

无线传感器网络最小覆盖集的贪婪近似算法*

陆克中^{1,2+}, 孙宏元³

¹(深圳大学 计算机与软件学院, 广东 深圳 518060)

²(深圳大学 国家高性能计算中心深圳分中心, 广东 深圳 518060)

³(深圳大学 ATR 国防科技重点实验室, 广东 深圳 518060)

Greedy Approximation Algorithm of Minimum Cover Set in Wireless Sensor Networks

LU Ke-Zhong^{1,2+}, SUN Hong-Yuan³

¹(College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China)

²(National High Performance Computing Center at Shenzhen, Shenzhen University, Shenzhen 518060, China)

³(ATR National Defense Technology Key Laboratory, Shenzhen University, Shenzhen 518060, China)

+ Corresponding author: E-mail: kzlu@szu.edu.cn, http://www.szu.edu.cn

Lu KZ, Sun HY. Greedy approximation algorithm of minimum cover set in wireless sensor networks. *Journal of Software*, 2010,21(10):2656–2665. <http://www.jos.org.cn/1000-9825/3670.htm>

Abstract: Network lifetime is a bottleneck that restricts the development of wireless sensor networks. One approach to save energy effectively and prolong network lifetime is to schedule some nodes to work and put other nodes into a low-powered sleep mode, while monitoring performance of network. The object of scheduling nodes is to obtain a minimum node set that can cover a monitored region. This is a NP-hard problem. Performances of present approximation algorithms have not been good. An approximation algorithm of a minimum cover set problem based on methodology is proposed. During the process of constructing a cover set, effective node that extends to maximal areas are selected to join the cover set. Theoretical analyses show that the algorithm can construct a cover set that perform well and has a time complexity that is $O(n)$, where n is initial node number. Experimental results show that performance of this new algorithm outperform that of present algorithms. The size of a cover set is decreased by 14.2%. Also, execution time is less than that of present algorithms. When initial nodes are deployed densely, the average degree of coverage obtained by the algorithm is below 1.75 and has an approximation ratio below 1.45.

Key words: wireless sensor network; network lifetime; nodes scheduling; minimum cover set; greedy algorithm; approximation algorithm

摘要: 网络生命期是限制无线传感器网络发展的一个瓶颈.在保证网络监控性能的前提下,仅调度部分节点工作而让其余节点处于低功耗的休眠状态,可以有效节省能耗,延长网络生命期.节点调度的目标是寻找一个能够覆盖

* Supported by the National Natural Science Foundation of China under Grant No.60602066 (国家自然科学基金); the Guangdong Provincial Natural Science Foundation of China under Grant No.2008254 (广东省自然科学基金); the Science and Technology Planning Project of Guangdong Province of China under Grant No.2006B36430001 (广东省科技计划项目)

Received 2009-01-15; Revised 2009-04-27; Accepted 2009-07-07

监控区域的最小节点集合,这是一个 NP 难问题,目前,其近似算法的性能较低.提出了一种基于贪婪法的最小覆盖集近似算法,在构造覆盖集的过程中,优先选择扩展面积最大的有效节点加入覆盖集.理论分析表明,该算法能够构造出较好的覆盖集,时间复杂度为 $O(n)$,其中, n 为初始节点总数.实验数据表明,该算法的性能要优于现有算法,得到的覆盖集的平均大小比现有算法减小了 14.2% 左右,且执行时间要短于现有算法.当初始节点分布较密时,该算法得到的平均覆盖度小于 1.75,近似比小于 1.45.

关键词: 无线传感器网络;网络生命期;节点调度;最小覆盖集;贪婪算法;近似算法

中图法分类号: TP393 文献标识码: A

无线传感器网络由大量具有感知能力、计算能力和通信能力的微型传感器节点以 Ad Hoc 方式组成,这种新型监控网络具有覆盖区域广泛、检测精度高、可快速部署、可自组织和高容错性等优点,可广泛地应用于国防军事、环境监测、交通管理、医疗卫生、制造业、反恐抗灾等领域^[1-4].

无线传感器网络的多数应用场合需要较长的网络生命期,网络生命期已成为无线传感器网络发展的一个瓶颈^[5,6].由于受到体积的限制,传感器节点由小容量电池供电,通常布置在较远或较危险的区域,如战场、动物栖息地^[7]、沙漠等,且传感器节点数目众多,更换电池非常不易,因此传感器节点的能量制约了无线传感器的网络生命期.

节点调度是延长网络生命期的一种有效方法.由于无线传感器网络中传感器节点数目众多,不可能在监控区域内以理想的蜂窝方式人工地逐个布置传感器节点,而是通过飞机或其他工具密集地撒落在监控区域内,这样,网络覆盖会出现一定程度的冗余.通过调度传感器节点,仅将部分节点投入工作,使得工作节点能够监控整个区域,而让其他节点处于低功耗的休眠状态,节点之间轮流工作,可以有效地节省整体能量消耗.每个节点都有其监控范围,称为覆盖区域.节点调度的目标就是得到监控区域上的最小覆盖集,让最少的节点工作,从而最大限度地节省能量消耗.

文献[8]提出了基于探测的分布式节点调度算法,若节点的探测范围内存在活动节点,则休眠.文献[9]提出了基于 Voronoi 区域面积的节点调度算法,若与节点相关的 Voronoi 区域面积小于某个阈值,则此节点可以休眠.文献[8,9]中的节点调度算法都不能保证工作节点完全覆盖监控区域.文献[10]根据节点的覆盖区域是否包含在邻居节点的覆盖区域内,提出了一种能够保持原始覆盖区域的节点调度算法,但此算法的性能较差.文献[11]通过节点覆盖区域之间的关系将原始区域划分成若干个互不相交的独立区域,每个独立区域有一个能够覆盖它的节点集合,从而转化为最小集合覆盖问题^[12].然而,最小集合覆盖问题的近似算法并没有考虑到无线传感器网络最小覆盖集问题的特性,其性能不太理想.文献[13]提出的一种基于目标区域 Voronoi 划分的集中式近似算法(CVT(centralized Voronoi tessellation)算法),是目前性能较好的最小覆盖集近似算法,它先通过 Voronoi 划分得到冗余节点,并构成一个冗余节点依赖图,再通过最大独立集近似算法^[14]求出可以同时关闭的冗余节点,算法得到的覆盖集的平均覆盖度为 2 左右,与理想值 1.21 仍相差较大.

本文设计了一种基于贪婪法的最小覆盖集近似算法.在构造覆盖集的过程中,没有将最小覆盖集问题转化为其他 NP 难问题,而是从最小覆盖集问题的自身特性出发,充分考虑节点覆盖区域的特点,基于贪婪法,优先选择扩展面积最大的有效节点加入覆盖集.本文理论分析了算法的性能,并通过实验对算法的性能进行了评估,实验数据表明,算法在覆盖集的大小和执行时间上要优于现有的 CVT 算法.

1 问题描述

无线传感器网络布置在二维平面上的目标区域 R 内,一般情况下, R 是一个矩形区域.传感器节点 s 的覆盖(感应)区域是以 s 为圆心、以感应距离 r 为半径的圆形区域(称为感应圆盘),我们记作 $SR(s)$. $SR(s) = \{p: \|p, s\| \leq r\}$, 其中, $\|p, s\|$ 表示点 p 和节点 s 之间的欧氏距离.若 S 是节点集合,则 S 的覆盖区域为 S 中所有节点覆盖范围的并,记作 $SR(S)$, $SR(S) = \bigcup_{s \in S} SR(s)$.

定义 1. 给定区域 R 和节点集合 S ,若 S 的覆盖区域能够包含区域 R ,即 $R \subseteq SR(S)$,则称节点集合 S 为区域 R

的覆盖集(cover set).

定义 2. 给定区域 R 和节点集合 S , 若 S 的子集 S' 是区域 R 的覆盖集, 且 S' 的任意真子集都不是区域 R 的覆盖集, 则称节点集合 S' 为极小覆盖集(minimal cover set). 其中, 含节点数最少的极小覆盖集称为最小覆盖集(minimum cover set).

给定区域 R 和节点集合 S , 最小覆盖集问题即是求出最小覆盖集. 可以看出, 若节点集合 S 是区域 R 的覆盖集, 则一定存在最小覆盖集. 我们可以假设初始撒在监控区域 R 内的传感器节点能够完全覆盖监控区域 R , 传感器节点调度的目标就是求出最小覆盖集, 让最少的节点工作. 文献[11]指出, 这是一个 NP 难问题, 目前不存在多项式时间内的算法. 下一节根据贪婪法, 给出了一种多项式时间内的近似算法.

2 贪婪算法

无线传感器网络的最小覆盖集问题是一个平面区域覆盖问题, 若两节点间的距离大于 $2r$, 则这两个节点的覆盖区域必不相交, 因此, 最小覆盖集问题具有区域独立的特征. 假设区域 R_1 和 R_2 是两个不相交的区域, 且它们之间的最短距离大于 $2r$, 若节点集 S_1 是区域 R_1 的最小覆盖集, 节点集 S_2 是区域 R_2 的最小覆盖集, 则节点集 $S_1 \cup S_2$ 是区域 $R_1 \cup R_2$ 的最小覆盖集. 可以看出, 两个不相交区域内节点集的覆盖区域之间相互影响很小. 在构造覆盖集时, 可以利用区域独立的特征, 先求局部区域内的覆盖集, 然后将此区域扩展至整个目标区域.

节点集的覆盖区域是此节点集中所有节点覆盖区域的并, 向节点集中添加其他节点后, 节点集的覆盖区域不会变小. 可以通过集合递增的方式来构造一个覆盖集: 不断地向节点集中添加节点, 直到此节点集覆盖整个目标区域. 假设目标区域为 R , 节点集合为 S , 构造覆盖集 C 的形式描述如下:

- (1) 初始时 $C = \emptyset$;
- (2) 从节点集 $S - C$ 中选择某节点 s , $C = C \cup \{s\}$;
- (3) 若节点集 C 是区域 R 的覆盖集, 则返回 C , 否则, 重复步骤(2).

集合递增方法构造覆盖集的关键在于步骤(2)中选择哪个节点加入集合中, 这直接影响到最终得到的覆盖集的大小. 从贪婪法的角度出发, 应选择扩展面积最大的节点, 然而这并不完全正确, 与原覆盖区域不相交的节点的扩展面积都最大, 原因在于没有考虑到新扩展区域和原覆盖区域的关系. 如图 1 所示, s_1 和 s_2 为已在覆盖集中的节点, 节点 s_3 的扩展面积虽然最大, 但与原覆盖区域的连接性并不好, 产生了一个狭窄的未覆盖区域, 随后仍需加入节点 s_4 来覆盖这个狭窄区域, 但节点 s_4 和节点 s_2, s_3 的重合性较大, 因此节点 s_3 的扩展区域效率并不高.

高效率扩展的关键在于要考虑到新扩展区域和原覆盖区域的关系. 如图 2 所示, 节点集在区域 R 中的覆盖区域的边界必由圆弧和线段组成, 圆弧之间或圆弧与线段的交点称为弧交点. 可以看出, 弧交点必定是凹点, 也就是局部区域最靠内侧、最难覆盖到的点. 在区域扩展的过程中, 弧交点必须被优先覆盖, 能够覆盖某弧交点的节点称为有效扩展节点. 在区域扩展过程的选择节点步骤, 先确定有效扩展节点, 再根据贪婪法则从中选择扩展面积最多的节点加入到覆盖集中. 因为覆盖目标区域外的区域是无意义的, 所以在计算节点和节点集的覆盖区域时, 仅指其在目标区域内的部分.

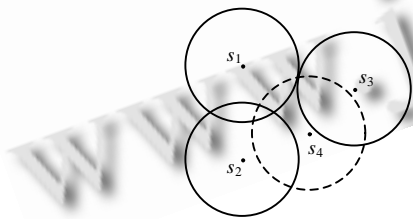


Fig.1 Extending node of low efficiency

图 1 效率低的扩展节点

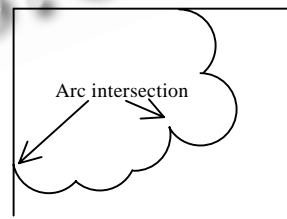


Fig.2 Border of covered region

图 2 覆盖区域的边界

当覆盖集为空或仅有一个节点时, 覆盖区域的边界可能没有弧交点, 上述节点选择方法并不适用. 由下节的

分析得知,当覆盖集中相邻节点间距离都等于 $\sqrt{3}r$ 时,此覆盖集是最优的.因此初始时,可在目标区域中心位置任意选择两个距离接近 $\sqrt{3}r$ 的节点加入覆盖集中.在算法中,我们按照逆时针顺序记录每个弧交点.对于每个弧交点,我们记录覆盖此弧交点且扩展面积最大的节点及扩展面积,扩展面积即是弧交点的权,每次扩展过程总是选择与优先级最大的弧交点关联的节点.新加一个节点后,因为此节点的覆盖区域仅与其周围节点的覆盖区域有重合,所以仅需更新与此节点关联的弧交点的前驱和后继弧交点的权.完整的覆盖集生成算法如下:

算法. GA 算法.

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

输出:区域 R 的覆盖集 C .

标记: A :弧交点集; N :新弧交点集.

第 1 步. $C=A=\emptyset$.

第 2 步.

第 2.1 步. 从 S 中选择距离区域 R 中心最近的节点 s .

第 2.2 步. 再从 S 中选择与 s 距离最接近 $\sqrt{3}r$ 的节点 s' .

第 2.3 步. $C=C\cup\{s, s'\}$, N 为 s 覆盖区域和 s' 覆盖区域的两交点, $A=A\cup N$.

第 3 步. While ($A\neq\emptyset$)

第 3.1 步. for each arc intersection $p_i\in N$

分别计算 $S-C$ 中覆盖 p_i ; p_i 前驱弧交点和 p_i 后继弧交点且扩展面积最大的节点;

更新 p_i ; p_i 前驱弧交点和 p_i 后继弧交点的权值.

第 3.2 步. 寻找 A 中优先级最大的弧交点 p , 并得到与 p 关联的节点 v , $C=C\cup\{v\}$.

第 3.3 步. 计算原覆盖区域与节点 v 覆盖区域的并, 并更新 A 和 N .

第 4 步. Return C .

定理 1. 如果初始节点能够覆盖目标区域, 则 GA 算法能够得到目标区域上的覆盖集.

证明:先证在 GA 算法第 2 步执行完后, A 为空, 等价于 C 是覆盖集. 若 C 是覆盖集, 则 C 的覆盖区域即是目标区域 R , 此时 A 为空; 反之, 若 A 为空, 则 C 的覆盖区域的边界中没有圆弧, 即全由圆或全由线段组成. 在 GA 算法第 2 步执行完毕后, 已经排除第 1 种情况, 而第 2 种情况即可说明 C 是覆盖集.

再证在 GA 算法第 3 步的每次循环过程中, 节点集 $S-C$ 中总是存在能够覆盖 A 中某弧交点的节点. 采用反证法, 假设 A 中所有弧交点都不能被 $S-C$ 覆盖. 设目标区域 R 中未被 C 覆盖的区域为 U , 易知区域 U 的边界的弧交点集也为 A . 因为初始节点集 S 能够覆盖区域 R , 易知 $S-C$ 也能覆盖区域 U . 因为每个节点的覆盖区域是一个闭区域, 所以 $S-C$ 的覆盖区域也是一个闭区域. 而一个区域被一个闭区域完全覆盖的必要条件是其边界能够被这个闭区域覆盖, 所以 U 的边界能被 $S-C$ 覆盖. 这与假设条件矛盾, 原命题得证.

最后证 GA 算法第 3 步的循环一定会终止. 前面已经证明, 在每次循环过程中, 总是能从 $S-C$ 中选择某节点加入到 C 中, 所以循环最多只能执行 $|S|-2$ 次, 即循环一定会终止. 而循环的终止条件为 $A=\emptyset$, 所以算法返回的 C 是目标区域上的覆盖集. \square

3 性能评估

3.1 时间复杂度分析

设初始节点总数为 n . 在区域 R 中, 任意边长为 $2r$ 的正方形内最多有 k 个节点, 算法得到的覆盖集大小为 d .

GA 算法第 2 步的时间为 $O(n)$. 下面来分析第 3 步循环的时间, 先对循环体内每一步的时间进行分析, 然后再分析循环的次数.

第 3.1 步: 新弧交点的个数至多为一个常数, 时间主要为计算覆盖弧交点且扩展面积最大的节点. 可以在初始时将区域 R 划分为边长 $2r$ 的网格, 计算每个节点所在的网格, 并建立索引. 这样, 计算覆盖弧交点且扩展面积最大的节点时, 仅需遍历此弧交点所在网格周围的 9 个网格内的节点. 故这一步的时间为 $O(k)$.

第 3.2 步:因为 A 的大小最多为 C 的大小的 2 倍,而寻找 A 中优先级最大的弧交点 p 仅需遍历 A ,所以这一步的时间为 $O(d)$.

第 3.3 步:计算原覆盖区域与节点 v 覆盖区域的并的时间为 $O(d)$.

因为每次循环加入一个节点,故最多循环 d 次,所以第 3 步循环的时间为 $O(d^2+dk)$.从以上分析可知,GA 算法总的复杂度为 $O(n+d^2+dk)$.后面的仿真实验结果表明,当初始节点总数 n 增大时,GA 算法得到的覆盖集的大小会小于某一常数.因此,一般情况下,GA 算法的时间复杂度为 $O(n)$.

3.2 覆盖集的性能分析

一般情况下,由于目标区域内传感器节点分布比较随机,因此很难分析 GA 算法的性能.因为目标区域内传感器节点分布较密,可先假设在目标区域内任意位置都存在传感器节点,分析此理想情况下 GA 算法的性能,再将结果推广至一般情况.

先分析理想情况下,对于某弧交点,当有效扩展节点处于何位置时,扩展面积最大.

定理 2. 对于某弧交点,理想情况下,扩展面积最大的有效扩展节点在此弧交点所在两圆的公共弦的延长线上,且与此弧交点的距离为 r .

证明:易知,节点距离弧交点越远,节点的覆盖区域和原覆盖区域的重叠部分就越小,扩展面积就越大.因此,理想情况下,扩展面积最大的有效扩展节点必在距离弧交点 r 处,此时,节点的覆盖区域恰好能够覆盖弧交点.下面再证在所有距离弧交点为 r 的节点中,扩展面积最大的节点位于此弧交点所在两圆的公共弦的延长线上.

如图 3 所示,弧交点为 p ,弧交点所在两圆的圆心分别为 s_1 和 s_2 ,节点 s 与点 p 的距离为 r , s 和 p 的连线与圆 s_1 和圆 s_2 的公共弦的夹角为 β ,角 α 称为弧交点 p 的圆心角,新弧交点 p_1 的圆心角为 γ ,新弧交点 p_2 的圆心角为 δ ,则有 $\gamma = \frac{\pi - \alpha}{2} + \beta$, $\delta = \frac{\pi - \alpha}{2} - \beta$,扩展面积 y 为

$$y = \pi r^2 - (\gamma - \sin \gamma)r^2 - (\delta - \sin \delta)r^2 = \alpha r^2 + 2 \cos \frac{\alpha}{2} \cos \beta r^2 \quad (1)$$

对 β 求导,得到 $y' = -2 \cos \frac{\alpha}{2} \sin \beta r^2$.

可以看出, β 取 0 时,即扩展节点在弧交点所在两圆的公共弦的延长线上时,扩展面积 y 取最大值.

定理 2 得证. \square

再分析理想情况下,当弧交点的圆心角为多少时,从该弧交点最优扩展后的覆盖面积最大.

定理 3. 理想情况下,当弧交点的圆心角为 $\pi/3$ 时,从该弧交点最优扩展后的覆盖面积最大.

证明:从定理 2 得知,最优扩展节点在弧交点所在两圆的公共弦的延长线上,且与此弧交点的距离为 r .如图 3 所示,此时 α 为变量,而 β 为 0,则 3 个节点 s_1, s_2 和 s 的覆盖面积 y 为

$$y = \alpha r^2 + 2r^2 \cos \frac{\alpha}{2} + 2\pi r^2 - (\alpha - \sin \alpha)r^2 = 2\pi r^2 + 2 \cos \frac{\alpha}{2} r^2 + \sin \alpha r^2 \quad (2)$$

对 α 求导,得到 $y' = \left(\cos \alpha - \sin \frac{\alpha}{2} \right) r^2$.

可以看出, α 取 $\pi/3$ 时, y 取最大值.定理 3 得证. \square

理想情况下,当弧交点的圆心角为 $\pi/3$ 时,从该弧交点最优扩展后,两个新弧交点的圆心角仍为 $\pi/3$.因此,理想情况下,GA 算法可得到如图 4 所示的最佳覆盖集,节点如蜂窝方式来布置,此最佳覆盖集具有以下两个特征:

- ① 每个弧交点的圆心角都为 $\pi/3$,即每个节点与相邻节点之间的距离都为 $\sqrt{3}r$;
- ② 每个节点都有 6 个相邻节点.

一般情况下,虽然传感器节点分布较密,但与理想情况仍有差距,所以 GA 算法无法得到理想情况下的最佳覆盖集.以下说明 GA 算法在扩展节点时,会向着理想情况下的扩展方式靠近,因此得到的覆盖集性能仍然较好.

如图 5 所示,弧的交点为 p ,弧交点所在两圆的圆心分别为 s_1 和 s_2 ,弧交点 p 的圆心角为 α .因为不是理想情

况,所以 α 可能不等于 $\pi/3$;从弧交点 p 最优扩展后,新增两个弧交点 p_1 和 p_2 , p_1 和 p_2 的圆心角都为 β ,则 $\beta = \frac{\pi - \alpha}{2}$, 得到

$$\left| \beta - \frac{\pi}{3} \right| = \frac{1}{2} \left| \alpha - \frac{\pi}{3} \right| \tag{3}$$

$$\frac{\alpha + \beta}{\frac{2}{3}\pi} = \frac{\frac{\alpha + \pi}{4}}{\frac{1}{3}\pi} = \frac{1}{4} \frac{\alpha}{\frac{1}{3}\pi} + \frac{3}{4} \tag{4}$$

由公式(3)和公式(4)得知:

- (1) β 比 α 更接近 $\pi/3$;
- (2) 与 α 与 $\pi/3$ 相比, $\alpha + \beta$ 与 $2\pi/3$ 更接近.

上述情况(1)和情况(2)分别说明,GA 算法在扩展节点时,向着理想情况下最佳覆盖集的特征①和特征②靠近.因此一般情况下,GA 算法在扩展节点的过程中,即使某次选择了效率低的扩展节点,随后的节点选择仍然会向着理想情况下的扩展方式靠近,进行某种程度的自动修正,从而得到较优的结果.

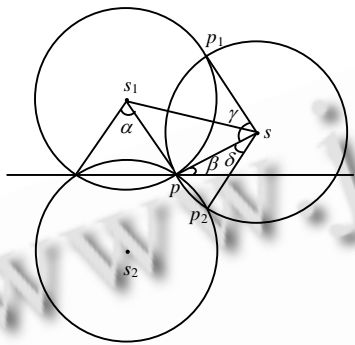


Fig.3 Example of effective extending
图 3 有效扩展示意

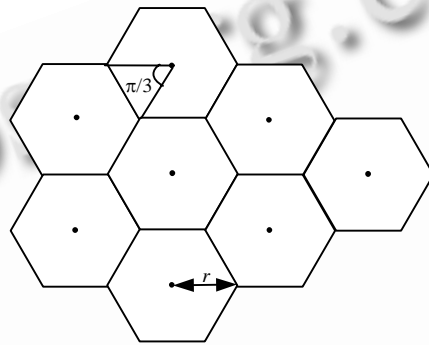


Fig.4 Optimal cover set under ideal case
图 4 理想情况下的最佳覆盖集

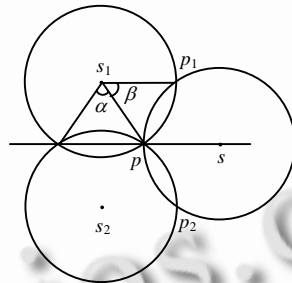


Fig.5 Extending under general case
图 5 一般情况下的扩展

4 仿真实验

为了评估我们算法的性能,用 C++语言设计了仿真程序,实现了本文提出的 GA 算法和文献[13]中提出的 CVT 算法.实验平台的性能参数为: Intel Core 2 Duo 2.66GHz CPU,2GB 内存.实验中,我们将 GA 算法和 CVT 算法进行对比分析.为了使得实验结果比较准确,每一个实验结果都为 100 次仿真实验的平均值.

最小覆盖集近似算法的输入参数有:目标区域的大小、初始节点总数、初始节点的分布以及节点的感应距离;输出为覆盖集.评估最小覆盖集近似算法的主要指标有:覆盖集的大小、平均覆盖度、近似比以及执行时间.

因为目标区域的增大可以看作节点的感应距离的等比例的缩小,所以目标区域的大小和节点的感应距离对算法输出结果的影响是一致的.在实验中,我们固定目标区域为 1000×1000 的矩形区域,通过改变初始节点总数、初始节点的分布以及节点的感应距离,观察 GA 算法在不同实验场景下的性能.

先观察初始节点的分布对 GA 算法性能的影响.实验中,目标区域为 1000×1000 的矩形区域,初始节点总数为 900,节点的感应距离为 100,节点的分布取 3 种典型的分布情况:均匀分布、随机分布和非均匀分布.在均匀分布过程中,节点沿着横向和纵向等距离地分布;在随机分布过程中,节点等概率地分布在目标区域的任意一点;在非均匀分布过程中,节点沿着纵向等距离地分布,沿着横向递增距离分布,节点密度从左到右逐渐变小.图 6 为这 3 种节点分布情况下的实验结果,细点表示初始节点,粗点表示 GA 算法得到的覆盖集中的节点,此外,还显示了覆盖集中节点构成的 Voronoi 图.

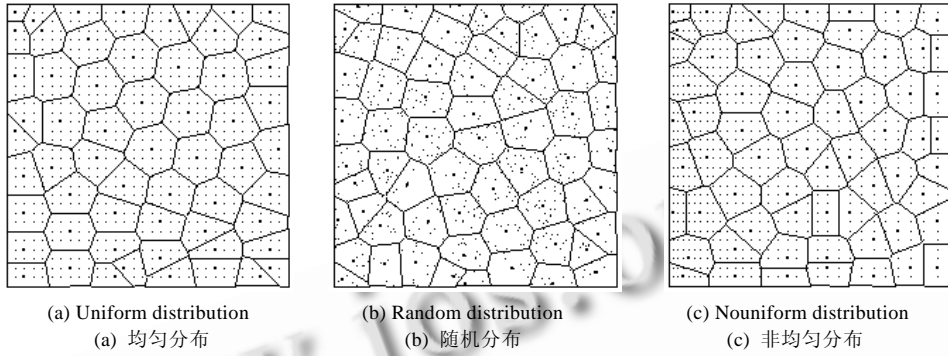


Fig.6 Cover sets obtained by GA algorithm under various node's distributions

图 6 不同节点分布情况下 GA 算法得到的覆盖集

可以看出,在不同节点分布情况下,GA 算法都得到了较小的覆盖集.其中,均匀分布情况下覆盖集的大小为 59,随机分布下覆盖集的大小为 63,非均匀分布情况下覆盖集的大小为 64.均匀分布情况下的覆盖集要比随机分布和非均匀分布略小,这是因为,在节点均匀分布情况下,目标区域内任意一点周围很短的距离内都存在节点,在 GA 算法的选择节点过程中,总是能够选择到比较好的节点.而在节点随机分布和非均匀分布的情况下,目标区域内会存在较多比均匀分布时要大的没有节点的空隙,这样就会选择到较差的节点.

再观察初始节点总数和节点的感应距离对 GA 算法性能的影响.实验中,目标区域为 1000×1000 的矩形区域,节点随机分布在目标区域内,先固定节点总数为 2 500,让节点的感应距离在 75~120 之间发生变化,每隔 5 取一个值,再固定节点的感应距离为 100,让初始节点总数在 500~5 000 之间发生变化,每隔 500 取一个值,图 7 和图 8 为实验结果.

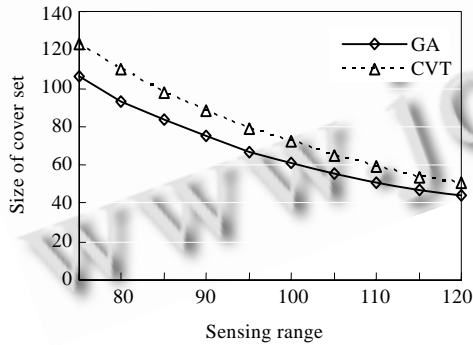


Fig.7 Sizes of cover sets under various sensing ranges

图 7 不同感应距离下覆盖集的大小

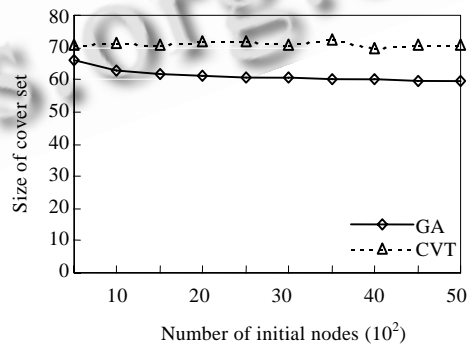


Fig.8 Sizes of cover sets under various numbers of initial nodes

图 8 不同初始节点总数下覆盖集的大小

从图 7 可以看出(初始节点总数为 2 500),GA 算法和 CVT 算法得到的覆盖集的大小都近似地与节点的感应距离的平方成反比.这是因为节点的覆盖区域是与节点的感应距离的平方成正比的,而覆盖集的大小取决于目标区域的面积与覆盖集中节点的平均独立覆盖区域面积之比.

从图 8 可以看出(节点的感应距离为 100),随着初始节点总数的增加,GA 算法得到的覆盖集会逐渐变小.当初始节点总数为 500 时,覆盖集的平均大小为 66.3;当初始节点总数增加到 1 000 时,覆盖集的大小减小得较多,减少了 5.3%;而当初始节点总数再增加时,覆盖集的大小则变化得不太明显,保持在 60 左右.这是因为,当节点密度增大时,GA 算法选择到较优位置的节点的概率会增大,当节点密度大到一定程度时,节点密度的增大对覆盖集大小的影响会变得非常小,此时覆盖集的大小就取决于 GA 算法本身的性能,而与初始节点总数的关系不大.从图 8 还可以看出,CVT 算法得到的覆盖集的大小保持在 71 左右,与初始节点总数基本无关.这是因为 CVT 算法是通过关闭冗余节点得到覆盖集,节点密度的增大并不能保证较优位置的节点不被关闭.图 7 和图 8 的实验数据表明,GA 算法得到的覆盖集的平均大小为 CVT 算法的 85.8%,比 CVT 算法减少了 14.2%.

目标区域内每点平均被多少个传感器节点覆盖称为平均覆盖度.平均覆盖度反映了网络中节点覆盖的冗余程度,是衡量最小覆盖集算法性能的重要指标,可以通过将所有节点在目标区域内的覆盖面积之和除以目标区域的面积求得.最佳覆盖集的平均覆盖度为 1.21,即最小覆盖集问题的最优解的平均覆盖度下界为 1.21.一般来说,最小覆盖集求解算法得到的覆盖集的平均覆盖度与目标区域及节点的感应距离基本无关.下面我们观察 GA 算法在不同实验场景下得到的覆盖集的平均覆盖度,其中,实验场景用初始节点覆盖密度来描述,即所有初始节点的覆盖面积之和除以目标区域的面积.

实验中,节点随机分布在 1000×1000 的矩形区域中,分别取节点的感应距离为 80,90,100,110 和 120,通过调节初始节点总数来控制初始节点覆盖密度.从图 9 的实验结果可以看出,当节点的感应距离取不同值时,平均覆盖度与初始节点覆盖密度的变化曲线基本一致.在初始节点覆盖密度较小时(小于 20),平均覆盖度随着初始节点覆盖密度的增大而减小得较快;而在初始节点覆盖密度较大时(大于 30),平均覆盖度则减小得较慢,会趋向于一个稳定值.当初始节点覆盖密度大于 20 时,GA 算法在不同实验场景下得到的覆盖集的平均覆盖度都小于 1.75.由于平均覆盖度近似正比于覆盖集的大小,因此近似比小于 1.45.

最后,我们通过实验来分析 GA 算法的执行时间.实验中,节点随机分布在 1000×1000 的矩形区域中,节点的感应距离为 100,通过调节初始节点总数来观察 GA 算法在不同实验场景下的执行时间,图 10 为实验结果(节点的感应距离为 100).可以看出,GA 算法的执行时间要小于 CVT 算法的执行时间.一般情况下,GA 算法都可以在很短的时间(小于 0.3s)内构造出覆盖集.CVt 算法的执行时间大致与初始节点总数的平方成正比,而 GA 算法的执行时间大致与初始节点总数成正比,这与前面关于 GA 算法时间复杂度的理论分析结果是一致的.

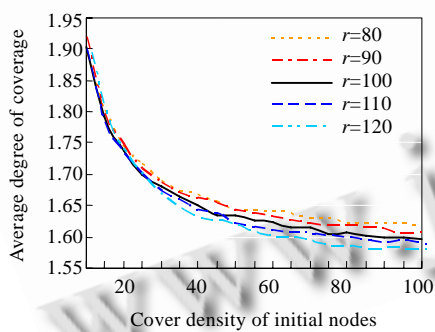


Fig.9 Average degree of coverage under various experiment scenes

图 9 不同实验场景下的平均覆盖度

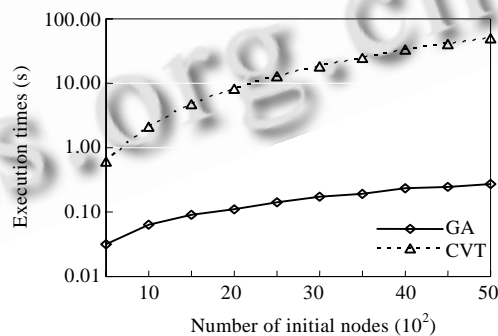


Fig.10 Execution times of algorithm under various numbers of initial nodes

图 10 不同初始节点总数下算法的执行时间

5 结 论

本文提出了一种基于贪婪法的最小覆盖集近似算法(简称 GA 算法),在构造覆盖集的过程中,优先选择扩展面积最大的有效节点加入覆盖集,GA 算法的时间复杂度为 $O(n)$,其中, n 为初始节点总数.理想情况下,GA 算法能得到最优覆盖集.理论分析表明,在节点分布较密的情况下,GA 算法在扩展节点的过程中仍然朝着理想情况下的扩展方式靠近,进行某种程度的自动修正,从而得到较优的结果.最后,通过仿真实验,将 GA 算法与目前性能较优的 CVT 算法进行了对比,GA 算法得到的覆盖集的平均大小为 CVT 算法的 85.8%左右,且执行时间要小于 CVT 算法.实验数据表明,一般情况下,GA 算法都可以在很短的时间(小于 0.3s)内构造出覆盖集.当初始节点覆盖密度大于 20 时,GA 算法在不同实验场景下得到的覆盖集的平均覆盖度都小于 1.75,近似比小于 1.45.此外,GA 算法的适应能力较强,在节点均匀布置、随机布置和非均匀布置的情况下,均能得到较小的覆盖集.

下一步的工作是理论分析本文提出的最小覆盖集近似算法的近似比,以及将目前的集中式算法改为分布式算法.

References:

- [1] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: A survey. *Computer Networks*, 2002,38(4): 393–422. [doi: 10.1016/S1389-1286(01)00302-4]
- [2] Seo HM, Moon Y, Park YK, Kim D, Kim DS, Lee YS, Won KH, Kim SD, Choi P. A low power fully CMOS integrated RF transceiver IC for wireless sensor networks. *IEEE Trans. on Very Large Scale Integration Systems*, 2007,15(2):227–231. [doi: 10.1109/TVLSI.2007.893586]
- [3] 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). [doi: 10.1360/crad20050121]
- [4] 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/ch/reader/view_abstract.aspx?flag=1&file_no=20030713&journal_id=jos
- [5] Dagher JC, Marcellin MW, Neifeld MA. A theory for maximizing the lifetime of sensor networks. *IEEE Trans. on Communications*, 2007,55(2):323–332. [doi: 10.1109/TCOMM.2006.887495]
- [6] Wang W, Srinivasan V, Chua KC. Extending the lifetime of wireless sensor networks through mobile relays. *ACM Trans. on Networking*, 2008,16(5):1108–1120. [doi: 10.1109/TNET.2007.906663]
- [7] Mainwaring A, Polastre J, Szewczyk R, Culler D, Anderson J. Wireless sensor networks for habitat monitoring. In: Raghavendrv CS, ed. *Proc. of the 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications*. New York: ACM Press, 2002. 88–97.
- [8] 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.
- [9] Vieira MAM, Vieira LFM, Ruiz LB, Loureiro AAF, Fernandes AO, Nogueira JMS. Scheduling nodes in wireless sensor networks: A Voronoi approach. In: Stiller B, George AD, eds. *Proc. of the 28th Annual IEEE Int'l Conf. on Local Computer Networks*. Washington: IEEE Computer Society, 2003. 423–429.
- [10] 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.
- [11] Slijepcevic S, Potkonjak M. Power efficient organization of wireless sensor networks. In: *Proc. of the IEEE Int'l Conf. on Communications*. Helsinki: IEEE Communication Society, 2001. 472–476. <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7452>
- [12] Chakravarty S, Shekhawat A. Parallel and serial heuristics for the minimum set cover problem. *The Journal of Supercomputing*, 1992,5(4):331–345. [doi: 10.1007/BF00127952]
- [13] 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> [doi: 10.1360/jos170175]
- [14] Hochbaum DS. *Approximation Algorithms for NP-Hard Problems*. Cambridge: PWS Publishing Company, 1995.

附中文参考文献:

- [3] 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. [doi: 10.1360/crad20050121]
- [4] 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/ch/reader/view_abstract.aspx?flag=1&file_no=20030713&journal_id=jos
- [13] 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> [doi: 10.1360/jos170175]



陆克中(1982-),男,安徽枞阳人,博士,副教授,CCF 会员,主要研究领域为高性能计算,无线传感器网络.



孙宏元(1962-),男,博士,教授,CCF 高级会员,主要研究领域为高性能计算,无线传感器网络.