

无线传感器网络最小连通覆盖集问题求解算法*

蒋杰¹⁺, 方力², 张鹤颖¹, 窦文华¹

¹(国防科学技术大学 计算机学院, 湖南 长沙 410073)

²(国防科学技术大学 网络信息中心, 湖南 长沙 410073)

An Algorithm for Minimal Connected Cover Set Problem in Wireless Sensor Networks

JIANG Jie¹⁺, FANG Li², ZHANG He-Ying¹, DOU Wen-Hua¹

¹(School of Computer, National University of Defense Technology, Changsha 410073, China)

²(Network Information Center, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: Phn: +86-731-4575830, Fax: +86-731-4575830, E-mail: jiangjie@nudt.edu.cn, <http://www.nudt.edu.cn>

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. <http://www.jos.org.cn/1000-9825/17/175.htm>

Abstract: Reducing power consumption to extend network lifetime is one of the most important challenges in designing wireless sensor networks. One promising approach to conserving system energy is to keep only a minimal number of sensors active and put others into low-powered sleep mode, while the active sensors can maintain the communication connectivity and cover the target region completely. The problem of computing such minimal active sensor set is NP-hard. In this paper, a centralized Voronoi tessellation (CVT) based approximate algorithm is proposed to construct a near optimal cover set of active sensors required to cover the target region completely. The communication graph induced by the cover set computed by CVT algorithm is connected if sensor's communication radius is at least twice of its sensing radius. In case of sensor's communication radius is smaller than twice of its sensing radius, a minimum spanning tree (MST) based connection algorithm is proposed to ensure the communication connectivity of the cover set. Finally, the performance of the proposed algorithm is evaluated through theoretical analysis and extensive numerical experiments. Experimental results show that the proposed algorithm outperforms the greedy algorithm in terms of the runtime and the size of the constructed connected cover set.

Key words: WSN (wireless sensor network); network lifetime; MCCS (minimal connected cover set); Voronoi tessellation; MIS (maximal independent set); MST (minimum spanning tree)

摘要: 降低能耗以延长网络生存时间是无线传感器网络设计中的一个重要挑战。在传感器节点高密度部署的环境中,在保证网络性能的前提下,仅将最少量的节点投入活跃工作状态,而将其余节点投入低功耗的睡眠状态,是一种节约系统能量的有效方法。如何计算同时满足“覆盖要求”(工作节点必须能够完全覆盖目标区域)和“连通性要求”(工作节点组成的通信网络必须是连通的)的最小节点集合,是一个NP难问题。设计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

* Supported by the National Natural Science Foundation of China under Grant No.90104001 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2003CB314802 (国家重点基础研究发展规划(973))

Received 2005-03-10; Accepted 2005-08-03

节点集.当节点通信半径大于等于 2 倍感知半径时,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(minimum spanning tree,简称 MST)的连通算法来计算确保 CVT 算法构造的覆盖集连通所需的辅助节点.理论分析和实验数据表明,CVT(+MST)算法的性能在时间复杂性和连通覆盖集大小方面都优于已有的贪婪算法.

关键词: 无线传感器网络;网络生存时间;最小连通覆盖集;Voronoi 划分;最大独立集;最小生成树
中图分类号: TP393 文献标识码: A

随着微传感器技术、无线网络技术以及嵌入式处理技术的发展,无线传感器网络(wireless sensor network,简称 WSN)在民用和军事领域得到了广泛的应用.典型应用包括环境监测、森林火灾预警、生物和化学攻击检测、战场态势感知等^[1-4].

由于体积和成本的限制,传感器节点通常采用容量有限的电池提供能量.此外,由于传感器网络中节点数量众多,或者由于传感器网络通常部署在战场、沙漠等危险环境中,替换节点电池或者对电池进行充电都是不可行的;另一方面,人们总是希望传感器网络的生存时间能够尽可能长.如何在单个节点能量受限、生存时间较短的情况下延长整个网络系统的生存时间,是无线传感器网络设计中的一个重要挑战.

密度控制是实现上述目的的重要而有效的手段.所谓密度控制,就是在不牺牲系统性能的前提下,将一部分节点投入低功耗的睡眠状态,只保留部分节点作为活跃工作节点.这样,可以降低网络中活跃节点的密度、降低感知数据的冗余性以及减少无线通信冲突和干扰,从而降低整个系统的能量消耗.当某个工作节点由于能量耗尽等原因而失效时,邻近的睡眠节点可以恢复工作代替该失效节点,继续维持网络的正常工作.因此,密度控制可以有效地延长无线传感器网络的生存时间.为保持网络原有性能,密度控制必须满足以下两个条件:(1) 保证目标区域的完全覆盖;(2) 保证通信网络的连通性,即所有工作节点组成的通信网络必须仍然是连通的.

文献[5]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.寻找最大数量的节点集是一个 NP 完全问题,Slijepcevic 和 Potkonjak 为此设计了一种集中式的启发式算法.文献[6]讨论了上述问题的一个变种.文献[7]提出了一种基于探测的分布式节点密度控制算法.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.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.文献[10]提出了一种分布式算法,以期在保持网络原始覆盖的前提下提高节点的能量利用效率并延长网络生存周期.文献[11]设计了能为不同区域提供不同监视质量的节点调度方法.上述工作在选择工作节点时都只考虑了覆盖要求,而没有考虑通信网络的连通性问题.文献[12]提出了一种解决 MCCS(minimal connected cover set)问题的集中式贪婪算法,与本文的工作最为接近.但本文提出的算法性能优于该贪婪算法.本文第 3 节给出了具体的性能比较.

本文讨论了如何在满足上述覆盖性和连通性要求的前提下,选择最少数量的工作节点问题.对于一个给定的目标区域,这是一个 NP 难问题.本文设计了一种多项式时间近似算法.算法的基本思想是:首先计算满足覆盖要求的近似最优的覆盖集.当节点的无线通信半径大于等于其 2 倍感知半径时,该覆盖集是连通的;当节点的无线通信半径小于 2 倍感知半径时,不能保证该覆盖集的连通性.为此,本文设计了一种基于最小生成树的算法来计算确保该覆盖集连通所需的辅助节点.最后,本文分析了算法的时间复杂性,并通过实验对算法的性能进行了评估.理论分析和实验数据表明,算法在时间复杂性方面优于贪婪算法,而且能求得更小的连通覆盖集.

1 问题描述

在二维平面 R^2 上,节点 s_i 的覆盖(感知)范围是以节点为圆心、半径等于感知半径 R_s 的一个圆形区域(称为“感知圆盘”),用 $S_i = \{p \in R^2 \mid d(p, s_i) \leq R_s\}$ 表示.其中 d 为欧氏距离.整个传感器网络的覆盖范围 C 是网络中所有节点覆盖范围的并集,用 $C = \bigcup_{i \in \{1, n\}} S_i$ 表示.点 p 被节点 s_i 覆盖,等价于 $d(p, s_i) \leq R_s$.目标区域 R 被传感器网络完全覆盖,

等价于 R 中的每个点至少被 1 个节点覆盖.节点 s_i 的直接通信范围为一个以 s_i 为圆心、通信距离 R_c 为半径的圆形区域.每个节点只能与位于该区域内的节点直接通信.

定义 1. 给定一个传感器节点集合 S .由该集中全部节点组成的通信网络 $G_c=(V_c,E_c)$ 是一个无向图,其中 $V_c=S, \forall s_i,s_j \in S, (s_i,s_j) \in E_c$ 当且仅当 $d(s_i,s_j) \leq R_c$.称通信图 G_c 由集合 S 导出.图 G_c 中的一条通信路径 $P=\{s_1,s_2,s_3,\dots\}$ 由一系列节点组成,其中每一对相邻节点互为通信邻居.如果通信图 G_c 中任意两个节点之间都存在一条通信路径,则称该通信图是连通的.

定义 2. 给定一个传感器节点集合 S 和目标区域 R .如果 R 内的每一个点至少被集合 S 内的一个节点覆盖,则称 S 为 R 的覆盖集.如果由集合 S 导出的通信图同时是连通的,则称集合 S 是 R 的连通覆盖集(connected cover set,简称 CCS).

在图 1 中,集合 $\{s_1,s_2,s_3,s_4,s_5\}$ 是目标区域(虚线围成的区域)的一个覆盖集.但该覆盖集不是连通的.加入节点 s_6,s_7,s_8 后才形成一个连通覆盖集.

最小连通覆盖集(MCCS)问题.考虑在目标区域 R 上部署的一个传感器网络 S .最小连通覆盖集问题就是寻找一个最小子集 $S' \subseteq S$,使得 $R \subseteq \bigcup_{s_i \in S'} S_i$ (即 S' 完全覆盖区域 R)并且由 S' 导出的通信图 G'_c 连通.

文献[12]指出 MCCS 问题是一个 NP 难问题.下一节给出求解 MCCS 问题的近似算法.

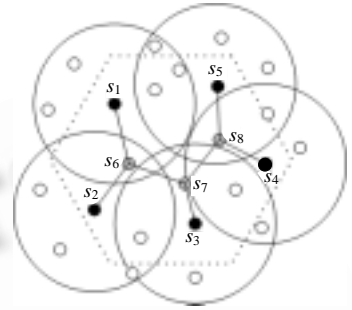


Fig.1 Connected cover set

图 1 连通覆盖集

2 近似算法

在本文给出的近似算法中,连通覆盖集的构造通过两个步骤完成.首先,在目标区域 R 的 Voronoi 划分基础上求出 R 的近似最小覆盖集.当 $R_c \geq 2R_s$ 时,该覆盖集同时就是连通覆盖集;当 $R_c < 2R_s$ 时,本文设计了一种基于最小生成树的算法,用于计算确保该覆盖集连通所需的辅助节点.

在算法设计过程中,假设每个节点知道自身的坐标信息.节点可以使用 GPS 或者其他定位手段获取位置信息.关于传感器网络的定位问题,在文献[13-15]中有具体的论述.此外,节点同构即每个节点的感知半径 R_s 相同、通信半径 R_c 相同,但比值 R_c/R_s 可变.目标区域是一个有界凸区域,并且初始网络是该区域的一个连通覆盖集.

2.1 CVT(centralized Voronoi tessellation)算法

CVT 算法用于构造近似最优覆盖集.其基本思想是:首先识别出网络中的冗余节点;然后计算出可以同时被关闭的冗余节点;最后,由所有的活跃工作节点组成目标区域的覆盖集.

2.1.1 识别冗余节点

定义 3. 如果一个节点的覆盖范围完全被其他节点的覆盖范围所包含,则称该节点为冗余节点.

定义 4. 目标区域中不被任何节点覆盖的点(或点集)称为覆盖漏洞.

一个节点为冗余节点,等价于关闭该节点后不会在网络中形成覆盖漏洞.覆盖漏洞的检测借助于目标区域的 Voronoi 划分.

定义 5. 给定二维平面上 R^2 的一个有限点集 $S=\{s_1,s_2,\dots,s_n\}$,定义与 s_i 相关联的 Voronoi 区域 $V_i:V_i=\{p \in R^2 \mid d(p,s_i) \leq d(p,s_j), j \neq i\}$,其中 d 为欧氏距离.点集 $\{s_i\}_{i=1}^n$ 称为 Voronoi 产生点,Voronoi 区域的边称为 Voronoi 边,其顶点称为 Voronoi 顶点,共享一条 Voronoi 边的两个点互为 Voronoi 邻居.集合 $\{V_i\}_{i=1}^n$ 称为 R^2 的 Voronoi 划分.Delaunay 三角剖分是 Voronoi 划分的对偶图.用 $G_D=(V_D,E_D)$ 表示点集 $S=\{s_1,s_2,\dots,s_n\}$ 形成的 Voronoi 划分所对应的 Delaunay 三角剖分.其中 $V_D=S, (s_i,s_j) \in E_D$ 当且仅当 s_i 与 s_j 互为 Voronoi 邻居^[16].

考虑部署在有界凸区域 R 上的传感器网络 $S=\{s_1,s_2,\dots,s_n\}$.将传感器节点作为 Voronoi 产生点,可以得到 R 的唯一 Voronoi 划分.在二维平面上,每个 V_i 均是凸多边形.如果 s_i 位于集合 S 的凸壳边界上,与其关联的 Voronoi 多边形无界^[13].但如果 R 有界,则无界 Voronoi 多边形将在目标区域边界的限定下变为有界.此时,所有 Voronoi

多边形都有界,从而得到有界 Voronoi 划分.

定义 6. 如果一个 Voronoi 划分中的 Voronoi 多边形全部有界,则该 Voronoi 划分称为有界 Voronoi 划分.

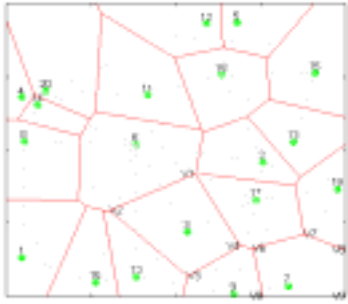


Fig.2 Bounded Voronoi tessellation
图 2 有界 Voronoi 划分

在下文中,使用 $BVT(S,R)$ 表示以传感器节点集合 S 为产生点构造的区域 R 的有界 Voronoi 划分,用 $BVC_i(S,R)$ 表示与节点 s_i 相关联的有界 Voronoi 多边形, $BVV_i(S,R)$ 表示 $BVC_i(S,R)$ 的顶点集,用 $BNV_i(S,R)$ 表示 s_i 的 Voronoi 邻居集.位于集合 S 凸壳边界上的节点称为边界节点,其他节点称为内部节点.内部节点 s_i 的 BVV_i 仅仅包括其 Voronoi 顶点.

图 2 是 20 个随机产生点形成的有界 Voronoi 划分. s_3 为内部节点, $BVV_3(S,R)=\{V_1,V_2,V_3,V_4\}$.若 s_i 为边界节点, BVV_i 不但包括 Voronoi 顶点,还包括其 Voronoi 边与目标区域边界之间的交点以及目标区域边界顶点. s_7 为边界节点, $BVV_7(S,R)=\{V_5,V_6,V_7,V_8,V_9\}$.其中, V_5 和 V_8 是其 Voronoi 边与目标区域边界之间的交点, V_9 是目标区域的顶点.在图 2 中,相应的 Delaunay 三角剖分用虚线表示.

在有界 Voronoi 划分中,每个 Voronoi 多边形都是凸多边形.注意到以下事实:一个凸多边形被一个圆完全包含,等价于该凸多边形的每一个顶点都位于该圆内部.

引理 1. 节点 s_i 的 $BVC_i(S,R)$ 被其感知圆盘 S_i 完全覆盖等价于 $BVC_i(S,R)$ 的全部顶点都位于 S_i 之内.

换言之,如果对于 $\forall v_i \in BVV_i(S,R), d(s_i, v_i) \leq R_s$ 均成立,则该 Voronoi 多边形完全被 S_i 覆盖.

定理 1. 有界凸区域 R 被传感器网络 S 完全覆盖,等价于网络中每个节点的感知圆盘都包含了与其相关联的有界 Voronoi 多边形的全部顶点.

证明:以传感器节点为产生点,构造 R 的有界 Voronoi 划分.对于任意节点 $s_i \in S$,如果 $BVV_i(S,R)$ 的元素均被 S_i 覆盖,根据引理 1 有 $BVC_i(S,R) \subseteq S_i$,所以 $R = \bigcup_{s_i \in S} BVC_i(S,R) \subseteq \bigcup_{s_i \in S} S_i$.因此,区域 R 被传感器网络完全覆盖.另一方面,如果 R 被 S 完全覆盖,则 R 中每个点至少被一个节点覆盖.根据 Voronoi 划分的定义,每个 Voronoi 顶点必然被其产生点覆盖.

定理 1 给出了判断节点集合 S 是否为区域 R 的覆盖集的方法.节点 $s_i \in S$ 为冗余节点等价于集合 $S \setminus s_i (S \setminus s_i$ 表示从 S 中删除 s_i 后的剩余节点集合)仍然是区域 R 的覆盖集,等价于对于任意节点 $s_j \in S \setminus s_i, BVT(S \setminus s_i, R)$ 划分中的 BVV_j 均被 S_j 完全覆盖.进一步观察到以下事实:只有 $BVC_j(S \setminus s_i, R), s_j \in BNV_i(S, R)$ 会不同于 $BVC_i(S, R)$,其他节点的有界 Voronoi 多边形在两次 Voronoi 划分中不会发生变化,因为从 S 中删除 s_i 并不会改变其 Voronoi 邻居.

冗余节点判别规则. 节点 $s_i \in S$ 为冗余节点等价于对于 $\forall s_j \in BNV_i(S, R), BVV_j(S \setminus s_i, R)$ 的全部元素均位于 S_j 内.

2.1.2 确定活跃工作节点

在根据上述规则判别冗余节点以后,进一步将冗余节点分为“独立冗余”和“依赖冗余”两个子类.

定义 7. 如果一个冗余节点的 Voronoi 邻居节点集中至少包含一个冗余节点,则该冗余节点为依赖冗余节点;如果其全部 Voronoi 邻居均为非冗余节点,则该冗余节点为独立冗余节点.

由于独立冗余节点的覆盖区域完全被其活跃的 Voronoi 邻居节点包含,因此,关闭独立冗余节点不会造成覆盖漏洞.依赖冗余节点的状态必须依赖其 Voronoi 邻居节点的状态.不能同时关闭互为 Voronoi 邻居的依赖冗余节点,否则将可能形成覆盖漏洞.为了解决上述依赖关系,将所有的依赖冗余节点组织成一个图,称为冗余依赖图(redundant dependency graph,简称 RDG).

定义 8. 冗余依赖图 $RDG(V_r, E_r)$ 是一个无向图,其中 V_r 是网络中全部依赖冗余节点的集合.对于 $\forall v_{ri}, v_{rj} \in V_r, (v_{ri}, v_{rj}) \in E_r$ 等价于节点 v_{ri} 和 v_{rj} 互为 Voronoi 邻居.依赖冗余节点的依赖度,等于 RDG 图中与该节点相邻接的边数.

RDG 图是 R 的有界 Voronoi 划分对应的 Delaunay 三角剖分的子图.从 Delaunay 三角剖分 $G_D=(V_D, E_D)$ 中删去所有非冗余节点、独立冗余节点及其关联边,就得到 RDG 图.可以同时被关闭的依赖冗余节点的数目决定了最终覆盖集的大小.而计算可以同时被安全关闭的依赖冗余节点数等价于计算 RDG 图的最大独立集(maximal

independent set,简称 MIS).图的最大独立集问题是一个 NP 难问题^[17].本文采用了与文献[17]中贪婪法类似的算法来寻找可以被同时关闭的依赖冗余节点.该算法首先在 RDG 图中选择一个依赖度最低的节点,将该节点加入集合 Q ,并从 RDG 图中删去该节点、该节点的全部邻居节点以及与这些节点相关联的边.重复这个过程直到 RDG 图为空为止.当算法终止时,集合 Q 构成 RDG 图的最大独立集.以图 3 为例,节点 A 首先被选中并加入集合 Q .然后从图中删去节点 A,B 以及边 AB,BC .接着选中节点 C (因为 C 在新图中度数最低),并删去节点 D 以及边 CD,DE 和 DF .最后,图中只剩下节点 E,F 以及边 EF .此时,可以在 E,F 中随机选择一个节点加入集合 Q .算法终止时,集合 $Q=\{A,C,F\}$ 就形成该图的最大独立集.

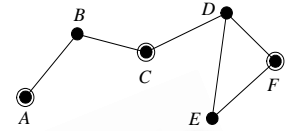


Fig.3 MIS of a graph
图 3 图的最大独立集

被选入 RDG 图最大独立集的依赖冗余节点可以被安全关闭,剩余的冗余依赖节点必须处于工作状态.为了使最终形成的覆盖集尽可能地小,重复上述标识冗余节点和计算最大独立集的过程,直到没有冗余节点可以被安全关闭为止.当算法结束时,所有的独立冗余节点以及最大独立集中的依赖冗余节点组成了可以被安全关闭的节点集合,称为安全集(safe set,简称 SS).其他所有剩余的活跃工作节点就形成目标区域的覆盖集.

下面给出构造区域近似最小覆盖集的 CVT 算法描述.

算法. CVT 算法.

输入:初始传感器节点集 $IS=\{s_1,s_2,\dots,s_n\}$ 和目标区域 R .

输出:区域 R 的覆盖集 CS .

标记: IRS :独立冗余节点集; DRS :依赖冗余节点集; SS :安全节点集; MIS :最大独立集; S :临时节点集.

第 1 步. $SS=CS=\emptyset, S=IS$.

第 2 步. 计算区域 R 的有界 Voronoi 划分 $BVT(S,R)$.

第 3 步. if $(R \not\subset \bigcup_{i \in S} S_i)$, Return $CS=\emptyset$.

第 4 步.

while $(S \neq \emptyset)$

第 4.1 步. $IRS=DRS=MIS=\emptyset$.

第 4.2 步. 识别 S 中的冗余节点.

第 4.3 步. 对冗余节点进行分类:

for each redundant node $s_i \in S$,

if $(s_i$ 为独立冗余节点), $IRS=IRS \cup \{s_i\}$;

else $DRS=DRS \cup \{s_i\}$.

end for;

第 4.4 步.

if $(DRS=\emptyset)$

计算 RDG 图的最大独立集 MIS ;

$SS=SS \cup IRS \cup MIS; S=IS-SS$;

else

$SS=SS \cup IRS$, 并退出 while 循环.

endwhile;

第 5 步. Return $CS=IS-SS$.

2.1.3 算法分析

从 CVT 算法的执行过程中可以看出,最后的覆盖集由初始覆盖集 IS 删除安全节点集 SS 组成.所以, CVT 算法构造的节点集合是目标区域 R 的一个覆盖集.

定理 2. 如果 $R_c \geq 2R_r$, 则 CVT 算法构造的覆盖集是连通的.

证明:考虑最后形成的覆盖集中的两个节点 s_i, s_j . 假定二者互为 Voronoi 邻居(如图 4 所示),且共享边 $\overline{v_1 v_2}$.

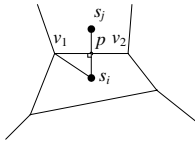


Fig.4 Proof of theorem 2

图 4 定理 2 的证明

根据 Voronoi 划分的定义,边 $\overline{v_1 v_2}$ 是线段 $\overline{s_i s_j}$ 的垂直平分线. 设 $\overline{v_1 v_2}$ 与 $\overline{s_i s_j}$ 相交于点 p . 由于节点 s_i 属于覆盖集, 则其有界 Voronoi 多边形的每个顶点都位于其感知圆盘内, 因此, $d(s_i, v_1) \leq R_s$, 从而 $d(s_i, s_j) = 2d(s_i, p) < 2d(s_i, v_1) \leq 2R_s$. 如果 $R_c \geq 2R_s$, 则 $d(s_i, s_j) < R_c$. 因此 CVT 算法构造的覆盖集中的每个节点都能够与其每一个 Voronoi 邻居节点直接通信, 则对应的 Delaunay 三角剖分图中的每条边长度都小于 R_c . 由于最小生成树是 Delaunay 图的子图, 从而该覆盖集是连通的.

CVT 算法的近似性能取决于求解图最大独立集问题的贪婪算法. 对于顶点数为 n , 边数为 l , 平均节点度为 δ 的任意图, 贪婪算法能够在最多 $O(l)$ 步内求得大小至少为 $n/(\delta+1)$ 的最大独立集^[17].

引理 2. 在二维平面 Voronoi 划分中, Voronoi 顶点数最大不超过 $2n-5$, 其中 n 为产生点数目^[16].

引理 3. 每个 Voronoi 多边形的平均 Voronoi 边数不超过 $6^{[16]}$.

引理 4. 构造 n 个产生点的平面 Voronoi 划分, “扫平面算法”最坏时间复杂性为 $O(n \log n)$, 平均时间复杂性为 $O(n)^{[16]}$.

定理 3. CVT 算法的最坏时间复杂性不超过 $O(mn^2 \log n)$, 平均时间复杂性不超过 $O(mn^2)$. 其中 m 是 CVT 算法第 4 步的最大循环次数, 并且 $m \leq \log_{7/6} n$, n 为最初部署的传感器节点数.

证明: 首先讨论最坏情况. CVT 算法使用“扫平面法”构造平面 Voronoi 划分. 算法第 2 步需要 $O(n \log n)$ 时间. 第 3 步需要 $O(n)$ 时间来判断初始网络部署是否构成目标区域的一个覆盖集. 步骤 4.2 需要 $O(n^2 \log n)$ 时间来识别所有冗余节点. 步骤 4.3 需要 $O(n)$ 时间. 步骤 4.4 需要 $O(k \log k)$ 时间来构造 RDG 图, 需要 $O(k)$ 时间来计算最大独立集 ($k = |DRS|$). 显然 $k \leq n$, 所以步骤 4.4 能在不超过 $O(n \log n)$ 时间内完成. 假定第 4 步的循环次数为 m , 则第 4 步的整个运行时间不超过 $O(mn^2 \log n)$. 根据引理 3 以及贪婪算法求解图的最大独立集的近似比, 可以得到在最坏情况下 $m \leq \log_{7/6} n$. 由于 CVT 算法的时间复杂性取决于步骤 4.2, 所以关于最坏时间复杂性的命题成立. 平均情况的时间复杂性分析与上述过程类似, 不同之处在于, 构造 n 个产生点的平面 Voronoi 划分平均需要 $O(n)$ 时间, 而不是 $O(n \log n)$.

2.2 基于最小生成树(minimum spanning tree, 简称MST)的连通算法

根据定理 2 的证明过程, 当 $R_c < 2R_s$ 时, 不能保证 CVT 算法构造的覆盖集的连通性. 为此, 本节给出一种基于图最小生成树(MST)的连通算法, 用于计算确保由 CVT 算法构造的覆盖集连通所需的辅助节点. 下文中, 基本节点是指由 CVT 算法构造的覆盖集中的节点, 其他节点称为辅助节点. 原始通信网络是网络中所有节点都处于活跃工作状态时形成的通信网络.

定义 9. 由 CVT 算法构造的覆盖集导出的通信图的最大连通子图称为基本连通片. 基本连通片仅由基本节点以及基本节点之间的边组成. 两个基本连通片之间的辅助路径是连接两个不同基本连通片的一条通信路径, 该路径除了两个端节点以外, 其他中间节点均为辅助节点.

假定由 CVT 算法构造的覆盖集包含 k 个分离的基本连通片(如图 5(a)所示, 基本节点由实心圆点表示), 这些基本连通片通过辅助节点(空心圆点表示)连接起来. 将每个基本连通片抽象成一个虚拟节点, 用 v_i 表示基本连通片 $comp_i$. 然后, 将原始通信图 G_c 中连接 $comp_i$ 中基本节点与辅助节点的边复制到 v_i 上, 并保持该边另一端的辅助节点不变. 通过这样的变换, 得到一个抽象通信图 $G'_c = (V'_c, E'_c)$ (如图 5(b)所示). 原始通信图 G_c 中连接两个基本连通片的最短辅助路径等价于抽象图 G'_c 中连接相应虚拟点的最短辅助路径. 使用宽度优先搜索(breadth-first search, 简称 BFS)算法可以在线性时间内计算出图 G'_c 中顶点 v_i 和 v_j 之间的最短辅助路径^[18].

计算出所有虚拟节点之间的最短辅助路径之后, 按如下方式创建加权图 $G_w = (V_w, E_w)$: 顶点集 V_w 是由所有代表基本连通片的虚拟节点组成, 因此, $|V_w| = k$. $\forall v_i, v_j \in V_w, e_{ij} = (v_i, v_j) \in E_w$ 当且仅当在图 G_c 中存在一条连接基本连通片 $comp_i$ 和 $comp_j$ 的辅助路径. 边 e_{ij} 的权重为图 G'_c 中 v_i 与 v_j 之间的最短辅助路径的长度. 路径长度等于路径上的辅助节点数. 图 5(c)是图 5(b)对应的加权图, 图中每条边旁边的数值是该边的权值, 表示最短辅助路径的长度;

括号中的字符串是该最短辅助路径上辅助节点的 id 标识.加权图 G_w 的最小生成树可用 Kruskal 算法求得.最后,与该最小生成树的边对应的辅助节点就是确保 CVT 算法构造的覆盖集连通所需要的辅助节点.

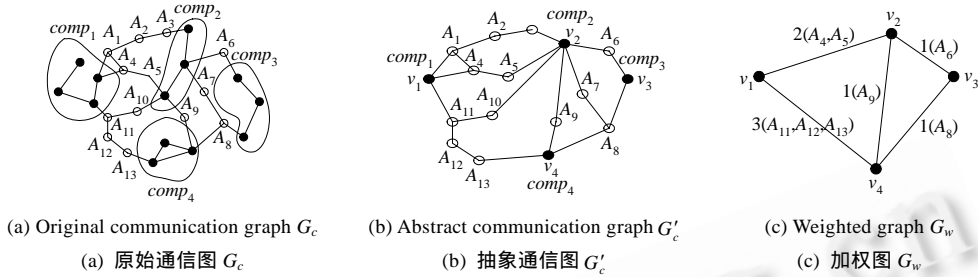


Fig.5 Constructing weighted graph based on original communication graph

图 5 从原始通信图构建加权图

将基于 MST 的连通算法描述如下:

算法. 基于 MST 的连通算法.

第 1 步. 在原始通信图的基础上构建加权图 G_w ;

第 2 步. 计算 G_w 的最小生成树;

第 3 步. 返回与该最小生成树的边相对应的辅助节点.

使用 BFS 算法来计算抽象通信图中任意两个基本连通片之间的最短辅助路径需要 $O(|V'_c| + |E'_c|)$ 时间.如果基本连通片的个数为 k ,则可以在 $O(k^2(|V'_c| + |E'_c|))$ 时间内得到加权图 G_w .Kruskal 算法可以在 $O(k \log k)$ 时间内计算出 G_w 的最小生成树.基于 MST 的连通算法的时间复杂性取决于加权图 G_w 的构造,因此算法的时间复杂性不超过 $O(k^2(|V'_c| + |E'_c|))$.

3 性能评估

为了评估算法性能,我们用 C 语言在 Linux 操作系统下设计了一个仿真程序.在仿真实验中,目标区域是一个 400×400 的矩形区域.节点使用在第 1 节中描述的感知模型和无线通信模型.默认 $R_s=30$.通过改变 R_c 实现不同的 R_c/R_s 比值,从而评估算法在不同环境下的性能.主要的性能指标是最终计算的连通覆盖集 CCS 的大小.CCS 越小,系统的总体能耗越小.为了保证初始部署的节点能够完全覆盖目标区域,并形成一个连通的通信网络,将目标区域分割为若干个边长为 40 的网格,并在每个网格的中心固定部署一个节点.在这样的部署方式下,能保证网络连通的最小 R_c 为 40.当 $R_s=30$ 时,位于网格中心的节点能够完全覆盖其所在网格,从而形成一个初始连通覆盖集.在这种设置下,最小连通覆盖集的大小近似为 100.除了这 100 个固定节点之外,仿真实验的每一个网络拓扑还包括大量随机部署的节点.每一个仿真结果均为 50 次仿真实验的平均值.

3.1 与贪婪算法的比较

文献[12]提出了解决 MCCS 问题的集中式贪婪算法.该算法使用网格作为目标区域的近似,并将网格点划分为若干个子区域,每个子区域被相同的传感器节点集覆盖.二维平面上 n 个圆所能形成的子区域的最大个数是 $n(n-1)+1$ [12].若要将 k 个采样点划分进不同的子区域,则产生子区域的预处理过程的时间复杂性为 $O(kn^3)$ [9].使用网格作为目标区域的近似,采样点的个数与目标区域的大小以及网格尺寸有关.而网格尺寸与覆盖精度密切相关.网格尺寸越大,则采样点的个数 k 越小,相应的预处理时间越短,但网络覆盖性能越差.在要求目标区域中的每个点至少被一个传感器节点覆盖的应用中,为了保证覆盖性能,网格尺寸必须足够小;另一方面,传感器网络需要监视的目标区域一般较大.因此,采样点的数目 k ,通常比网络中传感器节点数目大几个数量级,即 $k \gg n$.贪婪算法的时间复杂性取决于其划分子区域的预处理过程.因此,其时间复杂性高于本文设计的算法.仿真程序实现了文献[5]提出的子区域划分算法,以比较 CVT 算法和贪婪算法在计算时间上的差异.程序的运行环境是一台运行 Red Hat 8.0 Linux 的工作站,该工作站拥有一个 Pentium IV 2.4G HZ 的 CPU,512MB 内存以及 512MB 交

换空间.

表 1 列出了当网格尺寸为 0.1,0.5 和 1 以及 $R_s=30$ 时,不同节点密度下贪婪算法预处理过程的计算时间.表 2 列出了当 $R_c=40,60$ 时不同节点密度下 CVT 算法以及 CVT+MST 算法的总体执行时间.从中可以看出,CVT(+MST)算法在运行时间上明显优于贪婪算法.

Table 1 Preprocessing time of Greedy algorithm
表 1 贪婪算法预处理时间

Grid size	R_c	Node number	Preprocessing time
0.1	30	200	>1hour
0.1	30	300	>1hour
0.1	30	400	>1hour
0.5	30	200	33.9s
0.5	30	300	172.8s
0.5	30	400	988.5s
1	30	200	8.9s
1	30	300	51.5s
1	30	400	241.3s

Table 2 Runtime of CVT(+MST) algorithm
表 2 CVT(+MST)算法运行时间

R_c	R_s	Node number	Runtime
40	30	200	1.87s
40	30	300	4.68s
40	30	400	8.07s
60	30	200	1.76s
60	30	300	4.52s
60	30	400	7.86s

在不同节点密度下,贪婪算法和 CVT(+MST)算法构造的连通覆盖集的大小如图 6 所示.从图 6 中可以看出,无论 R_c/R_s 是否大于 2,本文设计的算法都能构造比贪婪算法包含更少工作节点的连通覆盖集.事实上,文献[12]使用的贪婪法是一种全局贪婪;而本文的算法只是在解析冗余节点之间的依赖关系时使用了贪婪算法.

综上所述,无论在运行时间方面,还是在最终构造的连通覆盖集的大小方面,本文设计的 CVT(+MST)算法都优于文献[12]提出的贪婪算法.

3.2 数值结果

图 7 是当 $R_c/R_s=2$ 时,CVT 算法所计算得到的 CCS 的大小(即工作节点数),以及关闭节点数随初始节点数的变化情况.当网络中仅包含 100 个固定节点时,所有节点都必须处于工作状态,以保证网络的连通性以及完全覆盖部署区域.注意到,当 $R_c/R_s \geq 2$ 时,CVT 算法构造的 CCS 的大小不受随机节点数的影响.事实上,当 $R_c/R_s \geq 2$ 时,CCS 的大小仅仅取决于节点的感知半径.因此,当 $R_s=30$ 且 $R_c \geq 2R_s$ 时,CVT 算法所构造的 CCS 的大小,在不同节点密度下是一个常量,而且非常接近于所期望的最小工作节点数(100).必须指出,这 100 个节点并不等同于网络中 100 个固定部署的节点.

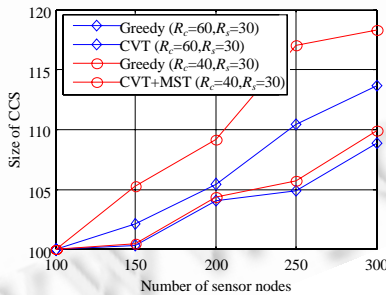


Fig.6 Performance comparison with Greedy

图 6 与 Greedy 算法的性能比较

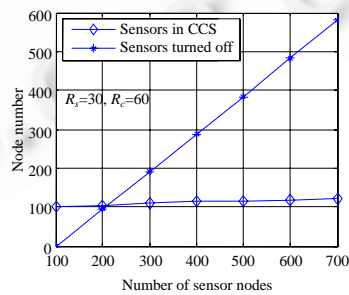


Fig.7 Solution of CVT vs. node number

图 7 CVT 算法解随节点数量的变化

图 8 是当 $R_s=30, R_c=40$ 时 CCS 的大小以及关闭节点数随初始节点数变化的曲线.实验结果以与 $R_c=60$ 条件下相应数值的比值的形势给出.虽然在图 8 中 CCS 中的节点数非常接近于 $R_c=60$ 时的数值,但当 $R_c < 2R_s$ 时,为确保覆盖集连通所需要的辅助节点数与比值 R_c/R_s 相关.因此,当 $R_c < 2R_s$ 时,算法计算得到的 CCS 的大小也与比值 R_c/R_s 相关.图 9 是辅助节点在最终求得的 CCS 中所占的比例,随节点数以及节点通信半径变化的情况.在这项实验中, $R_s=120, R_c$ 在 40~280 之间变化,从而 R_c/R_s 在 1/3 到 7/3 之间变化.如图 9 所示,在不同的节点密度下,当 $R_c \geq 160 (R_c/R_s \geq 4/3)$ 时,辅助节点在连通覆盖集中所占的比例很低.这个结果与图 8 的数据相吻合.但当 R_c/R_s 很小时,辅助节点在 CCS 中所占的比例是不容忽视的,有些情况下甚至超过 60%.从这个意义上讲,基于最小生成树

的连通算法的性能有待进一步提高.图 10 是 $R_s=80$ 、节点总数为 600 时,算法计算出的非工作节点数,以及 CCS 中的工作节点数随 R_c 变化的情况.当 $R_c/R_s=4/3$ 时,CCS 的大小基本上保持不变.

部署区域平均覆盖度是证明 CVT 算法能量高效的另一个指标.平均覆盖度是网络覆盖冗余程度的度量.平均覆盖度越高,就会有越多的节点同时向基站报告某点处发生的事件,造成不必要的网络负载和无线通信冲突,浪费节点能量.为了计算平均覆盖度,将部署区域分割为 1×1 的单元网格并假定在每个网格单元的中心都有一个事件发生.如果该中心同时被 k 个活跃节点覆盖,则称该点的覆盖度为 k .假定网络中共有 m 个事件点,事件点 i 的覆盖度为 k_i ,则平均覆盖度定义为 $\sum_{i=1}^m k_i / m$.如图 11 所示,尽管网络的原始平均覆盖度在 3~50 之间发生变化,使用 CVT 算法关闭部分节点以后,无论感知半径和节点密度如何变化,网络的平均覆盖度都维持在 2 左右.

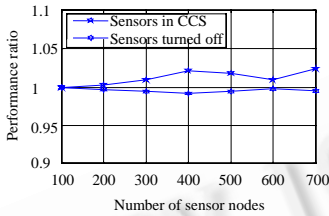


Fig.8 Performance ratio: $R_c=40/R_c=60$

图 8 性能比: $R_c=40/R_c=60$

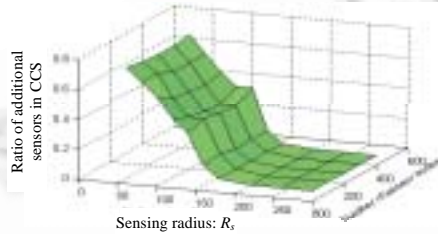


Fig.9 Ratio of additional sensors in CCS

图 9 辅助节点在 CCS 中的比例

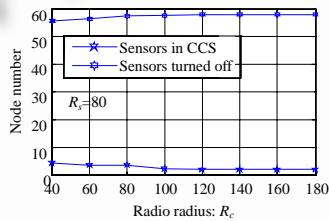


Fig.10 Performance vs. R_c

图 10 算法性能随 R_c 的变化

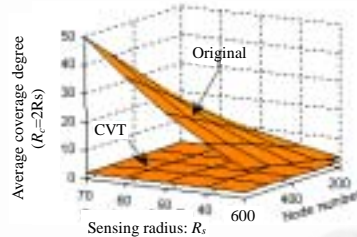


Fig.11 Average degree of coverage

图 11 平均覆盖度

4 结 论

本文讨论了无线传感器网络中的最小连通覆盖集问题.针对该问题,我们设计了一种有效的集中式算法.算法的基本思想是:首先构造覆盖集,然后确保该覆盖集连通.基于目标区域 Voronoi 划分的 CVT 算法可以构造目标区域的近似最小覆盖集.当 $R_c \geq 2R_s$ 时,该覆盖集是连通的;当 $R_c < 2R_s$ 时,不能保证由 CVT 算法构造的覆盖集的连通性.为此,我们设计了一种基于最小生成树的连通算法,用于计算确保该覆盖集连通所需的辅助节点.理论分析和仿真实验表明,我们设计的算法在时间复杂性和最终求得的连通覆盖集的大小方面,性能都优于文献 [12] 提出的贪婪算法.

References:

- [1] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: A survey. *Computer Networks*, 2002,38(4): 393-422.
- [2] Elson J, Estrin D. *Sensor Networks: A Bridge to the Physical World*. Norwell: Kluwer Academic Publishers, 2004. 3-20.
- [3] 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>
- [4] Cui L, Ju HL, Miao Y, Li TP, Liu W, Zhao Z. Overview of wireless sensor networks. *Journal of Computer Research and Development*, 2005,42(1):163-174 (in Chinese with English abstract).

- [5] Slijepcevic S, Potkonjak M. Power efficient organization of wireless sensor networks. In: Proc. of the Int'l Conf. on Communications. Helsinki: IEEE Communication Society, 2001. 472-476.
- [6] Abrams Z, Goel A, Plotkin S. Set k -cover algorithms for energy efficient monitoring in wireless sensor networks. In: Ramchandran K, Sztipanovits J, eds. Proc. of the 3rd Int'l Conf. on Information Processing in Sensor Networks. Berkeley: ACM Press, 2004. 424-432.
- [7] Ye F, Zhong G, Lu S, Zhang L. Peas: A robust energy conserving protocol for long-lived sensor networks. In: McKinley PK, Shatz S, eds. Proc. of the 23rd Int'l Conf. on Distributed Computing Systems. Providence: IEEE Press, 2003. 28-37.
- [8] Tian D, Georganas N. A coverage-preserving node scheduling scheme for large wireless sensor networks. In: Raghavendra CS, Sivalingam KM, eds. Proc. of the 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications. Atlanta: ACM Press, 2002. 32-41.
- [9] Chen H, Wu H, Tzeng N. Grid-Based approach for working node selection in wireless sensor networks. In: Viginier P, ed. Proc. of the Int'l Conf. on Communications. Paris: IEEE Press, 2004. 3673-3678.
- [10] Carbutar B, Grama A, Vitek J, Carbutar O. Coverage preserving redundancy elimination in sensor networks. In: Znati T, Raghavendra CS, eds. Proc. of the 1st IEEE Conf. on Sensor and Ad Hoc Communications and Networks. Santa Clara: IEEE Press, 2004. 377-386.
- [11] Yan T, He T, Stankovic J. Differentiated surveillance service for sensor networks. In: Akyildiz IF, Estion D, eds. Proc. of the 1st Int'l Conf. on Embedded Networked Sensor Systems. Los Angeles: ACM Press, 2003. 51-63.
- [12] Gupta H, Das SR, GU Q. Connected sensor cover: Self-Organization of sensor networks for efficient query execution. In: Gerla M, ed. Proc. of the ACM MobiHoc 2003. Annapolis: ACM Press, 2003. 189-200.
- [13] Bulusu N, Heidemann J, Estrin D. GPS-Less low cost outdoor localization for very small devices. IEEE Personal Communications Magazine, 2000,7(5):28-34.
- [14] He H, Huang C, Blum BM, Stankovic JA, Abdelzaher TF. Range-Free localization schemes in large scale sensor networks. In: Johnson DB, ed. Proc. of the ACM MobiCom 2003. San Diego: ACM Press, 2003. 81-95.
- [15] Romer K, Zurich E. The lighthouse location system for smart dust. In: Siewiorek D, ed. Proc. of the 1st Int'l Conf. on Mobile Systems, Applications, and Services. San Francisco: ACM Press, 2004. 15-30.
- [16] Okabe A, Boots B, Sugihara K, Chiu S. Spatial Tessellations: Concepts and Applications of Voronoi Diagram. 2nd ed., New York: John Wiley & Sons, 1999.
- [17] Hochbaum DS. Approximation Algorithms for NP-Hard Problems. Cambridge: PWS Publishing Company, 1995.
- [18] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms. 2nd ed., Cambridge: MIT Press, 2001.

附中文参考文献:

- [3] 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>
- [4] 崔莉, 鞠海玲, 苗勇, 李天璞, 刘巍, 赵泽. 无线传感器网络研究进展. 计算机研究与发展, 2005, 42(1): 163-174.



蒋杰(1976 -),男,重庆铜梁人,博士,主要研究领域为无线传感器网络,Ad-hoc网络.



张鹤颖(1976 -),女,博士,助理研究员,主要研究领域为网络拥塞控制.



方力(1976 -),女,工程师,主要研究领域为计算机网络.



奚文华(1946 -),男,教授,博士生导师,主要研究领域为高级计算机网络.