

果蝇启发的水下传感网部署算法

蒋鹏^{1,2}, 冯洋^{1,2}, 吴锋^{1,2}

(1. 杭州电子科技大学自动化学院, 浙江杭州 310018; 2. 浙江省物联感知与信息融合技术重点实验室, 浙江杭州 310018)

摘要: 在三维水下传感器网络中, 恶劣的环境容易导致传感器节点失效, 网络拓扑不稳定. 考虑到实际水环境中存在障碍物及边界, 提出一种果蝇群启发的水下传感网部署算法. 该方法以网络重连通为约束条件, 首先利用果蝇群的觅食行为求解节点移动位置, 同时加入欧琼鸟飞行机制保持移动过程中网络拓扑的相对稳定性, 最后采用障碍物局部感知移动模型进行避障. 结果表明, 该算法可以躲避障碍物, 提高覆盖和连通, 减少网络能耗.

关键词: 水下传感网; 部署; 果蝇优化; 避障

中图分类号: TP393

文献标识码: A

文章编号: 0372-2112 (2017)06-1403-05

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2017.06.018

Fruit Fly Inspired Underwater Sensor Network Deployment Algorithm

JIANG Peng^{1,2}, FENG Yang^{1,2}, WU Feng^{1,2}

(1. College of Automation, Hangzhou Dianzi University, Hangzhou, Zhejiang 310018, China;

2. Key Lab for IOT and Information Fusion Technology of Zhejiang, Hangzhou, Zhejiang 310018, China)

Abstract: In three-dimensional (3D) underwater wireless sensor network, the bad environment is easy to cause node invalidation and topology instability. Considering that obstacles and boundary exist in practical water environment, we put forward the fruit fly inspired underwater sensor network deployment algorithm. Firstly, the method regarded connectivity as the constraint condition, and it used foraging behavior of fruit fly to obtain the nodes location. Meanwhile, it added European starling flying features to keep the stability of network topology. Finally, it used obstacle local perceive model to avoid obstacles. The results show that the algorithm can avoid obstacles, further improve coverage and connectivity, and reduce energy consumption.

Key words: underwater sensor network; deployment; fruit fly optimization; avoid obstacles

1 引言

水下传感器网络^[1] (Underwater Wireless Sensor Network, UWSN) 具有自组织性, 特别适合在恶劣环境下组网, 形成对水下特定区域的监测. 由于水下节点常部署在规模较大且人迹罕至的远程水域, 节点成本高、电池不易替换^[2], 环境损坏造成的节点意外失效频繁出现, 所以实用化的部署算法不仅要满足网络覆盖和连通的基本条件, 同时也要考虑到水环境中障碍物等实际因素的影响. 因此, 设计符合实际的 UWSN 节点部署算法具有现实意义.

目前 UWSN 节点部署算法的研究主要集中于静态部署与动态部署. 静态部署^[3] 包括确定部署和随机部

署, 主要应用于没有移动节点的网络, 在随机部署场景中, 传感器节点通过飞机或其他辅助措施布撒在人类不方便到达的感兴趣的领域 (Area Of Interest, AOI), 但会导致大量节点冗余; 静态部署是已知传感器节点位置, 对监测区域进行覆盖, 缺点是部署效率较低. 而动态部署^[4-6] 可以解决上述两种部署方式的不足, 网络节点根据特定的算法自动调整位置, 直到达到网络的覆盖需求. 例如: Akkaya K 等^[4] 提出一种最大化覆盖的水下三维节点自组织部署算法, 通过调节节点在垂直方向上的距离, 减少节点间的冗余覆盖, 实现最优网络覆盖, 但该算法不能保证网络连通; Liu 等^[5] 考虑水流作用的影响, 通过增加新节点和移除冗余节点的方式解决了水下节点移动过程中覆盖率降低的问题, 但计算复杂

收稿日期: 2015-09-23; 修回日期: 2015-11-17; 责任编辑: 覃怀银

基金项目: 国家自然科学基金面上项目 (No. 61273072); 浙江省环保科技计划 (No. 2013A034); 国家重大科研仪器研制项目 (No. 61427808); 国家自然科学基金重点项目 (No. 61333009); 国家重点基础研究发展计划 (No. 2012CB821204); 浙江省科技厅公益技术研究社会发展项目 (No. 2014c33042)

度较高;Du H 等^[6]提出了粒子群启发的水下传感网部署算法(Particle Swarm Inspired Underwater Sensor Self-Deployment, PSSD), 通过结合粒子群寻优与拥挤度控制, 有效解决了网络覆盖问题, 且复杂度低、收敛速度快及分布可实现, 但不能保证网络连通. PSSD 算法启发于经典的群体智能算法——粒子群算法(Particle Swarm Optimization, PSO), 相比于传统优化算法, 它能简单有效的优化实际复杂问题, 但在后期易陷入局部极值. 而果蝇优化算法(Fruit Fly Optimization Algorithm, FFOA) 作为新型群体智能算法^[7], 除了拥有 PSO 的优点之外, 算法步骤更加简易, 适用于 UWSN 大规模复杂场景下节点部署的特点; 且可调参数少, 减少了因参数选取不当对 UWSN 网络性能的影响. 因此, 果蝇优化算法不失为一种有效解决 UWSN 部署优化问题的新方法.

另外, 现有的部署算法一般是不考虑水下环境中存在障碍物的情况, 而现实中确实存在障碍物阻碍节点移动. 例如, 海洋湖泊中潜藏的暗礁、游动的鱼群和密集的水草群等, 这些障碍物不但影响节点移动, 并且干扰节点之间传播信号. 因此, 本文将障碍物作为一个重要影响因素, 考虑到实际 UWSN 节点部署过程中.

针对上述 PSSD 算法存在的有效性和局限性, 同时考虑到监测水域中存在障碍物与边界, 提出果蝇启发的水下传感网部署算法(Fruit Fly inspired underwater Sensor network Deployment algorithm, FFSD). 首先, 该算法借鉴果蝇优化的核心思想, 具有简单易行、容错能力强、收敛速度快的优点; 再结合分组与欧琼鸟拓扑机制^[8], 避免节点陷入局部最优, 保证网络拓扑相对稳定; 次之, 部署过程中以网络重连通作为约束条件, 保证部署后期工作顺利展开; 最后运用障碍物局部感知移动模型进行避障, 防止节点因外力而失效. 仿真结果表明, 该算法在保证网络连通的情况下, 提高了覆盖质量, 也避免节点因外力作用而失效, 减少网络能耗.

2 问题定义

2.1 问题描述

本文将 UWSN 部署问题定义为: 对于初始随机分布在水中的节点, 在重连通的约束下, 实现最大网络覆盖率的目标优化问题.

假设在水下三维空间中随机部署 n 个无线传感器节点, SINK 节点分布在水面中心位置. 由于 UWSN 被应用在各种复杂环境下, 节点类型和网络模型多种多样, 本文对所采用网络情境作如下假设:

(1) 网络节点同构.

(2) 初始化网络时, 节点与监测事件随机分布, 节点具有感知自身及收发邻居节点位置信息的能力.

(3) 实际水环境中障碍物大小各异, 为便于计算,

将障碍物拟定为规则多面体. 障碍物初始随机分布在水下三维监测空间后不再移动, 节点在移动过程中, 若移动到障碍物区域, 部分节点移动受到阻碍, 其余则撞击到障碍物, 通信中断致使节点失效.

2.2 网络覆盖率

覆盖率计算采用布尔感知模型. 节点的感知范围是一个以节点为中心, R_s 为半径的球形区域. 其中, $P_j(a_j, b_j, c_j)$ 表示监测事件 P_j , $S_i(x_i, y_i, z_i)$ 代表第 i 个节点, 则 $d(s_i, p_j)$ 为节点到监测事件的距离, 见式(1):

$$d(s_i, p_j) = \sqrt{(x_i - a_j)^2 + (y_i - b_j)^2 + (z_i - c_j)^2} \quad (1)$$

$$f(s_i, p_j) = \begin{cases} 0, & d(s_i, p_j) > R_s \\ 1, & d(s_i, p_j) \leq R_s \end{cases} \quad (2)$$

其中, $f(s_i, p_j)$ 为节点 s_i 对监测事件 p_j 的覆盖概率. 若 s_i 与 p_j 的距离不大于节点感知半径, 则被覆盖; 反之未被覆盖.

覆盖率: 被节点覆盖的事件数目(e_c)与目标监测事件总数(e_t)的比值, 记为 C_{cov} , 见式(3):

$$C_{cov} = e_c / e_t \quad (3)$$

2.3 重连通性

为了顺利进行后期工作, UWSN 部署完成后必须保持网络连通. 然而有些节点到后期将无法正常工作, 网络不能维持连通状态, 因此, 本文要求部署后的网络是重连通的. 重连通被定义为: 如果网络中任意两个节点间有两条互斥的路径, 即使从网络中移去一个任意节点, 网络仍然连通^[9].

连通率: 通过单跳或多跳方式与 Sink 节点相连的节点数量(m)与节点总数量(n)的比值, 记为 C_{nec} , 见式(4):

$$C_{nec} = m / n \quad (4)$$

3 算法设计

3.1 FFSD 算法基本原理

首先, 采用蛙跳算法的分组思想将所有节点分组. 节点分成 t 组 memplex, 即 Y_1, Y_2, \dots, Y_m , 每一组 memplex 包含 s 个节点, 如式(5):

$$Y_t = [F(j), f_j | F(j) = F(t + m(j - 1))], \\ f_j = f(t + m(j - 1)), j = 1, 2, \dots, n] \quad (5)$$

其中, $F(j)$ 表示第 1 个节点进入 Y_1 , 第 2 个节点进入 Y_2 , 直到第 m 个节点进入 Y_m , 然后将第 $m + 1$ 个节点放入 Y_1 , 第 $m + 2$ 个节点放入 Y_2 , \dots , 第 $z + m$ 个节点放入 Y_m , 依次类推. 每轮移动结束后, 所有节点合并再重新分组, 多个组群同时运行, 避免算法陷入局部最优.

其次, 网络始终保持重连通. 一个节点当且仅当没有这个节点时, 其 P-hop 子图是不连通的, 那么这个节点被定义为关键节点. 如果一个节点发现自己是关键节点, 它会向所有邻居广播一个关键信息. 为了使网络

重连通,在节点移动后所有关键节点应该变成非关键节点.移动控制的基本思想是:在后期移动过程中,移动非关键节点,保持关键节点静态不动.

在重连通约束下,每组节点同时进行 FOA 进化,采用分布式覆盖.每个节点监测周围事件浓度,判断是否为最优节点.在此基础上,将欧琼鸟拓扑作用机制引入,使算法具有欧琼鸟飞行特征和更强的自适应性,保持稳定的网络拓扑结构.

最后,解决移动过程中存在障碍物和区域边界的问题.若在部署过程中遇到区域边界,节点将停止移动或向原来方向的反方向移动一步,避免节点移出监测区域;若遇到障碍物,将采用障碍物局部感知移动模型进行避障,因为此模型与节点感知范围有限的特点具有一致性.

3.2 FFSD 算法设计与流程图

根据 3.1 节阐述的基本原理,下面给出算法的具体实现步骤:

步骤 1 节点随机分布.随机生成节点群 F ,有 $n = r \times t$ 个节点.其中, r 表示每组 memplex 节点的数量, t 为 memplex 的数量,即子节点群的数量.

步骤 2 所有节点排序再分组.将 n 个节点按适应值 $F(x)$ 降序排列, $F(i)$ 为排在第 i 位的节点,当 $i = 1$ 时节点位置最好,按 3.1 节分组原理进行分组,直到节点分配完成.

步骤 3 每组进行 FOA 进化.在每组节点中,节点都会受到相邻节点影响,通过 FOA 进化使所有节点向目标位置靠近,具体进化过程如下:

(1) 计算每节点 u 到监测事件 w 的距离,即 $\text{Dist}(uw)$.

(2) 计算节点浓度值 $S(uw) = 1/\text{Dist}(uw)$.

(3) 判断组内最大 $S(uw)$ 值,即最优节点,该节点保持位置不变;反之,判断 $S(uw)$ 与 $1/R_s$ 的关系,若 $S(uw) \geq 1/R_s$,节点不再移动,否则节点向最优节点移动,见式(6)、(7):

$$p(u)^{(k+1)} = p(u)^{(k)} + \delta \cdot R_s \cdot v(u \rightarrow \max)^{(k)} \quad (6)$$

$$v(u \rightarrow \max) = \frac{p(u) - p(\max)}{|p(u) - p(\max)|} \quad (7)$$

其中, $p(u)^{(k+1)}$ 表示第 $k+1$ 次移动后节点的位置, $p(u)^{(k)}$ 表示第 k 次节点位置, $v(u \rightarrow \max)$ 表示第 u 个节点到最优节点的单位移动方向, δ 为节点移动速率.

引入欧琼鸟拓扑作用机制后,更新为式(8):

$$p(u)^{(k+1)} = p(u)^{(k)} + \delta \cdot R_s \cdot v(u \rightarrow \max)^{(k)} + c \cdot \text{Vtopologic}^{(k)} \quad (8)$$

$$\text{Vtopologic}^{(k+1)} = \frac{1}{N} \sum_{n \in 1} v_{nu}^{(k)} \quad (9)$$

其中, c 为拓扑因子; N 代表集合的势; T 是数量为 nu 且与第 i 个节点发生拓扑作用的节点集合.由于欧琼鸟拓

扑项式的引入,使传统 FOA 移动过程发生深刻的变化.

(4) 避障和防止移动到边界外.同 3.1 节,选择合适的避障路径;如遇到边界,再选一条其次最优的路径移动,或者停止运动.移动公式最终更新为式(10):

$$p(u)^{(k+1)} = p(u)^{(k)} + \delta \cdot R_s \cdot v(u \rightarrow \max)^{(k)} + c \cdot \text{Vtopologic}^{(k)} + X_{\text{area}} + X_{\text{obs}} \quad (10)$$

其中, X_{area} 为遇到边界时的位置变化,若没遇到边界其值为 0; X_{obs} 为遇到障碍物后的路径变化差值,其值为 0 时,代表移动路径不存在障碍物.

(5) 节点移动完成后,比较最优节点浓度值与移动后浓度值的关系.若移动后的值大于最大浓度值,则更新最优节点;反之,输出浓度值.

步骤 4 所有节点合并,即节点重新在各组跳跃.在每组都执行 FOA 进化后,将各子群 Y_1, Y_2, \dots, Y_m 重新合并为 F ,及时更新最优节点.

步骤 5 若满足终止条件,停止运行;否则,跳转至步骤 2.

4 仿真算例与分析

对西溪湿地水环境进行监测,假设水下监测范围为 $(150 \times 150 \times 150) \text{m}^3$,20 个障碍物分布于水中,大小、位置随机配置,同时布撒 150 个监测事件和 90 个节点.在 MATLAB 仿真环境下模拟 UWSN 节点部署优化过程,验证 FFSD 算法性能.表 1 为仿真参数含义和取值.

表 1 仿真参数

参数	取值	参数	取值
节点感知半径 R_s (m)	16	移动步长 s (m)	$0 < s < 30$
节点通信半径 R_c (m)	32	移动速率 δ (m/s)	1
Memplex 数量(组)	9	拓扑因子 c	0.7
Memplex 节点数量(个)	10	障碍物数目 Obs(个)	20

图 1 是 FFSD 算法与 PSSD 算法的覆盖率随网络迭代次数的变化图.可以看出,在前两次迭代中 PSSD 算法的覆盖率略高,而 FFSD 算法在中后期保持较高覆盖率,这是由于 PSSD 算法前期只追求高覆盖,使节点尽

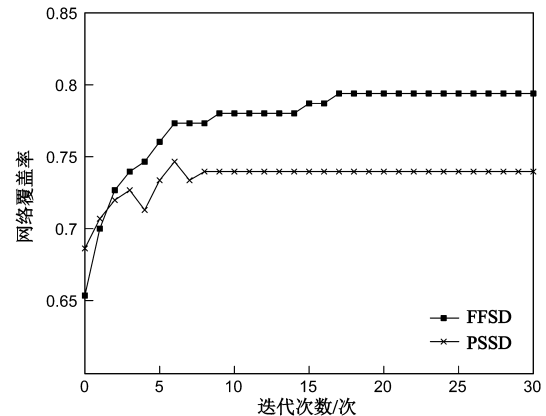


图1 覆盖率对比图

可能趋向事件分布. 而 PSSD 算法在第 5、7 次迭代中, 覆盖率有所降低, 是由于个别节点在已覆盖事件的情况下, 为了覆盖更多事件再次进行移动, 但在此过程中, 恰巧遇到障碍物致使该节点失效, 那么原本被覆盖的事件变为未被覆盖的状态, 导致覆盖率有所下降. FFSD 算法在保证节点不失效的情况下, 不断分组更新跳出局部最优解, 后期覆盖率仍小幅度提升, 达到 79.33% 后不再变化, 具有良好的覆盖性能.

图 2 是 FFSD 算法与 PSSD 算法有效节点数随网络迭代次数变化图. 首次迭代时, PSSD 算法 6 个节点失效, 最后失效 26 个节点. 由于采用障碍物局部感知移动模型来躲避水下障碍物, 使得 FFSD 算法失效节点为 0. 而 PSSD 算法在每次迭代中为了覆盖目标都有大规模节点移动, 这不可避免的会遇到障碍物, 移动的节点数量越多, 遇到障碍物的几率就越大; 若分布在障碍物周围的事件密度大, 则失效节点数目愈多. 因此, 在有障碍物的复杂情况下, FFSD 算法既能选择一条合适的路径覆盖事件, 又能有效地躲避障碍物, 避免节点因外力作用而失效.

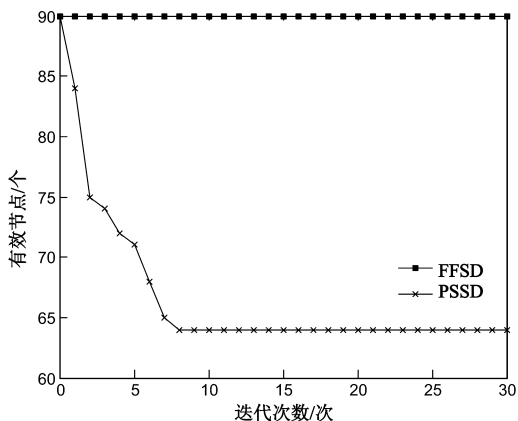


图2 有效节点对比图

图 3 是 FFSD 算法与 PSSD 算法重连通节点数目变化图. FFSD 算法在网络部署开始阶段, 节点随机分布在水下, 某些节点距离相隔较远, 达不到重连通的要求, 但

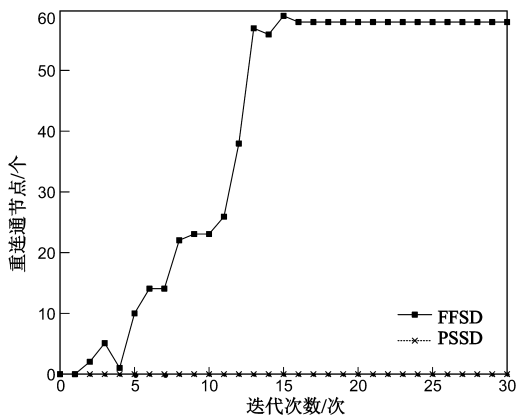


图3 重连通节点对比图

随着移动次数的逐渐增多, 实现重连通的节点数量随之增多, 由于水下存在障碍物的影响不能使整个网络实现重连通, 最终在 18 次移动后整个水下网络有且只有 58 个节点实现重连通. 然而 PSSD 算法在部署过程中, 没有考虑网络连通性, 节点移动后更加不能满足重连通的条件, 因此没有重连通节点. 由此看来, FFSD 算法能实现网络内大部分节点重连通, 维持网络后期拓扑的相对稳定.

图 4 为 FFSD 算法与 PSSD 算法网络连通率随迭代次数的对比图. 由图可以看出, FFSD 算法最终可以实现网络全连通, 由于网络将重连通作为约束条件, 实现了绝大多数节点的重连通, 更保证了整个网络的连通率为 100%; 另外, 欧琼鸟拓扑机制使节点在移动时始终与邻居节点保持同步, 这样避免了节点盲目移动, 保证整个网络拓扑相对稳定, 因此网络连通率稳中有升. 而 PSSD 算法节点的分布是趋向于事件分布的, 在移动过程中没有考虑网络连通性, 连通率只能达到 83.33%, 并不能实现全连通, 也不能保证顺利进行网络后期工作. 因此, FFSD 算法网络连通性优于 PSSD 算法.

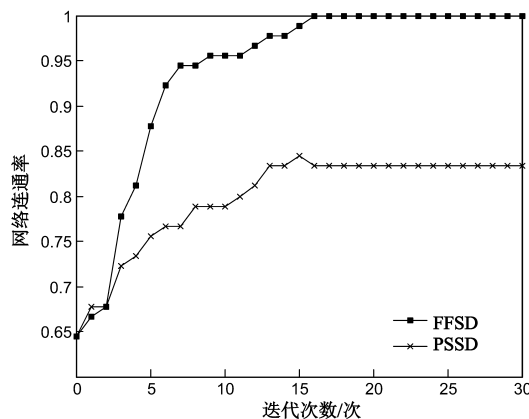


图4 网络连通性对比图

图 5 为 FFSD 算法与 PSSD 算法网络能耗随迭代次数变化的对比图. 本文采用文献 [10] 中以声波为媒介

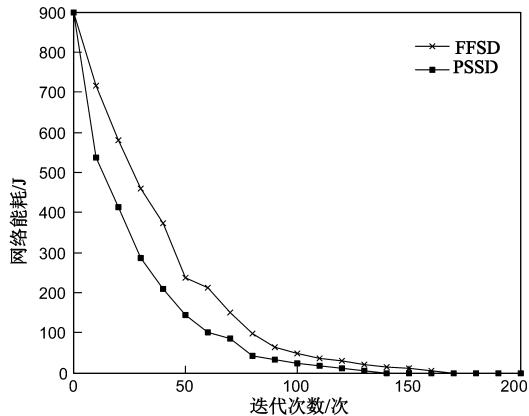


图5 网络能耗对比图

的 UWSN 数据通信能耗模型来计算网络能耗. 由图可以看出, FFSD 算法能耗始终高于 PSSD 算法; PSSD 算法在网络迭代 138 次后能耗为 0J, 而 FFSD 算法在网络迭代 162 次后能量耗尽. 这是由于 FFSD 算法的节点在移动过程中, 个别节点的移动只会影响周围邻居节点, 并不会造成全局节点移动, 减少了节点移动能耗; 与此同时, PSSD 算法在网络迭代 8 次时, 已有 26 个节点因障碍物失效, 那么这些节点自身的能量变为 0J, 造成了巨大的额外能量开销, 使得节点在网络运行前期阶段, 能耗远远低于 FFSD 算法. 由此看来, FFSD 算法网络生存时间延长, 且网络能耗有效降低.

5 结论

该文提出了一种果蝇启发的水下传感网部署算法 (FFSD), 解决了实际监测水域中存在障碍物和边界, 以及保证网络全连通的节点部署优化问题. FFSD 算法在 UWSN 部署过程中, 始终以网络重连通作为约束条件, 首次引入了果蝇优化的基本思想并加以改进, 采用障碍物局部感知移动模型进行避障, 不仅提高网络的覆盖能力, 保证全连通, 也降低了网络能耗; 另外, 水下障碍物的考虑使得 FFSD 算法更加具有实际意义. 仿真过程也体现出该算法简单易实现、计算量小以及复杂度低的优点. 作为今后工作, 将进一步深入对障碍物等水下复杂环境的研究.

参考文献

- [1] Kim D, Cano J C, Wang W, et al. Underwater wireless sensor networks [J]. *International Journal of Distributed Sensor Networks*, 2014, 52(7): 1-5.
- [2] Bicen A O, Sahin A B, Akan O B. Spectrum-aware underwater networks: cognitive acoustic communications [J]. *IEEE Transactions on Vehicular Technology Magazine*, 2012, 7(2): 34-40.
- [3] Ravelomanana V. Extremal properties of three-dimensional sensor networks with applications [J]. *IEEE Transactions on Mobile Computing*, 2004, 3(3): 246-257.
- [4] Akkaya K, Newell A. Self-deployment of sensors for maximized coverage in underwater acoustic sensor networks [J]. *Computer Communications*, 2009, 32(7): 1233-1244.
- [5] Liu B, Ren F, Lin C, et al. The redeployment issue in underwater sensor networks [A]. *IEEE Global Telecommunications Conference [C]*. New Orleans, LA, USA; IEEE, 2008. 1-6.
- [6] Du H, Xia N, Zheng R. Particle swarm inspired underwater sensor self-deployment [J]. *Sensors*, 2014, 14(8): 15262-15281.
- [7] Pan W T. A new fruit fly optimization algorithm: taking the financial distress model as an example [J]. *Knowledge-Based Systems*, 2012, 26: 69-74.
- [8] M Ballerini, N Cabibbo, R Candelier, et al. Interaction ruling animal collective behavior depends on topological rather than metric distance; Evidence from a field study [J]. *Proceedings of the National Academy of Sciences*, 2008, 105(4): 1232-1237.
- [9] Das S, Liu H, Nayak A, Stojmenovic I. A localized algorithm for bi-connectivity of connected mobile robots [J]. *Telecommunication Systems*, 2008, 40(34): 129-140.
- [10] 蒋鹏, 阮斌峰. 基于分簇的水下传感器网络覆盖保持路由算法 [J]. *电子学报*, 2013, 41(10): 2067-2073.
Jiang Peng, Ruan Bin-feng. Cluster-based coverage preserving routing algorithm for underwater sensor networks [J]. *Acta Electronica Sinica*, 2013, 41(10): 2067-2073. (in Chinese)

作者简介



蒋 鹏 (通信作者) 男, 1975 年生于浙江衢州. 现为杭州电子科技大学自动化学院教授、博士生导师. 主要研究方向为传感器网络、嵌入式系统及其应用、智能仪表.
E-mail: pjiang@hdu.edu.cn



冯 洋 女, 1991 年生于山东诸城. 现为杭州电子科技大学硕士生. 主要研究方向为水下传感器网络.
E-mail: suwan_fy@sina.cn



吴 锋 男, 1965 年生于安徽无为. 现为杭州电子科技大学副教授、硕士生导师. 主要研究方向为传感器网络、离散事件动态系统.
E-mail: fengwu@hdu.edu.cn