文章编号：2095-2163(2023)03-0148-10 中图分类号：TP274.2 文献标志码：A

一种基于 WSN 部署优化和 CPCRLB 的目标跟踪算法

高华金，江潇潇，王永琦
（上海工程技术大学 电子电气工程学院，上海 201620）

摘 要：针对传统的静态无线传感器网络(Wireless Sensor Network,WSN)中存在的覆盖率低、节点能量有限等问题,本文首先提出了一种基于麻雀搜索算法(Sparrow Search Algorithm,SSA)的节点优化部署方案,提高了整个 WSN 的覆盖率,并降低了网络通信能耗;其次,在节点优化部署的基础上,提出了一种基于条件后验克拉美-罗下界(Conditional Posterior Cramer-Rao Lower Bounds,CPCRLB)与能耗优化的目标跟踪算法,并设计了一种能同时兼顾跟踪精度与能耗的目标函数。仿真结果表明,本文所提算法能够有效提高网络覆盖率,并能在保证跟踪精度的条件下,最大限度地降低网络能耗。

关键词：WSN；麻雀搜索算法；覆盖率；能耗；节点优化部署；CPCRLB

A target tracking algorithm based on WSN deployment optimization and CPCRLB

GAO Huajin, JIANG Xiaoxiao, WANG Yongqi
(School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

【Abstract】 Aiming at the problems of low coverage and limited node energy in traditional static wireless sensor networks (Wireless Sensor Network, WSN), this paper first proposes a node optimization deployment scheme based on Sparrow Search Algorithm (SSA). The method improves the coverage of the entire WSN and reduces the energy consumption of network communication. Secondly, based on the optimized deployment of nodes, a Conditional Posterior Cramer-Rao Lower Bounds (CPCRLB) is proposed. The target tracking algorithm is optimized with energy consumption, and an objective function that can balance tracking accuracy and energy consumption is designed. The simulation results show that the algorithm proposed in this paper can effectively improve the network coverage, and can reduce the network energy consumption to the greatest extent under the condition of ensuring the tracking accuracy.

【Key words】 WSN; Sparrow Search Algorithm; coverage rate; energy consumption; node optimization deployment; CPCRLB

0 引 言

近年来,由于传感器节点具备感知、计算和通信等能力,WSN 的功能也越来越多样化,已经广泛地应用在各个领域中,如智能交通系统[1]、灾害预防检测[2]、生态环境监管[3]等场景中,不仅可以节约人力成本,还可以更高效地完成任务。目标跟踪是 WSN 最典型应用领域之一,而 WSN 中传感器节点的位置分布对目标跟踪的性能会产生至关重要的影响。研究可知,当采用传统的静态传感器网络结构进行目标跟踪时,其节点无法移动会带来一系列网络能耗问题,如网络能量分布不均、节点通信路径过长、能量空洞导致目标丢失等。因此,对 WSN 进行优化部署,可以提高网络利用率,延长网络寿命,从而更高效地执行目标跟踪任务。

目前,对传感器网络的优化部署主要考虑以下 2 方面:

(1)覆盖率。高覆盖率可以保证网络能够覆盖大部分监测区域,从而大大降低丢失跟踪目标的概率。

(2)网络通信能耗。能耗是传感器网络不可忽视的一个重要因素,将直接影响网络寿命。能耗过高会加速传感器节点能量消耗,并导致部分节点能量耗尽,继而不能正常工作,从而无法完成跟踪任务。

近年来,关于 WSN 节点部署问题的研究也取得

作者简介：高华金(1994-),男,硕士研究生,主要研究方向:目标定位与跟踪；江潇潇(1985-),女,博士,讲师,主要研究方向:目标定位与跟踪。
通讯作者：江潇潇 Email：jxx_simit@163.com

收稿日期：2022-05-05

了一定的成果。文献[4]中提到的虚拟力技术使每个节点受到 3 种力的影响，能够有效提高网络的部署效果。文献[5]中提出了一种基于范德华力的移动无线传感器网络传感器部署算法，将摩擦力引入到力方程中，与其他虚拟力相比，该算法具有更高的覆盖率、可接受的收敛时间和更均匀的配置，但是并没有考虑网络的连通性和能耗。文献[6]提出了一种基于虚拟平方网格进行节点部署的算法，该算法能通过减少节点数量缩短运行时间来实现更好的性能，并且能够保证整个网络的覆盖和连通性，但是该算法没有考虑节点可移动的情况。We 等学者[7]提出的将虚拟力与跟踪算法相结合，通过建立传感器节点与障碍物、目标和其他传感器节点间的虚拟力模型，根据受力平衡确定各节点位置，但最终的优化结果受网络中静态节点的影响，无法实现全局最优。

基于此，本文提出了一种新的基于 WSN 优化部署和条件后验克拉美罗下界(CPCRLB)的目标跟踪算法。该算法考虑将 WSN 建模为包含静态和动态节点的混合无线传感器网络，根据节点覆盖模型、信息传输能耗模型和最短路径算法建立目标函数，采用麻雀搜索算法对目标函数进行寻优计算，实现混合 WSN 的优化部署，在网络覆盖率和通信能耗上进行性能提升。接着，在对网络优化部署的基础上，设计了基于 CPCRLB 和能耗均衡的目标跟踪算法，通过二进制麻雀搜索算法完成了传感器节点选择。仿真结果表明，本文所提算法能够实现在跟踪精度和能耗之间的合理均衡，并优于其他算法。

本文结构如下：第一节详细介绍了本文算法所用的网络模型，包括节点覆盖模型、能耗模型和 Dijkstra 最短路径算法，第二节描述了基于麻雀搜索算法的网络部署优化过程，第三节在对网络优化部署的基础上，设计了基于 CPCRLB 和能耗均衡的目标跟踪算法，通过二进制麻雀搜索算法完成了传感器节点选择，并给出仿真结果与分析。

1　网络模型

1.1　节点覆盖模型

传感器节点对目标的检测能力随着其与目标距离的大小而变化[7]。假定网络中一共有 N 个传感器节点，所有传感器节点都具有相同的感知半径 r_s 和不确定的感知范围 $\delta \times r_s(\delta < 1)$，待检测目标 $T(x_t, y_t)$ 与某一传感器 $S_j(x_{sj}, y_{sj})$ 间的距离为 $d(S_j, T) = \sqrt{(x_t - x_{sj})^2 + (y_t - y_{sj})^2}$。其中，$j = 1, 2, \cdots, N$，表示传感器的标号。

由于实际工作环境存在噪声干扰，传感器节点对于目标的感知概率会根据距离的变化呈现一定的概率分布[8]，在该概率分布模型中，传感器节点 S_j 对目标 T 的感知概率表示为如下形式：

$$P_{Rj} = \begin{cases} 0 & d(S_j, T)/r_s - 1 \geqslant \delta \\ \lambda_2 \exp\left(-\dfrac{\lambda_1 \alpha_1^{\beta_1}}{\alpha_2^{\beta_2}}\right) & -\delta < \dfrac{d(S_j, T)}{r_s} - 1 < \delta \\ \lambda_2 & d(S_j, T)/r_s - 1 \leqslant -\delta \end{cases} \tag{1}$$

其中，λ_1、λ_2、β_1、β_2 是与传感器节点自身特性有关的参数，α_1、α_2 是与 r_s 和 $d(S_j, T)$ 有关的变量，可由如下公式进行描述：

$$\begin{aligned} \alpha_1 &= r_s \times (\delta - 1) + d(S_j, T) \\ \alpha_2 &= r_s \times (\delta + 1) - d(S_j, T) \end{aligned} \tag{2}$$

由此可知，传感器节点对目标的感知概率严重依赖于其到目标的距离，总体上讲，距离越大，感知概率越小，反之，感知概率越大。

通常情况下，单个传感器节点对目标的感知概率小于 1，因此，在监测过程中可以使用多个传感器节点来协同作业，以提高对目标的感知概率。假设在传感范围内，有 S 个节点同时对目标进行监测，则网络对该目标的检测概率为：

$$P_R = 1 - \prod_{j=1}^{S} (1 - P_{Rj}) \tag{3}$$

其中，S 为参与监测目标的节点个数，P_{Rj} 表示第 j 个节点对该目标的感知概率。由式(3)可得，增加传感器节点的个数，网络对目标的检测概率也会随之增大。

定义 $P_{R_{th}}$ 为目标感知概率的阈值，则目标能够被有效感知到的条件为：

$$P_R \geqslant P_{R_{th}} \tag{4}$$

满足上式的所有区域均为有效覆盖区域。

1.2　信息传输能耗模型

本文采用的能耗模型是基于传感器、微处理器和收发器等不同功能模块的功率和激活时间而设计的[9]，因此，任务节点(激活状态)消耗能量主要有 3 个方面，即感知目标、数据处理和数据通信[10]。其中，数据通信所消耗的能量最高，通常是由通信双方的位置决定。假设传感器节点在休眠状态下不消耗能量，而激活状态下传感器节点发送 r bit 数据到距离 d 所耗的能量[11]为：

$$e_t(r, d) = \begin{cases} r(e_{elec} + \varepsilon_a d^2) & d \leqslant r_{t\max} \\ +\infty & d > r_{t\max} \end{cases} \tag{5}$$

其中，e_{elec} 为电路损耗，为固定参数；ε_a 为功率放大能耗，为固定参数；r_{tmax} 为传感器节点的最大通信距离。相应地，传感器节点接收 r bit 数据所消耗的能量为：

$$e_r(r) = re_{elec} \tag{6}$$

文中，采用如下参数设置：e_{elec} = 50 nJ/bit，ε_a = 100 pJ/m^2/bit，r = 264 bit。

1.3 Dijkstra 最短路径算法

本文的网络由随机部署在 4 000 m×4 000 m 区域内的传感器节点构成。研究中，利用 K-means 聚类算法对网络中随机分布的节点进行聚类分簇，这里将整个网络划分为 4 个簇，并在每个簇的质心位置放置一个具有移动能力的动态传感器作为簇头节点，如图 1 所示，负责接收节点的量测数据并将数据发送到基站。

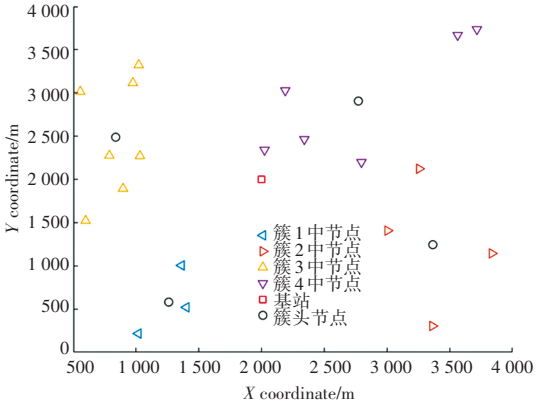


图 1 网络结构示意图

Fig. 1 Network structure diagram

由上文 1.2 节中的信息传输能耗模型可知，随着距离的增加，节点间通信能耗会越来越大。因此，有必要规划簇头节点到基站间通信的最短路径，从而降低网络能耗，节省网络资源。本文使用 Dijkstra 算法[12]来获得通信最短路径。簇头节点到基站采用多跳信息传输的方式，即簇头节点把量测信息传输给通信范围内的另一个节点，稍后这个节点再把接收到的量测信息传输给下一个节点，直到最后将量测信息传送到基站，因此，规划的通信路径即簇头节点到基站的多跳传输路径。

Dijkstra 算法是图论中实现寻求最短路径的有效方法，该算法可以解决簇头节点到基站间通信的最短路径规划问题，即最低通信能耗路径。假设某起源点传感器 o 的坐标为 $S_o = (x_o, y_o)$，且每个传感器节点都有一组标号 $[d_j, p_j]$，其中 d_j 表示从起源点传感器 o 到传感器节点 j 的当前最短路径长度，p_j 则是从 o 到 j 的最短路径中 j 的前一点，则从 i 到 j 的最短路径算法[13]基本过程如下：

(1)初始化。设置起源点 $d_o = 0, p_o = \varnothing$；其他节点 $d_i = +\infty, p_i = [\ ]$，$i = 1:N$，其中 N 是除起源点外的节点总数；并将起源点设置为已标记点 k，其他所有点设为未标记点。

(2)计算所有已标记点 k 到其直接连接且未标记的点 j 的距离，并记录 $d_j = \min[d_j, d_k + l_{kj}]$，其中 l_{kj} 是从点 k 到点 j 的直接连接距离。

(3)选取下一个点。令 $d_i = \min[d_j]$，则点 i 就被选为最短路径中的一点，并设为已标记点。

(4)找到点 i 的前一点。从已标记的点中找到直接连接到点 i 的点 j^*，作为前一点，并记录 $i = j^*$。

(5)标记点 i。如果所有点均已标记，则算法完成；否则，记录 $k = i$，转步骤(2)。

2 基于麻雀搜索算法的网络部署优化

2.1 麻雀搜索算法

麻雀搜索算法(Sparrow Search Algorithm，SSA)算法[14]是由 Xue Jiankai 和 Shen Bo 在 2020 年受麻雀的觅食行为和反捕食行为的启发而提出的一种群体智能优化算法。SSA 将整个种群划分为发现者、加入者和警戒者，警戒者根据安全阈值控制种群的搜寻方向，而发现者和加入者则会根据全局最优位置和全局最差位置来更新自己的位置。该算法具有寻优能力强、收敛速度快的优点。本文将麻雀搜索算法用于传感器网络的部署优化计算，以达到提高网络覆盖率并减少通信能耗的目的。这里，首先给出麻雀搜索算法的具体步骤。

假设种群中的一个个体代表一种传感器位置分布方案，当传感器数量为 N，粒子 id 在第 t 代的状态表示为：

$$X_{id}(t) = [x'_{id,1}(t), y'_{id,1}(t), \cdots, x'_{id,N}(t), y'_{id,N}(t)] = \{x^l_{id}(t) \mid l = 1, 2, \cdots, 2N\} \tag{7}$$

其中，id 表示种群成员序号；$id = 1, 2, \cdots, n_p$，n_p 为个体总数；t 表示当前迭代次数，$t = 1, 2, \cdots, N_{iter}$；$N_{iter}$ 为迭代总数；$x'_{id,1}(t)$，$y'_{id,1}(t)$ 分别表示第一个传感器的 X 轴位置、Y 轴位置；$x^l_{id}(t)$ 表示群成员 id 在第 t 代的位置向量 $X_{id}(t)$ 的第 l 个分量。

整个种群被划分为发现者和加入者两部分，这 2 部分中包含一定数量的警戒者，设种群的预警值和安全值分别为 RT 和 ST，警戒者负责当预警值 RT 超过安全值 ST 时，向整个种群传递危险信号，以使种群向安全的方向移动。假设发现者的数量为 n_d，

加入者的数量为 $n_p - n_d$，设警戒者的数量为 n_s，则在每次迭代中，发现者的位置更新如下：

$$x_{id}^{l}(t+1)=\begin{cases}x_{id}^{l}(t)\times\exp\left(-\dfrac{id}{\alpha\times N_{iter}}\right) & RT<ST\\ x_{id}^{l}(t)+Q\times\boldsymbol{L} & RT\geqslant ST\end{cases}\tag{8}$$

其中，$l=1,2,\cdots,n_d$；$\alpha\in(0,1]$ 是一个随机数；$RT\in[0,1]$ 是一个随机数，每次迭代时其值都不同；$ST\in[0.5,1]$ 是人为设定的常数，其值在迭代过程中保持不变；Q 是一个服从正态分布的随机数；$\boldsymbol{L}$ 表示一个 $1\times d$ 的矩阵，该矩阵内每个元素均为 1。当 $RT<ST$ 时，意味着此时的觅食环境周围没有捕食者，发现者可以执行广泛的搜索操作；当 $RT\geqslant ST$ 时，表示种群中的警戒者麻雀已经发现了捕食者，并向种群中其他麻雀发出了警报，此时所有麻雀都需要迅速飞到其他安全的地方进行觅食。

在觅食过程中，一些加入者会时刻监视着发现者，一旦察觉到发现者已经找到了更好的食物，就会立即离开现在的位置去争夺食物。如果赢了，就可以立即获得该发现者的食物，否则需要继续如下位置更新操作：

$$x_{id}^{l}(t+1)=\begin{cases}Q\times\exp\left(\dfrac{x_{worst}-x_{id}^{l}(t)}{i^2}\right) & id>n_p/2\\ x_p(t+1)+\left|x_{id}^{l}(t)-x_p(t+1)\right|.\boldsymbol{A}^{+}.\boldsymbol{L} & id\leqslant n_p/2\end{cases}\tag{9}$$

其中，$l=n_d+1,\cdots,n_p$；$x_p(t+1)$ 是目前发现者所占据的最优位置；x_{worst} 表示当前全局最差位置；$\boldsymbol{A}$ 表示一个 $1\times d$ 的矩阵，矩阵中各元素随机赋值为 1 或 -1，并且 $\boldsymbol{A}^{+}=\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}$。当 $id>n_p/2$ 时，表明适应度值较低的当前序号值为 id 的加入者没有获得食物，需要飞往其他地方觅食。

在搜寻过程中，随机初始化一些警戒者麻雀，其位置按如下方式更新：

$$x_{id}^{l}(t+1)=\begin{cases}x_{best}^{t}+\xi\times\left|x_{id}^{l}(t)-x_{best}^{t}\right| & f_i>f_g\\ x_{id}^{l}(t)+K\times\left(\dfrac{\left|x_{id}^{l}(t)-x_{worst}^{t}\right|}{(f_i-f_w)+\varepsilon}\right) & f_i=f_g\end{cases}\tag{10}$$

其中，x_{best}^{t} 是当前的全局最优位置；ξ 作为步长控制参数，是服从均值为 0，方差为 1 的正态分布的随机数；$K\in[-1,1]$ 是一个随机数；f_i 是当前麻雀个体的适应度值；f_g 和 f_w 分别是当前全局最佳和最差的适应度值；ε 是常数，以避免分母出现 0。

麻雀搜索算法的流程如图 2 所示。

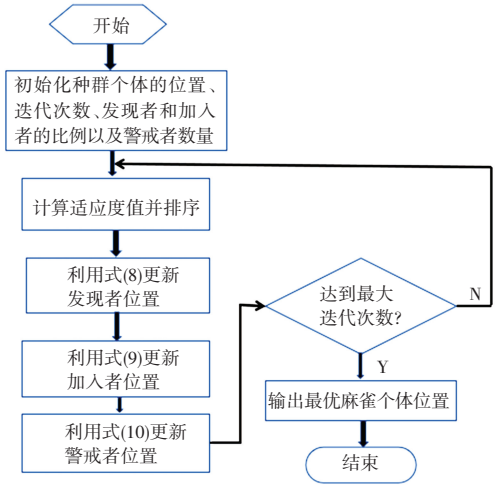


图 2　麻雀搜索算法流程图

Fig. 2　Flowchart of sparrow search algorithm

2.2　基于 SSA 的网络优化部署步骤

网络能耗和覆盖率是衡量传感器网络性能的 2 个关键指标，通过优化传感器网络的部署，能够有效提高网络覆盖率，缩短网络传输路径，节省网络能耗。同时，优化后的网络部署对目标跟踪性能的提高也有着重要的意义。这里，为了简化问题模型，本文做出如下假设：

(1)混合传感器网络由静态节点和动态节点共同构成。

(2)传感器节点随机分布在网络中。网络中的静态节点部署后静止不动，动态节点具有移动能力。

(3)传感器节点的覆盖模型采用前文 1.1 节所介绍的模型。

(4)传感器节点可以通过某种方式获取自身位置，基站可以获得所有节点的位置。

基于 SSA 的网络优化部署步骤如下：

步骤 1　系统初始化建模，首先随机分布 Ns 个静态节点。

步骤 2　采用 K-means 聚类算法对网络中随机分布的静态节点进行聚类，从而将整个网络划分为 c 个簇，并在每个簇聚类中心位置放置一个静态节点作为簇首。

步骤 3　在步骤 2 的基础上，再向网络中随机增加 (Nc) 个具有移动能力的动态节点。

步骤 4　按照公式(1)~(6)建立传感器节点的覆盖模型和信息传输能耗模型。

步骤 5　采用麻雀搜索算法对网络节点的部署进行优化：

(1)初始化种群。令每一个种群个体 $\{X_{id}(t)=$

$[x'_{id,1}(t), y'_{id,1}(t), \cdots, x'_{id,N}(t), y'_{id,N}(t)], id = 1:N\}$ 表示一种动态传感器的位置分布方案，并根据方案设置合适的发现者、加入者和警戒者的比例。

(2)计算适应度函数值。计算每个种群个体对应的适应度函数值，适应度函数按照式(11)来计算：

$$f_i = C_i / E_i \tag{11}$$

其中，C_i 表示第 i 个麻雀个体对应的传感器部署方案的网络覆盖率；E_i 表示通过 Dijkstra 算法获得的最优多跳通信路径的总能耗，即 4 个簇头节点通过多跳路径将数据传输到基站的能耗之和。

(3)更新麻雀个体状态。按照式(8)~(10)来依次更新发现者、加入者和警戒者的位置，并根据适应度值计算出全局最优位置和最差位置。

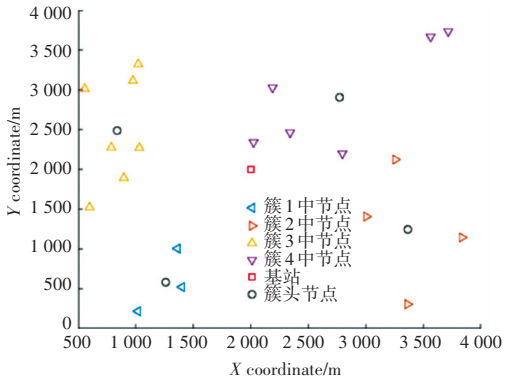
(4)判断迭代是否终止。判断当前迭代次数是否达到最大迭代次数，若达到，则输出最优麻雀个体对应的传感器位置分布，若未达到，则返回(2)。

2.3 仿真实验与分析

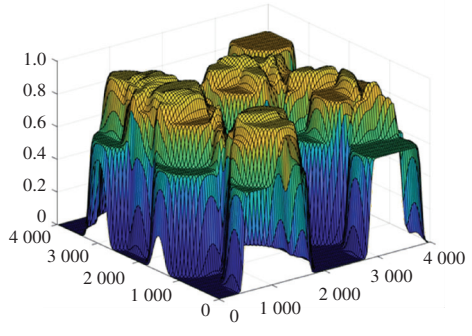
为了验证采用麻雀搜索算法对传感器的位置进行优化部署的性能，本文进行了如下实验：监视区域为[4 km×4 km]，随机分布着 24 个静态传感器节点和 14 个具有移动能力的动态传感器节点，并将其划分为 4 个簇。所有传感器节点的感知半径均为 1.2 km，不确定感知范围参数为 $\delta = 0.6$，覆盖率的计算方法为将整个网络按粒度为 40 m 划分为[100×100]个网格，计算各个网格中心点处感知概率大于 0.6 的网格占全部网格的比例、即覆盖率，节点覆盖模型的参数为 $\lambda_1 = 1, \lambda_2 = 0.6, \beta_1 = 3, \beta_2 = 2$，麻雀搜索算法的种群规模为 50，迭代次数为 100。

随机静态节点分布情况如图 3(a)所示。传感器被 K-means 聚类算法分为了 4 个簇，分别用 4 种颜色表示。图 3(b)表示随机静态节点的网络覆盖情况，水平坐标表示二维的地面传感器网络，垂直坐标表示整个网络对该点处的感知概率，可以看出，网络中很多地方感知概率小于 0.6，存在“真空”地带，容易导致最终无法有效检测目标的结果。

图 4(a)为加入了 14 个随机动态节点之后的网络节点分布，静态节点位置保持不变，图 4(b)表示此分布的网络覆盖情况。可以看出，网络中大部分区域对应的感知概率都超过了 0.6，只存在少部分感知概率低于 0.6 的区域，相对于随机静态节点的网络覆盖，加入了随机动态节点的网络的覆盖率有了明显的增加，网络中大部分区域都能保持相对较高的检测概率。



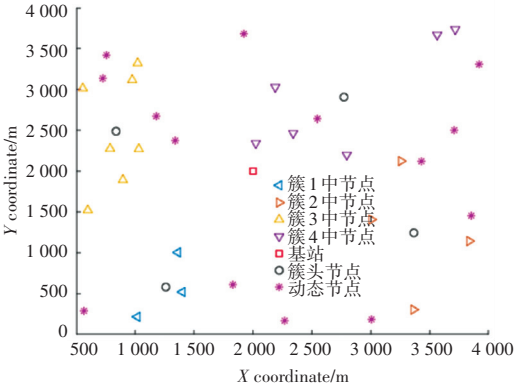
(a) 随机静态节点分布



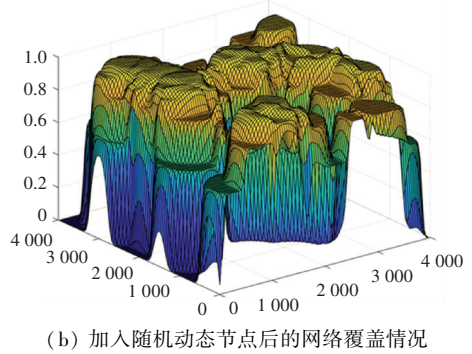
(b) 随机静态节点的网络覆盖情况

图 3 随机静态节点分布图及其网络覆盖情况

Fig. 3 Random static node distribution map and the corresponding network coverage



(a) 随机静态和动态节点分布

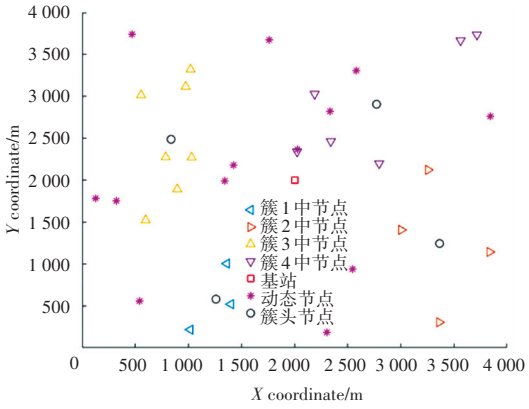


(b) 加入随机动态节点后的网络覆盖情况

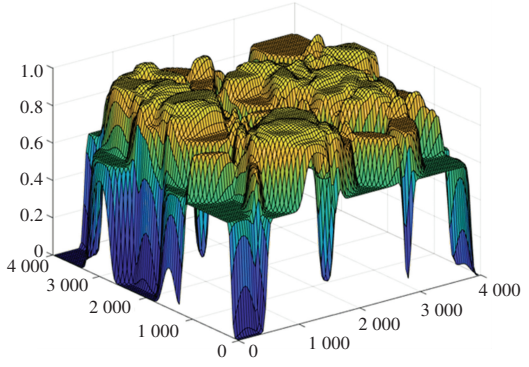
图 4 加入随机动态节点分布及其网络覆盖情况

Fig. 4 The distribution of adding random dynamic node and the corresponding network coverage

图 5(a)为采用 SSA 算法进行网络部署优化后的节点分布，可以看出，相比于随机静态和动态节点分布，优化后的节点分布明显更均匀了一些。图 5(b)为其网络覆盖情况，可以看出，网络覆盖率非常高，整个监测区域基本都在有效监测范围内。



(a) 优化后的节点分布



(b) 优化后的网络覆盖情况

图 5　优化后的节点分布及其网络覆盖情况

Fig. 5　The optimized node distribution and the corresponding network coverage

表 1 给出了传感器部署实验下覆盖率和通信能耗的具体数据，其中通信能耗计算的是各个簇的簇内节点将感知数据传送到簇头节点，并由簇头节点将数据打包发送到基站这一过程所消耗的能量。可以看出，相对其他 2 种情况，经过 SSA 优化部署后的节点分布的覆盖率有了显著提高，同时网络通信能耗也有了明显的降低，这为目标跟踪任务的实现提供了良好的基础。

表 1　传感器部署的覆盖率与通信能耗

Tab. 1　Sensor deployment coverage rate and communication energy consumption

	覆盖率/%	通信能耗/(10^8 nJ · bit^{-1})
随机静态节点	63.97	1.672 4
随机静态+动态节点	81.91	1.351 3
优化后的节点分布	90.14	1.085 8

3　网络优化部署下的机动目标跟踪

3.1　问题描述

3.1.1　系统模型

在目标跟踪中，对于在二维平面运动的目标，可以对其进行建模。考虑一个传感器节点数为 N 的网络。每个节点 S_j 的位置 (x_{sj},y_{sj}) 都是已知的，$j=1,2,\cdots,N$。k 时刻目标的状态向量表示为 $\boldsymbol{X}_k=[x_k,y_k,\dot{x}_k,\dot{y}_k]^{\mathrm{T}}$，$x_k,y_k$ 表示 k 时刻目标在 x 轴和 y 轴上的坐标位置，$\dot{x}_k$，$\dot{y}_k$ 表示 k 时刻目标在 x 轴和 y 轴上的速度。目标的状态方程和节点 j 对目标的量测方程如下：

$$\boldsymbol{X}_k=\boldsymbol{F}\boldsymbol{X}_{k-1}+\boldsymbol{W}_k \tag{12}$$

$$Z_k{}^j=\arctan((y_k-y_{sj})/(x_k-x_{sj}))+G_k{}^j \tag{13}$$

其中，$\boldsymbol{X}_k$ 为 k 时刻目标的状态；$\boldsymbol{F}$ 为状态转移矩阵；过程噪声 $\boldsymbol{W}_k$ 为零均值、协方差矩阵为 $\boldsymbol{Q}$ 的高斯白噪声；$Z_k{}^j$ 是 k 时刻第 j 个传感器节点对目标的量测值；$G_k{}^j$ 为量测方差为 σ^2 的量测噪声。

3.1.2　CPCRLB

目标跟踪是无线传感器网络的一个重要任务，但是由于能量的限制，并不是所有的传感器节点都能参与到目标跟踪。传感器管理就是选择一组最优的传感器节点参与目标跟踪，在获得较好的跟踪性能的同时，又能降低网络能量。

后验克拉美-罗下界(PCRLB)是目标状态估计的 Fisher 信息矩阵(FIM)的逆矩阵，给出了目标状态估计误差理论上的下界，但是 PCRLB 并没有利用实际的量测信息，完全由系统动态模型、量测模型和初始时刻状态的先验分布确定，因此对具体的目标航迹实现，PCRLB 不能完全反映目标跟踪的性能。文献[15]提出了条件 PCRLB(CPCRLB)。CPCRLB 以当前时刻的所有真实量测为条件，给出了目标状态估计的实际均方误差下界。由于 CPCRLB 包含了当前目标的真实航迹信息，因此适合用于传感器管理。

CPCRLB 是在已知过去所有 k 时刻的量测值 $Z_{1:k}$ 的条件下，得到在 $k+1$ 时刻一个新的量测 Z_{k+1} 时，$k+1$ 时刻所要估计的目标状态 X_{k+1} 的性能，即：

$$MSE(\hat{X}_{k+1}\mid Z_{1:k})=E\{[\hat{X}_{k+1}-X_{k+1}][\hat{X}_{k+1}-X_{k+1}]^{\mathrm{T}}\mid Z_{1:k}\}\geqslant \boldsymbol{L}^{-1}(X_{k+1}\mid Z_{1:k}) \tag{14}$$

这是 CPCRLB 的定义，即在上述情况下目标状态估计的均方误差下界。其中，$Z_{1:k}$ 是实际真实的

测量值,并不是一个随机向量, $\boldsymbol{L}(X_{k+1} \mid Z_{1:k})$ 表示的是目标状态估计值 $\hat{X}_{k+1}$ 的条件 Fisher 信息矩阵(Fisher Information Matrix, FIM)。文献[16]给出了条件 FIM 的一个迭代计算方法,但是该方法需要计算一个辅助 FIM,计算复杂度较高,为了简化计算,文献[17]提出了一种直接计算 CPCRLB 的方法:

$$\boldsymbol{L}(X_{k+1} \mid Z_{1:k}) \approx \boldsymbol{B}_k^{22} - \boldsymbol{B}_k^{21}[\boldsymbol{L}(X_{k+1} \mid Z_{1:k-1}) + \boldsymbol{B}_k^{11}]^{-1}\boldsymbol{B}_k^{12} \tag{15}$$

在确定的状态方程和量测方程的条件下,其 $\boldsymbol{B}_k^{11}$、$\boldsymbol{B}_k^{12}$、$\boldsymbol{B}_k^{22}$ 可由如下公式计算求得:

$$\boldsymbol{B}_k^{11} = \boldsymbol{F}^{\mathrm{T}}\boldsymbol{Q}^{-1}\boldsymbol{F} \tag{16}$$

$$\boldsymbol{B}_k^{12} = -\boldsymbol{F}^{\mathrm{T}}\boldsymbol{Q}^{-1} = (\boldsymbol{B}_k^{21})^{\mathrm{T}} \tag{17}$$

$$\boldsymbol{B}_k^{22} = \boldsymbol{Q}^{-1} + \boldsymbol{B}_k^{22,b} \tag{18}$$

对非线性量测模型, $\boldsymbol{B}_k^{22,b}$ 通常没有解析表达式。为计算 $\boldsymbol{B}_k^{22,b}$, 假设目标状态服从一阶 Markov 过程,采用重要性采样粒子滤波器进行滤波来近似目标状态估计,假设在 k 时刻有 Mp 个加权粒子 $\{X_k^{(l)}, \omega_k^{(l)}\}_{l=1}^{Mp}$, 并且在重采样步骤后,所有的粒子权重都为 $1/Mp$, 最后计算得出基于量测数据的条件 FIM 可以迭代逼近计算如下:

$$\boldsymbol{L}(X_{k+1} \mid Z_{1:k}) \approx [\boldsymbol{F}\boldsymbol{L}^{-1}(X_{k+1} \mid Z_{1:k-1})\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q}]^{-1} + \boldsymbol{B}_k^{22,b} \tag{19}$$

矩阵 $\boldsymbol{B}_k^{22,b}$ 的元素如下:

$$\boldsymbol{B}_k^{22,b}(1,1) = \frac{1}{S\sigma_v^2}\sum_{l=1}^{S}\sum_{j=1}^{N_s}\frac{(y_{k+1}-y_{sj})^2}{[(x_{k+1}-x_{sj})^2+(y_{k+1}-y_{sj})^2]^2}\Big|_{X_{k+1}=X_{k+1}^{(l)}} \tag{20}$$

$$\boldsymbol{B}_k^{22,b}(1,2) = \frac{1}{S\sigma_v^2}\sum_{l=1}^{S}\sum_{j=1}^{N_s}\frac{(y_{k+1}-y_{sj})(x_{k+1}-x_{sj})}{[(x_{k+1}-x_{sj})^2+(y_{k+1}-y_{sj})^2]^2}\Big|_{X_{k+1}=X_{k+1}^{(l)}} \tag{21}$$

$$\boldsymbol{B}_k^{22,b}(1,1) = \frac{1}{S\sigma_v^2}\sum_{l=1}^{S}\sum_{j=1}^{N_s}\frac{(x_{k+1}-x_{sj})^2}{[(x_{k+1}-x_{sj})^2+(y_{k+1}-y_{sj})^2]^2}\Big|_{X_{k+1}=X_{k+1}^{(l)}} \tag{22}$$

$$\boldsymbol{B}_k^{22,b}(1,3) = \boldsymbol{B}_k^{22,b}(1,4) = 0,\quad \boldsymbol{B}_k^{22,b}(2,3) = \boldsymbol{B}_k^{22,b}(2,4) = 0 \tag{23}$$

$$\boldsymbol{B}_k^{22,b}(3,3) = 0, \boldsymbol{B}_k^{22,b}(3,4) = 0, \boldsymbol{B}_k^{22,b}(4,4) = 0 \tag{24}$$

CPCRLB 能够比较精准地表示目标状态估计的性能边界,本文采用 CPCRLB 作为传感器管理准则,利用 CPCRLB 矩阵的逆中有关 x 方向和 y 方向的位置分量定义效用函数:

$$C_{k+1} = \frac{1}{L_{k+1}^{-1}(1,1) + L_{k+1}^{-1}(2,2)} \tag{25}$$

式(25)表示的是目标状态估计在 x 方向和 y 方向的位置坐标分量的边界和的倒数,利用此效用函数作为节点选择的目标函数。其中, $L_{k+1} = L(X_{k+1} \mid Z_{1:k})$, 可知传感器节点对应的 CPCRLB 越小,其效用函数值越大,则选择的传感器节点越优。

3.2 跟踪策略

在完成了网络中动态节点的优化部署之后,就可以进行目标跟踪。本文采用了二进制麻雀搜索算法。

当目标进入网络的监测范围内时,若目标处在某节点的感知范围内,则该传感器节点为候选节点。为了简化问题,建立节点选择模型如下:

$$A_i = \begin{cases} 1 & \text{选择节点 } i \text{ 来跟踪目标} \\ 0 & \text{不选择节点 } i \text{ 来跟踪目标} \end{cases} \tag{26}$$

其中, $i = 1,2,\cdots,Nm$, Nm 为候选节点的数量,用 0 或 1 来表示节点是否被选择。

节点选择需要计算大量候选节点的目标函数值,如果采用穷举搜索法不可避免地会进行大量的排列组合计算,这将严重消耗内存资源,因此有必要降低计算的复杂度。考虑到麻雀搜索算法的收敛速度较快,这里通过将该算法二进制化来选择建模节点,即采用二进制麻雀搜索算法(Binary Sparrow Search Algorithm, BSSA)在候选节点中选择参与跟踪任务的节点。值得注意的是,这里使用 BSSA 是为了进行最优节点选择,与前面使用 SSA 算法进行网络部署优化的意义是不同的。为了更清楚地说明节点选择任务的寻优计算,这里将进一步给出 BSSA 的具体流程。让每个种群个体代表一个候选节点,搜索空间的维数即为候选节点的个数,设候选传感器节点数量为 Nh, 粒子 id 在第 t 代的状态表示为:

$$M_{od}(t) = [A_{od,1}, A_{od,2}, \cdots, A_{od,Nh}] \tag{27}$$

其中, od 为种群成员序号, $od = 1,2,\cdots,n_t$, n_t 为种群个体总数; $A_{od,1}$ 表示第一个传感器的节点选择信息,其值为 0 或 1,代表该节点是否被选中; t 表示当前迭代次数, $t = 1,2,\cdots,N_{iter}$, N_{iter} 为迭代总数。

在 BSSA 中,更新了发现者、加入者及警戒者的位置信息后,利用 *Sigmoid* 函数把种群位置向量映射到[0,1]区间,按如下进行:

$$S(m_{od,k}(t)) = \frac{1}{1+e^{-a(b-x_{od,k}(t))}} \tag{28}$$

其中，a、b 为固定参数；od 为种群个体的序号；$k = 1,2,\cdots,Nm$。

此时，整个种群以二值的方式进行更新：

$$m_{od,k}(t) = \begin{cases} 1 & rand < S(x_{od,k}(t)) \\ 0 & rand \geqslant S(x_{od,k}(t)) \end{cases} \tag{29}$$

设 BSSA 的适应度函数为：

$$F = \varphi_1 \times D - \varphi_2 \times E \times \eta \tag{30}$$

其中，D 表示跟踪过程的精度控制部分，其值按式(25)来计算；E 表示每一时刻所选节点将量测数据传递给簇头节点所消耗的总能量；η 为数值调整系数；为了将精度部分和能耗部分的数值控制在相同的数量级，φ_1、φ_2 分别为精度和能耗的权重系数。

3.3　算法实现过程

算法的具体过程如下：

步骤 1　网络优化部署。根据 2.2 节的优化步骤完成对传感器网络节点的优化部署。

步骤 2　系统初始化。$t = 0$ 时刻，所有节点均处于休眠状态，并已知目标初始位置。

步骤 3　根据式(12)计算目标的预测位置，并分别计算目标预测位置到 4 个簇头节点的距离，选择其中距离最近的作为此刻的簇头，而其他 3 个簇头节点作为普通节点处理。

步骤 4　根据目标预测位置和节点的感知模型式(1)确定候选节点，以感知概率大于 0 的节点确定为候选节点，并根据二进制麻雀搜索算法来进行节点选择。

步骤 5　激活被选择的传感器节点，该节点会将量测数据直接发送到此刻的簇头节点，并依据 Dijkstra 路径优化算法选择最优多跳路径将数据传送到基站，再由基站根据所获得的量测数据进行粒子滤波，计算得到目标的状态估计。

步骤 6　$t = t + 1$ 时刻，重复步骤 3 ~ 步骤 5。

3.4　仿真分析

3.4.1　参数设置

本次实验针对单目标进行无线传感器网络下的目标跟踪，对本文提出的基于 WSN 部署优化和 CPCRLB 的目标跟踪算法进行性能验证与分析。如前文所述，本次跟踪实验的监视区域仍为[4 km×4 km]，量测模型采用纯方位模型，量测方差为 5°，基站位置为监测区域的中心，坐标为[2 000,2 000]。目标运动模型采用 CV 模型，其中状态转移矩阵 $\boldsymbol{F}$ 和过程噪声协方差矩阵 $\boldsymbol{Q}$ 分别为：

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{Q} = 0.5 \times \begin{bmatrix} \frac{t^3}{3} & 0 & \frac{t^2}{2} & 0 \\ 0 & \frac{t^3}{3} & 0 & \frac{t^2}{2} \\ \frac{t^2}{2} & 0 & t & 0 \\ 0 & \frac{t^2}{2} & 0 & t \end{bmatrix}$$

其中，采样周期 $t = 1$ s。目标函数的参数：$\varphi_1 = \varphi_2 = 0.5$，$\eta = 10^{-7}$。

本文采用粒子滤波器来完成目标状态估计，并采用 100 次蒙特卡洛仿真来验证算法结果。

3.4.2　仿真与分析

目标运动轨迹如图 6 所示。由图 6 可知，目标初始位置为[3 000,500]，向左前方做不规则运动。将本文算法与仅随机静态节点分布的网络、加入随机动态节点后的网络进行目标跟踪的对比，100 次蒙特卡洛仿真的结果如图 7~图 9 所示。图 7~图 9 中，Static 表示仅随机静态节点分布的网络，Hybrid 表示随机静态节点加入了随机动态节点后的网络，Optimized 表示动态节点优化后的网络。

图 7 展示了不同算法对目标运动的跟踪轨迹，可以看出，Optimized 算法对应的估计结果与目标实际轨迹最接近，优于其他算法。不同算法得到的平均 *RMSE* 见表 2。由表 2 可知，Optimized 算法的平均 *RMSE* 最低，说明其跟踪精度最高，优于 Static 算法和 Hybrid 算法；Hybrid 算法的平均 *RMSE* 的值介于 Optimized 算法和 Static 算法之间，比 Optimized 算法的平均 *RMSE* 的值大、并小于 Static 算法的平均 *RMSE*；Static 算法的平均 *RMSE* 的值最大，说明 Static 算法的跟踪精度最差。图 8 更直观地展示了这一过程，展示了不同算法在各个时刻对应的 *RMSE*，可以看出，节点优化部署后的跟踪方法的 *RMSE* 要明显小于其他 2 种算法的。这是因为优化后的网络具有很高的覆盖率，基本可以实现对网络范围内每一个点的有效监测；Hybrid 的覆盖率略逊于 Optimized 算法，所以在某些时刻会出现无法检测目标的情况；而 Static 的覆盖率明显最差，所以在跟踪的过程中，有很多个时刻无法检测到目标，导致出现“量测真空”，所以其跟踪误差也最大。可以看出，在 19 ~ 24 s 这一时间段内，Hybrid、Static 与 Optimized 的跟踪误差值相差最大，也是在这一过程中，Hybrid 和 Static 由于没有节点参与跟踪就会导致误差增大，而其余时间相差并不大。

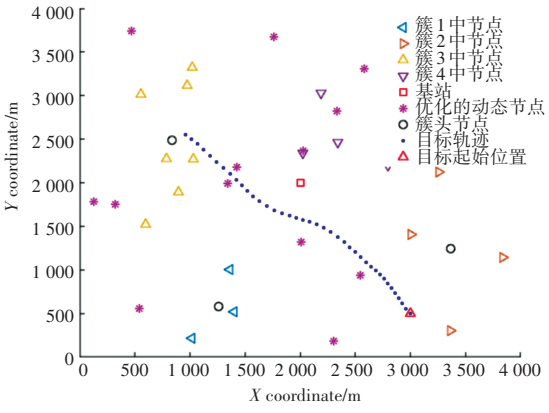


图 6 目标运动轨迹

Fig. 6 Target trajectory

图 9 给出了不同算法的跟踪能耗。由图 9 可以看出,Hybrid 算法能耗最大,而 Static 与 Optimized 算法能耗差异不大,但是 Static 是由于跟踪过程中出现了很多“量测真空”导致多个时刻无节点参与跟踪,所以能耗会比 Hybrid 要小,而 Optimized 每一时刻都有 3 个节点参与跟踪,其能耗也相对较小。

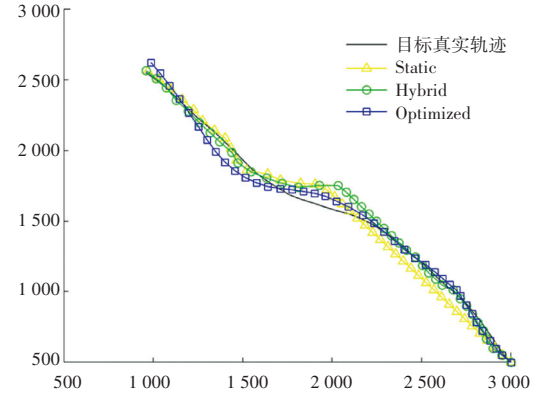


图 7 不同算法的目标跟踪结果与目标实际运动轨迹对比图

Fig. 7 Comparison of the target tracking results of different algorithms and the actual trajectory of the target

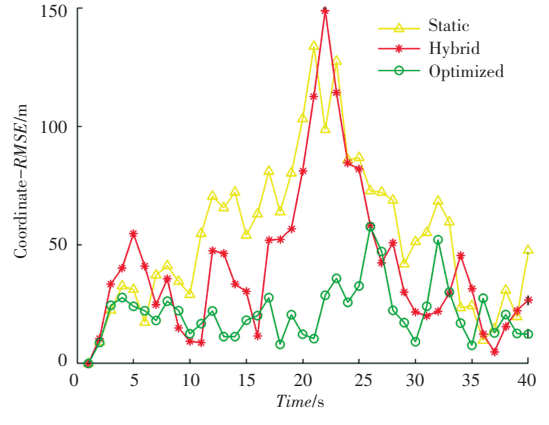


图 8 不同算法的跟踪误差对比图

Fig. 8 Comparison of tracking errors of different algorithms

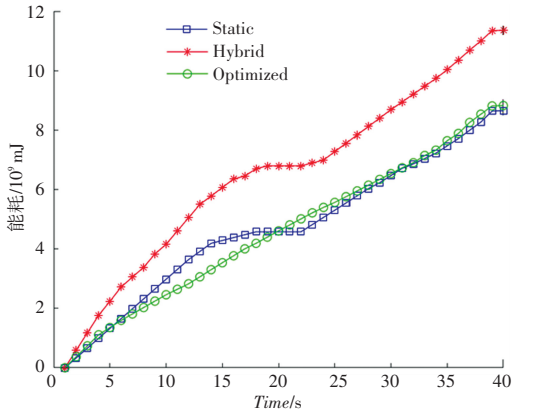


图 9 不同算法的目标跟踪能耗对比图

Fig. 9 Comparison chart of target tracking energy consumption of different algorithms

表 2 不同算法对应的平均 *RMSE*

Tab. 2 Average *RMSE* corresponding to different algorithms

	Static	Hybrid	Optimized
平均 *RMSE*	53.929 9	41.578 3	21.523 4

4 结束语

为了解决无线传感器网络存在的覆盖率低、通信能耗较大的问题,将 WSN 建模为包含动态和静态节点的混合网络,并采用麻雀搜索算法来对网络的部署进行优化,实现了网络覆盖率和通信能耗的性能提升,有效地强化了网络的性能。在此基础上,又设计了基于 CPCRLB 的目标跟踪算法,验证了本文所提基于 WSN 部署优化和 CPCRLB 的目标跟踪算法能够实现在跟踪精度和能耗之间的合理均衡,并优于其他算法。

参考文献

[1] 苏航. 面向智能交通的无线传感网络分簇算法[J]. 交通信息与安全, 2017, 35(03):74-79, 106.

[2] 梁桂兰, 周念东, 徐卫亚. 基于无线传感器网络的边坡实时安全监控系统:中国, CN101859478A[P]. 2010-10-13.

[3] ANITHA G, VIJAYAKUMARI V, MALATHY S, et al. Air pollution monitoring using WSN in cement factory[J]. Journal of Computational and Theoretical Nanoscience, 2018, 15(2):616-620.

[4] ABIDIN H Z, DIN N M, RADZI N A B M, et al. A review on sensor node placement techniques in wireless sensor networks[J]. International Journal on Advanced Science, Engineering andInformation Technology, 2017, 7(1) : 190-197.

[5] YU Xiangyu, LIU Ninghao, HUANG Weipeng, et al. A node deployment algorithm based on Van Der Waals force in wireless sensor networks[J]. International Journal of Distributed Sensor Networks,2013,9(10) : 505710-505710.

(下转第 162 页)