

基于虚拟力的无线传感器网络分簇部署策略

金仁成, 韦 宁, 徐 浩, 王立鼎

(大连理工大学 辽宁省微纳米技术及系统工程重点实验室, 辽宁 大连 116024)

摘 要: 为提高大规模密集部署的无线传感器网络节点覆盖率, 提出了一种基于虚拟力的节点分簇动态部署策略. 将传统基于虚拟力的节点动态部署分成簇间部署与簇内部署两阶段, 以达到打破网络中部节点受力平衡、降低部署过程中簇间干涉、提高节点覆盖率的目的. 利用二元感知模型对算法进行仿真, 结果表明, 相比于传统 VPF 算法及 VFA 算法, 该策略在随机部署大量密集节点的网络中具有较快的收敛速度与显著的优化效果, 覆盖率最终可达 90% 以上, 可以满足无线传感器网络的覆盖控制要求.

关 键 词: 无线传感器网络; 动态部署; 覆盖控制; 分簇; 虚拟力

中图分类号: TP 212.9; TN 929.5

文献标志码: A

文章编号: 1005-3026(2014)05-0640-05

Clustering Dynamic Deployment Strategy Based on Virtual Force in Wireless Sensor Networks

JIN Ren-cheng, WEI Ning, XU Hao, WANG Li-ding

(Key Laboratory for Micro/Nano Technology and System of Liaoning Province, Dalian University of Technology, Dalian 116024, China. Corresponding author: JIN Ren-cheng, E-mail: rcjin@dlut.edu.cn)

Abstract: In order to improve the coverage of wireless sensor networks (WSNs), a dynamic deployment strategy was presented based on virtual force (VF) clustering. Based on VF, the traditional algorithm was divided into two phases by the presented strategy. First phase was to deploy clusters and second phase was to deploy the nodes in same cluster, which could reduce the interference between clusters and improve the coverage of network. The strategy was simulated in binary perceptual model, which showed that the strategy had a faster convergence speed and significant effect of optimization comparing with traditional VPF and VFA algorithm in the environment of a large number of nodes. The coverage of network eventually could reach more than 90% which could meet the requirement of coverage control of WSNs.

Key words: wireless sensor networks; dynamic deployment; coverage control; clustering; virtual force

无线传感器网络(wireless sensor networks, WSNs), 近年来一直是学术界与工业界关注的焦点. 传感器节点的部署是无线传感器网络开展一系列相关工作的前提. 优化的节点部署可以有效降低区域内的监测空洞, 扩大监测范围, 降低部署成本, 增强网络的连通性能.

目前针对无线传感器网络节点部署的研究可分为两大类: 一类为人工静态部署; 另一类为动态部署. 显然, 动态部署具有更强的适应性.

1 虚拟力与节点动态部署

在解决有关机器人路径规划与避障问题时, 假定待规划物体受到由周围环境根据某种关系所产生的虚拟力的作用, 通过虚拟力作用在物体上所产生的效果, 达到优化的目的. 虚拟力算法具有实现简单、应用灵活、优化效果明显等优点. 因此, 有许多利用虚拟力算法解决无线传感器网络节点

动态部署的研究. Howard 等^[1]首先将 VPF 的概念引入到解决提高无线传感器网络的覆盖率的问题中. 将网络中的每个传感器节点视为带有等量同性电荷的粒子, 仿照库伦力设计节点间虚拟力.

Zou 等^[2]提出一种虚拟力算法 (virtual force algorithm, VFA), 网络中每个节点所受虚拟力不仅来自周围传感节点, 同时也包括障碍物与优先覆盖区域对节点所产生的斥力与引力. Wang 等^[3]提出一种由虚拟力引导的粒子群优化算法, 极大地提升了粒子群优化算法在节点在混合部署中的优化速度. Han 等^[4]提出了一种型心导向虚拟力自部署算法, 使节点所在 Voronoi 多边形型心对节点产生引力的作用, 实现较少的节点移动产生更好的网络部署效果.

前面介绍的虚拟力算法在无线传感器网络节点动态部署的应用中普遍存在以下几点问题:

1) 没有考虑无线传感器网络实际特点, 仿真条件过于理想化. 由于动态部署多用于抛洒部署的网络, 应按正态分布^[5]. 而已有相关算法在仿真过程中, 设定节点密度低、数量少, 且初期部署时覆盖率已较高.

2) 仅针对局部部署设计算法, 并未在全局范围内对算法进行验证. 如对于 VPF, VFA 等算法的仿真验证, 都只是局部验证. 对于处于中心位置节点出现受力均衡、簇间节点可能产生部署干涉等问题, 并未考虑.

3) 相关参数值不易选取. 由于虚拟力相关参数与通信半径等值的大小都会影响部署效果, 因此在事先无法判定随机部署状态的情况下, 无法选取合适参数.

2 感知与仿真模型

在对无线传感器网络节点部署进行研究前首先要明确节点感知模型. 传感器节点感知模型是节点感知质量的一种度量, 决定了网络监测能力与覆盖范围. 目前有两种较为常用的感知模型, 分别为二元感知模型、概率感知模型.

为简化模型, 本文采用二元感知模型. 即以 S_i 节点为感知区域圆心, 以节点的监测半径 R_s 作为圆形感知区域半径. 若事件 $P(x, y)$ 发生在圆形感知区域内, 则被监测的概率 $C_{x,y}(S_i)$ 为 1 否则为 0. $d(P, S_i) = \sqrt{(x - x_i)^2 - (y - y_i)^2}$ 为发生的事件 $P(x, y)$ 距节点 S_i 的平面距离.

$$C_{x,y}(S_i) = \begin{cases} 1, d(P, S_i) < R_s; \\ 0, d(P, S_i) \geq R_s. \end{cases} \quad (1)$$

确定节点感知模型后, 便可以对网络中节点的覆盖率进行计算. 网络节点覆盖率为所有节点集合覆盖的区域的面积与被监测区域总面积的比值. 本文在仿真中采用栅格法计算覆盖率, 即将监测区域划分面积相等的矩形栅格进行统计分析, 划分粒度由求解精度决定. 当栅格大小为待测区域大小的 4% ~ 0.25% 时, 计算值与精确值之间的误差为 0.5% ~ 0.1%^[6]. 因此, 本文将栅格大小设为 1.

$$\text{Coverage} = \frac{\cup C_i}{A} = \frac{\iint_A f(x, y) d\sigma}{A}, \quad (2)$$
$$f(x, y) = C_{x,y}(S_{OV}) = 1 - \prod_{S_i \in S_{OV}} (1 - C_{x,y}(S_i)). \quad (3)$$

式中 A 为监测区域面积, S_{OV} 为网络中节点集合. 为简化仿真模型, 针对无线传感器网络特点做如下假设:

- 1) 网络中所有节点可以通过节点定位等方式获取自身位置信息.
- 2) 所有节点都能够根据优化目标准确实现移动, 无位置偏差. 且节点所具有能量足够使部署进行结束.
- 3) 监测区域为二维平面且无障碍出现.

3 基于虚拟力的分簇部署策略

为了解决已提出的基于虚拟力的动态部署策略所存在的问题与不足, 提出一种基于虚拟力的无线传感器网络分簇部署策略, 见图 1. 将网络节点部署分为簇间部署与簇内部署两个阶段. 通过网络分簇确定参与簇间部署的移动单位规模, 从而对相关参数进行确定. 由于簇间部署后, 簇间距变大、原有单块聚集区域分散为多块聚集区域, 因此可以有效降低簇间干涉与节点受力平衡问题.

3.1 网络分簇算法设计

提出一种分布式基于地理位置的分簇算法, 将被监测的区域划分为若干大小相等的矩形区域, 在同一个矩形区域内拥有相同区域标识的节点才可以成为同一个簇的节点. 首先由汇聚节点得到网络需要监测的区域的位置以及大小, 并根据监测区域的大小及实际需求划分矩形区域, 给出矩形划分区域的边长, 将这些信息广播给整个网络. 如图 2 所示, 在面积为 100×100 的区域中, 划分了 100 个 10×10 的等面积区域, 中间矩形为汇聚节点. 此后网络中节点启用定位算法, 确定自身所在位置. 节点根据自身的位置, 就可确定所属划分区域, 设定自身区域标识 $[S_x, S_y]$.

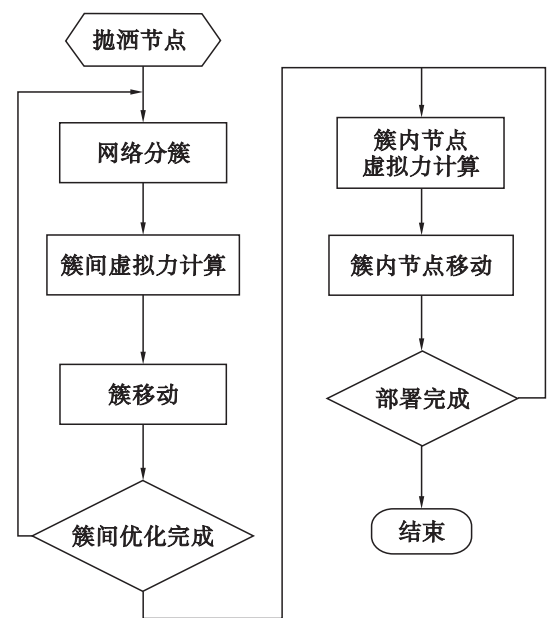


图 1 部署流程图

Fig. 1 Flow chart of the deployment

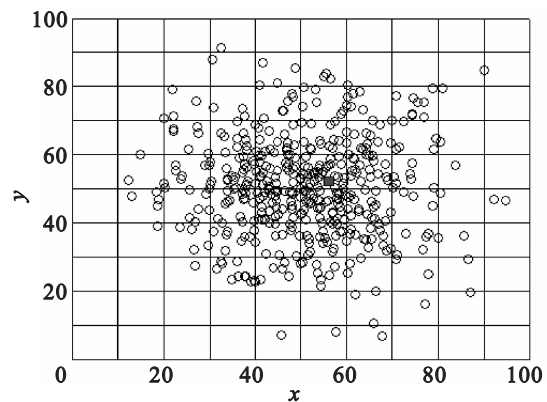


图 2 区域划分示意图

Fig. 2 Zoning map

$$[S_x, S_y] = \left[\left\lceil \frac{x_n - x_c}{l} \right\rceil, \left\lceil \frac{y_n - y_c}{l} \right\rceil \right]. \quad (4)$$

其中: x_n, y_n 分别为节点 n 的坐标值; x_c, y_c 分别为整个监测区域中心坐标值; l 为被划分矩形区域的边长; $\lceil \cdot \rceil$ 为向上取整。

此后任意节点 n 将进行时间为 $D(n)$ 的延时。

$$D(n) = \text{rand}(n) + T \times \left\lceil \frac{3\sqrt{2}d(n, c)}{l} \right\rceil. \quad (5)$$

其中: $\text{rand}(n)$ 为第 n 个节点产生的随机数值; T 为延时系数, 且 $T > \text{rand}(n)$; $d(n, c)$ 为节点 n 与节点所在划分区域中心间的距离。该公式将被划分区域分为三个拥有相同质心的矩形, 对角线长度比为 1:2:3, 处在中心区域的节点产生的时间延时较小, 相反则时间延时较大。为处于不同位置节点设定不同延时的目的为降低同一区域内的成簇时间。

在延时结束时, 若节点未收到与自身区域标识相同的其他节点的广播信息, 则节点以 $R_c = \sqrt{2}l$ 为半径广播自身位置信息以及区域标识, 并将自身设为簇首状态。若在延时时间内, 节点收到其他节点广播的信息, 则将收到的信息与自身信息进行比较。若节点自身与划分区域中心间距离小于发送信息节点与划分区域中心间距离, 则节点重新延时 $D(n)$ 时间, 并重新进行上述操作; 若大于, 则节点将自身设为等待成簇状态。

最终被选为簇首的节点在这一轮将进行成簇操作。簇首节点广播成簇命令, 在同一划分区域内的节点加入该簇, 成簇结束后, 簇首节点将会统计出本簇内的节点总数。对于簇间节点部署操作来说, 可能会出现划分区域间节点数相差过多的情况, 如图 2 所示, 有些划分区域内存在大量的节点, 而有些区域内有很少甚至没有节点。这样可能会造成簇内节点整体移动后, 大部分节点仍落在同一个划分区域的情况, 严重影响部署效果。

为避免这种情况的发生, 本文提出一种簇分裂策略, 即成簇后, 若簇首节点发现本簇内节点数量大于设定的某一阈值 N_{th} , 簇首将会在本簇内另外再选取一个簇首节点, 即一旦某一划分区域内节点数量过多, 那么在这一区域内将会出现两个规模相差不多、距离很近的簇。选取原则是, 距离原有簇首节点最近的普通节点将成为另一簇首节点。选取完毕后, 这两个簇首节点重新成簇。采取簇分裂策略后, 就可以有效避免在多次簇间部署后, 某一划分区域内仍有较多节点的情况。同时, 每一区域内也将设置一个成簇最少节点阈值 N_{th}^* , 当该区域内节点数量少于 N_{th}^* 时该区域内节点将不进行成簇操作。

3.2 簇间与节点间虚拟力计算

成簇后, 首先进行簇间部署, 即簇首节点通过计算来自周围簇首节点虚拟斥力的合力 \vec{F}_j^* , 控制本簇内所有节点朝相同方向进行等距离移动。

$$\vec{F}_j^* = \sum_{j=1, j \neq i}^n \vec{F}_{ij}^*. \quad (6)$$

成簇后, 每个簇的成员节点数量将不尽相同。相邻两个簇间, 节点数量的多少、距离的远近将直接影响到后续部署的效果。在两簇间距离相等的前提下, 两簇内节点数量越少, 对后续的部署就越有利; 而在两簇内节点数量不变的前提下, 簇间距离越远, 对后续的部署就越有利。这就需要使距离较近的两簇在移动后距离增大, 对于簇内节点数量较多的两簇则需移动更远的距离。因此, 将簇内成员节点数量与簇间距离作为参数引入到簇间

虚拟力的计算,参照库伦力,得到簇间斥力计算公式:

$$|\vec{F}_{ij}^*| = \begin{cases} K \frac{N_i N_j}{d_{ij}^{*2}}, d_{ij}^* < d_{th}^*; \\ 0, d_{ij}^* \geq d_{th}^*. \end{cases} \quad (7)$$

任意两个簇间作用力的方向 α_{ij} 为第 i 个簇的簇首与第 j 个簇的簇首的向量角度. K^* 为簇间虚拟力系数. N 为簇内成员节点数量. d_{ij}^* 为簇首节点间距离. d_{th}^* 为簇首间距离阈值,当簇首间距离 d_{ij}^* 大于或等于阈值时,簇间将不产生任何力.

在簇间部署结束后,将进行节点簇内部署.即每个簇内的节点都将各自的信息发送给簇首节点,由簇首节点根据簇内虚拟力计算公式确定簇内节点所受本簇内其他节点 i 的作用力.

$$|\vec{F}_{ij}| = \begin{cases} \omega_A (d_{ij} - d_{th}), d_{ij} > d_{th} \\ 0, d_{ij} = d_{th}; \\ \omega_R \frac{1}{d_{ij}}, d_{ij} < d_{th}. \end{cases} \quad (8)$$

其中: d_{ij} 为节点 i 与 j 间的欧氏距离; d_{th} 为阈值即节点间的最佳距离 ω_A, ω_R 分别为引力与斥力系数.

3.3 簇间与节点间移动策略

在计算完簇首间与簇内节点间的虚拟力后,网络中所有受力节点都将进行移动.节点从原有位置 $P(x_{old}, y_{old})$ 移动到新位置 $P(x_{new}, y_{new})$.

$$x_{new} = \begin{cases} x_{old}, |\Delta x| \leq x_{th}, \\ x_{old} + \Delta x, x_{th} < |\Delta x| < \text{Maxstep}, \\ x_{old} + \text{Maxstep}, |\Delta x| \geq \text{Maxstep}. \end{cases} \quad (9)$$

$$y_{new} = \begin{cases} y_{old}, |\Delta y| \leq y_{th}, \\ y_{old} + \Delta y, y_{th} < |\Delta y| < \text{Maxstep}, \\ y_{old} + \text{Maxstep}, |\Delta y| \geq \text{Maxstep}. \end{cases} \quad (10)$$

其中: $\Delta x, \Delta y$ 分别为节点或簇在 x 轴方向与 y 轴方向上的位置增量; x_{th} 与 y_{th} 为增量阈值,阈值的设定防止了节点或簇在某一位置震荡的发生,当节点或簇的位置增量很小时,这些增量将被忽略; Maxstep 为簇或节点在 x 轴或 y 轴方向的最大移动距离.簇间部署移动与簇内节点移动都设定了增量阈值与最大移动距离,但在不同阶段上述值的设置不应相等,在簇间部署阶段的值应大于簇内部署阶段值的设定.位置增量的计算符合牛顿运动方程.

$$\Delta x = \int \left(V_x + \frac{F_x - fV_x}{M} \right) dt, \quad (11)$$

$$\Delta y = \int \left(V_y + \frac{F_y - fV_y}{M} \right) dt. \quad (12)$$

其中: F_x 与 F_y 分别为这些合力在 x 轴与 y 轴上的分力; V_x 为簇或节点当前的移动速度; f 为阻尼系数,因此假设簇或节点处在一个具有阻尼的环境中移动,支撑移动的能量将被系统阻尼不断消耗,最终实现稳定; M 为虚拟质量; t 为每轮移动的时间.通过对速度的积分便可求出位移的大小.

此外,在簇与节点的移动过程中还要注意监测区域的边界问题.当节点过于靠近边界时,会产生边界效应,使覆盖率下降.因此,无论是簇的整体移动还是节点的单独移动都会距离监测边界一定的距离,即尚未达到监测边界时就已停止朝边界方向移动.

4 仿真与分析

仿真均在 100×100 区域内进行.设定节点的监测半径 R_s 为 4,若想在监测区域内实现全覆盖,则至少需要 225 个节点^[6],此时节点的覆盖效率最高.为保证网络的连通,设定节点间通信半径 R_c 为 15^[7].节点在监测区域呈正态分布.在 Matlab2011 环境下进行仿真.

图3为簇间部署覆盖率变化仿真结果.簇间部署取 K^* 值为 50, Maxstep 为 15, F_{th} 为 5,共进行 50 次优化.可以很明显看出,通过簇分裂增加簇间部署过程中簇的数量可以有效提高簇间部署阶段的部署效果.簇间部署最终可以达到稳定的状态,当达到稳定状态后,节点覆盖率并未达到很高的水平,因此需要进行簇内节点部署.

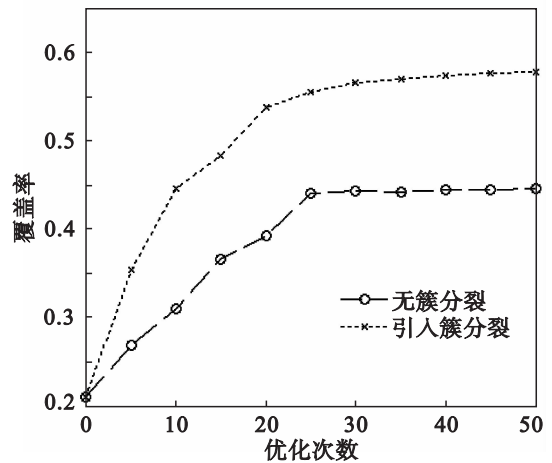


图3 簇间部署覆盖率

Fig. 3 Coverage changes in clusters deployment

图4为针对本文所提出策略的仿真结果.簇间部署参数不变,簇内部署 ω_A 为 1, ω_R 为 0.5,

Maxstep 为 5, F_{th} 为 2, 共进行 100 次优化. 由图 3 已知, 当簇间部署进行 25 次优化后, 节点覆盖率趋于稳定. 因此, 图 4 从第 25 次优化后, 开始实施簇内部署, 可以明显地看出簇分裂策略在簇内部署阶段也能明显提高节点覆盖率.

本文提出的基于虚拟力的分簇部署算法与 VPF 及 VFA 算法对比改善效果明显. 由于簇间部署的引进, 有效地改善了部署过程中簇间干涉的影响, 同时对于由节点过多过于密集所带来的虚拟力抵消现象也有很好的改善效果.

图 5 为通过本文所提出算法进行优化后的网络节点部署效果图. 可以明显地看出加入簇分裂策略后的算法优化效果十分明显.

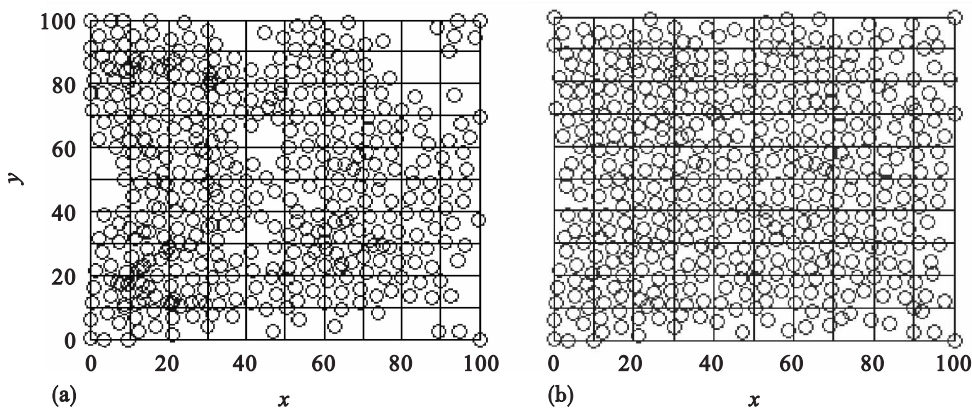


图 5 优化部署效果图
Fig. 5 Effect of deployment optimization
(a)—未加入簇分裂策略; (b)—加入簇分裂策略.

5 结 语

本文针对无线传感器网络随机部署的特点, 将网络的覆盖抽象为基于二元感知模型的圆盘覆盖问题, 提出了一种基于虚拟力的分簇部署算法. 该算法将基于虚拟力的部署分为两个阶段进行, 先进行网络分簇后的簇间部署, 再进行簇内节点间部署. 在分簇过程中引入簇分裂策略提高了本算法的优化性能. 仿真结果表明, 该算法有效改善了传统的基于虚拟力的 VFA, VPF 算法在部署过程中出现的簇间干涉、节点受力均衡等问题, 在初期有大量节点密集部署的网络中可以很好地改善原有随机部署网络的覆盖率.

参考文献:

[1] Howard A, Mataric M J, Sukhatme G S. Mobile sensor network deployment using potential field; a distributed scalable solution to the area coverage problem[C]//Distributed Autonomous

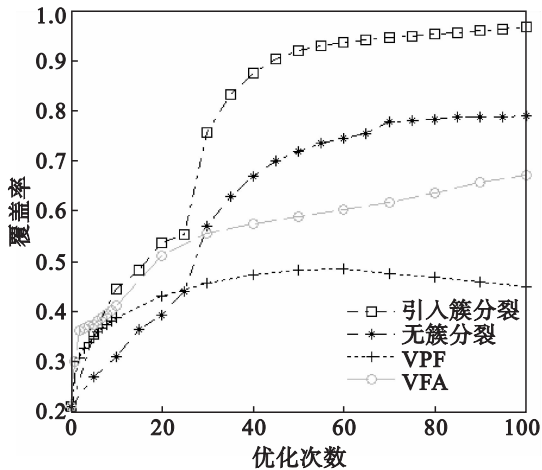


图 4 覆盖率优化对比
Fig. 4 Contrast of coverage optimization

Robotics Systems (DARS 2002). Tokyo: Springer-Verlag, 2002;299 – 308.

[2] Zou Y,Chakrabarty K. Sensor deployment and target localization in distributed sensor networks [J]. *ACM Transactions on Embedded Computing Systems*,2004,3(1) :61 – 91.

[3] Wang X, Wang S, Ma J J. An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment[J]. *Sensors*,2007,7:354 – 370.

[4] Han Y H, Kim Y H, Kim W T, et al. An energy-efficient self-deployment with the centroid-directed virtual force in mobile sensor networks[J]. *Simulation*,2012,88(2) :7 – 17.

[5] Choudhury R R, Kravets R. Location-independent coverage in wireless sensor networks[EB/OL] (2004 – 01 – 01) [2013 – 04 – 22] // http://www. crhc. uiuc. edu/~croy/pubs/, 2004.

[6] Wang X, Wang S, Ma J. Dynamic deployment opt-imization in wireless sensor networks[J]. *Lecture Notes in Control and Information Sciences*,2006,344:182 – 187.

[7] 曹峰,刘丽萍,王智. 能量有效的无线传感器网络部署[J]. *信息与控制*,2006,35(2) :147 – 153.

(Cao Feng, Liu Li-ping, Wang Zhi. A new energy-efficient WSN deployment algorithm [J]. *Information and Control*, 2006,35(2) :147 – 153.)