

文章编号:1001-2486(2011)04-0073-07

基于非合作博弈的栅栏信息覆盖虚拟节点组合算法*

班冬松,杨威,宋磊,蒋杰,窦文华
(国防科技大学 计算机学院,湖南长沙 410073)

摘要:邻近节点通过协作感知和信息融合组成虚拟节点,能够增加信息覆盖栅栏长度。基于非合作博弈模型,设计了一种完全分布式的虚拟节点组合算法 DVSF 算法。DVSF 算法面向栅栏信息覆盖应用,以尽量增加整个网络的信息覆盖栅栏长度为目标,将合适的邻近节点组合成虚拟节点,提升了整个网络的栅栏信息覆盖性能。仿真实验验证了 DVSF 算法的有效性和收敛性。

关键词:无线传感器网络;栅栏信息覆盖;虚拟节点;非合作博弈
中图分类号:TP393 **文献标识码:**A

Virtual Sensor Formation with Non-cooperative Game for Barrier Information Coverage in WSN

BAN Dong-song, YANG Wei, SONG Lei, JIANG Jie, DOU Wen-hua
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Neighboring sensors can collaborate with each other to form a virtual sensor based on cooperative sensing and data fusion, which may increase the length of information barrier. A distributed algorithm DVSF for virtual sensor formation based on non-cooperative games was devised. In order to increase the total length of information barrier of the networks, the DVSF chooses appropriate neighboring sensors to form virtual sensors, and it improves the performance of barrier information coverage of the networks. Simulations of the approach demonstrate the effectiveness and the convergence of DVSF.

Key words: wireless sensor networks; barrier information coverage; virtual sensor; non-cooperative game

无线传感器网络的栅栏覆盖^[1]是指通过相邻节点感知区域的相互交叠,在监控区域中形成传感器感知栅栏,监控穿越监控区域的移动目标。栅栏覆盖不必覆盖监控区域中的所有位置点,因而相对于区域覆盖策略,栅栏覆盖所需节点数目较少,更适用于检测移动目标是否穿越监控区域的应用场景。因此,近年来栅栏覆盖受到广泛关注^[1-6]。邻近的节点通过协作感知和信息融合,组合成一个虚拟节点,联合做出探测到目标的决策,这被称为信息覆盖^[9]。信息覆盖可以有效提高覆盖性能^[6-8],减少部署节点的数量。近年来,无线传感器网络栅栏覆盖领域已有许多研究工作,但大多数研究工作都是基于简单的二元圆形感知模型。文献[6]首次研究了栅栏信息覆盖(barrier information coverage),即基于信息覆盖模型的栅栏覆盖,针对虚拟节点成员数量 $k \leq 2$ 的情况,分析了虚拟节点栅栏投影长度的近似下界,并针对 2 个传感器节点($k \leq 2$)进行虚拟组合的情况

进行了研究,提出了寻找最少数量节点以栅栏信息覆盖监控区域的集中式算法。文献[7]对区域信息覆盖中网络中的传感器节点如何有效地组合成虚拟节点,增加覆盖面积,提出了一种虚拟节点组合的集中式算法。无线传感器网络节点能量有限、规模巨大,为了面向大规模无线传感器网络的应用,迫切需要低通信开销、低计算开销的分布式解决方法。据我们所知,目前尚没有面向栅栏信息覆盖,使网络中的节点能够有效地组合成虚拟节点以增加信息覆盖栅栏长度的分布式算法。

本文首先针对网络中的节点如何组合才能尽量增加整个网络的信息覆盖栅栏长度,提出了虚拟节点组合问题。然后基于非合作博弈模型,设计了一种完全分布式的虚拟节点组合算法 DVSF 算法,该算法通过将邻近节点组合成虚拟节点,有效地增加了网络的信息覆盖栅栏长度。最后利用详细仿真实验验证了该算法的有效性和收敛性。

* 收稿日期:2010-11-01
基金项目:国家自然科学基金项目(60603061,60903223)
作者简介:班冬松(1982—),男,博士生。

1 模型与问题描述

假设 N 个传感器节点随机部署于监控区域中, 节点集合记为 $S = \{s_1, s_2, \dots, s_N\}$ 。监控区域 \mathcal{B} 为矩形, 区域长为 l , 宽为 w 。节点能够获知自身地理位置信息, 网络是连通的。

目标发射或引起的信号强度随着距离的增加而递减。如果目标在点 t , 其被节点 s_i 探测到的信号强度如下^[6]:

$$w_i(t) = \begin{cases} \Omega & d_i < d_0 \\ \frac{\Omega}{(d(s_i, t)/d_0)^\alpha} & d_i \geq d_0 \end{cases} \quad (1)$$

式(1)中 Ω 是目标发射的信号强度, d_0 为一个很小常数, α 是一个已知的衰减系数。 $d(s_i, t)$ 为目标 t 到传感器节点 s_i 的距离。 本文将以 $\alpha = 1$ 的情况为例进行算法设计与分析, 本文设计的算法适用于任何衰减系数。

在每轮感知周期内, 每个活跃的传感器节点 s_i 搜集到的信号强度为 x_i 。 依据假设检验模型有下式:

$$\begin{cases} H_0: x_i = n \\ H_1: x_i = w_i(t) + n \end{cases} \quad (2)$$

式(2)中 H_0 表示目标不存在, H_1 则表示存在; $w_i(t)$ 是接收到的目标信号的强度, n 是背景噪声。 本文假设背景噪声服从正态分布, 独立于所有的传感器节点。 $\Upsilon(n)$ 代表背景噪声的累积分布函数。

一个传感器节点可以只依靠自身探测数据, 或者与邻居节点协作, 融合它们的感知数据, 最终做出探测决定。 协作感知有 2 种类型的融合机制: 数值融合和决策融合。 数值融合中融合节点融合邻居节点的感知能量数据, 决策融合中每个邻居节点首先独立判断目标是否存在, 然后将二进制的判断结果发送到融合节点进行融合。 本文使用基于数值融合的协作感知机制。

定义 1 虚拟节点 (virtual sensor): 一个虚拟节点 v_i , 由 k 个传感器节点 s_{i_1}, \dots, s_{i_k} 组成。 v_i 基于 k 个节点的感知数据 x_{i_1}, \dots, x_{i_k} 做出决策, k 是虚拟节点的协作度。 当感知数据之和大于等于决策阈值 T_k 时, 宣布目标被检测到:

$$\begin{cases} H_0: \sum_{m=1}^k x_{im} < T_k \\ H_1: \sum_{m=1}^k x_{im} \geq T_k \end{cases} \quad (3)$$

当 $k = 1$ 时, 一个虚拟节点等价于一个物理

传感器节点。 对点 t 来说, 一个虚拟节点 v_i 的信号感知强度是 k 个组成节点的感知强度之和。 一个未与其他节点 s_i 合作的节点被称为独立节点 (independent sensor)。

本文使用概率感知模型: 基于预先设定的决策阈值 T , 目标被一个虚拟节点或者独立节点以一定概率检测到, 检测概率 P_d 随着感知到的目标信号强度的增加而增加。 当节点 (虚拟节点或独立节点) 做出局部决策后, 它将决策 (1 个二进制值) 发送到 sink 节点。 在 sink 节点, 利用“或”规则进行融合判断, 即只要有一个虚拟节点或独立节点报告检测到目标, 系统认为检测到目标。 整个系统的误警率为:

$$P_{FD} = 1 - \prod_{i=1}^M (1 - p_i) \quad (4)$$

p_i 是一个节点的误警率, M 是节点总数 (包括虚拟节点和独立节点)。

定义 2 点信息覆盖: 给定系统最大误警率 P_{FD}^{\max} , 最小检测率 P_D^{\min} 。 点 p 被信息覆盖, 是指当点 p 存在目标 t 时, t 能够被至少一个节点以概率 P_d 检测到, $P_d \geq P_D^{\min}$, 并且系统的误警率 $P_F \leq P_{FD}^{\max}$ 。

定义 3 路径信息覆盖: 穿越监控区域 \mathcal{B} 宽度方向的路径 \mathcal{C} 被信息覆盖, 当且仅当 $\exists p \in \mathcal{C}$, p 被信息覆盖。

定义 4 栅栏信息覆盖: 通过监控区域 \mathcal{B} 的任意垂直穿越路径均能被部署的无线传感器网络信息覆盖, 则称 \mathcal{B} 被栅栏信息覆盖。

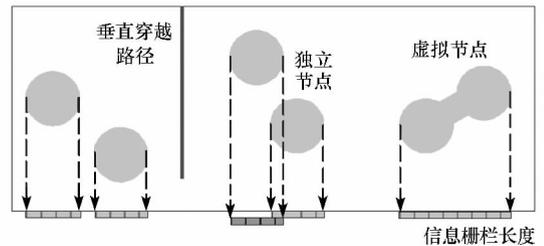


图 1 栅栏信息覆盖图示

Fig. 1 Illustration of barrier information coverage

如图 1 所示, 我们将一个节点在监控区域长度方向上的投影长度定义为该节点能够产生的信息覆盖栅栏长度。 监控区域 \mathcal{B} 被栅栏信息覆盖, 说明网络中的传感器节点在长度方向上的总的投影覆盖了整个监控区域。 下面介绍如何计算信息覆盖栅栏长度^[6]。

(1) 独立节点的信息覆盖栅栏长度

点 t 被独立节点 s_i 信息覆盖, 当且仅当 s_i 感知到的点 t 信号强度大于等于一个特定阈值 T_1 ,

由式(2)可得感知数据 $x_i = w(t) + n \geq T_1$, 检测率 $P_d = P(w(t) + n) \geq P_d^{\min}$, 代入背景噪声函数可得 $1 - \Upsilon(T_1 - w(t)) \geq P_d^{\min} \Rightarrow w(t) \geq T_1 - \Upsilon^{-1}(1 - P_d^{\min})$, 将 $T_1 - \Upsilon^{-1}(1 - P_d^{\min})$ 记为 I_1 。由式(1)可得独立节点 s_i 的信息覆盖区域 $A_i = \{t | d(s_i, t)\} \leq d_0 (\frac{\Omega}{I_1})^{\frac{1}{\alpha}}$, 覆盖区域为圆形, 半径 $R_i = d_0 (\frac{\Omega}{I_1})^{\frac{1}{\alpha}}$, 因此独立节点的信息覆盖栅栏长度为

$$BL(s_i, s_i) = 2R_i \quad (5)$$

(2) 协作度为 2 的虚拟节点的信息覆盖栅栏长度

文献[6]分析了协作度为 2 时虚拟节点的信息覆盖栅栏长度, 如式(6)和式(7)所示。

当信息覆盖区域 \mathcal{A} 连通时:

$$BL(s_1, s_2) = 2R_{\text{sub}} + L |\cos \theta| \quad (6)$$

当 A 不连通时:

$$BL(s_1, s_2)$$

$$= \begin{cases} 2R_{\text{sub}} + L(s_1, s_2) |\cos \theta| & |\cos \theta| \leq \frac{2R_{\text{sub}}}{L} \\ 2R_{\text{sub}} + 2\max\{Q \cos \theta, R_{\text{sub}}\} & \text{其他} \end{cases} \quad (7)$$

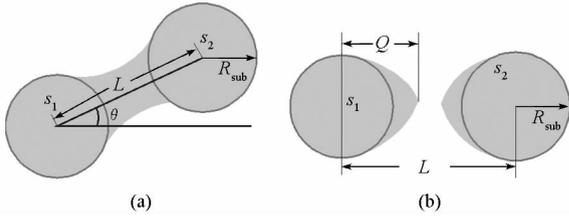


图2 (a) A 连通图示; (b) A 不连通的图示
Fig.2(a) A is connected; (b) A is not connected

由图 2(a) 和图 2(b) 可以看出, 2 个节点组合成一个新的虚拟节点, 通过协作感知增加了信息覆盖栅栏的长度。如果网络中尽量多的节点通过组成虚拟节点的方式, 增加信息覆盖栅栏长度, 则我们能用尽量少的节点实现监控区域的栅栏覆盖。因此本文将研究网络中的节点如何组合才能尽量增加整个网络的信息栅栏覆盖长度。本文将对虚拟节点协作度最多为 2 的情况进行分析研究。协作度大于 2 的情况, 由于更加复杂, 将在后续工作中研究。下面定义虚拟节点组合问题。

对任意的节点 $s_i \in S$, 如果 s_i 选择与 s_j 合作组成虚拟节点, 则节点的选 $c_i = s_j$ 择, 信息栅栏覆盖长度表示为 $BL(s_i, s_j)$ 。特别地, 如果 s_i 为独立节点, 则 $c_i = s_i$ 。令 $\mathbf{C} = S^N, \forall c_k \in \mathbf{C}, c_k = (c_{k_1}, c_{k_2}, \dots, c_{k_N})$ 代表各节点的一组选择。

定义 5 虚拟节点组合问题: 给定 N 个节点, 各节点如何无冲突地选择其他邻居节点组合成虚

拟节点, 使得网络中信息覆盖栅栏长度之和最大。该问题形式化描述为:

$$\text{Max } BL_{\text{total}}(\mathbf{c}_k) = \sum_{i=1}^N BL(s_i, c_{k_i}) \quad (8)$$

$$\text{Subject to } \mathbf{c}_k \in \mathbf{C} \quad (9)$$

$$c_{k_i} \neq c_{k_j}, \text{ if } s_i \neq s_j, \forall s_i, s_j \in S \quad (10)$$

式(9)表示 \mathbf{c}_k 为网络中所有节点的一组选择, 式(10)限制任意 2 个传感器的选择不能相同, 即 2 个不同的传感器节点不能同时选择同一个节点进行合作。

2 DVSF(Distributed Virtual Sensor Formation) 算法

本节利用非合作博弈论对虚拟节点组合问题进行分析, 设计了分布式虚拟节点组合算法 DVSF 算法。博弈论具有天然的分布式特性, 各节点基于自身利益最大化的目标, 通过互相博弈, 选择与能够形成最大信息栅栏长度的周边的邻居节点进行组合, 最终达到均衡状态。下面首先建立虚拟节点组合博弈模型。

2.1 虚拟节点组合博弈

局中人集合: 每个传感器节点 s_i 为一个局中人, 所有的传感器节点组成局中人集合 S 。策略集: 传感器节点 s_i 有 k_i 个邻居节点, s_i 有 $k_i + 1$ 种策略选择, 即 s_i 可以独立或 s_i 可以和 k_i 个邻居节点中的任意一个进行合作; s_i 的策略集是有限的, 记为 $C_i = \{c_{i_1}, c_{i_2}, \dots, c_{i_{k_i+1}}\}$ 。该博弈的一个局势为 $\mathbf{c} = (c_1, c_2, \dots, c_N) \in \prod_{i \in N} C_i$ 。

对于每个局势 \mathbf{c} , 局中人 s_i 得到的支付为 $P_i(\mathbf{c}) = P_i(c_1, c_2, \dots, c_N), i = 1, 2, \dots, n$ 。节点 s_i 的支付函数 P_i 定义为在一个局势 \mathbf{c} 下, s_i 能够获得的有效信息栅栏长度。由于节点 s_i 只可能与其邻居节点 s_j 中的某个节点组成虚拟节点, s_i 的支付函数只与其邻居节点 $N(i)$ 的策略选择相关, 更准确地说是与邻居节点中某个节点 s_j 的策略选择相关。支付函数定义如下:

若

$$c_i = s_i, P_i(\mathbf{c}) = P_i(c_i) = BL(s_i, s_i) = 2R_i \quad (11)$$

$$c_i = s_j, s_j \in N(i), \text{ 则有 } P_i(\mathbf{c}) = P_i(c_i, c_j)$$

$$= \begin{cases} \max\{\frac{BL(s_i, s_j)}{2}, 2R_l\} \\ c_j = s_j \text{ or } c_j = s_i \\ \frac{BL(s_i, s_j)}{2} \\ c_j = s_k, s_k \neq s_i, BL(s_i, s_j) > BL(s_j, s_k) \\ 2R_l \\ c_j = s_k, s_k \neq s_i, BL(s_i, s_j) \leq BL(s_j, s_k) \end{cases} \quad (12)$$

如果选择节点 s_i 作为独立节点,其能够获得的有效信息栅栏长度为 $2R_l$ 。如果 s_i 选择与某个邻居节点 s_j 组合:若 s_j 也选择 s_i 合作或者选择作为独立节点($c_j = s_j$ or $c_j = s_i$),则 s_i 与 s_j 的策略选择没有冲突,二者可以组合,有效信息栅栏长度为 $\frac{BL(s_i, s_j)}{2}$ 。若 s_j 已经与某个不同于 s_i 的节点 s_k 组合($c_j = s_k, s_k \neq s_i$),则 s_j 必须终止与 s_k 的组合,才能够与 s_i 进行组合。如果 s_i 与 s_j 的组合不能带来比原有组合更长的信息栅栏,则这个合作不可取, s_i 能获得的有效信息栅栏长度仍为 $2R_l$ 。否则, s_i 可以与 s_j 组合,获得的有效信息栅栏长度为 $\frac{BL(s_i, s_j)}{2}$ 。

假设每个节点是理性的,每个节点最大化自己的支付函数。传感器节点集合 S ,策略集 $\{C_i\}$,支付函数集 $\{P_i\}$,组成一个非合作博弈 $\Gamma = (S, \{C_i\}, \{P_i\})$,我们将此博弈称为虚拟节点组合博弈。

为了讨论问题方便,引进记号 $c \parallel t_i = (c_1, \dots, c_{i-1}, t_i, c_{i+1}, \dots, c_n)$,其中 $t_i \in C_i$ 。 $c \parallel t_i$ 表示局中人 s_i 把局势 c 中自己的策略 c_i 换成 t_i ,其他局中人策略保持不变,得到的新局势。

定义 6 Nash 均衡局势^[10]:设 c^* 是非合作 n 人博弈 $\Gamma = (N, \{C_i\}, \{P_i\})$ 的一个局势,如果 $P_i(c^* \parallel c_i) \leq P_i(c^*), \forall i \in N, \forall c_i \in C_i$,则称 c^* 为 Γ 的一个 Nash 均衡局势。

我们引入后悔值的概念,后悔值 $R_i(c \parallel c_i)$ 为节点 s_i 改变策略能够获得的最大获益与当前策略 c_i 的获益之间的差值。令 $c_{N(i)}$ 表示节点 s_i 及其邻居节点的策略的选择,后悔值定义如下:

$$R_i(c \parallel c_i) = \max_{c'_i \in C_i} (P_i(c \parallel c'_i) - P_i(c \parallel c_i)) = \max_{c'_i \in C_i} (P_i(c_{N(i)} \parallel c'_i) - P_i(c_{N(i)} \parallel c_i)) \quad (13)$$

在虚拟节点组合博弈中,每个节点的策略选择和支付函数只与邻居节点相关,因此节点 s_i 在

计算后悔值时只需考虑邻居节点的策略选择。

定理 1 对虚拟节点组合博弈 Γ ,若 $\forall s_i \in S, R_i = 0$,则 Γ 形成一个 Nash 均衡局势,反之亦然。

证明:若 $\forall s_i \in S, R_i = 0$,令此时 Γ 形成的局势为 $c = \{c_1, \dots, c_i, \dots, c_N\}$,则根据后悔值定义有 $\forall s_i \in S, \forall c'_i \in C_i, P_i(c \parallel c'_i) - P_i(c \parallel c_i) \leq 0$,即 $P_i(c \parallel c'_i) \leq P_i(c \parallel c_i) = P_i(c)$,根据 Nash 均衡定义, c 为 Nash 均衡局势 c^* 。反之,若 Γ 形成一个 Nash 均衡局势 $c^* = \{c_1, \dots, c_i, \dots, c_N\}$,则有 $\forall s_i \in S, \forall c'_i \in C_i, P_i(c^* \parallel c'_i) \leq P_i(c^*)$,推得 $P_i(c^* \parallel c'_i) - P_i(c^*) = P_i(c^* \parallel c'_i) - P_i(c^* \parallel c_i) \leq 0$,即 $R_i = 0$ 。得证。 □

2.2 DVSF 算法

对于任意一个局势 $c, \forall s_i \in S$,如果 s_i 的后悔值 $R_i > 0$,则节点 s_i 需要不断改变自己的策略,以使 $R_i = 0$ 。当所有节点后悔值为 0 时,根据定理 1,整个网络形成 Nash 均衡局势,网络处于稳定状态,各节点的策略不再变化。基于以上分析,我们提出了分布式的虚拟节点组合算法 DVSF 算法。假设算法执行前,所有节点通过通信获知邻居信息,包括邻居的位置与策略等。表 1 是 DVSF 算法的伪码描述。

DVSF 算法是完全分布式算法,各个节点通过与周围邻居节点交换信息,迭代运行该算法。DVSF 算法只针对活跃节点进行组合,休眠的邻居节点不进行组合,因为休眠节点没有监控能力。每轮迭代中,节点 s_i 首先计算自己的后悔值 R_i ,并且广播给邻居节点。每个节点都向邻居广播后悔值,则节点 s_i 通过信息交换可以获知邻居节点的后悔值。如果 s_i 的后悔值 $R_i = 0$,说明 s_i 满意其当前的策略,不会改变策略。如果后悔值 $R_i > 0$,则 s_i 需要更改策略。每个节点都是自私的,都想最大化自己的支付函数。但节点的策略不能同时更改,否则网络会出现冲突状态,比如 2 个节点同时选择同一个节点进行组合。本文设计了一种策略更改规则,让每个节点按照一定次序进行更改,保证网络的稳定性。该规则为:(1)如果 R_i 在所有邻居节点的后悔值中最大,则将策略更改为当前认为的最优策略 $c'_i = c_i^*$,并且通知邻居节点 s_k ,令 s_k 选择 s_i 构成组合。(2)如果 R_i 小于某个邻居节点 s_j 的后悔值 R_j ,则 s_j 具有优先更改权。如果 s_j 的最优选择为 s_i ,则 s_i 的策略选择更改为 s_j ;否则在本轮迭代中 s_i 保持策略不变。然后将

s_j 从邻居节点中删除。因为 s_j 已经形成了组合, 计算后悔值时不能再包含了。(3) 如果 $R_i = R_j$, 则根据节点 ID 号大小来决定策略的优先更改权。

表1 DVSF 算法伪码描述

Tab.1 DVSF algorithm

DVSF	Algorithm (node s_i)
	Foreach iteration t : ($t = 1, 2, \dots$)
1	c_i^{t-1} : the strategy of s_i in iteration $t-1$ $c_{N(i)}^{t-1}$: the strategies of s_i and its neighbors in iteration $t-1$
2	Start iteration t
3	Calculate regret $R_i (c_{N(i)}^{t-1} \parallel c_i^{t-1})$, $N^t(i)$ is neighbor set of s_i in iteration t
	Calculate best strategy
4	$c_i^* = \arg \max_{c_i \in C_i} (P_i(c_{N(i)} \parallel c_i^*) - P_i(c_{N(i)} \parallel c_i))$
5	Sends $R_i(c_{N(i)}^{t-1} \parallel c_i^{t-1})$ to its neighbors $N^t(i)$
6	if ($R_i(c_{N(i)}^{t-1} \parallel c_i^{t-1}) = 0$)
7	$c_i^t = c_i^{t-1}$
8	else
	Calculate s_j with maximum regret value in neighbor set $N^t(i)$: $s_j = \arg \max_{s_k \in N^t(i)} R_k$
9	/* Updating Strategy */
10	if $R_i > R_j$ then
11	$c_i^t = c_i^*$, $s_k = c_i^*$, $c_k^t = s_i$, $N^{t+1}(i) = N^t(i)$ endif
12	if ($R_i = R_j$)
13	Case1($i > j$): goto step 17
14	Case2($i < j$):
15	$c_i^t = c_i^*$, $s_k = c_i^*$, $c_k^t = s_i$, $N^{t+1}(i) = N^t(i)$
16	endif
17	if ($R_i < R_j$)
18	if ($c_j^* = s_i$)
19	$c_i^t = s_j$
20	else
21	$c_i^t = c_i^{t-1}$
22	endif
23	$N^{t+1}(i) = N^t(i) - s_j$
24	endif
25	Sends c_i^t to its neighbors $N^{t+1}(i)$
26	End iteration t

定理2 DVSF 算法能收敛到 Nash 均衡局势。

证明 令 c^t 为所有节点在第 t 轮迭代时的策略。假设 c^t 不是 Nash 均衡局势, 则 $\exists s_i \in S$, 使得 $R_i > 0$ 。根据 DVSF 算法, 在此轮迭代中, s_i 或者其邻居节点中一定有一个节点 s_k 将改变策略, 使得 $R_k = 0$ 。这将令网络的总的信息栅栏长度增加 R_k 。令 Φ 为整个网络的信息栅栏长度之和, 则在第 t 轮迭代中, Φ 的增量如下:

$$\Delta\Phi = \Phi(c^{t+1}) - \Phi(c^t) = \sum_{s_i \neq s_i^{t+1}, s_i \in S} R_i > 0$$

因此每轮迭代, 网络的信息栅栏长度之和是严格递增的。令 $\Phi^*(c^*)$ 在 Nash 均衡局势 c^* 下信息栅栏长度之和。由于传感器数量有限, 部署区域面积有限, 所以 $\Phi^* < \infty$ 。由于每轮迭代, Φ 是严格递增的, 因此 Φ 最终会达到 Φ^* , DVSF 算法将收敛。当算法收敛时, Φ 达到 Φ^* , 此时所有节点的后悔值均为 0。根据定理 1, 当前的策略 c^* 为一个 Nash 均衡局势。

定理 2 表明 DVSF 算法是收敛的。算法的收敛时间即迭代的轮数取决于邻居节点数量等因素。在实际部署中, 为确保算法收敛, 可以在部署前通过实验评估需要设置的迭代轮数 T_{\max} 。

每轮迭代中, 计算开销主要集中在后悔值的计算上, 令 M 为节点的最大邻居数量, 则每轮迭代中算法总的计算复杂度为 $O(M)$ 。DVSF 算法的计算复杂度只与邻居节点数量相关, 算法的计算开销很小。每次迭代, 每个节点首先广播自身的后悔值到邻居节点, 通信开销为 $O(N)$ 。然后改变了策略的节点广播新的策略到邻居节点, 最大通信开销为 $O(N)$ 。则每轮迭代的总的通信开销为 $O(N)$ 。算法迭代 k 轮终止, 则算法的总的通信开销为 $O(kN)$ 。后续实验表明, 迭代次数 k 远小于节点数量 N , 因此算法的通信开销较小。

3 实验仿真

本文使用 Matlab 评估 DVSF 算法性能。实验参数设置: 传感器节点按照 Poisson 分布随机部署于矩形区域中, 区域长为 20unit, 宽为 2unit。系统最大误警率 $P_{FD}^{\max} = 0.05$, 一个独立节点或虚拟节点的最小目标检测率 $P_D^{\min} = 0.9$ 。目标发射的信号功率 $\Omega = 100\text{mW}$, 常数距离 $d_0 = 0.01\text{unit}$ 。假设背景噪声服从均值为 0, 方差为 $\frac{\sqrt{2}}{2}\text{mW}$ 的高斯分布, 信号衰减系数 $\alpha = 1$ 。

3.1 信息覆盖栅栏长度

本文面向栅栏信息覆盖应用,首次基于非合作博弈理论提出了一种完全分布式的虚拟节点组合算法 DVSF,目前尚无可直接比较的分布式解决方法。为了评估 DVSF 算法的性能,本文设计了一种寻找虚拟节点组合的集中式的贪婪(Greedy)算法。Greedy 算法基于文献[6]提出的 BICS 算法,主要思想为:扫描所有节点,(1)如果节点 s_i 为独立节点,则在 s_i 的邻居节点中寻找与 s_i 组合,能得到最大信息栅栏长度的节点 s_j ,并且 $BL(s_i, s_j) > 4 \times R_f$,则令 s_j 与 s_i 组合为一个虚拟节点。如果这样的 s_j 不存在,则 s_i 为独立节点。(2)如果 s_i 已是组合节点,则跳过。图 3 比较了应用分布式 DVSF 算法、Greedy 算法以及 All Independent 情况下的信息栅栏长度,all Independent 表示不应用虚拟节点组合算法。DVSF 算法显著增加了网络的信息覆盖栅栏长度,而且随着部署节点数量增加,增加幅度增大:在部署节点数量为 200 时, DVSF 算法产生的信息覆盖栅栏长度为 141.9, All Independent 为 118。与 All Independent 相比, DVSF 算法使得网络信息覆盖栅栏总长度增加幅度约 20%。在部署节点数量为 1000 时,与 All Independent 相比, DVSF 算法增加幅度达 30%。从图 3 可以看出,在节点部署密度较低时, DVSF 算法与集中式的 Greedy 算法性能接近,并且 DVSF 算法具有完全分布式特性,适合应用于节点能量有限的大规模无线传感器网络。

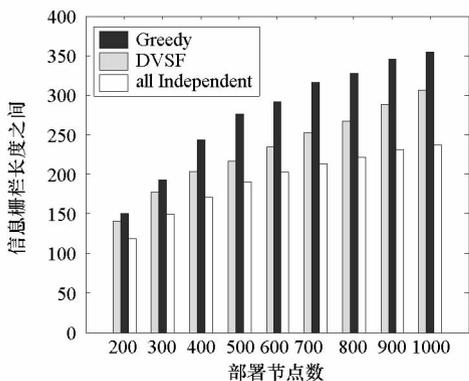


图 3 信息栅栏长度之和与部署节点数关系
Fig.3 Total length of barrier information coverage vs. deployed sensors

Greedy 算法的计算时间复杂度为 $O(N)$ 。DVSF 算法中每个节点的时间复杂度为 $O(tM)$,其中 M 为一个节点的邻居节点的最大数量, t 为算法收敛时的最大迭代次数。图 4 显示了 Greedy 算法与 DVSF 算法的时间复杂度的比较情况,从图 4

可以看出,在保持部署密度不变的情况下,随着网络节点规模的增加, Greedy 算法的计算时间复杂度大幅增加,而 DVSF 算法基本保持不变,适合于能量有限的无线传感器节点。

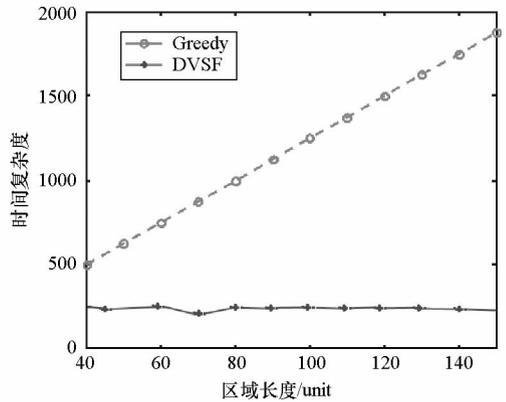


图 4 时间复杂度与网络规模关系
Fig.4 computation complexity vs. region length

图 5 显示了在保持部署的节点数量 400 不变,区域面积扩大时,平均每个节点能够获得的信息栅栏长度的变化情况。随着区域长度增加,节点部署密度降低。从图 5 可以看出,初始时,平均每个节点的信息栅栏长度增加;但当区域长度大于 30unit 后,即节点密度降低到一定值后,平均信息栅栏长度迅速下降,最终与 All Independent 情况相同。这是因为当初始密度降低时,通过虚拟节点组合,可以弥补节点间的空隙,增加信息栅栏长度。但当密度超过一定值时,节点间距离过大,组合成虚拟节点无法获得比独立时更长的信息覆盖栅栏,则能够组合的节点越来越少,最终所有节点都无法和其他节点进行组合,与 All Independent 情况相同。上述特点说明 DVSF 算法在较稀疏的网络(而不是非常稠密的网络)中具有较好的性能。

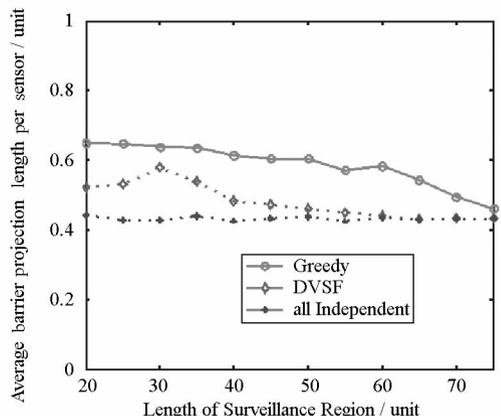


图 5 平均信息栅栏长度与区域长度关系
Fig.5 Average barrier length vs. region length

3.2 收敛过程

DVSF 算法分布式迭代运行,最终将收敛到

Nash 均衡局势。图 6 显示了迭代次数与部署节点数量的关系:算法最终都收敛,但随着部署节点数量增加,收敛的迭代次数也增加。因为节点密度越大,邻居节点数目越多,需要交互信息更改策略的次数越多。当部署节点为 200 时,迭代次数约 16 次,由图 3 可知,此时 DVSF 算法增加信息覆盖栅栏长度约 20%。因此 DVSF 算法适用于较稀疏的网络,能够以适中的迭代次数,较大幅度地增加信息覆盖栅栏长度。在栅栏覆盖的应用场景中,由于不需覆盖监控区域的所有位置点,无线传感器网络的部署密度通常不是很高,属于较稀疏的网络。因此 DVSF 算法适合于较低节点部署密度的栅栏覆盖应用场景。保持节点部署密度不变,增加部署区域面积,则网络节点规模将增加。图 7 显示当节点部署密度不变时,随着网络节点规模的增加,DVSF 收敛的迭代次数基本不变。这说明 DVSF 算法收敛性网络规模无关,该算法能够适用于大规模无线传感器网络。

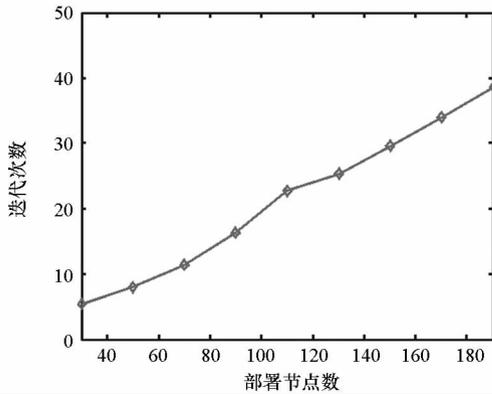


图 6 迭代次数与部署节点数关系

Fig. 6 Iterations vs. deployed sensors

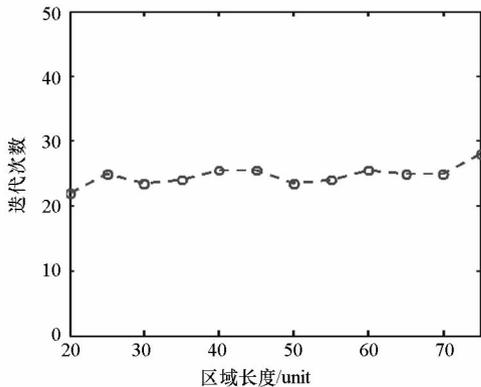


图 7 迭代次数与区域长度关系

Fig. 7 Iterations vs. region length

4 结束语

本文提出了一种完全分布式的虚拟节点组合算法 DVSF 算法。DVSF 算法基于非合作博弈模型,每个节点最大化自身的利益,通过邻近节点间的相互通信,迭代生成虚拟节点组合。实验表明,DVSF 算法有效地增加了网络的信息覆盖栅栏长度,并具有良好的收敛性和可扩展性,可适用于大规模的无线传感器网络。下一步研究将针对协作度大于 2 的虚拟节点组合情况,设计有效的分布式算法。

参考文献:

- [1] Kumar S, Lai T H, Arora A. Barrier Coverage with Wireless Sensors [C]//Proc. of Mobicon05, ACM, 2005: 284 - 298.
- [2] Chen A, Kumar S, Lai T H. Designing Localized Algorithms for Barrier Coverage[C]//Proc. of Mobicon07, ACM, 2007: 63 - 74.
- [3] Balister P, Bollobas B, Sarkar A, et al. Reliable Density Estimates for Achieving Coverage and Connectivity in Thin Strips of Finite Length [C]//Proc. of Mobicon07, ACM, 2007: 75 - 86.
- [4] Liu B Y, Olivier Dousse, Wang J. Strong Barrier Coverage of Wireless Sensor Networks[C]//Proc. of Mobihoc08, ACM, 2008: 411 - 420.
- [5] Anwar Saipulla, Benyuan Liu, Jie Wang. Barrier Coverage of Line-Based Deployed Wireless Sensor Networks [C]//Proc. of Infocom09, IEEE, 2009: 127 - 135.
- [6] Yang G Q, Qiao D J. Barrier Information Coverage with Wireless Sensors[C]//Proc. of INFOCOM09, IEEE, 2009: 918 - 926.
- [7] Wang W, Srinivasan V, Chua K C, et al. Energy-efficient Coverage for Target Detection in Wireless Sensor Networks [C]//Proc. of IPSN07, ACM, 2007: 313 - 322.
- [8] Xing G L, Tan R, Liu B Y, et al. Data Fusion Improves the Coverage of Wireless Sensor Networks[C]//Proc. of Mobicon09, ACM, 2009: 284 - 298.
- [9] Wang B, Srinivasan V, Chua K C, et al. Information Coverage for Wireless Sensor Networks [J]. IEEE Communications Letters, 2005, 9(11): 967 - 969.
- [10] 谢政. 对策论导论[M]. 北京: 科学出版社, 2010.