

# 基于 VF-CS 的移动传感器网络覆盖优化算法

李光辉<sup>1,2,3</sup>, 胡世红<sup>1,3</sup>

(1. 江南大学物联网工程学院, 江苏 无锡 214122; 2. 江苏省无线传感网高技术研究重点实验室, 江苏 南京 210003;  
3. 物联网技术应用教育部工程技术研究中心, 江苏 无锡 214122)

**摘 要:** 在野外环境部署大规模传感器网络时, 往往采用随机部署方式, 导致覆盖率不高。为此提出一种基于虚拟力 (virtual force) 扰动和布谷鸟搜索 (CS, Cuckoo search) 的移动传感器网络覆盖优化算法 (VF-CS)。首先, 对传感器节点进行 Voronoi 图划分, 形成独立的泰森多边形 (Thiessen polygon)。其次, 对泰森多边形内的节点进行虚拟力的分析, 将多边形顶点和邻居节点的作用力作为布谷鸟搜索位置更新的扰动因子。最后, 通过布谷鸟搜索引导节点移动实现覆盖优化。仿真实验结果表明, 与以往基于 Voronoi 图的覆盖优化算法相比, VF-CS 算法提高了覆盖率, 减少了节点平均移动距离。

**关键词:** 移动传感网络; 虚拟力; 布谷鸟搜索; 覆盖率; 优化

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2018039

## Coverage optimization algorithm based on VF-CS in mobile sensor network

LI Guanghui<sup>1,2,3</sup>, HU Shihong<sup>1,3</sup>

1. School of Computer Technology, Jiangnan University, Wuxi 214122, China

2. Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China

3. Research Center of IoT Technology Application Engineering (MOE), Wuxi 214122, China

**Abstract:** A random placement of large-scale sensor network in the outdoor environment often causes low coverage. An area coverage optimization algorithm of mobile sensor network (MSN) based on virtual force perturbation and Cuckoo search (VF-CS) was proposed. Firstly, the virtual force of the sensor nodes within the Thiessen polygon was analyzed based on the partitioning of Voronoi diagram of the monitoring area. Secondly, the force of polygon vertices and neighbor nodes was taken as the perturbation factor for updating the node's location of the Cuckoo search (CS). Finally, the VF-CS guided the node to move so as to achieve the optimal coverage. The simulation results demonstrate that the proposed algorithm has higher coverage and shorter average moving distance of nodes than the Voronoi diagram based algorithms in literatures.

**Key words:** mobile sensor network, virtual force, Cuckoo search, coverage, optimization

## 1 引言

区域覆盖是无线传感器网络中的一个基本问

题, 它直接影响了网络的服务质量。当网络拓扑结构发生变化时, 保持或增加网络的整体覆盖率十分重要<sup>[1]</sup>。移动传感器网络是由配备有移动平台的传

收稿日期: 2017-10-12; 修回日期: 2018-02-14

通信作者: 李光辉, ghli@jiangnan.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61472368, No.61174023); 江苏省重点研发计划基金资助项目 (No.BE2016627); 中央高校基本科研业务费专项资金基金资助项目 (No.RP51635B); 无锡市国际科技研发合作基金资助项目 (No.CZE02H1706)

**Foundation Items:** The National Natural Science Foundation of China (No.61472368, No.61174023), The Key Project of the Jiangsu Provincial Research and Development (No.BE2016627), The Fundamental Research Funds for the Central Universities (No.RP51635B), International Scientific and Technological Cooperation Projects of Wuxi (No.CZE02H1706)

传感器节点组成,以便在初始部署后允许传感器节点移动<sup>[2]</sup>。越来越多的应用场合需要移动传感器网络,如智能交通系统、安全系统、社会交互等复杂场景<sup>[3,4]</sup>。移动传感器网络以其自然优势能够很好地适应网络拓扑结构变化,并将节点移动到正确的位置,从而提高区域覆盖率。

目前,针对传感器网络覆盖优化问题,已经有很多研究成果<sup>[5~15]</sup>。例如,丁旭等<sup>[9]</sup>通过研究区域覆盖的特征,提出特征点集的概念并改进了粒子群算法,将传统的区域覆盖转化为基于特征点集优化覆盖问题。李劲等<sup>[10]</sup>结合博弈论提出一种分布式的覆盖优化算法,算法收敛时网络能达到较高的覆盖率。近年来,由于移动传感器网络相比于静态的传感器网络在应对拓扑变化方面更有优势<sup>[11]</sup>,移动节点的部署优化研究领域也取得了重要进展<sup>[16~22]</sup>。其中,Voronoi 图是移动传感器网络中常用的覆盖分析方法,涂志亮等<sup>[17]</sup>针对移动传感器网络中动态目标的监测优化问题,建立基于 Voronoi 剖分的监测性能评价函数,提高网络覆盖质量,提出基于群集控制的传感器节点部署分布式控制方法。Boukerche 等<sup>[18]</sup>提出一种基于 Voronoi 图的技术,在传感器节点位置未知的条件下,通过定向天线获取邻居节点位置以及局部平面扫描算法寻找覆盖漏洞,从而提高全局覆盖。Lee 等<sup>[19]</sup>提出了一种基于 Voronoi 多边形形心的部署策略(CBS, centroid-based scheme),将区域覆盖问题转化为每个传感器节点所属泰森多边形的覆盖优化问题,但 CBS 算法没有考虑邻居传感器节点的覆盖,容易出现覆盖重叠等问题。方伟等<sup>[20]</sup>在 CBS 的基础上分析了 Voronoi 多边形盲区情况,提出一种基于 Voronoi 多边形盲区的覆盖控制部署策略(BCBS, blind-zone centroid-based scheme),有效提高了覆盖率,但由于其在分析盲区构造与多边形盲区相近的多边形时,计算复杂度高,导致算法耗时偏长。Abo-Zahhad 等<sup>[21]</sup>提出一种基于 Voronoi 图的集中式免疫部署算法(CIVDA, centralized immune-Voronoi deployment algorithm),其利用 Voronoi 图的特性折中异构传感器网络的覆盖与能量消耗,算法收敛快,但存在“早熟”现象,而且在同构网络中,该算法在覆盖率方面还有待提高。

针对以往研究中存在的问题,本文提出了一种基于虚拟力扰动和布谷鸟搜索的移动传感器网络覆盖优化算法,该算法在 Voronoi 图划分的基础上,

对泰森多边形内的传感器节点进行虚拟力的分析,不仅考虑了多边形顶点的作用力以分析覆盖漏洞区域方向,同时也考虑了邻居节点的作用力以减少重叠覆盖面积,将节点所受的总虚拟力作为布谷鸟搜索位置更新的扰动因子,引导节点更新位置朝着存在覆盖漏洞区域的方向移动,并且移动范围局限在所属的泰森多边形,加快了全局收敛速度,提高了整体覆盖率,也减少了节点的平均移动距离。

## 2 预备知识

### 2.1 问题模型

在二维正方形监测区域内,监测区域大小  $\Gamma$  为  $L \times L$ ,在  $\Gamma$  的范围内随机抛撒  $N$  个传感器节点,传感器节点集合  $S = \{S_1, S_2, \dots, S_N\}$ 。所有传感器节点具有相同的性能,节点的感知半径为  $R_s$ ,通信半径为  $R_c$ ,点  $S_i$  位置坐标记为  $(x_i, y_i)$ ,  $i=1, 2, \dots, N$ 。为了有效计算网络覆盖率,将监测区域网格化为  $m \times n$  个像素点,记为  $H_j = (x_j, y_j)$ ,  $j \in \{1, 2, \dots, m \times n\}$ ,用感知模型测量每个像素点是否被传感器节点覆盖<sup>[19]</sup>。节点  $S_i$  与像素点  $H_j$  的距离记为

$$D(S_i, H_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

传感器节点的感知范围是一个以节点坐标为圆心、 $R_s$  为半径的封闭圆盘区域。像素点  $H_j$  被点  $S_i$  感知的概率  $p(S_i, H_j)$  为

$$p(S_i, H_j) = \begin{cases} 1, & D(S_i, H_j) \leq R_s \\ 0, & D(S_i, H_j) > R_s \end{cases} \quad (2)$$

所有传感器节点同时对像素点进行感知的联合感知概率为

$$C(S, H_j) = 1 - \prod_{i=1}^N (1 - p(S_i, H_j)) \quad (3)$$

一般情况下,像素点可以同时被多个传感器节点同时感知,只要有一个节点覆盖了像素点,则称该像素点被覆盖。由此,监测区域的覆盖率  $Q_{cov}$  为传感器节点集覆盖的像素点的总数与监测区域  $\Gamma$  的总像素点的比值,记为

$$Q_{cov} = \frac{\sum_{j=1}^{m \times n} C(S, H_j)}{m \times n} \quad (4)$$

网络中所有节点覆盖的监测区域面积与节点

感知范围面积总和的比值称为节点的覆盖效率。节点覆盖效率  $C_e$  反映网络中节点的冗余程度,  $C_e$  越大表示节点的冗余程度越小, 节点分布越均匀。具体计算式为

$$C_e = \frac{Q_{\text{cov}} \times L \times L}{N \times (\pi \times R_s^2)} \quad (5)$$

## 2.2 Voronoi 图及泰森多边形

**定义 1** Voronoi 图。给定一组传感器节点  $S = \{S_1, S_2, \dots, S_N\}$ , 将二维平面  $\Gamma$  分成  $N$  个单元  $G = \{G_1, G_2, \dots, G_N\}$ , 使单元  $G_i (i=1, 2, \dots, N)$  中恰好包含一个传感器节点  $S_i$ ,  $S_i$  称为  $G_i$  的生成节点, 并且单元  $G_i$  中的任意点比其他传感器节点都更接近其生成节点  $S_i$  [18], 即

$$G_i = \bigcap_{i \neq j, j=1}^N \{p | D(S_i, p) < D(S_j, p) : p \in \Gamma\} \quad (6)$$

其中,  $p$  为监测区域  $\Gamma$  中的任意一点, 单元  $G_i$  组成的图称为监测区域的 Voronoi 图。例如, 当  $8 \text{ m} \times 8 \text{ m}$  大小的监测区域内随机部署了 15 个传感器节点, 其 Voronoi 图如图 1 所示, 每个传感器节点对应一个单元。

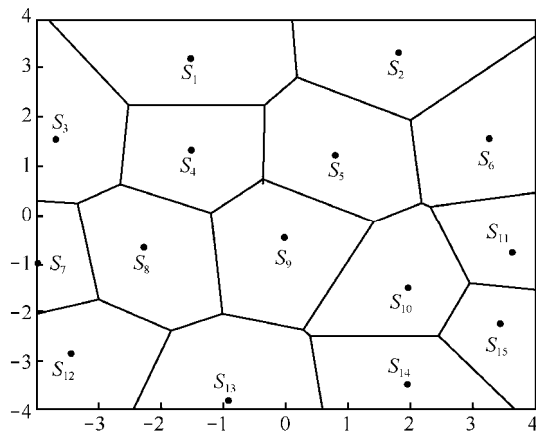


图1 Voronoi 图

**定义 2** 泰森多边形。Voronoi 图将二维平面划分成  $N$  个单元, 且每个单元只包含一个传感器节点, 称每个单元为传感器节点所属的泰森多边形。

如图 2 所示, 节点  $S_i$  所属的泰森多边形为  $G_i = \langle V_i, E_i \rangle$ , 其中,  $V_i$  表示泰森多边形顶点集合,  $E_i$  表示泰森多边形的边集合, 节点  $S_i$  的泰森多边形顶点集合  $V_i = \{V_{i1}, V_{i2}, V_{i3}, V_{i4}, V_{i5}\}$ , 用  $N_i$  表示节点  $S_i$  的邻居节点集合,  $N_i = \{S_{i1}, S_{i2}, S_{i3}, S_{i4}, S_{i5}\}$ ,  $i=1, 2, \dots, N$ 。

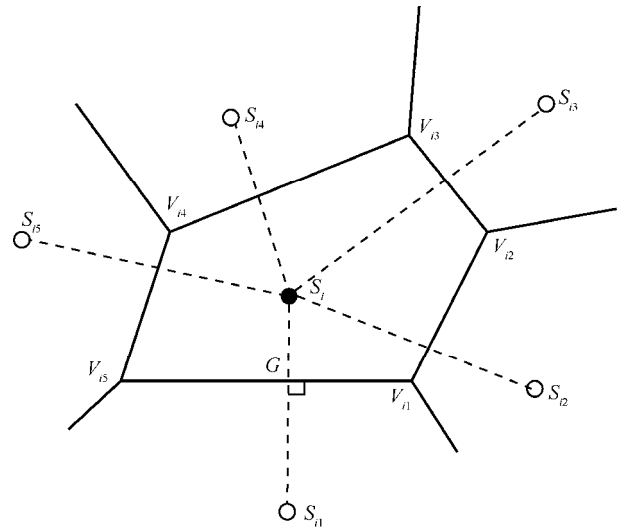


图2 泰森多边形

**定义 3** 泰森多边形形心。泰森多边形形心是指将多边形分成面积相等的 2 个部分所有直线的交点 [19], 如图 3 所示。

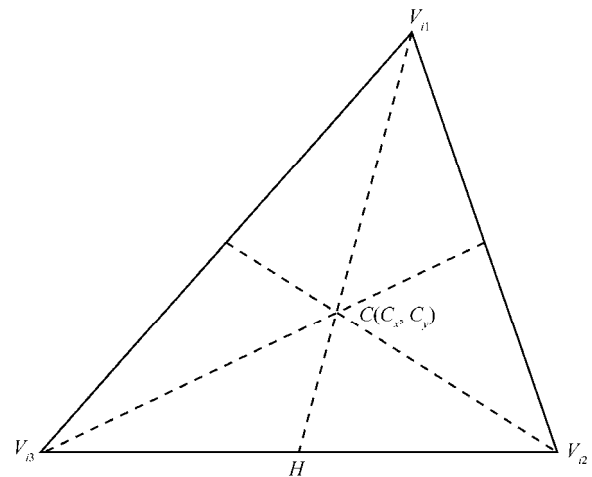


图3 多边形形心

下面以图 3 的三角形为例, 说明形心的计算过程。这里,  $C(C_x, C_y)$  为三角形的形心, 直线  $V_{i1}H$  将  $\Delta V_{i1}V_{i2}V_{i3}$  分成面积相等的 2 个部分:  $\Delta V_{i1}V_{i3}H$  和  $\Delta V_{i1}V_{i2}H$ 。节点  $S_i$  所属泰森多边形的顶点集合  $V_i = \{V_{ij} = (x_{ij}, y_{ij}) | j=1, 2, \dots, n\}$ , 则形心  $C(C_x, C_y)$  的计算为

$$\begin{cases} C_x = \frac{1}{6A} \sum_{j=0}^{n-1} (x_{ij} + x_{i(j+1)}) (x_{ij}y_{i(j+1)} - x_{i(j+1)}y_{ij}) \\ C_y = \frac{1}{6A} \sum_{j=0}^{n-1} (y_{ij} + y_{i(j+1)}) (x_{ij}y_{i(j+1)} - x_{i(j+1)}y_{ij}) \end{cases} \quad (7)$$

$$A = \frac{1}{2} \sum_{j=0}^{n-1} (x_{ij} y_{i(j+1)} - x_{i(j+1)} y_{ij}) \quad (8)$$

Voronoi 图在传感器节点部署算法中常用于检测覆盖漏洞<sup>[23,24]</sup>。本文算法基于 Voronoi 图,使每个传感器节点以其所在的泰森多边形为移动范围区域,实行以覆盖率最大化为目标的优化算法,具体将在第 3 节进一步介绍。

### 2.3 布谷鸟搜索算法

布谷鸟搜索是由 Deb 等<sup>[25]</sup>提出的一种基于布谷鸟寻窝孵蛋的繁殖习性以及 Levy 飞行特性的新型优化算法。CS 算法具有参数设置少、随机搜索路径优、收敛速度快等优点,已成功应用于工程优化等实际问题中<sup>[26,27]</sup>。

CS 算法有以下 3 个规则<sup>[25]</sup>。

1) 每个布谷鸟每次只产一个蛋,并随机选择鸟窝孵化。

2) 在随机选择的一组鸟窝中,最好的鸟窝将被保留到下一代。

3) 可利用的鸟窝数量  $m$  是固定可选的,一个鸟窝的主人发现外来鸟蛋的概率  $p_a \in [0,1]$ ,  $p_a$  一般取固定值<sup>[25]</sup>。

在以上 3 个基本规则下,布谷鸟寻窝的路径和位置更新式为

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\lambda), i=1,2,\dots,m; t=1,2,\dots \quad (9)$$

其中,  $x_i^{(t)}$  是第  $t$  代的第  $i$  个鸟窝的位置,  $\alpha$  是步长控制量,其取值与求解的实际问题规模有关,一般令  $\alpha=1$ 。式(9)本质上是一种随机游走方程,一般情况下,一个随机游走方程是一个马尔可夫链,其未来位置由当前位置(方程的第一项)和转移概率(方程的第二项)所决定;  $\oplus$  表示点对点乘法,  $Levy(\lambda)$  为莱维随机搜索路径,莱维行走的步长满足一个重尾的稳定分布,是一种短距离和偶尔较长距离相间隔的行走方式,并且满足

$$Levy(\lambda) \sim \frac{\Phi \times u}{|v|^{\frac{1}{\beta}}}, \beta \in (1,3] \quad (10)$$

这里,  $u$  和  $v$  都服从标准正态分布,即

$$\begin{cases} u \sim N(0, \Phi^2) \\ v \sim N(0, \Phi^2) \end{cases} \quad (11)$$

$\Phi$  满足

$$\Phi = \left\{ \frac{\Gamma(1+\beta) \sin\left(\pi \frac{\beta}{2}\right)}{\Gamma\left\{\left[\frac{1+\beta}{2}\right] \beta 2^{\frac{\beta-1}{2}}\right\}} \right\} \quad (12)$$

由此可得,布谷鸟寻窝的路径和位置更新式为

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \frac{\Phi u}{|v|^{\frac{1}{\beta}}}, i=1,2,\dots,m \quad (13)$$

### 2.4 最佳部署定理

在传感器网络中,用最少数量的传感器节点部署监测区域,同时满足节点间没有覆盖空隙,从而区域达到满覆盖率,称这样的节点部署为最佳部署<sup>[28]</sup>。针对无线传感器网络的覆盖率、连通度及容错性能优化, Ammari 等<sup>[28]</sup>提出了以下的最佳部署定理。

**定理 1** 传感器网络最佳部署定理。如图 4 所示, 3 个相邻节点  $S_1$ 、 $S_2$  和  $S_3$ , 节点感知圆盘间无间隙,任何一对相邻节点感知圆盘有恒定大小的重叠, 3 个相邻节点感知圆盘相交于一个点,连接 3 个节点位置组成一个等边三角形,节点间距离为  $\sqrt{3}R_s$ ,按照这样的方式配置传感器节点的部署策略是最佳的<sup>[28]</sup>。

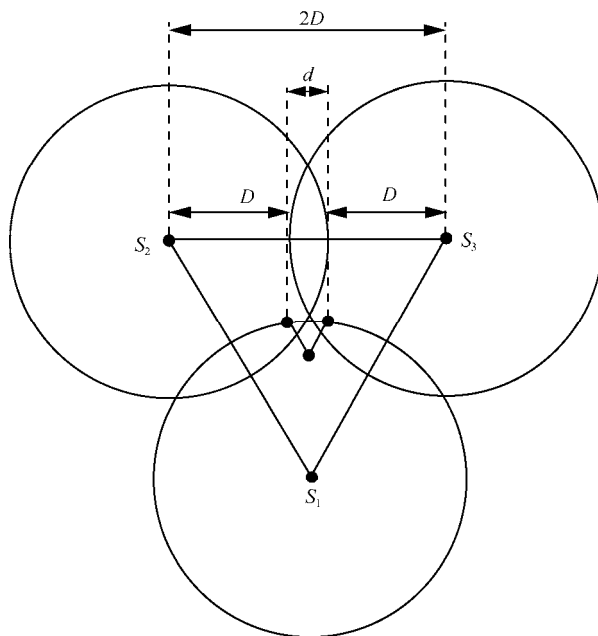


图 4 最佳部署示意

**证明** 考虑图 5 所示的传感器配置,传感器节点的感知圆盘间没有重叠,理论上来说是最大化利用了传感器节点的感知范围,则部署的传感器节点

数量将达到最小。但任意 3 个节点的感知圆盘间都存在间隙区域，在间隙区域有一个等边三角形，即  $\triangle ABC$ ，为了覆盖所有的间隙区域满足全覆盖，需要覆盖  $\triangle ABC$ 。为了达到此目的，传感器节点  $S_2$  和  $S_3$  应当水平并相对移动直到它们的感知圆盘圆周相交于  $\triangle ABC$  的重心位置  $g$ ，但  $\triangle ABC$  还没有完全被覆盖，为了把剩下的间隙覆盖， $S_1$  应当垂直并自下向上移动直到感知圆盘圆周触及到点  $g$ 。实际上，间隙区域中距离节点  $S_1$ 、 $S_2$  和  $S_3$  最远的点就是点  $g$ ，因此，要想覆盖所有间隙，就需要覆盖  $\triangle ABC$  的重心。这样就得到了如图 4 所示的最佳部署配置，即任何一对相邻传感器节点的感知圆盘有恒定大小的重叠，并且任意 3 个相邻传感器节点的感知圆盘圆周相交于一个点，按照这样的方式部署传感器节点数量最少并且达到全覆盖，故为最佳部署。

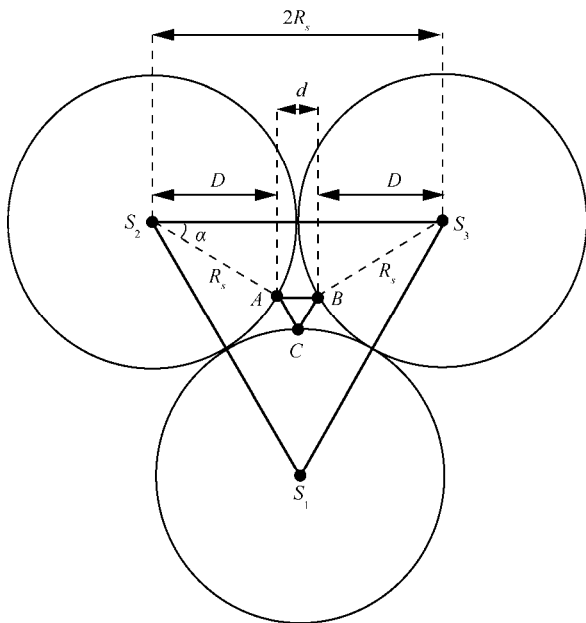


图5 传感器感知圆盘相切的最小间隙

图 5 中任意一对感知圆盘相切的传感器节点之间的最大距离为  $2R_s$ ，为了覆盖  $S_1$ 、 $S_2$  和  $S_3$  之间的覆盖间隙，节点间的距离应为  $2R_s - d$ ，因为  $D = R_s \cos(\alpha) = \frac{\sqrt{3}R_s}{2}$ ， $\alpha = \frac{\pi}{6}$ ，且  $d = 2R_s - 2D$ ，即  $d = 2R_s - \sqrt{3}R_s$ ，由此得到节点间的距离为  $2R_s - d = \sqrt{3}R_s$ 。

### 3 基于 Voronoi 图的 VF-CS 算法

本文提出了一种基于虚拟力扰动和布谷鸟搜

索的覆盖优化 (VF-CS) 算法。布谷鸟搜索中的莱维飞行过程采用随机步长，为防止其在搜索过程中跳出特定区域，使位置更新后节点的移动距离过大，VF-CS 算法利用了泰森多边形的概念，把泰森多边形中的节点受力作为布谷鸟搜索中位置更新的扰动因子，引导节点朝着存在覆盖漏洞区域的方向移动，并且移动范围局限在节点所属的泰森多边形，从而减少平均移动距离。VF-CS 算法在 Voronoi 图的基础上，假设节点所属的泰森多边形为其移动范围，采用布谷鸟搜索寻找最优移动位置，并将虚拟力引入布谷鸟搜索路径中，优化路径，防止算法出现“早熟”现象，同时加快全局收敛速度。

#### 3.1 虚拟力分析

由于设定节点只在所属泰森多边形内移动，所以可假设其只受所属泰森多边形顶点以及相邻节点的作用力。监测区域内的节点完成 Voronoi 图分割后，节点所属泰森多边形内的覆盖情况会根据节点的位置以及相邻节点的部署发生变化。为了提高多边形内的覆盖率，节点应往多边形中存在覆盖漏洞的方向移动，同时为了减少覆盖重叠，相邻节点间需保持一定的距离。

##### 3.1.1 多边形顶点的作用力

如图 6 所示，节点  $S_i$  所属的泰森多边形  $G_i$ ，顶点集合  $V_i = \{V_{i1}, V_{i2}, V_{i3}, V_{i4}, V_{i5}\}$ ，若顶点在节点  $S_i$  感知范围外，则顶点对节点产生引力，反之产生斥力，具体作用力为

$$F_{vij} = \begin{cases} (w_a(D(V_{ij}, S_i) - R_s), \alpha_{ij}), & D(V_{ij}, S_i) > R_s \\ 0, & D(V_{ij}, S_i) = R_s \\ (w_r(R_s - D(V_{ij}, S_i)), \alpha_{ij} + \pi), & D(V_{ij}, S_i) < R_s \end{cases} \quad (14)$$

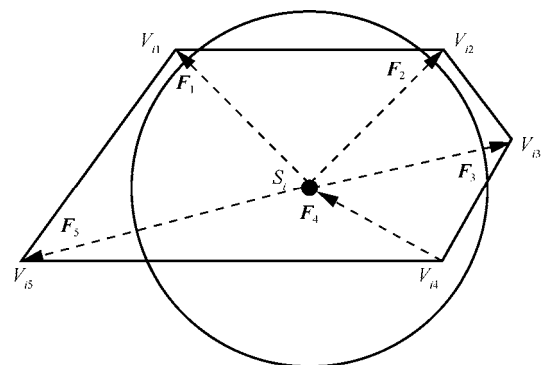


图6 多边形顶点对节点的作用力

其中,  $w_a(w_r)$  表示对引力(斥力)的度量,  $\alpha_{ij}$  表示向量  $F_{vij}$  的方向。由此可分析, 图 6 中顶点  $V_{i1}$ 、 $V_{i2}$ 、 $V_{i3}$ 、 $V_{i5}$  都位于节点  $S_i$  感知范围外, 对节点产生引力; 顶点  $V_{i4}$  位于节点  $S_i$  感知范围内, 对节点产生斥力。

### 3.1.2 相邻节点间的作用力

传感器节点的覆盖范围是以节点位置为圆心、 $R_s$  为半径的圆盘, 由定理 1 可知, 图 4 所示的传感器配置为最佳部署, 此时圆心距为  $\sqrt{3}R_s$ 。

如图 7 所示, 节点  $S_i$  所属的泰森多边形  $G_i$ , 其邻居节点集  $N_i = \{S_{i1}, S_{i2}, S_{i3}, S_{i4}, S_{i5}\}$ 。由 2.4 节最佳部署定理知, 相邻节点间的欧氏距离保持在  $\sqrt{3}R_s$  可以达到最大的覆盖效率, 因此, 假设相邻节点间存在作用力, 若节点间的欧氏距离大于  $\sqrt{3}R_s$  且小于  $R_c$ , 则产生引力; 若欧氏距离小于  $\sqrt{3}R_s$ , 则产生斥力, 具体作用力为

$$F_{sij} = \begin{cases} (k_a(D(S_{ij}, S_i) - R_s), \beta_{ij}), & \sqrt{3}R_s < D(S_{ij}, S_i) < R_c \\ 0, & D(S_{ij}, S_i) = \sqrt{3}R_s \\ (k_r(R_s - D(S_{ij}, S_i)), \beta_{ij} + \pi), & D(S_{ij}, S_i) < \sqrt{3}R_s \end{cases} \quad (15)$$

其中,  $k_a(k_r)$  表示对引力(斥力)的度量,  $k_{ij}$  表示向量  $F_{sij}$  的方向。由此可分析, 图 7 中节点  $S_{i1}$ 、 $S_{i4}$  和  $S_{i5}$  与节点  $S_i$  的欧氏距离大于  $\sqrt{3}R_s$ , 对节点产生引力; 节点  $S_{i2}$  与节点  $S_i$  的欧氏距离为  $\sqrt{3}R_s$ , 则不产生任何作用力; 节点  $S_{i3}$  与节点  $S_i$  欧氏距离小于  $\sqrt{3}R_s$ , 则产生斥力。

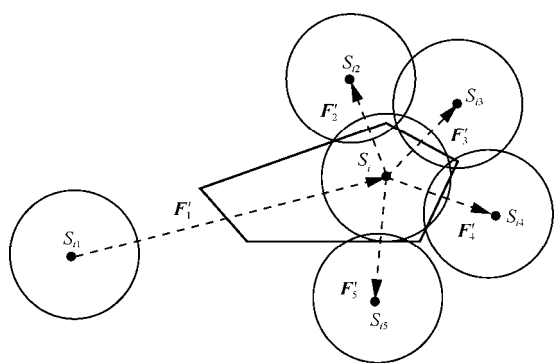


图 7 邻居节点作用力

### 3.1.3 虚拟力扰动因子的计算

由以上分析可知, 节点  $S_i$  所受的总作用力  $F_i$  为

$$F_i = \sum_{j=1}^{N(V_i)} F_{vij} + \sum_{j=1}^{N(N_i)} F_{sij} \quad (16)$$

其中,  $N(V_i)$  表示节点  $S_i$  所在泰森多边形顶点的个数,  $N(N_i)$  表示节点  $S_i$  邻居节点的个数。为了使布谷鸟搜索算法搜索路径优化, 本文在式(13)中引入虚拟力的影响因子, 得到虚拟力扰动的传感器节点位置更新式为

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \frac{\Phi \times u}{|v|^{\frac{1}{\beta}}} + \alpha f_i, i=1, 2, \dots, n \quad (17)$$

$$f_i = \begin{cases} \frac{F_{ix}}{F_i} \times \max\_dis \times e^{\frac{-1}{|F_i|}} \\ \frac{F_{iy}}{F_i} \times \max\_dis \times e^{\frac{-1}{|F_i|}} \end{cases} \quad (18)$$

其中,  $f_i$  为节点  $S_i$  受到虚拟作用力的移动距离, 称为虚拟力扰动因子,  $\max\_dis$  表示节点的最大移动距离, 具体算法如算法 1 所示。

#### 算法 1 虚拟力扰动因子计算

输入 传感器节点  $S_i$ , 所属泰森多边形顶点集  $V_i$ , 邻居节点集  $N_i$

输出 节点的虚拟力扰动因子  $f_i$

- 1)  $F_{vi} = 0$ ,  $F_{si} = 0$
- 2) for  $j$  from 1 to  $N(V_i)$
- 3) 计算  $D(S_i, V_{ij})$ ;
- 4) 根据式(14)计算  $F_{vij}$ ,  $F_{vi} = F_{vi} + F_{vij}$ ;
- 5) end for
- 6) for  $j$  from 1 to  $N(N_i)$
- 7) 计算  $D(S_i, S_{ij})$ ;
- 8) 根据式(15)计算  $F_{sij}$ ,  $F_{si} = F_{si} + F_{sij}$ ;
- 9) end for
- 10) 由式(16), 得到  $F_i$ , 代入式(18), 求得虚拟力扰动因子  $f_i$

### 3.2 虚拟力扰动的布谷鸟搜索算法

该算法应用于覆盖优化, 寻找最优的移动位置。由 2.3 节可知, 可利用的鸟窝数量  $m$  是固定的, 表示节点可选择的移动位置数; 鸟窝位置  $X_j(t)$ ,  $j=1, 2, \dots, m$  表示节点的坐标; 鸟窝的适应值  $q_{-f_i}$  表示节点在所属泰森多边形内的覆盖率;  $q_{-f_{\max}}$  表示每组鸟窝最优适应值。详细算法如算法 2 所示。

#### 算法 2 VF-CS 算法

输入 节点  $S_i'$  的初始鸟窝位置集合  $P_i^{(0)}$ , 所属泰森多边形顶点集  $V_i$ , 邻居节点集  $N_i$

**输出** 节点的最佳移动位置  $S'_i$

- 1) 计算  $q\_f(P_i^{(0)})$ , 找出  $X_a^{(0)}$
- 2) while  $t < t_{\max}$
- 3) 保留  $X_a^{(t-1)}$ ;
- 4) for  $j$  from 1 to  $m$  and  $j \neq a$
- 5) 根据式(13), 更新位置;
- 6) end for
- 7) 增加  $X_a^{(t-1)}$ , 得到  $new\_Q_i^{(t)}$ ;
- 8) 计算  $q\_f(new\_Q_i^{(t)})$ , 与  $q\_f(P_i^{(0)})$  比较, 保留适应值大的鸟窝位置, 得到  $Q_i^{(t)}$ ;
- 9) for  $j$  from 1 to  $m$
- 10) if  $r < p_a$
- 11) 保留  $X_j^{(t)}$ ;
- 12) else 根据式(13), 更新位置;
- 13) end if
- 14) end for
- 15) 得到  $new\_Q_i^{(t)}$ ;
- 16) 计算  $q\_f(new\_Q_i^{(t)})$  与  $q\_f(Q_i^{(t)})$  比较, 保留适应值大的鸟窝位置, 得到  $Q_i^{(t)}$ ;
- 17) for  $j$  from 1 to  $m$
- 18) 调用虚拟力扰动因子算法;
- 19) 根据式(17), 更新位置;
- 20) end for
- 21) 得到  $new\_P_i^{(t)}$ ;
- 22) 计算  $q\_f(new\_P_i^{(t)})$ , 与  $q\_f(Q_i^{(t)})$  比较, 保留适应值大的鸟窝位置, 可得  $P_i^{(t)}$ ;
- 23) 计算  $q\_f(P_i^{(t)})$ , 找出  $X_a^{(t)}$ , 对应的适应值  $q\_f_{\max}^{(t)}$ ;
- 24) if  $\frac{q\_f_{\max}^{(t)} - q\_f_{\max}^{(t-1)}}{q\_f_{\max}^{(t-1)}} > 0.0001$
- 25)  $S'_i = X_a^{(t)}$ ; break
- 26) end if
- 27) end while
- 28)  $S'_i = X_a^{(t)}$

### 3.2.1 算法描述

首先, 找到初始鸟窝集合  $P_i^{(0)} = [X_1^{(0)}, X_2^{(0)}, \dots, X_m^{(0)}]$  中适应值最大的鸟窝  $X_a^{(0)}$ ,  $a \in \{0, 1, \dots, m\}$ , 进入循环,  $t$  为循环代数, 保留上代适应值最大的鸟窝位置  $X_a^{(t-1)}$ , 并用式(13)对其他鸟窝位置进行位置更新,

得到一组新的鸟窝位置, 对新的鸟窝位置计算适应值, 与上一代的鸟窝位置  $P_i^{(t-1)}$  比较, 保留适应值大的鸟窝位置, 得到一组较优的鸟窝位置  $Q_i^{(t)} = [X_1^{(t)}, X_2^{(t)}, \dots, X_m^{(t)}]$ ; 鸟窝主人判断外来鸟蛋的概率为一个随机数  $r, r \in [0, 1]$ , 将  $r$  与  $p_a$  比较, 若小于  $p_a$ , 则表示宿主发现外来鸟蛋, 该鸟窝位置将被舍弃, 不作为节点的候选移动位置, 即保留发现概率小的鸟窝, 并随机改变发现概率大的鸟窝位置, 得到一组新的鸟窝位置, 再对新的鸟窝计算适应值, 与  $Q_i^{(t)}$  比较, 保留适应值大的鸟窝位置, 得到一组较优的鸟窝位置  $Q_i^{(t)}$ ; 利用式(17)对鸟窝位置进行虚拟力扰动, 得到一组新的鸟窝位置  $Q_i^{(t)}$ , 并与  $Q_i^{(t)}$  比较, 保留适应值大的鸟窝位置, 从而得到一组更优的鸟窝位置  $P_i^{(t)}$ ; 找出  $P_i^{(t)}$  中最优的鸟窝位置  $X_a^{(t)}$ , 判断其适应值是否满足停止条件, 若满足则输出最优位置, 否则继续循环更新。

### 3.3 基于 VF-CS 算法的覆盖优化策略

本文所提的覆盖优化策略基于 Voronoi 图, 由 Lee 等<sup>[19]</sup>提出的 CBS 算法利用泰森多边形形心的性质有效地提高了覆盖率, 故将泰森多边形形心位置用于 VF-CS 算法的初始化, 整个覆盖优化策略步骤如下。

**Step1** 确定监测区域大小  $\Gamma$  以及要部署的传感器节点数量  $N$ , 在监测区域内随机部署节点, 节点集合为  $S = \{S_1, S_2, \dots, S_N\}$ 。

**Step2** 在监测区域划分 Voronoi 图, 构成泰森多边形集合  $V = \{V_1, V_2, \dots, V_N\}$ 。

**Step3** 计算节点  $S_i$  所属  $V_i$  的覆盖情况, 若全覆盖则转 Step6; 若存在覆盖漏洞, 进入 Step4。

**Step4** 计算节点  $S_i$  所属  $V_i$  的形心  $C_i$ , 在  $V_i$  内随机生成  $m-2$  个节点位置, 连同  $S_i$ 、 $C_i$  形成集合  $P_i^{(0)} = [X_1^{(0)}, X_2^{(0)}, \dots, X_m^{(0)}]$ , 其中,  $X_{(m-1)}^{(0)} = S_i$ ,  $X_{(m)}^{(0)} = C_i$ 。

**Step5** 通过 VF-CS 算法找到节点  $S_i$  的最优移动位置  $S'_i$ , 转 Step7。

**Step6**  $S_i$  位置保持不变, 即  $S'_i = S_i$ 。

**Step7** 重复 Step3~Step5, 直至所有节点在所属泰森多边形内找到最优移动位置并进行一次性移动。

**Step8** 重复 Step2~Step7, 直至全局收敛, 产生最终覆盖率。

### 3.4 算法时间复杂度分析

设  $N$  为节点个数,  $m$  为布谷鸟初始群体数,  $t_{\max}$  为最大迭代次数。VF-CS 算法首先划分 Voronoi 图, 其时间复杂度为  $O(N)$ ; 此后, 算法的主要计算过程在于布谷鸟搜索阶段, 对于每个节点, 按照 VF-CS 算法流程, 每迭代一次, 最多需要更新 3 次位置, 更新位置的时间复杂度为  $O(\max(m) \times N)$ , 所以在节点迭代最多的情况下所需的时间复杂度为  $O(t_{\max} \times N(\max(m) \times N))$ , 总的覆盖策略的最坏情况时间复杂度则为  $O(N + t_{\max} \times N(\max(m) \times N))$ 。而方伟等<sup>[20]</sup>的 BCBS 算法的最大时间复杂度为  $O(N + t_{\max} \times N(N^2))$ , 当  $m$  不大于  $N$  时,  $O(N + t_{\max} \times N(\max(m) \times N))$   $O(N + t_{\max} \times N(N^2))$ , 一般情况下,  $m$  取值都小于  $N$ , 所以 VF-CS 算法时间复杂度比 BCBS 算法要低。

## 4 实验分析

为了验证本文提出的 VF-CS 算法性能, 假设在不同大小的监测区域内随机投放了不同数量的传感器节点, 分别完成仿真实验。为了排除随机性引起的误差, 每种实验场景都进行了 20 次初始化, 每种算法运行 30 次, 取最终的平均结果进行比较。传感器节点的感知半径  $R_s$  为 7 m, 通信半径  $R_c$  为  $2R_s$ , 布谷鸟初始群体  $m$  取 10, 鸟窝主人发现外来鸟蛋的概率  $p_a$  一般取 0.25, 虚拟力下节点移动的最大距离  $\max\_dis$  为 5 m。实验是在 Intel® core i7 双核 CPU, 主频 3.2 GHz, 内存 8 GB, 操作系统 Windows 10 环境下进行的, 编程语言采用 Matlab 2017b。在相同的实验环境下, 选择 CBS<sup>[19]</sup>、BCBS<sup>[20]</sup> 以及 CIVDA<sup>[21]</sup> 算法进行对比实验。

### 4.1 仿真场景及参数设置

为了更全面地进行对比, 本文设计了 2 组仿真实验。实验 1 考虑了 3 种不同大小的监测区域在相同的检测区域部署不同数量的传感器节点情形。1) 100 m×100 m 区域, 分别部署 90、80、70、60、50 和 40 个传感器节点; 2) 200 m×200 m 区域, 分别部署 340、320、300、280、260 和 240 个传感器节点; 3) 350 m×350 m 区域, 分别部署 1 000、900、800、700、600 和 500 个传感器节点。将 VF-CS 算法和以往 3 种同类算法在不同部署环境中、不同节点规模下的覆盖率变化趋势进行横向对比, 并对 4 种算法的平均移动距离和算法耗时进行比较。实验 2 在实验 1 设置的 3 种部署环境中分别选取一种节点规模进行纵向对比, 100 m×100 m 取  $N=90$ , 200 m×200 m

取  $N=340$ , 350 m×350 m 取  $N=1\ 000$ , 观察 VF-CS 算法和以往同类算法在 3 种部署环境下覆盖率的变化趋势, 对节点平均移动距离和算法耗时进行比较。具体的实验参数设置如表 1 所示。

表 1 参数设置		
仿真	区域大小	节点个数 ( $N$ )
实验 1	100 m×100 m	90、80、70、60、50、40
	200 m×200 m	340、320、300、280、260、240
	350 m×350 m	1 000、900、800、700、600、500
实验 2	100 m×100 m	90
	200 m×200 m	340
	350 m×350 m	1 000

### 4.2 覆盖率

图 8 为实验 1 中监测区域为 100 m×100 m 的其中 4 种情形 ( $N=90$ 、80、70、60) 下的覆盖率比较结果。图 8(a)~图 8(d)分别给出了 4 种算法在不同节点规模下, 随着迭代次数 (round) 的增加, 覆盖率 (coverage) 的变化趋势, 当覆盖率不再增加时, 算法停止迭代。从图 8 可以发现, 覆盖率在前几轮迭代中急剧变化, 后慢慢趋于平稳。本文提出的 VF-CS 算法较其他 3 种算法, 覆盖率都有提高。例如图 8(c), 当  $N=80$  时, VF-CS 最终覆盖率达到 96.55%, BCBS 算法、CBS 算法以及 CIVDA 算法最终覆盖率分别为 95.70%、94.76% 和 85.16%, VF-CS 算法比 BCBS 算法覆盖率提高了 0.9%, 比 CBS 算法覆盖率提高了 1.9%。CBS 算法没有考虑邻居节点对泰森多边形的覆盖, 无法保证泰森多边形的形心位于多边形的覆盖漏洞区域中, 而 BCBS 算法分析了泰森多边形的覆盖漏洞, 并构造了覆盖漏洞多边形, 将节点移动到其形心, 从而在 CBS 算法的基础上提高了覆盖率。CIVDA 的覆盖率最低, 由图 8 可看出其收敛速度快, 是因为该算法出现了“早熟”现象, 导致其过早收敛, 难以获得全局最优结果。

表 2 给出了实验 1 所有情形下取得的最终覆盖率, 3 种部署环境下, 随着部署传感器节点数量的增加, 覆盖率也随之提高, 由表 2 可看出, 本文 VF-CS 算法的覆盖率高于其他 3 种算法, 且当部署节点数量较小时, VF-CS 算法取得的覆盖率优势更为明显。例如, 在 200 m×200 m 的情况下, 当  $N=240$  时, VF-CS 算法的最终覆盖率比 BCBS 提高 2.99%, 比 CBS 提高 4.3%, 比 CIVDA 提高 13.56%, 平均提高约 6.95%; 在 350 m×350 m 的情况下, 当  $N=1\ 000$  时,



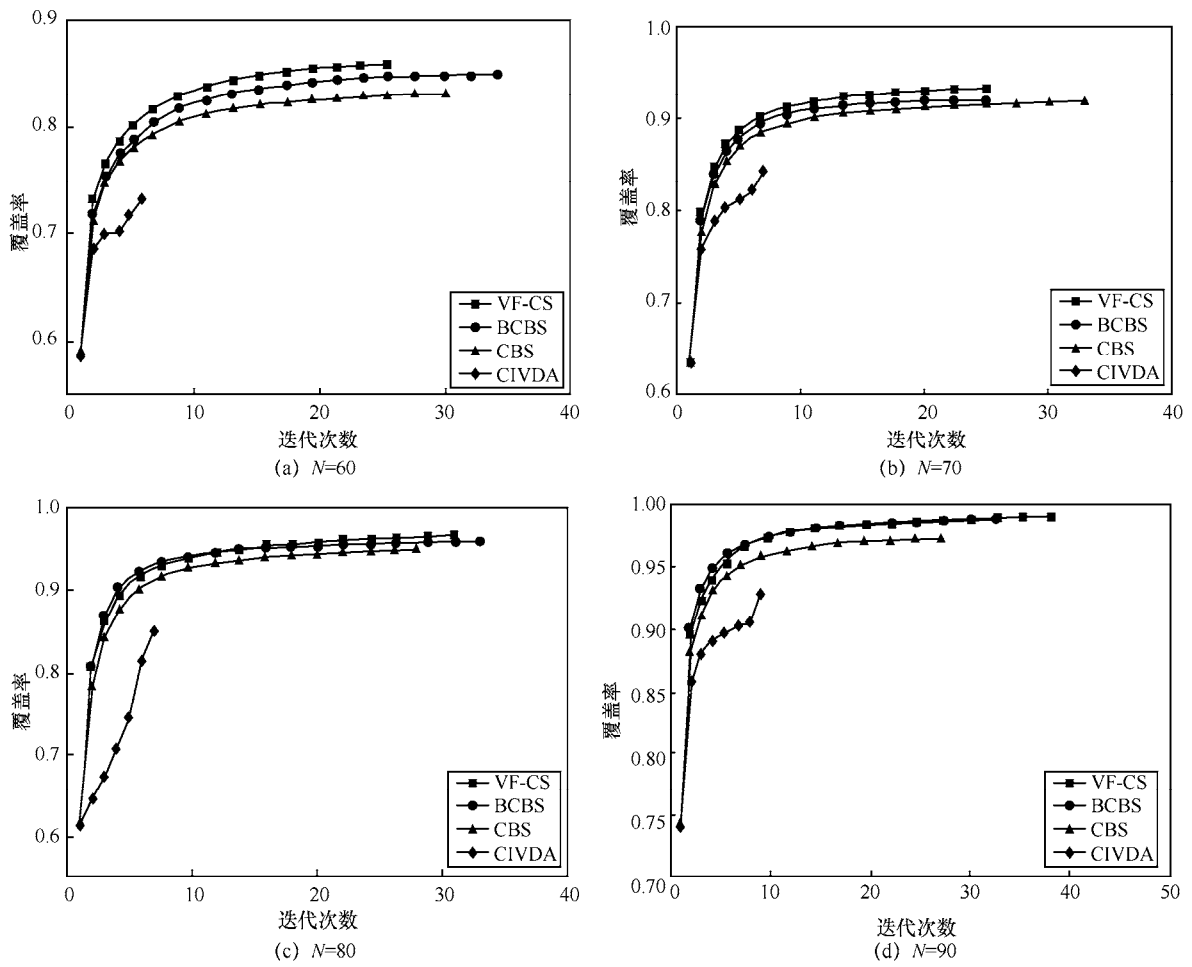


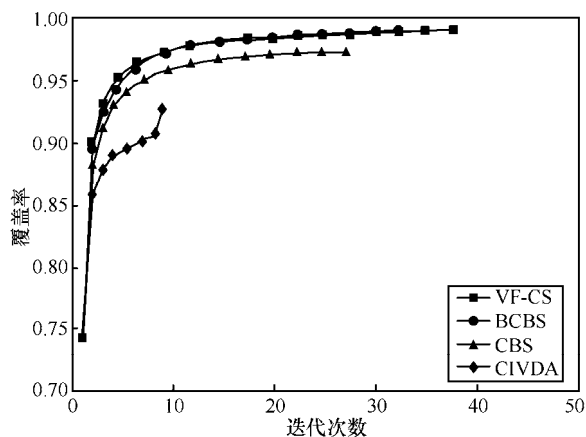
图 8 不同规模的传感器网络覆盖率随着迭代次数的变化趋势

表 2 不同传感器节点数量下的区域最终覆盖率

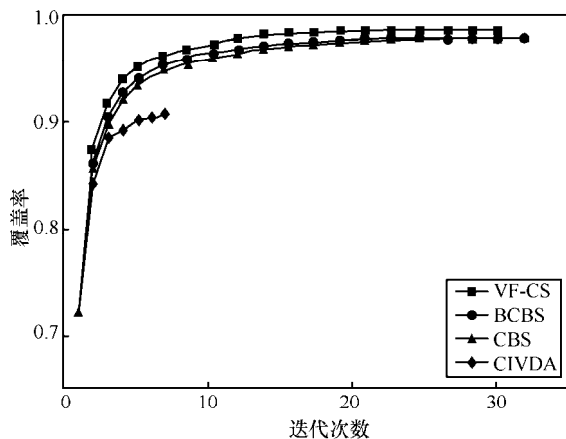
区域大小	节点个数 (N)	VF-CS 算法覆盖率	BCBS 算法覆盖率	CBS 算法覆盖率	CIVDA 算法覆盖率
100 m×100 m	90	0.989 9	0.989 8	0.972 0	0.928 3
	80	0.965 5	0.957 0	0.947 6	0.872 6
	70	0.933 9	0.919 3	0.920 6	0.844 0
	60	0.858 6	0.849 5	0.830 9	0.783 8
	50	0.809 0	0.773 2	0.754 4	0.698 6
	40	0.702 0	0.700 3	0.683 2	0.612 7
200 m×200 m	340	0.987 2	0.978 6	0.978 8	0.907 7
	320	0.965 1	0.955 4	0.952 3	0.884 5
	300	0.959 7	0.942 9	0.943 3	0.845 6
	280	0.939 0	0.922 1	0.917 2	0.808 0
	260	0.904 3	0.891 3	0.883 7	0.799 3
	240	0.882 1	0.856 5	0.845 9	0.776 8
350 m×350 m	1 000	0.983 6	0.974 6	0.968 7	0.879 9
	900	0.969 9	0.949 2	0.932 3	0.833 2
	800	0.931 2	0.905 1	0.894 4	0.775 9
	700	0.853 7	0.835 1	0.827 4	0.646 3
	600	0.776 8	0.746 4	0.714 4	0.597 6
	500	0.659 3	0.628 7	0.606 5	0.553 6

VF-CS 算法的最终覆盖率比 BCBS 提高 0.92%，比 CBS 提高 1.54%，比 CIVDA 提高 11.79%。

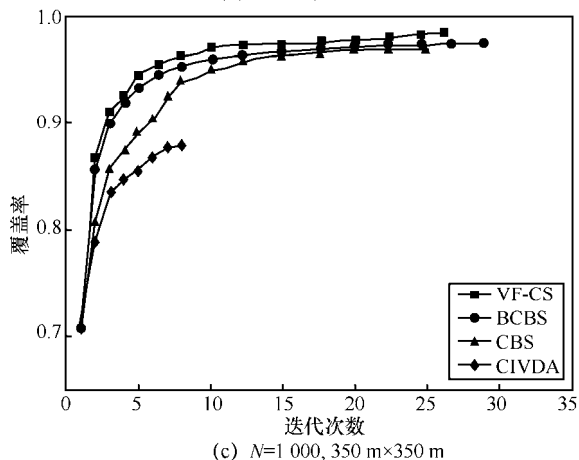
图 9 为实验 2 的覆盖率比较结果。从图 9 可看出，对于 3 种部署环境，VF-CS 算法的覆盖率都比其他算法高。例如，监测区域大小为  $200\text{ m} \times 200\text{ m}$ ， $N$  为 340 个，VF-CS 算法最终覆盖率达 98.72%，BCBS 算法、CBS 算法以及 CIVDA 算法最终覆盖率分别为 97.86%、97.88% 和 90.77%。



(a)  $N=90, 100\text{ m} \times 100\text{ m}$



(b)  $N=340, 200\text{ m} \times 200\text{ m}$

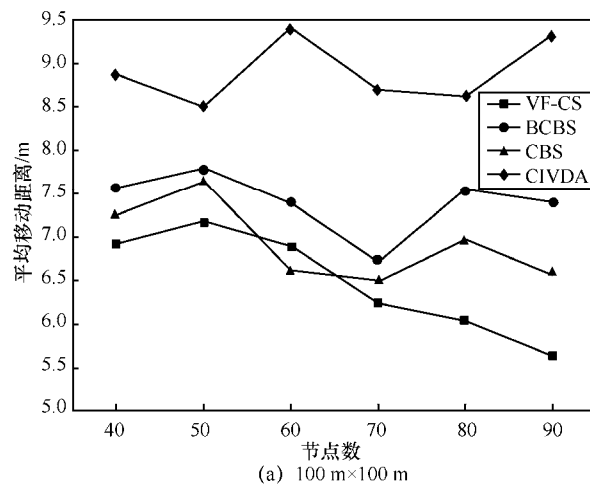


(c)  $N=1000, 350\text{ m} \times 350\text{ m}$

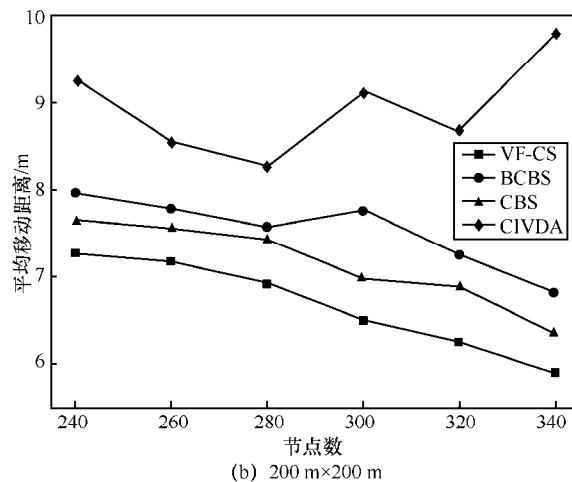
图 9 不同规模的传感器网络覆盖率随着迭代次数的变化趋势

### 4.3 节点平均移动距离

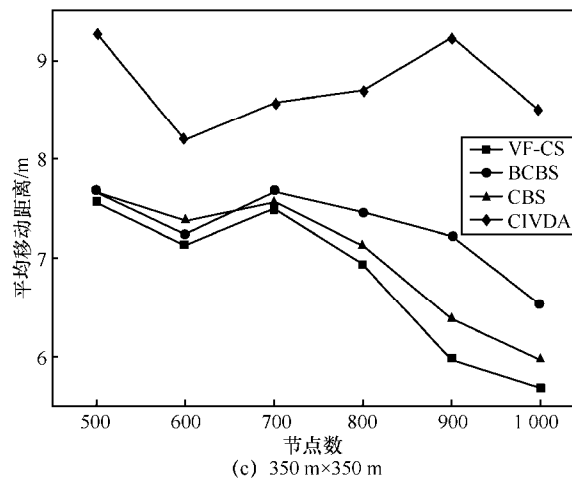
在实验 1 和实验 2 的基础上，对传感器节点的移动距离进行了记录，取平均移动距离作为传感器节点能量消耗的衡量指标，即平均移动距离越小，能耗越低。图 10 和图 11 分别给出了实验 1 和实验 2 的节点平均移动距离的比较结果。



(a)  $100\text{ m} \times 100\text{ m}$



(b)  $200\text{ m} \times 200\text{ m}$



(c)  $350\text{ m} \times 350\text{ m}$

图 10 实验 1 的平均移动距离对比

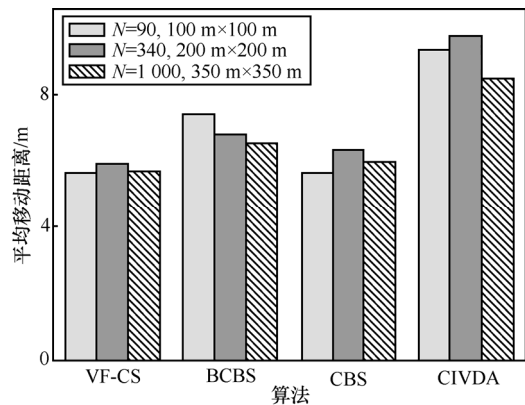


图 11 实验 2 的平均移动距离对比

由图 10 可知,3 种部署环境下,VF-CS 算法的平均移动距离均比其他算法小,例如图 10(a),部署区域大小为  $100\text{ m}\times 100\text{ m}$ ,当  $N=60$  时,VF-CS 算法平均移动距离比 CBS 算法略大,比其他 3 种算法小,但当  $N=40、50、70、80、90$  时,VF-CS 算法的平均移动距离都明显小于其他 3 种算法,其中,CIVDA 的平均移动距离最大,而且随着传感器节点数量的增加,VF-CS 的平均移动距离随之减小,比其他 3 种算法稳定。由于 BCBS 算法在覆盖率方面的表现与本文提出的 VF-CS 算法相差不多,将 VF-CS 算法的平均移动距离与 BCBS 算法进行单独比较可以发现,VF-CS 算法的平均移动距离比 BCBS 算法小很多,如图 10(b)所示,部署区域大小为  $200\text{ m}\times 200\text{ m}$ ,当  $N=300$  时,VF-CS 的平均移动

距离比 BCBS 的平均移动距离小近  $1.5\text{ m}$ 。由于 CBS 和 BCBS 算法都是在几何计算的基础上进行覆盖优化,节点移动范围虽也是局限于泰森多边形内,但在寻找移动位置的过程中,其优化进程不如 VF-CS 算法快,所以平均移动距离都高于 VF-CS 算法,而 CIVDA 中节点的移动范围并没有受到限制,导致其平均移动距离远大于其他 3 个算法。由图 11 可知,在  $100\text{ m}\times 100\text{ m}$ 、90 个节点, $200\text{ m}\times 200\text{ m}$ 、340 个节点, $350\text{ m}\times 350\text{ m}$ 、1 000 个节点这 3 种情形下,VF-CS 算法的节点平均移动距离都比其他算法小。

4.4 算法耗时

表 3 给出了 4 种覆盖优化算法在 3 种部署环境下不同节点规模中的计算时间。由表 3 可知,VF-CS 算法平均耗时比 BCBS 小很多,例如,部署区域为  $350\text{ m}\times 350\text{ m}$ ,当  $N=1\,000$  时,VF-CS 算法耗时  $4\,228.239\text{ s}$ ,BCBS 算法耗时  $11\,507.883\text{ s}$ ,CBS 算法耗时  $6\,826.56\text{ s}$ ,BCBS 算法耗时是 VF-CS 算法耗时的 2.72 倍,CBS 算法耗时是 VF-CS 算法耗时的 1.61 倍。由此可见,VF-CS 算法大大提高了全局收敛速度,尽管 VF-CS 算法比 CIVDA 算法耗时略多一些,但从 4.2 节的实验分析可知,VF-CS 算法的覆盖率比 CIVDA 要高很多。CIVDA 算法耗时最少,主要是由于该算法的全局优化能力弱,出现“早熟”现象所致,但其覆盖率是最低的。而 BCBS 算法由于其在分析盲区构造与多边形盲区相近的

表 3 实验 1 算法耗时

区域大小	节点个数 (N)	算法			
		VF-CS 耗时/s	BCBS 耗时/s	CBS 耗时/s	CIVDA 耗时/s
100 m×100 m	90	59.046 2	251.723 7	52.758 0	16.911 0
	80	57.353 8	178.180 0	47.789 2	14.983 1
	70	39.680 0	118.417 4	50.691 6	15.829 2
	60	44.431 9	131.474 0	39.778 4	14.601 2
	50	43.961 0	124.783 5	36.496 7	12.962 3
	40	33.454 0	108.393 2	33.846 3	11.335 9
200 m×200 m	340	742.342 0	1 936.584 3	1 256.473 3	287.458 1
	320	704.561 0	1 843.874 0	1 189.490 0	260.569 0
	300	650.460 0	1 729.332 0	1 054.730 0	252.650 0
	280	639.212 0	1 696.370 0	998.543 0	238.791 0
	260	611.851 0	1 509.321 0	943.560 0	227.650 0
	240	598.443 0	1 393.428 0	885.430 0	216.314 0
350 m×350 m	1 000	4 228.239 0	11 507.883 0	6 826.560 0	628.381 0
	900	3 832.467 0	10 458.660 0	6 243.489 0	537.246 0
	800	2 548.260 0	6 644.470 0	4 573.630 0	483.620 0
	700	2 003.724 0	5 075.567 0	3 229.692 0	448.493 0
	600	1 337.370 0	3 911.877 0	2 866.482 0	403.720 0
	500	1 072.951 0	3 407.689 0	2 247.266 0	377.633 0

表 4 节点覆盖效率对比

算法	覆盖效率 (100 m×100 m)			覆盖效率 (200 m×200 m)			覆盖效率 (350 m×350 m)		
	N=90	N=70	N=50	N=340	N=300	N=260	N=1 000	N=800	N=600
VF-CS	0.678 1	0.789 2	0.800 2	0.667 8	0.797 7	0.849 6	0.703 4	0.776 5	0.825 3
BCBS	0.659 9	0.748 7	0.799 8	0.655 5	0.786 1	0.832 1	0.684 5	0.749 3	0.812 6

多边形时, 计算复杂度高, 导致算法耗时偏长。

#### 4.5 节点覆盖效率

为了检验算法在不同环境的适应能力, 实验测试了算法的覆盖效率, 以检验算法在不同的网络节点分布密度情况下的性能。表 4 给出了 VF-CS 和 BCBS 这 2 种覆盖优化算法在 3 种部署环境下不同节点规模中的覆盖效率。

由表 4 可知, 在 3 种环境下 VF-CS 算法的覆盖效率  $C_e$  均大于 BCBS 算法, 证明 VF-CS 算法的节点冗余度低于 BCBS 算法, 网络中节点的分布更加均匀。例如, 监测区域大小为 350 m×350 m、N=1 000 时, VF-CS 算法最终的覆盖效率为 0.703 4, 比 BCBS 算法大 2.69%。在每种部署环境下, 随着节点数量的增加, 覆盖效率降低, 表明节点冗余程度增大。例如, 监测区域大小为 350 m×350 m, 当 N 由 600 增加到 1 000 时, VF-CS 算法的覆盖效率由 0.825 3 降低到 0.703 4, 降低了 14.8%; 而 BCBS 算法覆盖效率由 0.812 6 降低到 0.684 5, 降低了 15.8%。由此说明, 在部署环境节点数量变化的情况下, VF-CS 算法相比于 BCBS 算法, 覆盖效率能保持更高的水平。

## 5 结束语

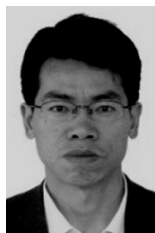
本文针对移动传感器网络提出了一种基于 Voronoi 图和改进布谷鸟搜索的覆盖优化算法。该算法在传统的 Voronoi 图划分的基础上, 对传感器节点所在的泰森多边形的顶点以及邻居节点进行虚拟力分析, 将传感器节点受到总作用力作为该节点进行布谷鸟搜索位置更新时的扰动因子, 从而加快优化进程; 在布谷鸟搜索最佳移动位置过程中, 结合 Voronoi 多边形形心在覆盖优化中的有效作用, 将每个节点所在的泰森多边形形心位置考虑进初始化位置集合中, 提高了整体覆盖率。实验结果表明, 相比于其他 3 种基于 Voronoi 图的节点部署算法, 本文算法提高了网络覆盖率, 减少了平均移动距离。

#### 参考文献:

- [1] ISBITIREN G, AKAN O B. Three-dimensional underwater target tracking with acoustic sensor networks[J]. IEEE Transactions on Vehicular Technology, 2011, 60(8):3897-3906.
- [2] SHAIMAA M, MOHAMED, HAITHAM S, et al. Coverage in mobile wireless sensor networks (M-WSN): a survey[J]. Computer Communications, 2017, 1(66):133-150.
- [3] ZHU C, SHU L, HARA T, et al. Research issues on mobile sensor networks[C]//International ICST Conference on Communications and NETWORKING. 2010:1-6.
- [4] MUNIR S A, REN B, JIAO W, et al. Mobile wireless sensor network: architecture and enabling technologies for ubiquitous computing[C]//International Conference on Advanced Information NETWORKING and Applications Workshops. 2007:113-120.
- [5] 刘惠, 柴志杰, 杜军朝, 等. 基于组合虚拟力的传感器网络三维空间重部署算法研究[J]. 自动化学报, 2011, 37(6):713-723.
- [6] LIU H, CHAI Z J, DU J C, et al. Sensor redeployment algorithm based on combined virtual forces in three dimensional space[J]. Acta Automatica Sinica, 2011, 37(6):713-723.
- [7] 石为人, 袁久银, 雷璐宁. 无线传感器网络覆盖控制算法研究[J]. 自动化学报, 2009, 35(5):540-545.
- [8] SHI W R, YUAN J Y, LEI L N. Research on wireless sensor network coverage control algorithm[J]. Acta Automatica Sinica, 2009, 35(5):540-545.
- [9] AHMAD P A, MAHMOUD M, OMAR M H. Virtual force algorithm and cuckoo search algorithm for node placement technique in wireless sensor network[C]//The 4th International Conference on Computing and Informatics. 2013:28-30.
- [10] JIN L, CHANG G, JIA J. Mobile sensor networks node distribution optimization based on minimum redundant coverage[C]//Chinese Control Conference. 2010:4851-4856.
- [11] 丁旭, 吴晓蓓, 黄成. 基于改进粒子群算法和特征点集的无线传感器网络覆盖问题研究[J]. 电子学报, 2016, 44(4):967-973.
- [12] DING X, WU X B, HUANG C. Area coverage problem based on improved PSO algorithm and feature point set in wireless sensor networks[J]. Acta Electronica Sinica, 2016, 44(4):967-973.
- [13] 李劲, 岳昆, 刘惟一. 基于融合的无线传感器网络 k-集覆盖的分布式算法[J]. 电子学报, 2013, 41(4):659-665.
- [14] LI J, YUE K, LIU W Y. Distributed set k-cover algorithms for fusion-based coverage in wireless sensor networks[J]. Acta Electronica Sinica, 2013, 41(4):659-665.
- [15] 庄曜铭, 吴成东, 张云洲, 等. 无线传感器网络中复合事件栅栏覆盖问题[J]. 通信学报, 2017, 38(6):75-84.
- [16] ZHUANG Z M, WU C D, ZHANG Y Z, et al. Compound event barrier coverage in wireless sensor network[J]. Journal on Communica-

- tions, 2017, 38(6):75-84.
- [12] ADULYASAS A, SUN Z, WANG N. Connected coverage optimization for sensor scheduling in wireless sensor networks[J]. IEEE Sensors Journal, 2015, 15(7):3877-3892.
- [13] ALDURAIBI F, LASLA N, YOUNIS M. Coverage-based node placement optimization in wireless sensor network with linear topology[C]//IEEE International Conference on Communications. 2016: 107-124.
- [14] XIA J. Coverage optimization strategy of wireless sensor network based on swarm intelligence algorithm[C]//International Conference on Smart City and Systems Engineering. 2017:179-182.
- [15] DAOUDI A, DETIENNE B, AZOUZI R E, et al. Robust coverage optimization approach in wireless sensor networks[C]//International Conference on Wireless Networks and Mobile Communications. 2017:1-7.
- [16] SHEN Z, CHANG Y, JIANG H, et al. A generic framework for optimal mobile sensor redeployment[J]. IEEE Transactions on Vehicular Technology, 2010, 59(8):4043-4057.
- [17] 涂志亮, 王强, 沈毅. 移动传感器网络中目标跟踪与监测的同步优化[J]. 自动化学报, 2012, 38(3):452-461.
- TU Z L, WANG Q, SHEN Y. A distributed simultaneous optimization algorithm for tracking and monitoring of moving target in mobile sensor networks[J]. Acta Automatica Sinica, 2012, 38(3):452-461.
- [18] BOUKERCHE A, XIN F. A voronoi approach for coverage protocols in wireless sensor networks[C]//Global Telecommunications Conference. 2007:5190-5194.
- [19] LEE H J, KIM Y H, HAN Y H, et al. Centroid-based movement assisted sensor deployment schemes in wireless sensor networks[C]//Vehicular Technology Conference Fall. 2009:1-5.
- [20] 方伟, 宋鑫宏. 基于 Voronoi 图盲区的无线传感器网络覆盖控制部署策略[J]. 物理学报, 2014, 63(22):128-137.
- FANG W, SONG X H. A deployment strategy for coverage control in wireless sensor networks based on the blind-zone of voronoi diagram[J]. Acta Physica Sinica, 2014, 63(22):128-137.
- [21] ABO-ZAHHAD M, SABOR N, SASAKI S, et al. A centralized immune-Voronoi deployment algorithm for coverage maximization and energy conservation in mobile wireless sensor networks[J]. Information Fusion, 2016, 30(C):36-51.
- [22] 周彤, 洪炳镨, 朴松昊. 基于虚拟力的混合感知网节点部署[J]. 计算机研究与发展, 2007, 44(6): 965-972.
- ZHOU T, HONG B R, PU S H. Hybrid sensor networks deployment based on virtual force[J]. Journal of Computer Research and Development, 2007, 44(6):965-972.
- [23] MAHBOUBI H, AGHDAM A G. Distributed deployment algorithms for coverage improvement in a network of wireless mobile sensors: relocation by virtual force[J]. IEEE Transactions on Control of Network Systems, 2016, PP (99):1-14.
- [24] MAHBOUBI H, AGHDAM A G. An energy-efficient strategy to improve coverage in a network of wireless mobile sensors with non-identical sensing ranges[J]. Vehicular Technology Conference (VTC Spring), 2013, 14(2382):1-5.
- [25] DEB S, YANG X S. Cuckoo search via levy flights[C]//World Congress on Nature & Biologically Inspired Computing. 2009: 210-214.
- [26] LIU C, CHUNMING Y E. Cuckoo search algorithm for the problem of permutation flow shop scheduling[J]. Journal of University of Shanghai for Science & Technology, 2013, 35(1):17-20.
- [27] YANG X S, DEB S. Multiobjective cuckoo search for design optimization[J]. Computers & Operations Research, 2013, 40(6): 1616-1624.
- [28] AMMARI H M, DAS S K. Coverage, connectivity, and fault tolerance measures of wireless sensor networks[M]. Stabilization, Safety, and Security of Distributed Systems. Heidelberg: Springer, 2006:35-49.

#### [作者简介]



李光辉 (1970-), 男, 湖南郴州人, 博士, 江南大学教授、博士生导师, 主要研究方向为无线传感器网络、容错计算、无损检测技术。



胡世红 (1993-), 女, 江苏连云港人, 江南大学硕士生, 主要研究方向为无线传感器网络覆盖优化。