

# 无线传感器网络的元胞自动机自组织算法

樊棠怀<sup>1,2</sup>, 肖贤建<sup>1</sup>, 殷玲玲<sup>1</sup>, 徐立中<sup>1</sup>

(1. 河海大学计算机及信息工程学院, 南京 210098; 2. 南昌工程学院计算机科学与技术系, 南昌 330099)

**摘要:** 提出一种分布式、自适应的无线传感器网络元胞自动机(CA)自组织算法, 将网络中的每个节点映射成 CA 中的元胞, 通过控制节点在不同时间的状态转换(活跃/睡眠), 消除与邻居节点间的频繁通信, 从而降低能源消耗。仿真实验结果表明, 该算法具有较强的目标检测能力。

**关键词:** 无线传感器网络; 元胞自动机; 自组织算法

## Cellular Automata Self-organization Algorithm for Wireless Sensor Network

FAN Tang-huai<sup>1,2</sup>, XIAO Xian-jian<sup>1</sup>, YIN Ling-ling<sup>1</sup>, XU Li-zhong<sup>1</sup>

(1. College of Computer and Information Engineering, Hohai University, Nanjing 210098;

2. Department of Computer Science and Technology, Nanchang Institute of Technology, Nanchang 330099)

**【Abstract】** An adaptive and distributive Cellular Automata(CA) machine self-organization algorithm is proposed in Wireless Sensor Network(WSN). Each node of the network is assumed to be equipped with CA machine, which controls the decision on its state transition (active/standby) at different time, and avoids frequent communication with its neighbors. The power consumption is reduced. Simulation experimental results show this algorithm has stronger target detection capability.

**【Key words】** Wireless Sensor Network(WSN); Cellular Automata(CA); self-organization algorithm

### 1 概述

无线传感器网络具有节点能源受限、节点数量众多, 稠密布置、随机撒布等形态特征, 这对网络和节点的软硬件设计提出挑战。合理设计节点自组织机制能够为在未知环境中运行的网络带来灵活性、适应性和有效性的平衡。通过对无线传感器网络的自组织机制的研究, 能够解决快速响应、可靠性与耗能之间的问题。

基于簇的自组织算法是无线传感器网络节点常见的自组织算法<sup>[1]</sup>。文献[2]提出一种动态簇组织算法, 通过簇首轮换和簇重组, 平均了节点能耗。文献[3]描述一种分布式简单簇组织算法, 在成簇率、簇成员分布、成簇时间和拓扑控制等方面取得较好的结果, 尤其对大规模无线传感器网络, 可以获得较高的成簇率和较快的成簇速度, 有良好的适应能力。但簇组织需要全局信息的介入, 而在一个大规模布署的网络中, 受到诸多因素的制约, 节点都掌握全局信息是不现实的。所以, 根据范围受限的本地信息来进行节点的自组织是个研究方向, 元胞自动机算法就是个较好的解决办法。

### 2 相关问题

元胞自动机算法是种简单而又具有广泛应用前景的算法, 采用一种新思想处理复杂系统的仿真和系统发展的预测。元胞自动机的优点在于能够描写具有局部相互作用的多体系统所表现的集体行为及其时间演化。先将空间分割成由共同特点的元胞组成的规则的点阵, 像围棋组成蜂窝一样。其中, 每个元胞对应着有限组数值, 以描写该元胞的状态。这些数值与一定的时刻相对应, 同步地随着分立的时间步骤( $t=0, 1, \dots$ )按照一定规则演化。所谓“规则”主要是指近距离内

元胞间的相互作用即局域作用。按这种规则, 某元胞在某瞬时的取值决定于该元胞和几个相邻元胞在前一时刻的取值。

如果将 WSN 的海量节点视为元胞自动机中的元胞, 则每个元胞分别对应于一个固定位置的传感器节点, 每个节点具有一定的检测和计算能力。通过设计合理的活跃机制和元胞间的协同, 就有可能以群体智能的方式实现 WSN 自组织。本文构造一种基于元胞自动机的自组织算法(Cellular Automata-based Self-organization, CAS), 将 WSN 自组织问题映射为分布式群体智能优化方法, 通过优化节点的唤醒机制, 实现在消耗较少能量的前提下, 对目标保持较好的检测能力。

### 3 CAS 算法

#### 3.1 元胞自动机模型设计

元胞自动机是个 4 元组  $(L, S, N, f)$ , 其中,  $L$  是规则的网格, 构成这个网格的要素被称为元胞; 将无线传感器网络定义为一个元胞自动机,  $L$  则对应于 WSN 网络, 构成网格的要素即为无线传感器节点;  $S$  是有限的状态集合, 该应用中有 2 个状态: 0-睡眠, 1-活跃;  $N$  是邻居半径的集合, 本文取传感器节点的单跳通信距离;  $f$  为状态转移函数, 即  $S^n \rightarrow S^{n+1}$  的过渡函数。

将无线传感器网络定义为一个元胞自动机, 它具有如下

**基金项目:** 教育部科学技术研究基金资助重点项目(107057); 江西省教育厅科技计划基金资助项目(赣教技字[2007]338号)

**作者简介:** 樊棠怀(1962-), 男, 副教授、博士研究生, 主研方向: 信息获取与处理, 无线传感器网络; 肖贤建, 博士研究生; 殷玲玲, 硕士研究生; 徐立中, 教授、博士生导师

**收稿日期:** 2009-06-02 **E-mail:** fantanghuai@tom.com

特性<sup>[4]</sup>：

- (1)每个元胞的位置是固定的，对应于一个传感器节点；
- (2)节点有活跃/睡眠 2 种状态，用 1/0 表示；
- (3)每个节点都将有机会成为活跃节点；
- (4)每个节点的状态(活跃/睡眠)由上个周期周围节点的状态决定；

(5)某个节点在一个周期(时间步)内是活跃的，则下个周期不可以是活跃的，以减少特定节点的电池能量消耗。

设感知区域为  $D$ ，在  $D$  内均匀撒布  $n$  个元胞，其集合可表示为  $S = \{S_1, S_2, \dots, S_n\}$ ，其中， $S_k$  表示第  $k$  个传感器节点。设节点的空间位置向量为  $L = \{L_1, L_2, \dots, L_n\}$ ，其中， $L_k$  为  $S_k$  的空间位置向量。

每个元胞都具有独立的感知、交流能力(对应于传感器节点的检测、通信能力)，分别定义如下：

#### (1)通信方式

本文考虑传感器节点采用 S-MAC 协议<sup>[5]</sup>，即每个元胞在每个 *Round* 周期按照以下 2 个阶段的方式工作：活跃/睡眠阶段。元胞根据唤醒规则选择自己处于活跃状态或睡眠状态：当处于活跃状态时，则对周围环境进行检测并进行必要的处理；而处于睡眠状态时，元胞将进入睡眠状态以节省能量。

由于多跳通信不仅消耗了过多的能量而且还会造成网络通信延迟及误码率增高等不良后果，因此仅考虑单跳范围内元胞之间的通信协同。

设元胞  $S_k$  的邻居集合  $neighbor(k)$  为

$$neighbor(k) = \{S_j \mid j \text{ 满足 } \|l_j - l_k\| \leq R_c, j = 1, 2, \dots, n\} \quad (1)$$

其中， $R_c$  大小为传感器节点单跳通信距离。

#### (2)检测感知

对于元胞  $S_k$ ，其检测范围为一个半径为  $R_s$  的圆形区域  $V_k^\phi$ ，当目标出现在该圆形区域内时，元胞会以一定的概率检测到它。检测区域的集合  $sensor(i)$  为

$$sensor(i) = \{S_k \mid L_k \in V_k^\phi\} \quad (2)$$

每个元胞应能存储其所有邻居节点的能量值  $E$ 、节点位置  $LOC$ 、当前状态  $G(t)$  等。

设元胞节点  $S_k$  在  $t$  时刻活跃状态标志位为  $G_k(t)$ ：

$$G_k(t) = \begin{cases} 1 & \text{时刻 } S_k \text{ 处于活跃状态} \\ 0 & \text{时刻 } S_k \text{ 处于睡眠状态} \end{cases} \quad (3)$$

$S_k$  在  $t$  时刻唤醒机制的设计通过参数  $W(t)$  表述。设  $S_k$  的唤醒概率为  $w_k(t) \triangleq P\{G_k(t) = 1\}$ ，检测概率为  $Pd_k$  (即当在  $S_k$  的检测范围内出现目标时，以  $Pd_k$  的概率发现目标)，虚警概率为  $Pf_k$  (即使  $S_k$  的检测范围内没有目标出现，仍以  $Pf_k$  的概率错误的宣告发现目标)。设  $S_k$  在  $t$  时刻判定其是否发现目标的标志为  $\phi_k(t)$ ：

$$\phi_k(t) = \begin{cases} 1 & \text{时刻 } S_k \text{ 判定目标出现} \\ 0 & \text{其他} \end{cases} \quad (4)$$

显然，当  $S_k$  的检测范围内存在目标时， $\phi_k(t) = 1$  的发生概率为  $w_k(t) \cdot Pd_k$ ，而当其检测范围内不存在目标时， $\phi_k(t) = 1$  的发生概率为  $w_k Pf_k$ 。

### 3.2 算法设计

当无线传感器网络的感知区域内存在目标时，需有较多的传感器节点能够参与检测，以保证检测效果，而没有目标存在时，为节省能量，则需有较多的节点处于睡眠状态。

WSN 系统中的每个元胞存在 2 个操作对象：演化规则和节点单元状态。

本文定义节点的演化规则如下：若节点在  $t$  时刻为 1 状态，则在  $t+1$  时刻会转换为 0 状态；若节点在  $t$  时刻为 0 状态，通过计算  $W(t)$  的值，超过某个预定的阈值  $W_{th}$  时，则该节点状态转换为 1，并小范围的广播一个 *signal* 信号，通知邻居节点其状态改变了。接收到信号的邻居节点会在  $t+1$  时刻在其存储单元内把该节点的状态值由 0 改为 1。

影响节点取 0/1 这 2 种状态的因素具体表述如下：

#### (1)节点距离检测目标的距离

每个传感器节点的检测范围是有限的，为了提高检测的准确性，希望目标周围的节点能较多的处于活跃状态。因此，离目标的距离越近，节点为 1 的可能性越大；反之越小。

#### (2)上个时间周期本节点的状态

每个节点有 0 或 1 这 2 种可能的状态。若在  $t-1$  时刻为 1，则  $t$  时刻即变为 0。若  $t-1$  时刻为 0，通过计算  $W(t)$  的值， $t$  时刻可能变为 1，或者继续保持为 0。

#### (3)上个时间周期邻居节点的状态

邻居节点距离本节点越近，其状态对本节点的影响越大；同时，希望活跃节点的分布是均匀的，在减少能耗的同时又要有较好的检测效果。因此在  $t-1$  时刻，邻居节点为 1 的数目越多，本节点为 1 的概率越小；邻居节点为 0 的节点数目越多，本节点为 1 的概率越大；另外，距离目标越近的邻居节点，它的状态对本节点的影响也会越大。

#### (4)本节点的剩余能量

WSN 节点的能量一般是有限的，但必须要有足够的处于活跃状态的节点来组成网络的主干，所以，能量消耗较多的节点应该让其尽可能多的处于睡眠状态，即节点的剩余能量越多，为 1 的概率越大，反之越小。

当节点的能量为初始值的 1% 时，设该节点的状态永久为 0，即节点失效。

因此，本文构造元胞自动机元胞  $S_k$  的活跃权值计算式为

$$W_k(t) = \bar{G} \left( \frac{3}{e^{0.1D}} + 10 \left( \frac{1}{e^{0.1d_1}} + \dots + \frac{1}{e^{0.1d_n}} \right) + 3E \right) \quad (5)$$

其中， $G$  为元胞在  $t-1$  时刻的状态； $D$  为元胞离目标的距离； $d$  是  $t-1$  时刻状态为 0 的邻居距离本节点的距离； $E$  为  $t-1$  时刻节点的剩余能量。

### 3.3 性能指标设计

为说明算法的性能，设计性能指标如下：

#### (1)节点唤醒概率 $w_k(t)$

定义节点的唤醒概率为

$$w_k(t) = n_w(t) / N \quad (6)$$

其中， $n_w(t)$  表示  $t$  时刻所有处于活跃状态的元胞总数，实际上是能量消耗指标， $n_w(t)$  越大，处于活跃态的节点越多，总的能量消耗也越多； $N$  为整个无线传感器网络元胞的数目。

#### (2)节点利用率 $n_s(t)$

$t$  时刻节点利用率  $n_s(t)$  为在  $t$  时刻检测到目标的节点总数与处于活跃态的节点总数之比，由下式计算：

$$n_s(t) = n_d(t) / n_w(t) \quad (7)$$

其中， $n_d(t)$  表示  $t$  时刻检测到目标的元胞节点总数； $n_s(t)$  实际上表示的是资源的有效利用率， $n_s(t)$  越大说明算法效率越高，合理提高资源利用率是算法设计的目标，也是衡量算法性能的指标之一。

## 4 仿真分析

### 4.1 仿真环境

感知区域大小为  $100\text{ m} \times 100\text{ m}$ ，其中随机撒布 150 个性能指标相同的传感器节点，如图 1 所示。目标以  $v=3\text{ m/s}$  作匀速直线运动，起点为  $[0,0]$ ，终点为  $[100,80]$ ，运动轨迹在图中表示为实线，可以看出，该条轨迹展现了运动目标从进入感知区域到离开的完整过程。

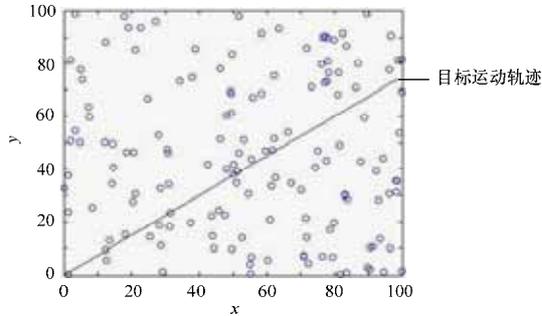


图 1 仿真场景

实验参数取值如表 1 所示。

表 1 仿真实验参数

参数	意义	取值
$R_s$	有效检测半径	15 m
$R_c$	有效通信半径	30 m
$P_D$	检测概率	0.9
$P_F$	虚警概率	0.05
$W_{th}$	活跃阈值	0.3
Round	工作周期	1 s

### 4.2 仿真结果分析

图 2~图 5 分别是 1, 8, 16, 32 时刻节点的活跃情况。

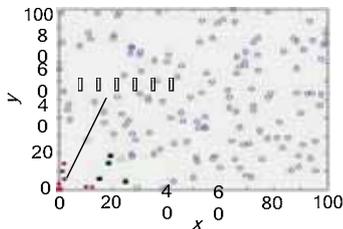


图 2 时刻 1 时的节点唤醒情况

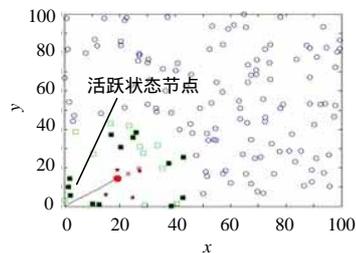


图 3 时刻 8 时的节点唤醒情况

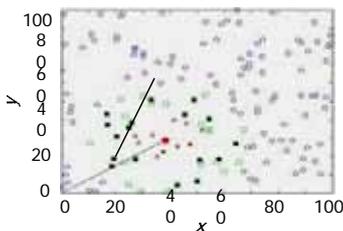


图 4 时刻 16 时的节点唤醒情况

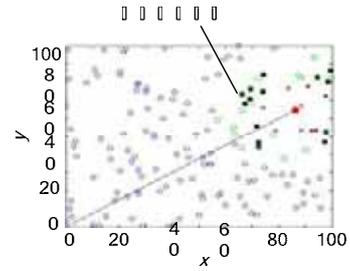


图 5 时刻 36 时的节点唤醒情况

从图 2~图 5 可以看出，节点的唤醒情况可以根据目标自适应，距离目标越近，处于活跃状态的节点密度越大，且能均匀分布。距离目标越远，处于活跃状态的节点密度越小。表明在目标附近的区域能有较高的检测概率，在其他区域检测概率保持在较低的水平。随着目标的移动，沿途的节点随着目标的状态而自适应地变化。这说明本文算法在目标观测范围内的较多节点被唤醒参与感知，而其余节点则多处于睡眠状态。目标周围活跃的节点越多，节点的检测概率越大，该单元内目标被检测到的概率也越大，说明检测效果越好，对感知区域有较好的检测效果是无线传感器网络实现功能的保障。本文算法使群体智能得到一定程度的体现。

将本文算法与 RA 算法进行比较，其中，算法 A 表示所有节点的唤醒概率  $w=1$ ；算法 B 表示所有节点的唤醒概率  $w=0.49$ ；算法 C 为本文算法；算法 D 表示所有节点的唤醒概率  $w=w_{\min}=0.035$ 。4 种算法中其余参数取值均按表 1 选取。实验结果如图 6~图 8 所示。可以看出，算法 A 中所有节点都按照概率 1 被唤醒，显然消耗的能量最多。算法 B 中所有节点的唤醒概率取为 0.49，表明算法 B 被唤醒节点数较少，能量消耗也较少。算法 C 中节点按照本文算法自适应的分布活跃节点。算法 D 中所有节点都按照最小概率  $w_{\min}$  被唤醒，算法 D 只有最少的节点被唤醒，能量消耗最少。

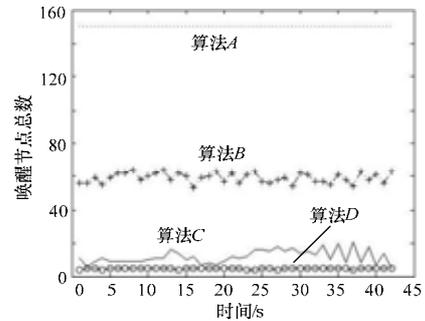


图 6 算法唤醒节点总数比较

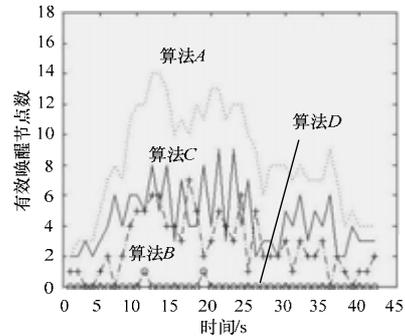


图 7 有效唤醒节点数比较

(下转第 32 页)