

基于学习自动机的异构有向传感器节点调度算法

李 明^{1,2}, 胡江平²

(1. 重庆工商大学 重庆市检测控制集成系统工程实验室, 重庆 400067;

2. 电子科技大学 自动化工程学院, 成都 611731)

摘 要: 现有有向传感器网络调度算法多数只面向同构传感器节点, 而未考虑节点异构性对算法性能的影响。为此, 提出一种基于学习自动机的异构有向节点调度算法。将节点调度问题转化为集合覆盖问题后, 利用学习自动机的特性自适应地更新所选取感知方向的概率, 从而构建多个满足条件的节点覆盖集合。仿真结果表明, 与贪婪算法相比, 该算法能有效减少能量消耗并延长网络寿命。

关键词: 有向传感器网络; 节点异构性; 节点调度; 学习自动机; 贪婪算法

中文引用格式: 李明, 胡江平. 基于学习自动机的异构有向传感器节点调度算法[J]. 计算机工程, 2018, 44(8): 199-203.

英文引用格式: LI Ming, HU Jiangping. Scheduling algorithm for heterogeneous directed sensor nodes based on learning automata[J]. Computer Engineering, 2018, 44(8): 199-203.

Scheduling Algorithm for Heterogeneous Directed Sensor Nodes Based on Learning Automata

LI Ming^{1,2}, HU Jiangping²

(1. Chongqing Engineering Laboratory for Detection Control and Integrated System,

Chongqing Technology and Business University, Chongqing 400067, China;

2. School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

【Abstract】 Most existing directed sensor network scheduling algorithms only focus on isomorphic sensor nodes, without considering the influence of node heterogeneity on the performance of the algorithm. To solve this problem, a heterogeneous nodes scheduling algorithm based on learning automata is proposed. The node scheduling problem is transformed into a set coverage problem, and the probability of the selected perception direction is adaptively updated by using the characteristics of learning automata to construct a number of node coverage sets with satisfying conditions. Simulation results show that compared with greedy algorithm, the proposed algorithm can effectively reduce energy consumption and prolong network lifetime.

【Key words】 directed sensor network; node heterogeneity; node scheduling; learning automata; greedy algorithm

DOI: 10.19678/j.issn.1000-3428.0047712

0 概述

随着微机电技术的发展, 由视频传感器、红外传感器和超声波传感器等构成的有向传感器网络在实践中得到了广泛应用^[1], 且受到越来越多研究者的重视。如何在保证监测质量的前提下减少网络能耗、延长网络工作时间, 成为有向传感器网络领域的重点研究方向之一。节点调度技术通过对传感器节点的资源进行时间和空间的优化, 实现了网络工作时间的延长^[2-4]。文献[5]提出一种邻居节点调度协议, 引入局部覆盖集的概念, 通过局部覆盖集来判断

当前节点是否为冗余节点, 并在考虑节点剩余能量时决定其是否可以转为休眠状态。该调度协议允许一个节点加入多个覆盖集, 覆盖集轮流工作使网络生存期达到最大化。文献[6]研究最大有向区域覆盖问题, 通过调整有向传感器的工作方向使覆盖的区域面积达到最大值, 但其在调整过程中未考虑能量消耗。文献[7]提出 2 种启发式有向传感器网络节点调度算法, 以解决监测目标在有优先级情况下的网络覆盖问题并延长网络寿命, 但该算法需要大量的消息交换, 算法复杂度较高。文献[8]提出一种基于遗传算法的节点调度策略, 用其来延长网络的

基金项目: 重庆教委科学技术研究项目(KJ1600627, KJZH17124); 重庆市检测控制集成系统工程实验室开放课题(611315002); 重庆市社会科学规划项目(2017YBGL142)。

作者简介: 李明(1982—), 男, 副教授、博士, 主研方向为智能算法、传感器网络; 胡江平, 教授、博士。

收稿日期: 2017-06-26 **修回日期:** 2017-08-08 **E-mail:** sshjlm@163.com

生命周期。文献[9]将网络中的节点划分为互不相交的子集合,每个子集合都能将数据传送到 sink 节点,每次只有一个子集合处于工作状态,其他子集合处于休眠状态,通过这种方式来延长网络的生命周期。文献[10]提出一种基于遗传算法的有向传感器节点调度方法,可以延长网络寿命。文献[11]提出一种基于网格划分的有向传感器最大覆盖调度算法。

上述方法都是针对感知角度、感知半径和携带相同能量的同构有向传感器网络的,均未考虑节点异构性特别是能量异构性对节点调度算法的影响,且算法计算复杂度均较高,未考虑传感器节点资源的局限性。此外,由同构节点构成的网络扩展性较差,不利于在出现节点故障后对网络进行维护^[12]。尽管文献[13]针对无线传感器网络服务质量会随着网络运行而下降这一现象,提出了一种基于目标覆盖的异构有向传感器网络分布式节点调度策略,但该策略需要频繁地进行消息交换,通信消耗较大,影响了节点的寿命。

针对上述算法存在的不足,本文提出一种基于学习自动机的异构有向传感器网络节点调度算法。利用学习自动机的自适应能力将网络节点划分成互相独立的子集合,使每个子集合都能保证网络完全覆盖,并通过不同子集合休眠、工作状态的切换来实现网络生命周期的延长。

1 异构节点调度算法的原理与实现

1.1 系统模型与假设

在二维平面内随机分布着 M 个需要监测的目标点 $T_i (i=1,2,\dots,M)$,目标点位置信息已知。在二维平面内部署 N 个有向传感器节点 $S_i (i=1,2,\dots,N)$,其感知半径为 $RS_i (i=1,2,\dots,N)$,通信半径为 $RC_i (i=1,2,\dots,N)$,感知方向 D_i 的数量为 $|D_i| (i=1,2,\dots,N)$,感知角度为 $A_i (i=1,2,\dots,N)$,携带能量为 $E_i (i=1,2,\dots,N)$,所在位置 $P_i (i=1,2,\dots,N)$ 已知或者可以通过其他定位技术获得^[14]。这些有向传感器节点的系统参数,即感知半径、通信半径、感知方向的数量、感知角度和所携带能量均不相同。假定有向传感器节点在同一时刻只能在一个感知方向上处于工作状态,即同一时刻某个节点只有一个感知方向处于工作状态,其他感知方向均处于休眠状态。

定义 1 (目标点 T_i 被有向传感器节点 S_i 覆盖) 目标点 T_i 与有向传感器节点 S_i 所在位置 P_i 的距离小于等于 S_i 的感知半径,且 S_i 与目标点 T_i 之间的连线与 S_i 感知方向的夹角小于 S_i 感知角度的一半。

本文要解决的问题是如何对有向传感器节点进行集合划分,使得划分的集合数最多,同时保证集合中的传感器节点能够满足目标覆盖的要求,进而实现网络生命周期的最大化。该问题形式化描述为:

$$\max_{k=1}^K \sum t_k \quad (1)$$

$$\text{s. t. } \sum_{k=1}^K \sum_{j=1}^{|D_i|} x_{i,j,k} \cdot t_k \leq L_i, \forall S_i \in S \quad (2)$$

$$\sum_{j=1}^{|D_i|} x_{i,j,k} \leq 1, \forall S_i \in S, k=1,2,\dots,K \quad (3)$$

$$\sum_{D_{i,j} \in C_{T_m}} x_{i,j,k} \geq 1, \forall T_m \in T, k=1,2,\dots,K \quad (4)$$

$$x_{i,j,k} = \begin{cases} 1, & D_{i,j} \in C_k, t_k \geq 0 \\ 0, & \text{其他} \end{cases}$$

其中, t_k 表示每个集合的工作时间, K 表示划分的覆盖集合的最大数量, $D_{i,j}$ 表示节点 S_i 的第 j 个感知方向, C_k 表示第 k 个满足条件的覆盖集合, L_i 表示节点 S_i 的寿命, C_{T_m} 表示能覆盖监测目标 T_m 的感知方向的集合, $m=1,2,\dots,M$ 。式(2)保证节点的工作时间不会超过其寿命;式(3)保证在满足条件的集合中最多只有一个感知方向处于工作状态;式(4)保证每个监测目标都能被覆盖集合中的至少一个感知方向所监测到。

1.2 学习自动机算法的原理与实现

1.2.1 学习自动机基本原理

学习自动机是一种能够根据外界环境自适应选择优化行为的智能单元,其主要通过与环境的不断交互来改进自己的行为,从而在当前的环境条件下获得环境奖励的最大动作^[15-16]。一般情况下,学习自动机可用一个四元组 $\{\alpha, \beta, p, G\}$ 来描述。各部分意义为: α 表示候选动作的集合 $\{\alpha_1, \alpha_2, \dots, \alpha_r\}$, r 为动作个数; β 表示环境的反馈集合,在学习自动机中,环境可分为连续环境和离散环境。在连续环境(也称为 S 型环境)中, β 取值为 $[0, 1]$ 之间的小数,在离散环境(也称为 P 型环境)中, β 取值为 0 或 1; p 为与候选动作集一一对应的概率集合 $\{p_1(n), p_2(n), \dots, p_r(n)\}$, $p_i(n)$ 表示在时刻 n 动作 α_i 被选中的概率,一般情况下,集合 p 中的元素需要满足 $0 \leq p_i(n) \leq 1$, 各元素的和 $\sum_{i=1}^r p_i(n) = 1$; G 表示学习自动机的更新规则,其映射关系为 $p_i(n+1) = G[\alpha(n), \beta(n), p_i(n)]$ 。

学习自动机在学习过程中基于以下规则更新其动作概率^[17]:

1) 对于有利的响应,根据式(5)来更新动作概率。

$$\begin{cases} p_i(n+1) = p_i(n) + a \times [1 - p_i(n)] \\ p_j(n+1) = p_j(n) - a \times p_j(n) \end{cases}, \forall j \neq i \quad (5)$$

2) 对于不利的响应,根据式(6)来更新动作概率。

$$\begin{cases} p_i(n+1) = p_i(n) - b \times p_i(n) \\ p_j(n+1) = p_j(n) + b \times \left[\frac{1}{r-1} - p_j(n) \right] \end{cases}, \forall j \neq i \quad (6)$$

其中, a, b 分别表示奖励参数与惩罚参数。

学习自动机的结构可以分为固定结构和可变结构,可变结构学习自动机的行为数量可以发生变化。学习自动机的控制规则简单易用,无需耗费计算资源,适用于传感器这种内存和计算资源受限的硬件节点。此外,学习自动机具有收敛性能好、优化能力强的特点,能迅速通过与环境的交互在反馈学习中找到优化的解。因此,本文采用可变结构的学习自动机来进行问题求解。

1.2.2 基于学习自动机的异构节点调度算法

现有的节点调度算法多数针对参数相同的节点,而本文提出的基于学习自动机的异构节点调度算法借助学习自动机与环境的交互和候选动作集的奖惩,能够实现网络覆盖率和网络寿命的优化。基于学习自动机的异构节点调度算法分为3个步骤:算法初始化,求解满足覆盖要求的异构节点集合,算法参数更新。各步骤具体内容如下:

1) 算法初始化主要对学习自动机的一些参数进行初始化。根据求解问题的特点,定义候选动作集为所有有向传感器节点的感知方向,即 $\alpha = \{D_{i,j}\}$, $i = 1, 2, \dots, N; j = 1, 2, \dots, |D_i|$ 。相应地,取候选动作集的概率为该感知方向覆盖的监测目标数与所有需要监测的目标数的比值,即:

$$p = \{p_1, p_2, \dots, p_t\}, t = \sum_{i=1}^N |D_i|$$

$$p_u = \frac{|T_{i,j}|}{M}$$

$$u = 1, 2, \dots, t; i = 1, 2, \dots, N; j = 1, 2, \dots, |D_i|$$

其中, $|T_{i,j}|$ 表示第 i 个节点第 j 个感知方向覆盖的监测目标数。

设置2个全局变量 $G_{\text{uncovered}}$ 和 G_{loop} 分别用于存储未覆盖的监测目标数和最大迭代次数, $G_{\text{uncovered}}$ 的初值设为所有的监测目标数。

2) 求解满足覆盖要求的异构节点集合时,算法以迭代的方式运行,用变量 k 记录当前的迭代次数,并赋其初值为1。每次迭代能得到一个满足覆盖要求的节点感知方向集合,若新得到的感知方向集合的基(即集合中元素的个数)小于已得到的、最好的满足覆盖要求的集合中的基,则新集合中感知方向对应的候选动作概率将得到奖励(变大),并按照式(5)进行更新,否则,将按照式(6)进行更新。然后,将已得到的、最好的满足覆盖要求的集合更新为新得到的集合,并更新节点的剩余能量,若该节点能量已耗尽,则将该节点涉及的所有感知方向从候选动作集中删除,相应地,将其对应的概率设为0。最后,将当前迭代次数 k 加1,若 k 的值已大于最大迭代次数 G_{loop} ,则算法结束,否则,进入下一次迭代。

每次迭代过程的具体步骤为:

步骤1 学习自动机随机选择一个节点的一个感知方向。

步骤2 根据该感知方向信息计算得到覆盖的目标信息。

步骤3 更新 $G_{\text{uncovered}}$ 变量,将集合中已覆盖的目标信息删去。

步骤4 若该变量已变成空集,说明所有的目标都已覆盖,则该次迭代结束,重新将变量 $G_{\text{uncovered}}$ 赋值为监测目标的集合,否则,继续步骤1。

有向传感器节点某个特定时刻只能工作在一个感知方向,因此,为提高算法效率,在上述步骤1中,若该节点的某个感知方向已被选择,则将候选动作集中该节点的其他感知方向删去,并将其对应的动作概率设为0。同时,为避免已经选择的感知方向被重复选择,将已被选中的感知方向从候选动作集中删去,并将其对应的动作概率设为0。此外,在节点能量消耗方面,用 t_w 表示每个集合的工作时间,本文假定节点消耗的能量与节点工作时间成正比,即 $E_{s_i} = \varepsilon \cdot t_w$, ε 为比例系数,本文设为1。

从上述算法步骤可以看出,本文基于自动学习机的节点调度算法通过自动学习机与环境的交互和结果反馈,借助对动作概率更新并经过一定次数的强化学习后,能够得到问题的解。相比现有的节点调度算法,该算法运算规则简单,适用于传感器这类计算资源受限的网络。

1.3 运用贪婪算法对节点调度问题进行求解

运用贪婪算法对有向异构节点调度问题进行求解,主要分为3个步骤:

步骤1 参数初始化,对节点的能量、剩余能量和网络工作时间进行初始化。

步骤2 求解满足条件的节点集合。在每次迭代中,选取节点感知方向时综合考虑节点对监测目标的覆盖情况和节点自身的能量状况,即按照式(7)的贪婪策略进行选择。

$$\text{cost}(d_{i,j}) = B_1 \times \frac{N_{\text{mc}}(d_{i,j})}{N_{\text{uncovered}} + \mu} + B_2 \times \frac{E_{\text{re}}(S_i)}{E(S_i)} \quad (7)$$

其中, B_1 、 B_2 表示权重系数,取值范围均为(0,1)且 $B_1 + B_2 = 1$, $N_{\text{mc}}(d_{i,j})$ 表示在当前的迭代次数还未被覆盖的目标中,感知方向 $d_{i,j}$ 所能覆盖的目标数, $N_{\text{uncovered}}$ 表示还未被覆盖的目标数, $E_{\text{re}}(S_i)$ 表示节点 S_i 的剩余能量, $E(S_i)$ 表示节点 S_i 的初始能量。

步骤3 节点和监测目标状态更新。每次选取符合条件的感知方向后,将该节点的其他感知方向从候选集合中删去。每次迭代结束后,更新节点的能量状态,将能量为0的节点的所有感知方向从候选集合中删去。

贪婪算法的具体步骤如下:

步骤 1 判断当前可用的感知方向集合是否能完全覆盖目标,若能完全覆盖,则执行步骤 2,否则,执行步骤 7。

步骤 2 设置变量 t_w 和集合 E, T, F 并初始化 $t_w = \text{常数}, E = T, F = \text{空}$ 。其中, t_w 为每个集合的工作时间, T 表示全体监测目标集合, E 和 F 是用来记录算法中间结果的临时变量集合,分别存储监测目标和满足条件的感知方向。

步骤 3 判断集合 E 是否为空,若为空,则执行步骤 6,否则,执行步骤 4。

步骤 4 对当前的感知方向集合按照式(7)计算每个感知方向的 $cost$ 值。

步骤 5 选取最大 $cost$ 值所对应的感知方向,将该感知方向加入集合 F 中,同时将该感知方向对应的覆盖目标从集合 E 中删去,并将该感知方向所对应节点的其他感知方向从当前感知方向集合中删去,然后返回步骤 3。

步骤 6 将 $\{F, t_w\}$ 加入集合 $Result_set$ 中,更新 F 集合中感知方向所对应的节点的剩余能量,若剩余能量为 0,则将该感知方向对应的节点的所有感知方向从当前可用的感知方向集合中删去,然后返回步骤 1。

步骤 7 返回结果 $Result_set$ 。

2 仿真结果与性能分析

为分析本文算法的性能,在 Windows 7 64 位和 MATLAB 2013 平台上进行仿真实验。其中,每个实验数据均是 30 次实验结果的平均值。

2.1 仿真参数

假定 $40\text{ m} \times 40\text{ m}$ 区域内均匀分布 120 个监测目标,随机部署 2 种类型的有向传感器节点,为保证一定的节点冗余度,节点数目设为 210(2 种类型的节点数目相同)。节点部分参数设置如表 1 所示。

表 1 有向传感器节点参数设置

类型	感知半径/m	能量/J	感知角度
1	10	1.5	$\pi/2$
2	20	2.0	$2\pi/3$

其他参数设置为:式(7)中 B_1, B_2, μ 的取值分别为 0.50、0.50、0.01。网络寿命定义为传感器节点能完全覆盖监测目标的时间。

2.2 结果分析

2.2.1 网络寿命与集合工作时间 t_w 的关系

为验证网络寿命与集合工作时间之间的关系,将部署节点数分别设置为 50 和 75,将 t_w 分别设置为 0.25 s、0.50 s、0.75 s、1.00 s,图 1 所示为网络寿命随集合工作时间的变化曲线。

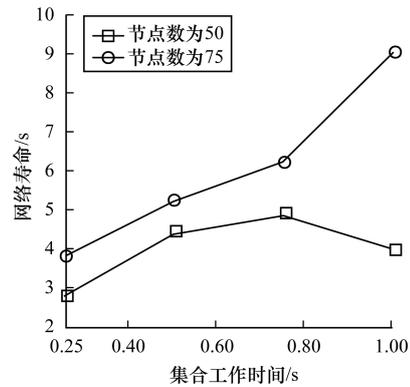


图 1 网络寿命与集合工作时间的关系

从图 1 中可以看出,每个覆盖集合工作时间不同,网络的寿命也不同。原因是每个节点的能量消耗在不同的覆盖集合中,每个集合工作时间不同,消耗的能量也不同,其中较短的集合工作时间导致其能够构建较多的满足条件的覆盖集合,从而使网络的寿命得以延长。此外,在相同的覆盖集合工作时间条件下,节点数目较多的传感器网络寿命较长。

2.2.2 算法运行时间与集合工作时间 t_w 的关系

为验证算法运行时间与集合工作时间之间的关系,将部署节点数分别设置为 50 和 75,将 t_w 分别设置为 0.25 s、0.50 s、0.75 s、1.00 s,图 2 所示为算法运行时间随集合工作时间的变化曲线。

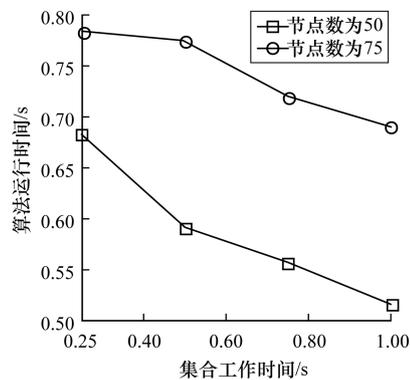


图 2 算法运行时间与集合工作时间的关系

从图 2 中可以看出,随着集合工作时间的增加,算法运行时间减少。原因是集合工作时间增加,即节点在该覆盖集合中消耗的能量增加,导致其构建的符合要求的覆盖集合减少,从而使得算法很快收敛。在相同的集合工作时间条件下,节点数目较多的网络需要较多的时间去构建符合条件的覆盖集合。

2.2.3 网络寿命与节点个数的关系

为比较本文基于自动学习机的节点调度算法和贪婪算法的性能,将节点数目分别设为 50、75、90、100,每个覆盖集合的工作时间设为 0.25 s,即 $t_w = 0.25\text{ s}$,监测目标数设为 15,然后分析网络寿命与节点个数之间的关系。图 3 所示为网络寿命随节点数目的变化曲线。

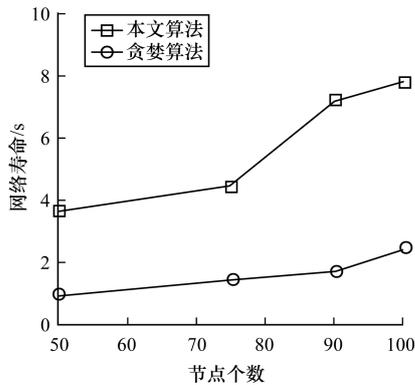


图3 网络寿命与节点个数的关系

从图3中可以看出,节点个数增加时2种算法的网络寿命均延长。但在部署节点数相同的条件下,本文算法的网络寿命明显长于贪婪算法,该结果验证了本文算法的有效性。

2.2.4 网络寿命与监测目标个数的关系

为分析监测目标个数对算法网络寿命的影响,将监测目标个数分别设为10、15、20、25,每个覆盖集合的工作时间设为0.25s,部署节点数设为100,然后分析网络寿命与监测目标个数的关系。图4所示为网络寿命随监测目标个数的变化曲线。

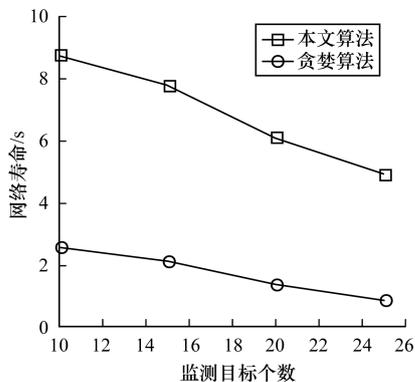


图4 网络寿命与监测目标个数的关系

从图4中可以看出,监测目标个数增加时2种算法的网络寿命均减少。但在监测目标个数相同的情况下,本文算法网络寿命明显长于贪婪算法。

3 结束语

本文提出一种基于学习自动机和贪婪算法的异构有向节点调度算法。学习自动机算法通过与环境的交互来更新节点感知方向选取概率,以达到覆盖集合的优化;贪婪算法根据节点覆盖目标的情况和节点能量情况构建符合要求的感知方向集合,从而达到延长网络寿命的目的。仿真结果表明,与贪婪算法相比,本文算法能有效延长网络寿命。下一步将设计基于连通覆盖的异构有向节点调度算法。

参考文献

- [1] GUVENSAN M A, YAVUZ A G. On coverage issues in directional sensor networks: a survey [J]. *Ad Hoc Networks*, 2011, 9(7): 1238-1255.
- [2] MUSILEK P, KRÖMER P, BARTOŇ T. Review of nature-inspired methods for wake-up scheduling in wireless sensor networks [J]. *Swarm and Evolutionary Computation*, 2015, 25(12): 100-118.
- [3] RENOLD A P, CHANDRAKALA S. Survey on state scheduling-based topology control in unattended wireless sensor networks [J]. *Computers and Electrical Engineering*, 2016, 56(1): 334-349.
- [4] VALERA A C, SOH W S, TAN H P. Survey on wakeup scheduling for environmentally-powered wireless sensor networks [J]. *Computer Communications*, 2014, 52(5): 21-36.
- [5] 温俊, 蒋杰, 窦文华. 公平的有向传感器网络方向优化和节点调度算法 [J]. *软件学报*, 2009, 20(3): 644-659.
- [6] 程卫芳, 廖湘科, 沈昌祥. 有向传感器网络最大覆盖调度算法 [J]. *软件学报*, 2009, 20(4): 975-984.
- [7] HOOSMAND M, SOROUSHMEHR S M R, KHADIVI P, et al. Visual sensor network lifetime maximization by prioritized scheduling of nodes [J]. *Journal of Network and Computer Applications*, 2013, 36(1): 409-419.
- [8] GIL J M, HAN Y H. A target coverage scheduling scheme based on genetic algorithms in directional sensor networks [J]. *Sensors*, 2011, 11(2): 1888-1906.
- [9] KIM Y H, HAN Y H, JEONG Y S, et al. Lifetime maximization considering target coverage and connectivity in directional image/video sensor networks [J]. *Journal of Supercomputing*, 2013, 65(1): 365-382.
- [10] WANG Jian, NIU Changyong, SHEN Ruimin. Priority-based target coverage in directional sensor networks using a genetic algorithm [J]. *Computers and Mathematics with Applications*, 2009, 57(11): 1915-1922.
- [11] 韩崇, 孙力娟, 郭剑. 一种基于网格划分的有向传感网时空覆盖调度算法 [J]. *南京邮电大学学报(自然科学版)*, 2013, 33(5): 82-87.
- [12] GUPTA P, KUMAR P R. The capacity of wireless networks [J]. *IEEE Transactions on Information Theory*, 2000, 46(2): 388-404.
- [13] 王力立, 吴晓蓓, 黄成, 等. 基于目标覆盖的异构有向传感器网络分布式节点调度策略 [J]. *控制与决策*, 2016, 31(12): 2140-2146.
- [14] 杨铮, 吴陈沭, 刘云浩. 位置计算: 无线网络定位与可定位性 [M]. 北京: 清华大学出版社, 2014.
- [15] TORKESTANI J A. Adaptive learning to rank algorithm: learning automata approach [J]. *Decision Support Systems*, 2012, 54(1): 574-583.
- [16] NICOPOLITIDIS P, PAPADIMITRIOU G I. Adaptive wireless networks using learning automata [J]. *Wireless Communications*, 2011, 18(2): 75-81.
- [17] LOTF J J, NAZHAD S H H, HOSSEINZADEH M. Applications of learning automata in wireless sensor networks [J]. *Procedia Technology*, 2012(1): 77-84.