



Partial coverage generation algorithm in wireless sensor networks

Boyao Wei

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

June 20, 2019

无线传感器网络中部分覆盖的生成算法*

魏博垚¹

1(首都师范大学 信息工程学院, 北京 100048)

摘要: 在无线传感器网络(wireless sensor network)中,通过睡眠机制可以让一些节点休眠,从而节省能量,延长网络寿命.此外在多数应用场景下无线传感器网络(WSN)只需要部分覆盖目标区域,即可实现对整个目标区域的监测目的.无线传感器网络(WSN)在随机抛洒过程中由于节点之间相互重叠易产生大量的覆盖冗余,如何计算节点监测区域的面积,如何激活尽可能少的节点以满足覆盖要求,是一个难点.本文通过建立网格模型来表示节点监测区域的面积,并通过调整网格粗细粒度,衡量节点间的冗余程度.在此基础上提出基于最大独立集的部分覆盖休眠调度机制(NSA),该算法通过贪心的策略,首先找到一个最大独立集,然后按照节点的冗余程度与覆盖贡献通过增删极个别节点从而达到监测要求的目的.通过合理的激活节点,满足部分覆盖要求,从而让更多的节点进入休眠状态,使能量最大程度得到保留.

关键词: 覆盖模型;网格;最大独立集;无线传感器网络

Partial coverage generation algorithm in wireless sensor networks

CHEN Wen-Long¹⁺, WEI Bo-yao¹, TANG XiaoLan¹

¹(Information Engineering College, Capital Normal University, Beijing 100048,China)

Abstract: In the wireless sensor network, some nodes can sleep by the sleep mechanism, which saves energy and prolongs the network life. In addition, in most application scenarios, the wireless sensor network (WSN) only needs to partially cover the target area. Monitoring requirements for the entire target area. Wireless sensor networks (WSNs) are likely to generate a large amount of coverage redundancy due to overlapping between nodes during random flooding. How to calculate the size of the node monitoring area and how to meet the coverage requirements, so that the activated nodes As little as possible, it is a difficult point. This paper establishes the grid model to represent the area of the monitoring area of the node, and adjusts the granularity of the grid to measure the overlap between nodes. Based on the maximum The part of the independent set covers the sleep scheduling mechanism (NSA). The algorithm first finds a maximum independent set through the greedy strategy, and then adds and deletes nodes according to the redundancy degree and coverage contribution of the node to meet the monitoring requirements. Through reasonable activation nodes, Meet some of the coverage requirements, allowing more nodes to go to sleep, so that The largest extent preserved.

Key words: Cover model; grid; maximum independent set; wireless sensor network

1 引言

无线传感器网络(wireless sensor network)由于其成本低,灵活多样,且能够在恶劣的自然环境中工作(如深海,外太空和战场等环境),所以通过大面积部署无线传感器节点,进行数据的收集与整理.随着通讯技术的不断发展,更加高效,便捷的通讯方式不断出现,使得人们对数据的收集与感知变得越来越迫切.由于传感器节点携带电池能量有限,无线传感器网络一旦部署,在某些情况下更换电池极其困难.因此如何在满足监测和通讯的前提下节省能耗,延长整个网络寿命一直是传感器网络的研究热门之一.

无线传感器网络覆盖模型按照分布方式分为随机性覆盖模型和固定性覆盖模型,固定性覆盖模型是根据预先设定好的位置确定网络拓扑结构,而随机性覆盖则是在节点位置随机分布的情况下,完成对目标的覆盖;覆盖模型按照覆盖比例又分为完全覆盖与部分覆盖.完全覆盖指的是对目标区域的监测需要 100%的覆盖率,而部分覆盖指的是对目标区域只需要按一定比例进行覆盖.

在传感器网络中节点的能量大部分用于数据传输,其中由于簇头节点负责本簇节点与簇外节点通讯,因此当选簇头的节点能量消耗较快.有数据显示^[1],传输一个比特的数据所消耗的

作者简介: 魏博垚,男,硕士研究生,主要研究领域为无线传感器网络;

能量相当于一个传感器节点执行上千次指令的能耗.所以在传感器网络监测的区域当中,只需要激活一部分的节点负责监测功能,并向簇头节点传输收集的数据,就能满足正常工作.只要负责监测的节点能量未耗尽,就能保证此区域的监测功能正常运转.从未激活节点中轮流选择一个节点充当簇头,只负责收集与转发监测节点数据,而其余未激活的节点进入休眠,就能更大程度上保证网络寿命.因此调度哪些节点负责数据采集,使哪些节点负责数据转发,使哪些节点处于休眠状态是一个极为复杂的问题.

无线传感器网络部分覆盖休眠调度策略是提高节点能效的有效手段之一,即在满足覆盖要求时,使其余节点进入休眠状态,进而降低能耗.例如,当无线传感器网络(WSN)应用于监测环境的湿度或温度时,只需要覆盖目标区域的一部分即可完成整个区域的监测要求,其余节点可以进入休眠状态.因此,通过部分覆盖休眠调度策略可延长无线传感器网络寿命.

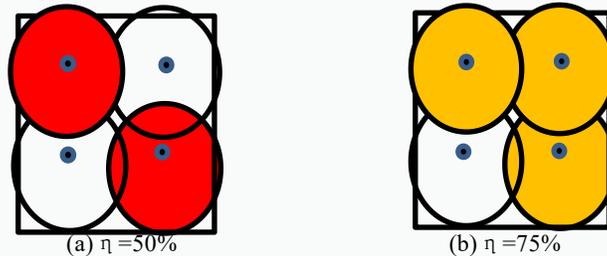


图1 部分覆盖图

图1是最低覆盖率为 $\eta=50\%$ 和 $\eta=75\%$ 部分覆盖图示意图,其中阴影部分是两个区域分别在不同覆盖率 η 条件下激活最少节点所组成的工作节点集.

在无线传感器网络(WSN)中,由于节点随机抛撒,节点位置无法控制,且一旦抛洒则无法移动,所以合理的调度节点极为重要.为了能够有效的监测目标区域,需要对目标区域大面积抛洒节点,因此节点相互之间就会存在重叠的情形,在对目标区域监测时,产生监测冗余.当负责监测的节点产生的监测冗余数据非常大时,会产生大量的资源浪费.

本文目标就是在随机抛撒的无线传感器节点里选取一组能够满足覆盖率且节点个数尽可能少的节点集合,激活此集合使其进行数据的采集工作,从剩余的节点中轮流选取节点充当簇头,负责数据的传输.此外本文还提出了基于网格的计算方法,用于计算节点监测面积.

本文是建立在分完簇的基础上,在簇内讨论随机性部分覆盖.在此条件下提出了基于最大独立集的部分覆盖算法,该算法充分考虑部分覆盖的特点,首先快速选取一组传感器节点集,此集合的特点是所有节点相互之间没有或者只有很少的监测重叠.在此基础上根据不同的覆盖要求,判断是否满足覆盖率,如果满足覆盖率,需判断是否能够在不影响监测要求的情况下删除冗余节点,直到再删除任意节点均无法满足覆盖要求.如果不满足覆盖率则在最大独立集节点的基础上,加入监测贡献区域最大的节点,直到满足最小覆盖要求,当恰好满足覆盖要求时,则直接激活此集合,其余节点轮流选择节点充当簇头,剩余节点进入休眠.当超过最低覆盖要求以后,仍需进行删除判断,删除操作同上.当算法终止后,激活此节点集合,充当监测节点集,从剩余节点中轮流选择节点充当簇头.

第二部分介绍相关工作,第三部分讲述无线传感器网络(WSN)相关模型与假设.第四部分介绍基于最大独立集的部分覆盖快速生成算法.第五部分仿真实验.第六部分总结.

2 相关工作

在无线传感器网络通讯协议中,LEACH 协议^[4]最初在 2000 年提出,该算法基本思想是:以循环的方式随机选择簇头节点,将整个网络的能量负载平均分配到每个传感器节点中,从而达到降低网络能源消耗、提高网络整体生存时间的目的.但是该协议通过随机安排节点轮流当选簇头十分容易造成簇头节点的负载不均衡;后续针对 LEACH 的改进版本众多,但是很少

考虑到节点监测区域重叠的情况.文献[5]提出了一种基于目标区域 Voronoi 划分的集中式近似算法(centralized Voronoi tessellation,简称 CVT)用于计算完全覆盖目标区域所需要的近似最小节点集,其基本思想是利用 Voronoi 图,把传感器网络划分为 Voronoi 多边形,利用贪心的策略,计算完全覆盖目标区域所需要的近似最小覆盖集,但其未考虑节点异构的情形;与此同时,该文献也并未指出当激活节点工作时,如何利用休眠节点,此方法也未考虑部分覆盖策略.文献[6]针对传统网络覆盖模型仅以区域覆盖率作为评价标准,而未考察不同覆盖模型下节点能量有效性问题,在协作覆盖模型的基础上,提出了能量有效的分层协作覆盖模型 EEHCCM,该算法基于启发式算法当中的蚁群算法;该算法只是指出传感器网络的分层模型,但是此研究并未涉及节点的调度机制.文献[7]首先讨论了如何利用节点的覆盖冗余来延长网络生存时间.其基本思想是:将所有节点划分为若干互不相交的节点集合,其中每个节点集都能够完全覆盖目标区域.通过不同节点集合之间的交替工作,可以延长网络生存时间.互不相交的集合越多,网络的生存时间越长,但是未考虑部分覆盖模型,以及分簇的情况.文献[8]根据邻居节点之间的覆盖交叠关系,提出了一种旨在延长网络生存时间的分布式节点调度算法,并未涉及部分覆盖.文献[9]提出了一种基于目标区域网格划分的工作节点选择算法,只给出网络的精准度,并未考虑节点间的冗余情况.文献[10]提出的 PCLA 算法,仍然有大量的监测冗余,尤其是当覆盖限制条件越苛刻时,此算法的性能大打折扣,文章也并未提出如何计算面积覆盖.

3 模型与假设

3.1 环境描述

无线传感器部署分两种情况,第一种情况是分布的节点集覆盖的区域刚好达到对目标区域最低监测要求,无冗余节点,则此情况下,从负责目标区域监测的节点集当中选择节点轮流当选簇头;第二种情况对目标区域覆盖在满足最小监测要求(如覆盖监测区域 80%)情况下,有较大的冗余节点产生,删除冗余节点,只需激活满足最小覆盖要求的节点进行数据的采集并传输信息给簇头节点,在冗余节点当中轮流选择一个节点当簇头,只负责数据的传输,其余节点进入休眠.只需要保证满足最小覆盖的节点能够正常进行工作,其余冗余节点对于整个网络的正常工作无紧要关系,因此能够更大程度上延长网络的工作时间.本文主要工作在于如何快速找到满足覆盖要求的节点集,且节点集个数较少,对其余节点进行合理的调度,最大限度地延长网络生存时间.由于无线传感器网络受到各种因素影响,以下是本文中无线传感器的条件假设模型:

- (1) 本算法在簇内讨论,且随机分布在簇内的节点集 S 能够完全覆盖此簇区域;
- (2) 簇内每个节点都可以与簇外某节点进行通讯;
- (3) 每个节点具有 ID 号且具有位置感知能力,知道自己与邻居节点的坐标;
- (4) 每个节点初始能量相同.不同节点的感知半径可以不同.

3.2 网络覆盖模型

每个传感器节点都有感知半径 r ,在随机抛洒过程中,传感器节点会产生大量监测冗余,本文提出建立网格模型的方法,通过计算节点覆盖网格交叉点的方式,来计算节点的监测面积与覆盖率.

在二维平面中,目标区域为 R ,传感器节点 s_i 的监测区域是以 s_i 为圆心,监测半径为 r 的圆形区域.簇内传感器节点集合表示为 $S = \{s_1, s_2, \dots, s_n\}$.利用横纵坐标把区域 R 划分为粒度为 b ,大小均匀的网格,通过计算不同节点所包含的交叉点的个数来表示不同传感器节点的覆盖面积.若区域 R 内任意交叉点 p ,都能够在集合 S 中的找到某一节点 s_i ,使得 s_i 的监测区域能够覆盖点 p ,则称集合 S 是对区域 R 的完全覆盖;不同应用环境下,对覆盖要求不相同,一个簇内

所要求的最低覆盖率用 η 表示,即覆盖率低于 η 则无法对目标区域进行有效的监测.

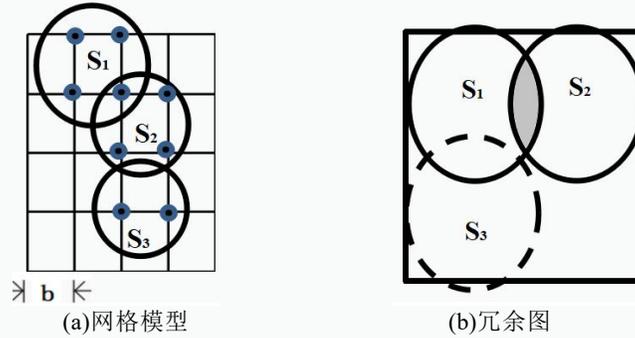


图2 覆盖模型图

定义1 (节点覆盖区域):在网格覆盖模型中,一个传感器节点包含 n 个交叉点,则此传感器覆盖面积为 n 个单位,其中 b 代表网络粗细粒度.

定义2 (节点监测冗余):给定区域 R 和节点集合 S ,若两个传感器公共覆盖区域有交叉点 p ,则判断两个传感器节点监测区域有重叠,重叠区域称之为监测冗余.

在本文中用 $A(S')$ 表示 S' 所覆盖的面积大小,若节点 s_i 与 s_j 具有重叠关系则用 $E(s_i, s_j) = 1$ 来表示,如无重叠关系则 $E(s_i, s_j) = 0$.如图2(a)所示, s_1 与 s_2 节点分别包含4个交叉点,则 $A(s_1) = 4$, $A(s_2) = 4$, $A(s_1, s_2) = 8$. s_1 与 s_2 的公共区域有交叉点则判断这两个节点相交, $E(s_1, s_2) = 1$, s_2 与 s_3 公共区域无交叉点,则 $E(s_2, s_3) = 0$.用 $int(s_i)$ 表示与 s_i 有重叠关系的节点集合, $N(s_j) = |int(s_j)|$ 表示此集合的节点个数.

定义3 (节点贡献区域):若2个激活节点的监测范围有公共区域,节点的非公共区称为这个节点贡献区.

如图1(b)实线所画圆代表激活的负责监测的节点,虚线所画的圆代表休眠或未负责监测的节点,阴影部分代表 s_1, s_2 节点的监测冗余(监测冗余只在负责监测的激活节点之间产生),而 s_1 与 s_3 并无监测冗余, s_1 未被其他负责监测的激活节点所覆盖的区域称之为 s_1 的监测贡献区.

由于传感器节点之间会产生重叠冗余,显然当激活节点进行监测工作时,如果节点间的重叠冗余只有很少的一部分,对整体性能产生的影响则可以忽略不计,但是如果重叠冗余很大则必然导致监测资源的浪费.如何衡量节点覆盖冗余情况,本文通过调整网格粗细粒度的方式来判断此冗余的影响.所以网格粗细粒度一方面可以计算无线传感器节点的覆盖面积,另一方面可以判断节点之间的重叠情况.当粒度 b 的值越大,则判断两节点之间有重叠冗余的概率越小,也就是说这两个传感器节点的重叠区域可以忽略不计,因此在创建最大独立集的时候可以取得较好的结果.但是粒度 b 偏大时会直接影响覆盖面积的计算精度,导致误差.如果粒度 b 越小,虽然会提高覆盖面积的计算精度,同时也会对于传感器节点的冗余判断过于敏感,在创建最大独立集时,导致生成的最大独立集元素个数过少,对节点的增加和删操作频繁,增加算法的时间复杂度.本文通过实验,得出粒度 b 的合理取值区间.

3.3 传感器能量消耗模型

在传感器网络中节点的能量大部分用于数据传输,其中簇头节点负责本簇与簇外节点通讯,因此当选的簇头节点能量消耗最快.所以在传感器网络监测的区域当中,只需要保障负责监测的节点满足监测功能,在负责监测的节点能量未耗尽前,就能保证整个区域能够正常运转,而未激活节点轮流当选簇头,可以提高网络生存时间.

根据文献[1],向节点传输 q 比特的消息,且传输距离为 d 时,所消耗的能量表示为:

$$E_{TE}(q, d) = \begin{cases} qE_e + qE_{te}d^2 & \text{if } d < d_0 \\ qE_e + qE_{te}d^4 & \text{if } d \geq d_0 \end{cases} \quad (1)$$

式中 E_e , E_{te} 均为传输能量消耗参数,相应的接受 q 比特数据的能量消耗公式为:

$$E_{rx}(q) = qE_e \quad (2)$$

在一个簇内,通常充当簇头的节点数据传输距离大于等于 d_0 ,而簇内普通节点的数据传输距离小于 d_0 .而簇头需要接受簇内节点的数据,因此单个簇头节点能量消耗与单个普通监测节点的能量消耗之差表示为:

$$E_c - E_p = qE_{te}(d^4 - d^2) + n(qE_e) \quad (3)$$

其中 E_c 代表簇头节点能量消耗, E_p 代表普通监测节点能量消耗, n 代表向簇头节点传输数据的节点个数.

由此可见,簇头节点比普通节点能量消耗更多的,充当簇头的节点更容易造成节点的能量消耗,导致节点无法正常工作,影响整个无线传感器(WSN)网络的寿命.

4 节点调度算法(NSA)

4.1 最大独立集介绍

定义 4 (最大独立集):独立集是指节点集合 S 中两两互不重叠的节点构成的集合,其中集合中节点个数最多的称之为最大独立集,简称最大集.

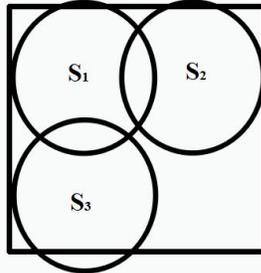


图 3 节点分布图

在图 3 当中,组成的独立集为 $C = \{s_2, s_3\}$,此集合内节点互不重叠,且节点个数最多,则称其为最大独立集,简称最大集.

研究表明,寻找最大独立集是一个 NP 问题,本文提出了基于贪心算法的策略来寻找最大独立集,通过找到的最大独立集,对最大独立集的覆盖率进行计算,判断是否满足覆盖率要求,如果不满足覆盖率,则向已经计算出来的最大独立集里面添加节点,如果满足覆盖率则判断是否可以删除冗余节点.如果刚好满足覆盖监测要求,则直接激活此集合充当监测节点.其余节点轮流充当簇头节点.

4.2 最大独立集生成算法(MSS)

本文的目的是以一种相对快速的方式找到最大独立集,以最大独立集为基础,通过增加和删除一些节点,进而建立一个满足部分覆盖要求的无线传感器网络.

MSS 算法通过采取贪心的策略,通过节点与其他节点的重叠个数多少找出最大独立集 C .首先用矩阵 M 来建立重叠关系矩阵,如果判断两个节点 s_i, s_j 间有重叠关系,则 $E(s_i, s_j) = 1$,

否则 $E(s_i, s_j) = 0$ 其次对目标区域里的所有节点按照相交节点个数 $N(s_j)$ 的大小,从小到大排序形成排序表 $Q < q_a > = < q_1, q_2, \dots, q_n >$ 使用 $C_{k,i}$ 表示每次迭代产生的独立集合,其中 k 表示迭代的次数, i 表示产生的分支编号; $P_{k,i}$ 表示以 Q 为初值的禁忌排序候选表,当按序选择节点 s_j 加入 $C_{k,i}$ 后,则从 $P_{k,i}$ 当中删除 s_j 和 $int(s_j)$. 直到 $P_{k,i}$ 为空时则选择节点个数最多一支做为最大独立集输出,如果有个数相同的情况则随机一个作为最大独立集输出,未加入此集合的节点加入 C' 集合.

初始化候选独立集 $C_{k,i} = \{\emptyset\}$; 初始时候选禁忌排序表 $P_{k,i} = Q$, 当排序候选表 $P_{k,i}$ 为空时,结束本分支计算; 每次从 $P_{k,i}$ 依照从小到大的顺序选择节点 s_j 加入候选最大独立集 $C_{k,i}$, 若 $N(s_j) = N(s_{j+1})$, 且 $E(s_j, s_{j+1}) = 0$, 则 $C_{k,i} = s_j + s_{j+1}$; 否则产生分支分别为 $C_{k,i}, C_{k,i+1}$, 分别更新每个分支对应的候选禁忌排序表 $P_{k,i}, P_{k,i+1}$, 再从各自更新的候选禁忌排序表 $P_{k,i}$ 当中选择下一个节点 s_{j+1} 加入各自的候选独立集, 直到对应的 $P_{k,i}$ 表为空时则结束此分支; 以下给出 MSS 算法描述:

算法 1

- ① 输入: 传感器节点集 $S = \{s_1, s_2, \dots, s_n\}$ 和目标区域 R .
- ② 标记: $C_{k,i}$: 每次迭代产生的独立集; $P_{k,i}$: 禁忌排序候选表; Q : 排序表;
- ③ 首先: 初始化 $C_{0,0} = \{\emptyset\}, P_{0,0} = Q$;
- ④ While($P_{k,i} \neq \{\emptyset\}$)
- ⑤ 从 $P_{k,i}$ 选择 s_j 加入 $C_{k,i+1}$
- ⑥ If($N(s_j) = N(s_{j+1}) \parallel E(s_j, s_{j+1}) = 0$)
- ⑦ $C_{k,i} = C_{k,i} \cup \{s_j, s_{j+1}\}$ 更新对应 $P_{k,i}$ // 从 $P_{k,i}$ 删除 s_j 和 s_j 有重叠的节点.
- ⑧ Else
- ⑨ $C_{k,i} = s_j$, 更新对应 $P_{k,i}$ // 产生分支
- ⑩ $C_{k,i+1} = s_{j+1}$ 更新对应 $P_{k,i+1}$
- ⑪ End if
- ⑫ $C = \max(C_{k,i})$
- ⑬ Return C // 返回节点个数最多的集合, 如果元素个数相等则随机返回
- ⑭ End while

4.3 节点覆盖调度算法

在最大独立集的基础上, 算出此最大独立集的覆盖率, 由于不同的应用环境, 具有不同的覆盖要求, 因此在 MSS 算法的基础上, 需要进行增删节点操作. 节点调度算法(NSA)描述如下:

算法 2

- ① If($\frac{A(C)}{A(R)} = \eta$) 直接激活集合 C 进行监测, 从 C' 中选择节点轮流充当簇头

- ② If($\frac{A(C)}{A(R)} < \eta$) // η 代表最低覆盖要求
- ③ 从未加入集合 C' 的节点当中选择与 C 产生节点贡献最大的节点加入其中
- ④ Until $\frac{A(C)}{A(R)} > \eta$
- ⑤ If($\frac{A(C)}{A(R)} > \eta$)
- ⑥ 删除贡献区域最少的节点
- ⑦ Until
- ⑧ 再删任意一个节点,都无法满足 $\frac{A(C)}{A(R)} > \eta$
- ⑨ End if
- ⑩ 得到激活节点集合 C 进行监测工作
- ⑪ If($C' \neq \emptyset$)
- ⑫ 从 C' 选择一个节点充当簇头节点
- ⑬ Else
- ⑭ 从 C 选择节点轮流选择节点充当簇头

4.4 举例

本文以二维矩阵,表示节点与节点之间的相交关系,如果两个节点 s_i 与 s_j 之间有重叠, $E(s_i, s_j) = 1$, 否则 $E(s_i, s_j) = 0$.

以 7 个节点为例,建立相交关系矩阵 M ;

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

首先根据矩阵 M 中节点的相交个数 $N(s_j)$ 的大小对节点矩阵进行从小到大排序,初始化形成排序表 $Q = \langle s_1, s_2, s_7, s_3, s_4, s_6, s_5 \rangle$, $P_{0,0} = Q$, $C_{0,0} = \{\emptyset\}$.

其次选择节点 s_1 加入候选独立集 $C_{1,0} = (s_1)$, 同时更新 $P_{1,0} = P_{0,0} - \langle s_1, s_2 \rangle$, 此时从 $P_{1,0}$ 当中依次选择节点 s_7 加入候选独立集 $C_{2,0} = \{s_1, s_7\}$, 同时更新 $P_{2,0} = P_{1,0} - \langle s_7, s_5, s_6 \rangle$, 此时继续从 $P_{2,0}$ 中继续选择节点加入候选独立集, 此时由于 $N(s_3) = N(s_4)$ 且 $E(s_3, s_4) = 1$, 则产生分支 $C_{3,0} = (s_1, s_7, s_3)$, $C_{3,1} = (s_1, s_7, s_4)$. 分别更新对应的候选表 $P_{3,0} = P_{2,0} - \langle s_3, s_4 \rangle$, 此时 $P_{3,0}$ 为空,

则停止更新, $P_{3,1} = P_{2,0} - \langle s_3, s_4 \rangle$ 此时, $P_{3,1}$ 为空, 停止更新. 最终产生 2 个最大独立集 $C_{3,0} = (s_1, s_7, s_3)$ 与 $C_{3,1} = (s_1, s_7, s_4)$, 输出集合个数最多的分支, 如果有多个最大独立集, 则随机输出一个集合作为最终结果 C 输出, 其余未加入 C 的节点进入集合 C' .

得出的最大独立集 C , 进行覆盖率计算, 如果 $\frac{A(C)}{A(R)} < \eta$, 从剩余未激活的节点集 C' 当中选

择与最大独立集 C 节点产生覆盖冗余最少的节点加入节点集 C 当中, 直到满足覆盖要求, 当满足覆盖要求后, 对激活节点集进行冗余判定, 删除激活节点集当中贡献区域最少的节点, 直到删除任意节点都无法满足覆盖要求, 则输出激活节点集, 此集合作为激活节点负责数据监测.

5 实验

5.1 仿真

本文使用 C++ 与 MATLAB 进行模拟分析, 下图通过 C++ 产生的实验数据图, 图 4 描述了 NSA 机制. 假定传感节点分布于 $R=400 \times 400$ 区域, 传感器半径取值范围为 50~90 单位, 粒度 b 为 20 单位, 区域覆盖要求为 80%.

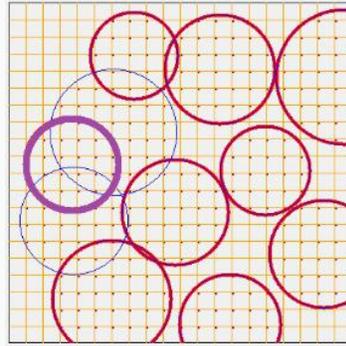


图 4 激活节点图

实验结果如图 4 所示, 首先通过 MSS 算法得出最大独立集(图 4 中红色与紫色节点), 由于不满足覆盖要求 η ($\eta=80\%$), 则通过 NSA 算法需要进行增加节点(图 4 中蓝色节点). 通过 NSA 算法得到满足覆盖要求的节点后, 进行覆盖判断, 当大于覆盖要求时, 进行冗余节点(图 4 中紫色节点)的删除操作, 此时无法再删除任意节点, 则作为最终激活节点集(图 4 中蓝色与红色节点), 进行区域的监测节点, 从休眠节点当中轮流选择节点充当簇头节点.

5.2 节点激活率仿真

由于节点是随机撒播, 如果完全覆盖区域 R 会导致抛洒的节点数目趋向于无穷大, 所以当覆盖率达到 95% 则认定为是对目标区域 A 的完全覆盖. 本文使用 θ 表示占总抛洒节点比例, η 表示目标区域的覆盖要求. 为了更好的分析 MSS 算法, 选择文献[11]所提出的 CDS 算法, 文献[12]提出的 DFS 算法作比较并进行仿真. 覆盖率 η 分别等于 $\eta=0.6$, $\eta=0.8$. 传感器半径取值范围为 50~90 单位, 粒度 b 为 20 单位. 实验数据如图所示.

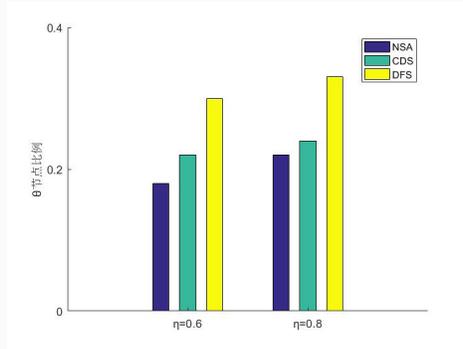


图5 不同覆盖率激活节点对比图

如图5所示,通过与CDS和DFS算法相比,本文提出的NSA算法的在 $\eta=0.6, \eta=0.8$ 的情况下,激活节点率相比较低,由于本文面向的是部分覆盖,并非是全覆盖问题,所以未考虑全覆盖情况。

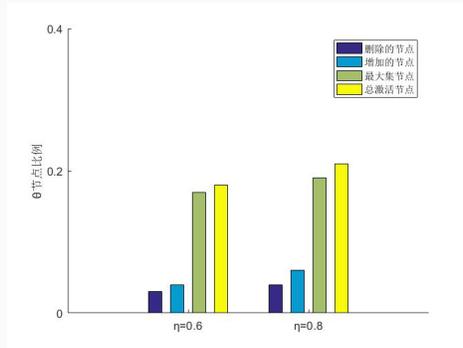


图6 增删节点率

图6是在生成最大独立集后,增加与删除节点的比例图.由图6可以看出,在生成最大独立集后,仅需要增加或者删除极少的节点就能满足覆盖要求。

5.3 网格粒度实验

在覆盖率分别为 $\eta=0.6, \eta=0.8$ 的情况下.在实验当中,使用的区域大小与传感器半径均是固定值,单纯使用粒度 b 的大小无法体现粒度与半径的关系,因此引用变量粒度比 $B = \frac{b}{r}$, b 代表网格单位粒度, r 是传感器半径,由于节点半径不同,这里取节点半径平均值.实验结果如图7所示:

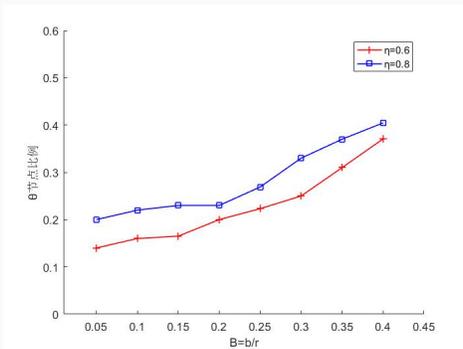


图7 不同网格粒度的节点激活率

根据图 7 结果所示,当 B 的比值越小,计算越精确,节点激活率较低.当粒度 B 增大时,节点激活率会增高.原因在于当 B 较小时,为了满足 95%的初始覆盖,播撒节点过于密集,所以激活节点占总抛洒节点比重低.随着粒度比增高,满足初始覆盖的总抛洒节点个数也会下降,当粒度继续增大时,抛洒节点个数会偏少,激活的节点所占比例会增大.

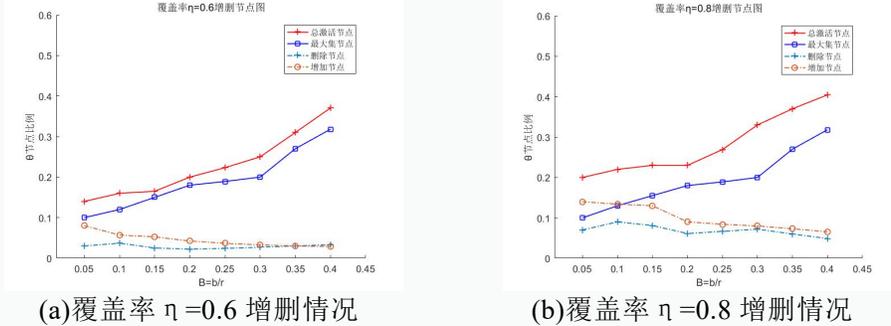


图 8 不同粒度的节点增删图

如图 8(a)与图 8(b)所示当粒度比较小时,节点冗余判断过于敏感,影响最大集的生成,导致增删操作较为频繁,当粒度比增大时,虽然总激活的节点比例逐渐增高,但是只需要较少增加或删除节点,因此较大的粒度比与较小的粒度都会影响算法性能,所以当粒度比在 0.2 左右时,效果较佳.

6 总结

本文讨论了无线传感器网络(WSN)的部分覆盖问题,由于无线传感器网络(WSN)在随机抛洒过程中节点监测区域相互重叠易产生大量的覆盖冗余,针对此问题提出了 NSA 算法,首先创立最大独立集,然后以此集合为基础,针对不同覆盖率的要求,节点通过计算与最大独立集的冗余情况,进行增删操作,进而达到覆盖要求.根据实验数据显示,NSA 能够有效地减少活动节点数,使得更多节点进入休眠状态,从而达到延长整个网络寿命的目的.其次,本文也通过调整网格粒度之比,对冗余区间进行度量,进而通过实验得出网格粒度的最佳取值范围.

参考文献:

- [1] 魏永红,李科杰. 层次拓扑结构的无线传感器网络能量模型[J]. 计算机应用, 2010, 30(07):1731-1735.
- [2] 刘晓爽. 无线传感器网络覆盖与优化技术研究[D]. 2015.
- [3] 王珺. 无线传感器网络能量有效性的研究[D]. 南京大学, 2012.
- [4] Heinzelman W R, Chandrakasan A P, Balakrishnan H. Energy-efficient communication protocol for wireless sensor networks[C]// System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on. IEEE, 2000.
- [5] 蒋杰,方力,张鹤颖, et al. 无线传感器网络最小连通覆盖集问题求解算法[J]. 软件学报, 2006, 17(2):175-184.
- [6] 杨勇,夏士雄,周勇. WSN 中能量有效的分层协作覆盖模型[J]. 软件学报, 2012, 23(10):2783-2794.
- [7] Slijepcevic S, Potkonjak M. Power efficient organization of wireless sensor networks. [C]// IEEE International Conference on Communications. IEEE, 2001.
- [8] Chen H N, Wu H Y, Tzeng N F. Grid-based approach for working node selection in wireless sensor networks[C]// Communications, 2004 IEEE International Conference on. IEEE, 2004.
- [9] Mostafaei H, Montieri A, Persico V, et al. A sleep scheduling approach based on learning automata for WSN partial coverage[J]. Journal of Network and Computer Applications, 2017, 80:67-78.
- [10] 舒坚,郭凯,刘群. 机会传感网络连通性参数研究[J]. 计算机学报, 2016, 39(5):1067-1080.
- [11] 马华东,袁培燕,赵东. 移动机会网络路由问题研究进展[J]. 软件学报, 2015, 26(3):600-616.
- [12] 崔莉,鞠海玲,苗勇, et al. 无线传感器网络研究进展[J]. 计算机研究与发展, 2005, 42(1):163-174.
- [13] 任丰原,黄海宁,林闯. 无线传感器网络[J]. 软件学报, 2003, 14(7).

- [14] 郑国强, 李建东, 李红艳, et al. 多跳无线传感器网络的高效中继节点快速选择算法[J]. 通信学报, 2010, 31(11).
- [15] 陆克中, 孙宏元. 无线传感器网络最小覆盖集的贪婪近似算法[J]. 软件学报, 2010, 21(10).
- [16] 徐鹏飞, 陈志刚, 邓晓衡. 无线传感器网络中的分布式 Voronoi 覆盖控制算法[J]. 通信学报, 2010(8):16-25.