

一种基于 A-means 聚类算法的 Sweep Coverage 机制^{*}

刘晨光, 林 锋, 周激流
(四川大学 计算机学院, 成都 610064)

摘 要: 为了改进 Sweep Coverage 机制在真实场景中的性能, 提出了一种基于聚类的 Sweep Coverage 机制 (clustering-based sweep coverage, CBSC), 该机制引入一种自适应的聚类算法 A-means (self-adaptive K-means algorithm) 来控制节点的移动。实验结果表明, 在真实的监控场景中, CBSC 机制较以往的覆盖机制取得了更好的性能表现。

关键词: 无线传感器网络; 覆盖机制; 扫描覆盖; 移动控制; 聚类; K-均值

中图分类号: TP301 **文献标志码:** A **文章编号:** 1001-3695(2012)03-1051-03

doi:10.3969/j.issn.1001-3695.2012.03.068

Sweep Coverage approach with A-means clustering algorithm

LIU Chen-guang, LIN Feng, ZHOU Ji-liu
(College of Computer Science, Sichuan University, Chengdu 610064, China)

Abstract: The key issue of sweep coverage scheme is to schedule the sensor's mobility effectively in real scene. In this work, this paper proposed a novel Sweep Coverage approach with A-means (self-adaptive K-means algorithm) clustering algorithm named CBSC. The simulation results show that the CBSC achieves better performance than existing approaches in real scene.

Key words: wireless sensor networks (WSN); coverage scheme; Sweep Coverage; mobility control; clustering; K-means

0 引言

无线传感器网络是由部署在监测区域内的大量传感器节点形成的一个自组织网络, 网络中的节点相互协作感知、采集和处理目标数据并传输给汇聚节点。为了保证对信息的有效感知和采集, 节点需要覆盖到信息源区域。在传统的全覆盖^[1]和栅栏覆盖^[2]的应用中, 监控区域需要被静止的节点实时覆盖。而在一些特殊的应用中, 节点只需在给定的时间间隔内覆盖监控区域内的 POI (points of interest) 即可保证对信息的感知和采集。由于 POI 并不需要被实时覆盖, 因此使用移动传感器节点间歇性地覆盖 POI 可以有效地减少对节点的使用。

针对这种情况, Cheng 等人^[3]提出了无线传感器网络的 Sweep Coverage 问题: 在满足 POI 覆盖间隔需求的前提下, 如何有效地部署和控制移动节点, 以减少覆盖所需的节点数量, 降低应用成本。同时, Cheng 等人论证了 Sweep Coverage 问题是 NP-Hard 问题。此外, 文献[3]还提出了两种 Sweep Coverage 机制: 集中式的 CSWEEP 机制和分布式的 DSWEEP 机制。CSWEEP 机制建立一条遍历所有 POI 的 TSP 回路 P , 同时将 P 划分为 $L_0 = \frac{vt}{2}$ (t 为 POI 的最短覆盖间隔需求) 长度的段, 每段由一个节点往返覆盖。这种机制实现较为简单, 但当 POI 覆盖时间需求差异较大时, 需要相对较多的节点数量。DSWEEP 机制由 Epidemic Exchange 和 Next-POI 两部分组成, 前者用于节点之间交换和更新 POI 覆盖信息; 后者用于决策节点下一个要覆盖的 POI。尽管该机制应用较为灵活, 但是节点总倾向于距

自身较近的 POI 移动, 部分 POI 可能因此长期失去覆盖, 从而降低了对 POI 的覆盖率。文献[4]提出了一种新的 Sweep Coverage 机制 VRP-SWEEP: 首先使用 VRPTW (vehicle routing problem with time windows) 的一个插入式启发算法, 生成节点的初始移动路径; 接着使用模拟退火算法对初始路径进行优化, 生成节点的最终移动路径。尽管该机制在节点使用数量和覆盖率上均优于 CSWEEP 和 DSWEEP 机制, 但其在 POI 非均匀分布的场景中, 容易将相距较远的 POI 包含在同一条路径中, 增加路径长度, 从而需要更多的节点才能满足 POI 的覆盖间隔需求。

鉴于目前提出的 Sweep Coverage 机制的缺点和不足, 本文引入聚类算法来控制节点的移动, 提出了一种基于聚类的 Sweep Coverage 机制 (CBSC)。通过引入一种自适应的聚类算法, CBSC 机制将相距较近的一类 POI 包含在同一条路径中, 并为每条路径分配一定数量的移动传感器节点, 以满足 POI 的覆盖间隔需求。

1 基于聚类的 Sweep Coverage 机制

因为节点移动路径的长度直接影响着节点的使用数量, 所以将距离较近的 POI 包含在同一条路径, 而将相距较远的 POI 分属不同的路径, 可以有效地减少路径长度, 从而降低节点的使用数量。为此, 在 CBSC 机制中, 引入聚类算法, 把距离相近的 POI 聚为一类, 并在此基础上形成节点的移动路径。

1.1 问题定义

给定无线传感器网络的工作区域, 边界大小为 $X \times Y$, 工作

收稿日期: 2011-06-16; 修回日期: 2011-07-25 基金项目: 国家自然科学基金资助项目 (60773168)

作者简介: 刘晨光 (1988-), 男, 安徽五河人, 硕士研究生, 主要研究方向为无线传感器网络 (scujklcg@gmail.com); 林锋 (1975-), 男, 讲师, 博士, 主要研究方向为无线传感器网络; 周激流 (1963-), 教授, 博导, 博士, 主要研究方向为计算机图形图像处理、计算智能。

区域内存在点集 $P = \{p_0, p_1, \dots, p_n\}$, 其中, p_0 表示汇聚节点, p_i ($i \neq 0$) 表示 POI. P 中各元素坐标分别为 $[x_i, y_i]$ ($i \in \{0, 1, \dots, n\}$). T_i ($i \neq 0$) 表示 p_i 的覆盖间隔需求. 当前有一组传感器节点, 其移动速度和数据缓冲区大小分别设为 v 和 cap . 于是, 在 CBSC 机制中 Sweep Coverage 问题可以定义为: 在给定上述条件的情况下, 且在满足 POI 覆盖间隔需求的前提下, 求解如何控制节点的移动, 使其保证消耗最少的节点可将信息投递到汇聚节点.

1.2 聚类算法

由于经典的 K-means^[5] 算法需要用户事先给出要生成的类的数目 K , 所以很难适用于 POI 分布未知的情况. 针对 K-means 算法的缺点, 文献[6]提出一种改进的 K-means 算法 X-means. 该算法只需用户提供 K 值的上限 $K_{\max}$, 即可自动地确定 K 值. 然而, 合适的上限值很难确定, 若 $K_{\max}$ 过大, 算法效率变低; 若 $K_{\max}$ 过小, 算法可能错过最优的 K 值. 所以本文对 X-means 进行改进, 提出一种自适应的 K-means 算法 A-means.

A-means 算法从 K 值的下限开始, 不断地增加中心点直到聚类效果达到最佳. 本文通过计算 BIC (Bayesian information criterion) 评分来判断聚类结果的好坏, BIC 分数越大, 聚类结果越好. 为了判断聚类结果是否达到最优, 引入聚类结果未改进计数 count. 在算法的运行过程中, 如果当前聚类结果的 BIC 分数大于最优解, 则更新最优解为当前聚类结果, 同时将 count 置为零; 否则 count 值加 1. 如果 count 达到其上限值, 说明聚类结果已经连续多次没有改进, 算法结束, 输出最优解. 算法的实现流程如下:

- 令 $K = K_{\min}$, 任意选择 K 个初始聚类中心点, 初始化聚类结果未改进计数 $count = 0$, 迭代次数 $step = 0$.
- 运行 K-means 直到收敛, 得到中心点集合 C_k . 如果 $BIC(C_k) > BIC_{\max}$, $BIC_{\max} = BIC(C_k)$, $count = 0$; 否则, $count = count + 1$; 如果 $count = count_{\max}$, 转到步骤 d).
- Improve-Structure: $step = step + 1$. 如果 $step \% del_ratio = 0$, 删除某些中心点; 否则, 分裂某些中心点. 转到步骤 b).
- 算法结束, 输出最佳的聚类结果.

用一个例子来说明 Improve-Structure 操作的详细步骤, 图 1 表示了 K-means ($K=4$) 的运行结果. 首先, 对于任意一个中心点 μ_i , 将其分裂为两个子中心点 μ_i' 和 μ_i'' , 如图 2 所示. 然后对 μ_i 拥有的 POI 集合以 μ_i' 和 μ_i'' 为初始聚类中心点, 运行局部的 K-means ($K=2$) 算法. 当所有的 2-means 算法都收敛后 (图 3), 计算 μ_i 的分裂值: $BIC(\mu_i) - BIC(\mu_i', \mu_i'')$, 记为 $birth_value_i$. 由于 BIC 评分越大聚类效果越好, 所以, 如果 $birth_value_i < 0$, 说明 μ_i' 和 μ_i'' 比 μ_i 更符合 POI 的分布, 则保留 μ_i' 和 μ_i'' 而删除 μ_i ; 否则, 保留 μ_i 而删除 μ_i' 和 μ_i'' . 图 4 显示了存活下来的中心点. 当所有中心点的分裂值都大于零时, 算法获得了一个局部的最优解 (图 4); 为了得到全局最优解, 算法将分裂值最小的 $K/2$ 个中心点进行分裂, 并删除在这一过程中产生的不符合 POI 分布的中心点, 如图 5~7 所示.

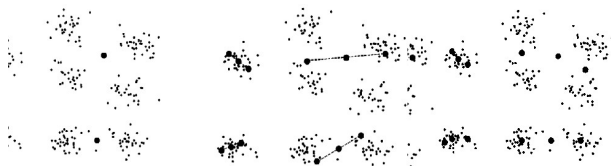


图 1 K-means ($k=4$) 运行结果 图 2 每个中心点分裂为两个子中心点 图 3 局部 2-means 收敛结果

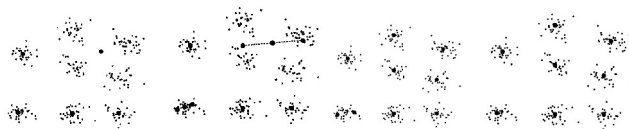


图 4 局部最优解

图 5 继续分裂

图 6 跳出局部最优解

图 7 全局最优解

本文使用 BIC 来衡量聚类结果的好坏. 对于给定的 POI 集合 D , 依据不同的 K 值 A-means 算法得到聚类结果 M_k . 根据 K-means 算法的假设, M_k 中的每个类满足球状高斯分布. 为了选择最佳的聚类结果, 使用贝叶斯后验概率 $Pr[M_k|D]$ 来判断 M_k 与 D 拟合的好坏, 概率值越大, M_k 的聚类效果则越好. 文献[7]提出, 后验概率的值可用 BIC 来估算:

$$BIC(M_k) = l(D) - \frac{n_k}{2} \times \log R \quad (1)$$

其中: $l(D)$ 为似然函数 $l(D)$ 在其参数 σ 取最大似然估计值 $\hat{\sigma}$ 时的值, n_k 表示 M_k 中变量的个数, $R = |D|$.

同时, 定义 $\mu_{(i)}$ 为 p_i 的中心点, $D_{(i)}$ 为 $\mu_{(i)}$ 拥有的 POI 集合, $R_{(i)} = |D_{(i)}|$. 所以, 方差的最大似然估计值为

$$\hat{\sigma}^2 = \frac{1}{R} \sum_i \|x_i - \mu_{(i)}\|^2 \quad (2)$$

p_i 的概率为

$$Pr(p_i) = \frac{R_{(i)}}{R} \times \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2} \|p_i - \mu_{(i)}\|^2\right) \quad (3)$$

D 的似然函数为

$$l(D) = \log \prod_i Pr(p_i) = \sum_i \left(\log \frac{1}{2\pi\sigma^2} - \frac{1}{2\sigma^2} \|p_i - \mu_{(i)}\|^2 + \log \frac{R_{(i)}}{R} \right) \quad (4)$$

对于中心点 μ_m ($1 \leq m \leq K$) 拥有的数据集合 D_m , 其似然函数的最值为

$$l(D_m) = -R_m \log(2\pi) - R_m \log(\sigma^2) - \frac{R_m}{2} + R_m \log R_m - R_m \log R \quad (5)$$

则 $l(D) = \sum_i l(D_i)$.

1.3 节点移动路径的生成

在 1.2 节中, 使用 A-means 算法将 POI 集合划分为 K 个子集. 在本节中, 建立了 K 条从汇聚节点出发的移动路径, 每条路径遍历同一子集内的所有 POI, 并最终回到汇聚节点; 同时在路径上部署一定数量的移动传感器节点来负责 POI 处信息的收集处理, 并将信息传递到汇聚节点. 在汇聚节点与一个 POI 子集之间建立一条最短的回路是典型的 TSP 问题, 本文使用 TSP 问题的一种多项式时间近似算法 PTAS (polynomial-time approximation scheme)^[8] 来生成路径.

设某个节点所在回路为 $circ_x$, 其长度 L_x 必须满足以下条件: 回路上所有 p_i ($i \neq 0$) 都有 $\frac{L_x}{v n_x} \leq T_i$, 其中 n_x 为回路 $circ_x$ 上移动节点的数量. 因此, 单条 POI 遍历路径上所需节点数量 n_x 由式(6)确定:

$$n_x = \lceil L_x / v T_{\min}^x \rceil \quad (6)$$

其中: $T_{\min}^x$ 表示该条路径上所有 POI 中最小的覆盖间隔需求.

2 实验结果及分析

2.1 实验设置

为了检验上述 Sweep Coverage 机制的性能, 本文以真实场景的数据 CNFDB (Canadian national fire database)^[9] 进行了仿

真。本文选取受害森林面积大于 200 公顷的重大火灾作为 POI,每个 POI 的覆盖间隔需求在 24 ~ 72 h 间随机选取。数据采集节点置于小型无人机(飞行速度 70 ~ 220 km/h)上,且数据缓冲区足够大,不存在溢出。本文分别以不同年份的数据、不同的节点移动速度以及不同的汇聚节点位置进行了实验,所有实验结果均为十次仿真的平均值。

2.2 CBSC、VRP-SWEEP、CSWEEP 节点数量对比

为了比较 CBSC、VRP-SWEEP 和 CSWEEP 三种机制所需要的节点数量,本文选取 1990—1999 年的数据进行了实验,节点移动速度设置为 100 km/h,汇聚节点位于场景中央。由于 CSWEEP 机制只应用于所有 POI 覆盖间隔要求相同的场景,故实验时以最小的覆盖间隔要求为基准来进行计算。从表 1 可以看出,CBSC 算法计算出的节点数目明显少于 VRP-SWEEP 和 CSWEEP 算法。这是由于在 CBSC 机制中,距离较近的 POI 总是包含在同一条路径中,从而降低了路径长度,有效地减少了节点的使用数量。

表 1 节点数目对比

年份	POI 数量	所需传感器节点数量		
		CBSC	VRP-SWEEP	CSWEEP
1990	255	21	31	43
1991	319	22	34	48
1992	153	14	21	32
1993	212	17	25	38
1994	395	21	36	49
1995	422	22	38	50
1996	461	23	42	58
1997	127	12	18	25
1998	514	29	51	66
1999	284	21	32	44

2.3 节点移动速度对 Sweep Coverage 机制的影响

为了研究节点移动速度对 Sweep Coverage 机制的影响,本文选取 1999 年的数据进行了实验,汇聚节点位于场景中央,并设置节点的移动速度为 90 ~ 180 km/h。从图 8 可以看出,随着节点移动速度的增加,三种机制所需节点数量呈显著下降趋势。这个现象表明,通过提升移动节点的移动速度,可以有效地减少节点的使用数量。



图 6 跳出局部最优解

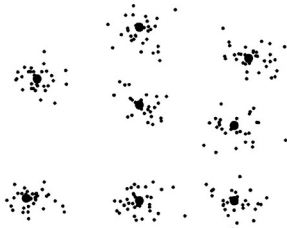


图 7 全局最优解

2.4 汇聚节点位置对 Sweep Coverage 机制的影响

为了研究汇聚节点位置对 Sweep Coverage 机制的影响,本文改变汇聚节点的位置,分别对 CBSC、VRP-SWEEP 以及 CSWEEP 机制进行了实验;从 1999 年的数据中随机选取 100 ~ 280 个 POI 进行实验,节点的移动速度设为 100 km/h。如图 9 所示,将汇聚节点置于场景中央,CBSC 机制所需的节点数量明显少于 VRP-SWEEP 和 CSWEEP 机制,且随着 POI 数量的增多,节点数量的增长也较为缓慢。从图 10 可以看出,将汇聚节点置于场景边缘时,CSWEEP 机制受到的影响不大,CBSC 机制所需的节点数量略有增加,而 VRP-SWEEP 机制所需的节点数量

则明显增多,几乎与 CSWEEP 机制相同。以上实验表明,CBSC 机制对 POI 数量的变化适应性较好,CSWEEP 机制可以较好地适应汇聚节点位置的变化,而 VRP-SWEEP 机制对汇聚节点的位置变化则较为敏感。

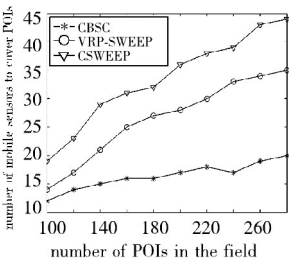


图 9 汇聚节点位于场景中央

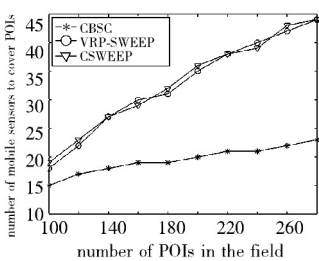


图 10 汇聚节点位于场景边缘

3 结束语

本文提出了一种基于聚类的 Sweep Coverage 机制 CBSC。使用一种自适应的聚类算法 A-means 将 POI 集合划分为 K 类;使用 TSP 问题的算法,在汇聚节点与每个类之间都建立一条节点的移动路径。实验结果表明,在真实的监控场景下,CBSC 机制不仅使用更少的节点,而且更好地适应了 POI 数量的变化。笔者下一步计划在真实的环境中验证本文的覆盖机制,同时将进一步研究节点移动对覆盖机制造成的影响,对本文的 Sweep Coverage 机制加以完善。

参考文献:

[1] BAI Xiao-le, XUAN Dong, YUN Zi-qiu, *et al.* Complete optimal deployment patterns for full-coverage and k -connectivity ($k \leq 6$) wireless sensor networks[C]//Proc of the 9th ACM International Symposium on Mobile Ad hoc Networking and Computing. New York: ACM Press, 2008: 401-410.

[2] YANG Guan-qun, QIAO Da-ji. Barrier information coverage with wireless sensors[C]//Proc of IEEE Conference on Computer Communications. 2009:918-926.

[3] CHENG Wei-fang, LI Mo, LIU Ke-bin, *et al.* Sweep coverage with mobile sensors[C]//Proc of IEEE International Symposium on Parallel and Distributed Processing. Washington DC: IEEE Computer Society, 2008: 1-9.

[4] 王伟,林峰,周激流. Sweep Coverage 中的节点移动控制[J]. 四川大学学报:自然科学版,2010,47(5):1015-1019.

[5] JAIN A. Data clustering: 50 years beyond K-means[J]. Pattern Recognition Letters, 2010, 31(8): 651-666.

[6] PELLEGRINI D, MOORE A. X-means: extending K-means with efficient estimation of the number of clusters[C]//Proc of the 17th International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 2000: 727-734.

[7] KASS R E, WASSERMAN L. A reference Bayesian test for nested hypothesis and its relationship to the Schwarz criterion[J]. Journal of the American Statistical Association, 1995, 90(431): 928-934.

[8] ARORA S. Polynomial time approximation schemes for Euclidean TSP and other geometric problems[C]//Proc of the 37th Annual IEEE Symposium on Foundations of Computer Science. [S. l.]: IEEE Computer Society, 1996: 2-11.

[9] Canadian Forest Service. Canadian national fire database: agency fire data[EB/OL]. http://cwfs.cfs.nrcan.gc.ca/en_CA/nfdb.