流不小于侦测门限值 λ 。这是一个最小费用最大流问题，也是一个 NP 难问题，可取该问题的一个特殊例子来分析。图 4 表示网络流图 G ，其中源点 s 和汇点 t 间仅设一跳节点，每个节点设有权值 ω_i ，节点之间的边界容量为 c_i ，需要在网络中找到某个路径使得网络的权值最小且网络的最大流大于或等于侦测门限值 λ 。若该问题不是 NP 难问题，则其复杂度是多项式时间的，此问题可以用式(2)表示：

$$\min \sum_{i=1}^n \omega_i x_i, \text{ s.t. } \min \sum_{i=1}^n c_i x_i, x_i \text{ 为 } 0 \text{ 或 } 1 \quad (2)$$

式(2)是典型的 0-1 整数规划问题，该问题已被证明是 NP 难问题，由此可证明前述问题也是 NP 难问题。由于无法为 NP 难问题找到一个多项式时间的求解算法，PSCA 算法仅是该问题的一个较优解。算法可以找出一组节点，使所形成的栅栏覆盖网络拥有最小的侦测权值，并且最终组成的覆盖网络的最大流大于或等于侦测门限值 λ 。

3.3.2 算法具体设计 算法首先清理网络节点，剩余能量少于能量门限值 θ 的节点将被删除，其中 θ 的值根据网络的实际需求来设定。而后寻找从源点 s 到汇点 t 的最短路径，再使用最大流算法得到最终结果。

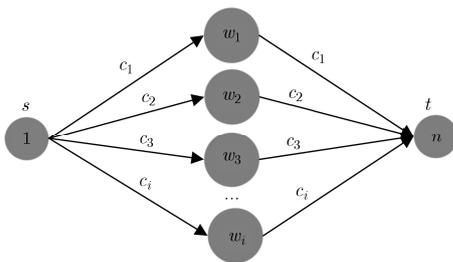


图4 网络流图特例示意图

(1)求解最短路径：本研究中的 WSN 属于单源最短路径求解问题。经典的迪杰斯特拉算法只能求解一条最短路径，本算法经过改进可得到多条最短路径，满足复杂网络的需求。假设对于单源正权图，其中每条边的权值为 1，源点为 s ，改进后的迪杰斯特拉(Dijkstra)算法可用表 2 所示的伪代码表示，该算法可以计算出多条最短路径，其时间复杂度为 $O(|V|^2)$ 。

表 2 改进的迪杰斯特拉算法的伪代码

1	for v in V :
2	Initialize $\text{dist}[v]$, $\text{previous}[v]$;
3	end for
4	Initialize Queue as the set of all nodes in G
5	while Queue is not empty:
6	Initialize u as the vertex has the smallest distance in $\text{dist}[]$
7	remove u from Queue
8	if $\text{dist}[u]$ is 0
9	break;
10	end if
11	for each neighbor w of u :
12	Initialize $\text{connector} = \text{dist}[u] + \text{distBetween}(u, w)$
13	if $\text{connector} < \text{dist}[w]$:
14	$\text{dist}[w] = \text{connector}$
15	$\text{previous}[w] = u$
16	end if
17	else if($\text{connector} == \text{dist}[w]$):
18	copy one vertex as dup equals to w
19	add vertex to Queue
20	end if
21	end for
22	end while

(2)使用改进的加权最大流算法：通过计算最短路径得到新网络流图之后执行改进的最大流算法。算法首先利用 Edmonds-Karp 算法得到网络流图 $G = (V, E)$ 的最大流 $\text{Flow}(G)$ ，并计算出当网络流量最大时每个节点通过的流量。若存在多条最短路径，则按节点移除策略移除 V 中的节点，并将移除节点后的新网络流图表示为 $G' = (V', E')$ 。若移除节点 i 后，网络 G' 的最大流 $\text{Flow}(G')$ 不小于侦测门限值 λ ，则节点 i 可移除，否则不可移除。当不存在可移除的节点时，算法结束。剩余的节点集 V' 即为最终的节点集，是需要开启侦测功能的候选节点，它们构成

的覆盖网络即为满足条件的 1-栅栏覆盖网络, 其他节点则进入睡眠状态。该算法的时间复杂度是多项式时间的, 由于枚举了 V 中每一个节点并且每个节点的时间花费是 $O(|V||E|^2)$, 所以算法的总时间复杂度为 $O(|V|^2|E|^2)$ 。

(3)节点的移除策略: 为使最终构成的网络生存时间最大且花费最小, 因此从每个节点的流量和能量出发来制定策略。将网络流图 $G = (V, E)$ 的最大流量表示为 $\text{Flow}(G)$, 此时节点 i 通过的流量表示为 $\text{Flow}_G(i)$ 。移除节点 i 后形成的新网络流图表示为 G' , 此时最大流表示为 $\text{Flow}(G')$ 。根据最大流理论可知 $\text{Flow}(G') \geq \text{Flow}(G) - \text{Flow}_G(i)$ 。很明显节点移除后网络最大流的衰减度受移除节点 i 的流量的影响。因此在门限值范围内, 每次选择流量较小的节点进行移除最安全可靠。

除考虑流量值外, 移除节点时还需考虑每个节点的能量使用状况, 将之表示为节点的花费, 显然不应选择花费过高的节点。假设所有节点初始能量相同并表示为 B , 节点的剩余能量表示为 B_i , 则节点的花费 Cost_i 可表示为式(3)。其中 θ 表示节点能量门限值, 按实际情况将其定义为一个常量。由于剩余能量低于 θ 的节点不建议使用, 因此将其花费定义为 B , 表示其花费最高, 达到了节点初始化的全部能量。

$$\text{Cost}_i = \begin{cases} B, & B_i < \theta \\ B - B_i, & B_i \geq \theta \end{cases} \quad (3)$$

移除节点时综合考虑 Cost_i 和 $\text{Flow}_G(i)$, 显然应将花费高且流量小的节点先移除。由此定义节点的移除依据 $R(i)$, 如式(4)所示, 每次只要选择 $R(i)$ 最大的节点即可。结合网络流量, 移除节点后网络 G' 的最大流量大于或等于网络侦测门限值 λ , 则该节点是可以移除的, 反之则不可移除。

$$R(i) = \text{Cost}_i / \text{Flow}_G(i) \quad (4)$$

上述过程的伪代码如表 3 所示。

4 仿真实验及结果分析

本仿真实验使用 Python 语言以及开源的数字扩展库 NumPy, SciPy 和绘图库 Matplotlib 来实现。并选取前述的 ONSA 算法以及 LBCP 算法为参照算法。

4.1 网络环境及参数设置

定义矩形部署区域大小为 $500 \text{ m} \times 20 \text{ m}$, 抛洒的传感器数量为 N , 部署网络内的节点密度为 ρ , 穿越门限值 λ 分别设为 0.1, 0.3, 0.5, 0.7, 0.9, 所有节点剩余能量门限值 θ 设为 5, 每个节点剩余能量 B_i 值为 0~30 的随机值, 节点初始能量 B 为 30, 处

表 3 改进后的加权最大流算法的伪代码

1	Initialize graph G and set C is empty
2	use Edmonds-Karp Algorithm to calculate the max-flow of graph G
3	for node i in set V
4	calculate $R(i) = \text{Cost}_i / \text{Flow}_G(i)$
5	if V is empty
6	return C
7	end if
8	choose node i with maximum $R(i)$ and remove i from V and has new graph G'
9	update existing nodes and edges E'
10	calcite the max-flow $F(G')$ of graph G'
11	if $\text{Flow}(G') < \lambda$
12	add node i to C
13	end if
14	end for

于激活状态即侦测状态的节点在每个时隙里消耗 1 单元的能量。同时设每个传感器节点的感知和通信范围分别为 5 m 和 10 m, 传感器侦测的不确定性 r_c 为 3 m, 衰减系数 λ 及 β 均为 0.5。实验针对不同的传感器数量进行 20 次实验, 将所得结果平均, 每次实验均为不同的网络环境。过程中影响衡量指标的主要参数是 k-HOP 分簇节点的跳数 k 以及初始抛撒到部署区域的节点数量 N , 需要通过仿真实验分别找到各个参数对于网络性能的影响。

4.2 算法性能衡量指标

衡量性能的指标主要包括栅栏覆盖网络的构建成本和生存时间两个方面。其中衡量网络构建成本的参数包括启动节点的个数和传递的信息量两个方面。启动节点的个数是指执行完算法后开启侦测功能的传感器个数。构建栅栏覆盖启动的节点越少, 所需成本越低。传递的信息量是指算法执行途中节点之间传递信息的数量。传递的信息量越多, 则所需成本越大、效能越低。另一衡量指标网络生存时间则越长越好。在进行仿真实验时, 初始为每个节点随机分配一个初始剩余能量 B_i , 随后可得到每个节点的能量花费情况以及移除依据 $R(i)$ 。

4.3 网络的构建成本分析

构建 1-栅栏覆盖网络时, 选择部分节点激活, 使它们构成度为 1 的强栅栏覆盖。激活的节点越少, 成本越低。同时网络覆盖的性能受 3 个因素的影响: 成簇过程的跳数 k 、抛撒节点的总数 N 和栅栏覆盖网络侦测概率门限值 λ , 实验时分别取 3 个参数的默认值为 15, 500 及 0.7。同时, 对其中一个参数进行实验时, 其他参数也设定为默认值。

4.3.1 跳数 k 对网络构建的影响 WSN 网络中激活的节点数与簇中跳数 k 的关系如图 5 所示。由图可见激活节点数随着跳数 k 的增加而降低，当 k 大到覆盖整个网络之时，网络激活的节点数最少，此时达到全局最优。但如图 6 所示，网络中信息的传输量与 k 值的变化成正比关系，随着跳数 k 的增加，信息的传输量也大大增加，这将大大降低网络的效率，并减少当前激活节点的使用寿命。因此对于构建网络而言，传输的信息量和激活的节点数需根据需要进行权衡。由图 7，进一步考虑跳数 k 与网络生存时间的关系， k 越大网络的生存时间也越长。若 k 值大到覆盖整个网络，则网络的生存时间达到最优，这是由于 k 越大簇越少，激活的节点数也越少，使得网络生存时间得以延长。结合以上三方面， k 的取值不宜过大也不宜过小，为了便于研究，在进行其他参数的实验时，选取 15 作为 k 取值的默认值。

4.3.2 激活节点数与部署节点数的关系 在相同参数下，将本算法与 ONSA 算法以及 LBCP 算法在激活节点数与总节点数 N 的关系上进行比较，结果如图 8 所示。显然，当在区域内随机抛洒节点数为 300 时，网络节点部署密度不高，各算法几乎都需激活所有节点来维持网络的正常运行。随着部署节点的增加，本算法激活的节点数目与其他两种算法相比明显减少。尤其当节点部署数目达到 800 时，由于具有更多的簇内最短路径的选择，因此需要激活的节点更少。对于 ONSA 算法，它将 Sink 节点与普通节点共同部署，在初始节点数较少的情况下其算法性能较好，但随着初始部署节点的增加，需要有更多的节点来连通 Sink 节点，导致需激活的节点数也相应增加，因而在初始节点数众多的情况下算法性能有限；对于 LBCP 算法，该算法假设入侵者的穿越路径是在一定范围内的，此处按照文献[9]所设置的穿越范围 $d = 100$ m 来设定，结果表明 LBCP 算法优于 ONSA 算法，但与本算法相比仍有差距。

因此本算法在激活节点数性能指标上更适用于大型网络的部署。

4.3.3 传输信息量与部署节点数的关系 侦测节点间传输的信息量与总抛撒的节点数量的关系如图 9 所示。在部署节点数比较少的情况下，网络构建率不高，因此算法间的表现相差不大，网络传输信息量变化也不大。随着部署节点数量的增加，DBCA 算法比 ONSA 算法传输的信息量平均减少 25% 左右，比 LBCP 算法平均减少 40% 左右。同时，本算法是分布式的，网络被分成簇结构，随着布署节点数量的增加，传递信息量的增加比另外两种算法平滑很多，给节点的负担相对小，且无论布署多少节点，单个簇结构的规模都不会变大，簇结构里的信息传递量也不会很大，不会招致节点的信息过载，因此该分布式算法可以应用在大型无线传感器网络中。

4.4 栅栏覆盖网络的生存时间分析

DBCA 算法将网络时间分为一个个时隙，初始时每个节点拥有一个初始剩余能量 B_i ，在每一个时隙里，选取一部分节点集合 C 作为最终选定的节点集合，将这些节点置为激活节点并构建成满足条件的度为 1 的强栅栏覆盖。激活的节点消耗 1 点的能量，睡眠节点不消耗能量。整个网络的生存时间维持到网络不能组建成为度为 1 的强栅栏覆盖为止。

网络覆盖生存时间与初始抛撒的总节点数 N 之间的关系如图 10 所示。由结果可知，节点数量较少时，由于网络的构建率不高，3 种算法的表现相差不大，但随着部署节点数量的增加，本算法的生存时间要远远高于另外两种算法。其中 ONSA 算法由于对 Sink 节点的依赖，影响了网络的生存时间。而 LBCP 算法由于采用局部覆盖的方式使其生存时间高于 ONSA 算法。本算法通过 PSCA 算法尽可能激活最少的节点来实现网络覆盖，在部署节点越来越多的情况下所要激活的节点反而减少，因而随着部署节点数的增长，其网络覆盖生存时间大大提高。

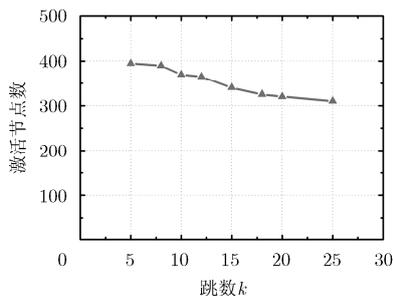


图 5 激活节点数与跳数 k 之间的关系

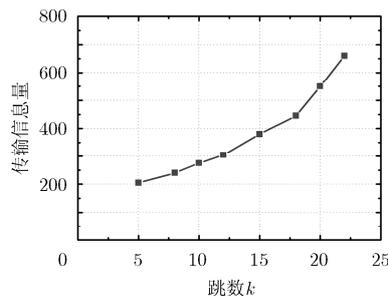


图 6 传输信息量与跳数 k 之间的关系

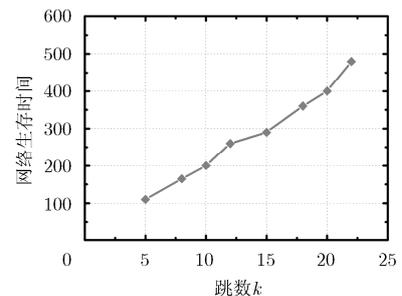


图 7 网络生存时间与跳数 k 之间的关系

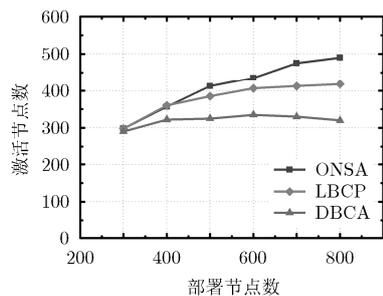


图 8 激活节点数与部署节点数关系

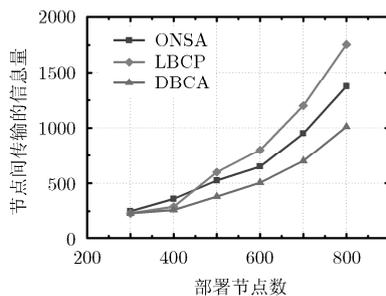


图 9 节点间传输的信息量与部署节点数关系

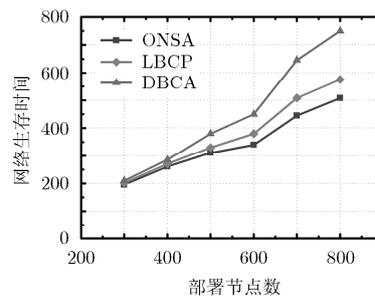


图 10 网络生存时间与部署节点数的关系对比图

5 结束语

本文从网络覆盖成本和生存时间两个角度出发来研究 WSN 的栅栏覆盖问题, 并设计了一种分布式的 1-栅栏覆盖算法 DBCA 算法。通过在传统集中式算法中结合 k-HOP 分簇思想, 使得 DBCA 算法能在部署成本和生存时间方面得到优化, 并可用于大型的传感网络。同时, 在仿真实验中与 LBCP 以及 ONSA 算法进行比较, 可知在部署节点达到 700 时, DBCA 算法分别比 ONSA 算法和 LBCP 算法传输的信息量减少 25% 和 41.6%, 生存时间分别比 ONSA 和 LBCP 算法多 44% 和 30%, 在大型栅栏覆盖网络中, 对网络生存时间的延长效果尤其明显。但本研究假定的是理想环境, 且所有节点均拥有 360° 的侦测范围, 而实际应用中节点的侦测效果往往具有方向性, 节点数目众多时节点摆放方向的不同对结果有很大的影响。因此在实际部署中, 对于侦测传感器节点的选择是一个比较重要的问题, 同时对于不同的感知方式也需采用不同的部署形式。而在 k-HOP 分簇算法中, 不同的网络规模对于 k 取值的关系也是未来工作的研究重点, 对于一定数量的部署节点, 如何设置 k 值使得结果最优也是另一个极具有挑战的问题。

参考文献

- [1] 杜晓玉, 孙力娟, 郭剑, 等. 异构无线传感器网络覆盖优化算法[J]. 电子与信息学报, 2014, 36(3): 696-702. doi: 10.3724/SP.J.1146.2013.00730.
DU Xiaoyu, SUN Lijuan, GUO Jian, et al. Coverage optimization algorithm for heterogeneous WSNs[J]. *Journal of Electronics & Information Technology*, 2014, 36(3): 696-702. doi: 10.3724/SP.J.1146.2013.00730.
- [2] LI Mo, LI Zhenjiang, and ATHANASIOS V. A survey on topology control in wireless sensor networks: taxonomy, comparative study, and open issues[J]. *Proceedings of the IEEE*, 2013, 25(10): 2367-2380. doi: 10.1109/JPROC.2013.2257631.
- [3] AHMED N, KANHERE S S, and JHA S. Probabilistic coverage in wireless sensor networks[C]. *IEEE Conference on Local Computer Networks*, Sydney, 2005: 672-681. doi: 10.1109/LCN.2005.109.
- [4] SANTOSH Kumar, LAI T H, and ANISH Arora. Barrier coverage with wireless sensors[J]. *Wireless Networks*, 2007, 13(6): 817-834.
- [5] TAO Dan and WU Tinyu. A survey on barrier coverage problem in directional sensor networks[J]. *IEEE Sensors Journal*, 2015, 15(2): 876-885. doi: 10.1109/JSEN.2014.2310180.
- [6] YANG Guanqun and QIAO D. Barrier information coverage with wireless sensors[C]. *INFOCOM 2009*, Rio de Janeiro, 2009: 918-926. doi: 10.1109/INFCOM.2009.5062002.
- [7] LAI Y L and JIANG J R. Sink-connected barrier coverage optimization for wireless sensor networks[C]. *The Seventh International Conference on Wireless and Mobile Communications*, Wuhan, 2011: 198-203.
- [8] FAN Haosheng, LI Minming, SUN Xianwei, et al. Barrier coverage by sensors with adjustable ranges[J]. *ACM Transactions on Sensor Networks*, 2014, 11(1): 14(1)-14(20).
- [9] CHEN Ai, SANTOSH Kumar, and LAI T H. Local barrier coverage in wireless sensor networks[J]. *IEEE Transactions on Mobile Computing*, 2010, 9(4): 491-504. doi: 10.1109/TMC.2009.147.
- [10] LIU B, DOUSSE O, NAIN P, et al. Dynamic coverage of mobile sensor networks[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2013, 24(2): 301-311. doi: 10.1109/TPDS.2012.141.
- [11] 罗卿, 林亚平, 王雷, 等. 传感器网络中基于数据融合的栅栏覆盖控制研究[J]. 电子与信息学报, 2012, 34(4): 826-831. doi: 10.3724/SP.J.1146.2011.00446.
LUO Qing, LIN Yaping, WANG Lei, et al. Barrier coverage control based on data fusion for wireless sensor network[J].

- Journal of Electronics & Information Technology*, 2012, 34(4): 826–831. doi: 10.3724/SP.J.1146.2011.00446.
- [12] HOU T C and LI V O K. Transmission range control in multihop packet radio networks[J]. *IEEE Transactions on Communications*, 1986, 34(1): 38–44. doi: 10.1109/TCOM.1986.1096436.
- [13] ZOU Y and CHAKRABARTY K. Sensor deployment and target localization in distributed sensor networks[J]. *ACM Transactions on Embedded Computing Systems*, 2004, 3(1): 61–91.
- [14] LI J, CHEN J, and LAI T H. Energy-efficient intrusion detection with a barrier of probabilistic sensors[C]. *INFOCOM 2012, Orlando, 2012*: 118–126. doi: 10.1109/INFOCOM.2012.6195479.
- [15] JING Yuan, BAE Egil, and TAI Xuecheng. A study on continuous max-flow and min-cut approaches[C]. *IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, 2010*: 2217–2224. doi: 10.1109/CVPR.2010.5539903.
- 应可珍：女，1978 年生，讲师，主要研究方向为无线传感器网络、覆盖与数据融合。
- 陈庆章：男，1956 年生，教授，主要研究方向为无线传感器网络，分布式处理与协同工作等。