

基于遗传算法的传感器网络动态联盟研究^{*}

张石 张哲 朱吉昌

(东北大学信息学院 沈阳 110004)

摘要 动态联盟又称“虚拟企业”,是一种企业管理手段。它指一些相互独立的商业过程或企业组成的暂时联合。这里,将其理念应用于无线传感器网络中随机散布模式的节点动态组合研究,并基于遗传算法设计了过程模型,很好地解决了大规模传感器网络在应用过程中的节能规划问题。最后,本文针对该算法给出了仿真结果并对其行了分析,以说明算法的有效性。

关键词 遗传算法,动态联盟,无线传感器网络

Dynamic Alliance Based on Genetic Algorithms in Wireless Sensor Networks

ZHANG Shi ZHANG Zhe ZHU Ji-Chang

(Institute of Electronic Information Engineering Northeastern University, Shenyang 110044)

Abstract Dynamic Alliance (DA), namely, Virtual Corporations (VCs), is an enterprise management method. It means a temporary union formed by some independent commercial processes or corporations. In this paper, we apply this method to the research of nodes dynamic scheduling in Wireless Sensor Networks (WSN). We propose a DA model based on Genetic Algorithms (GA) and a typical simulation is conducted to verify the effectiveness of our model.

Keywords Genetic algorithms, Dynamic alliance, Wireless sensor networks

1 引言

目前,由于无线传感器网络广泛的应用领域,其研究吸引了众多学者的注意。传感器网络的初期节点分布有两种策略,一种是大规模的随机散布(Scattering Mode),另一种是针对特定用途进行有目的的设置(Planning Mode)^[1]。其中,大规模的随机散布模式的传感器网络在军事国防、环境监测、生物医疗、抢险救灾、危险区域远程控制等许多重要领域都有非常现实的应用前景。在这种模式下,为了得到较好的网络分布,必须投入远大于实际需要数目的冗余节点以获得较好的覆盖率。此时,如何有效地进行节点组合管理,在确保网络可靠性的前提下提高整体网络的生命周期,成为随机散布模式传感器网络面对的重要问题之一。本文设计了一种基于遗传算法的动态联盟^[2]过程对节点进行组织管理,并最后对算法性能进行了仿真分析。

2 问题的假设及分析

为了方便对传感器节点组织管理问题的研究,本文针对大规模随机散布模式下无线传感器网络的特点,进行了如下假设:

1) 侦测区域 A 可抽象为二维平面,在侦测区域内节点是随机布撒的。在布撒的过程中,节点的数目对于侦测区域的完全覆盖是存在冗余的。

2) 节点在完成布撒后是不可移动的,并且每个节点的坐标 $\{x_i, y_i\}$ 是可知的。

3) 所有的传感器节点具有相同的感知半径 r , 并且感知半径在节点的整个生命周期内保持不变。

4) 节点间的通信距离小于于节点有效感知半径的两倍(保证在节点对侦测区域完全覆盖的情况下网络是可靠连通

的^[3])。

基于上面四点假设,我们提出了如下节点动态联盟设计思想:

在布撒的节点中选择一部分节点组成动态联盟实现对侦测区域的完全覆盖,其它的节点进入休眠态。当工作节点中有部分节点死亡时,侦测区域内出现盲点,这时,唤醒网内的全部有效节点,重新进行动态联盟的选择。网络重复进行以上过程,直到网内的全部有效节点不足以构成对侦测区域的完全覆盖。此时,传感器网络失效。

3 三点问题

3.1 节点集覆盖率

现定义侦测区域 A 被离散化为 $m \times n$ 个像素,且每个像素的面积为 1; 侦测区域上的传感器节点集为 $N, N = \{n_1, n_2, \dots, n_i\}$, 且每个传感器节点 n_i 的覆盖模型为以节点坐标 $\{x_i, y_i\}$ 为圆心, r 为半径的圆。则侦测区域上的像素点 (x, y) 被传感器节点 n_i 所覆盖的这一事件 r_i 的概率 $P_{ov}(x, y, n_i)$ 为一个二值函数:

$$P\{r_i\} = P_{ov}(x, y, n_i) = \begin{cases} 1 & (x-x_i)^2 + (y-y_i)^2 \leq r^2 \\ 0 & \text{其他} \end{cases} \quad (1)$$

也就是说,如果一个像素点 (x, y) 到传感器节点 n_i 的距离不大于传感范围 r , 则我们认为该像素点 (x, y) 被传感器节点 n_i 覆盖。

假设所有的随机事件 r_i 是相互独立的,即 r_i 和 r_j 无关, $i, j \in [1, N]$ 且 $i \neq j$, 则存在如下关系:

$$P\{\bar{r}_i\} = 1 - P\{r_i\} = 1 - P_{ov}(x, y, n_i) \quad (2)$$

$$P\{r_i \cup r_j\} = 1 - P\{\bar{r}_i \cap \bar{r}_j\} = 1 - P\{\bar{r}_i\} \cdot P\{\bar{r}_j\} \quad (3)$$

其中, \bar{r}_i 为 r_i 的补,表示像素点 (x, y) 未被传感器节点 n_i 所

^{*} 基金项目:国家自然科学基金(60434030);国家自然科学基金(60374072)。张石 教授,博士。

覆盖这一事件。节点集中只要有一个节点覆盖了像素点 (x, y) ,我们就认为该像素点 (x, y) 被节点集覆盖。因此,像素点 (x, y) 被节点集所覆盖的概率即为 r_i 的并集。节点集 N 的覆盖率可以用公式(4)来进行计算:

$$P_{cov}(x, y, N) = P\{\bigcup_{i=1}^N r_i\} = 1 - P\{\bigcap_{i=1}^N \bar{r}_i\} \\ = 1 - \prod_{i=1}^N (1 - P_{cov}(x, y, n_i)) \quad (4)$$

上式表明,如果所有的节点都没有覆盖到像素点 (x, y) ,则该像素点 (x, y) 为未覆盖点,否则,便认为该像素点 (x, y) 被节点集覆盖。

定义节点集 N 的区域覆盖率 $R_{area}(N)$ 为其覆盖面积 $A_{area}(N)$ 与侦测区域 A 的总面积 A_s 之比,则:

$$R_{area}(N) = \frac{A_{area}(N)}{A_s} = \frac{\sum_{x=1}^m \sum_{y=1}^n P_{cov}(x, y, N)}{m \times n} \quad (5)$$

3.2 侦测区域的分块策略

对于整个侦测区域,我们可以把其分成若干适当的小块。对于处于同一小块内的多个传感器节点,可视其在对侦测区域的完全覆盖上是存在部分冗余的。因此,基于动态联盟的思想,每次从每个小块中至多选择两个节点进行结盟,完成对侦测区域的覆盖。

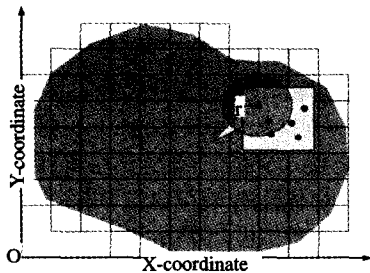


图1 分块策略示意图

在本算法中,如图1把被侦测区域化分为 J 个正方形的小块。正方形的边长 a 范围为 $r\sqrt{2}/2$ 到 $r\sqrt{\pi}$,其中 r 为传感器节点的感知半径,通常取 $a = r\sqrt{2}$ 。由于节点坐标已知,每一正方形小块内的节点数 I 亦可知。把位于相同正方形内的节点定为一组,并把第 j 个小正方形内的第 i 个节点表示为 $N_i^{(j)}$ 。每组每次至多选择两个节点组成动态联盟。当经过若干轮联盟后,可能出现某一正方形小块内的有效节点数为零并且无法选出一组动态联盟完成对侦测区域完全覆盖,这时,把分块的起始点 O 沿横、纵坐标分别平移 $r\sqrt{2}/2$ 后重新分块。

3.3 结盟过程的数学描述

从总体上来说,节点动态联盟的任务就是在保持网络连接的前提下,从大量的传感器节点中选出一组数目较少的节点,以较小的组网代价完成对侦测区域的完全覆盖。因此,动态联盟过程可以抽象为以下多目标优化问题:

目标函数1:动态联盟总的能耗成本 C 最小。这里的成本 C 包括两部分:一是分布式计算的内在成本 C_{cal} ,另一个是重新选择路由的运行成本 C_{rou} 。一般地,该目标函数可以表示为:

$$\min C = \min (C_{cal} + C_{rou}) = \min \left[\sum_{j=1}^J \sum_{i=1}^I C_i^{(j)} H_i^j + \frac{1}{2} \sum_{i=1}^I C_i^{(j')} H_i^{j'} H_i^{j''} \right] \quad (6)$$

其中: $H_i^j = \begin{cases} 1, & \text{选择 } N_i^{(j)} \text{ 参加动态联盟} \\ 0, & \text{不选择 } N_i^{(j)} \text{ 参加动态联盟} \end{cases}$

$C_i^{(j)}$ 为节点 $N_i^{(j)}$ 动态联盟计算的内在成本, $C_i^{(j')}$ 为节点 $N_i^{(j')}$ 和 $N_i^{(j')}$ 之间建立路由的通信成本; $i', i'' \in [1, I], j', j'' \in [1, J]$ 。

目标函数2:动态联盟所用的节点数目最少。

$$\min N = \min \sum_{j=1}^J \sum_{i=1}^I H_i^j \quad (7)$$

其中: $H_i^j = \begin{cases} 1, & \text{选择 } N_i^{(j)} \text{ 参加动态联盟} \\ 0, & \text{不选择 } N_i^{(j)} \text{ 参加动态联盟} \end{cases}$

约束条件:

1 覆盖率大于某一门限 G ,即: $R_{area}(C) \geq G$

2 根据传感器网络动态联盟的实际意义,在每个分出的正方形小块内至多选择两个节点参加组成动态联盟,即: $\sum_{i=1}^I H_i^j \in [0, 2]$ 。

4 基于遗传算法的动态联盟过程模型设计

遗传算法是一种基于自然选择与遗传机理的随机搜索算法;动态联盟过程是一个典型的多目标优化问题。根据遗传算法在求解多目标问题的天然优势,设计了如下算法模型。

4.1 基于遗传算法的动态联盟过程模型

4.1.1 节点的动态联盟在遗传算法中的编码方式

根据遗传算法,第一组节点动态联盟组合可以用图2所示的代码串 L 表示,其中码串的每一位代表一个候选节点的状态,位于一个小正方形内的一组节点定义为一个字段。图中, $N_i^{(j)} = 1$ 表示第 j 个小正方形中的第 i 个候选节点被选中组成动态联盟, $N_i^{(j)} = 0$ 表示未被选中。显然,代码串的长度 L 取决于所考察的候选节点个数,总的群体个数 $P = 2^L$ 。

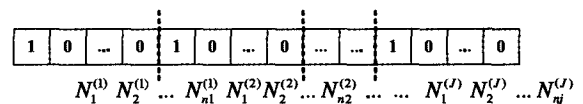


图2 节点的动态联盟在遗传算法中的编码方式

4.1.2 适应度函数 $f(t)$ 的构造

采用 Goal Programming 法构造多目标决策问题的适应度函数。对于多目标优化问题,算法设计者通常难以清晰、准确地给出各个目标的最优或最坏值。但是,由选用的路由协议和结盟算法的复杂度参数,可估算组网能耗成本的理想值。同时,被侦测区域的面积和每个节点的感知面积已知,这样每次结盟选择节点数目的理想值亦可知。因此,可以在状态空间中定义一个正理想点 (C^+, N^+) 和一个负理想点 (C^-, N^-) ,从而可以构造适应度函数 $f(t)$ 如下:

$$f(t) = \frac{d^-}{d^+ + d^-} \quad (8)$$

$$d^-(t) = \| ((C(t) - C^-), (N(t) - N^-)) \|$$

$$d^+(t) = \| ((C(t) - C^+), (N(t) - N^+)) \|$$

其中, $\| \cdot \|$ 表示取范数, t 为遗传算法中的代数, d^- 为到负理想点的距离测度, d^+ 为到正理想点的距离测度。

4.2 遗传算子与修正机制设计

在遗传算法中,复制、交叉和变异算子直接影响着遗传算法的效率和收敛速度。另外,考虑到动态联盟问题的具体约束条件,算法也需要加以相应改进。

4.2.1 选择算子

本算法先对种群中的个体按照适应度从大到小进行排序,再采用轮盘选择法,即假设每一个个体(代码串)的适应度为 $f_k (k=1, \dots, L)$,群体总的适应度为 $\sum_{k=1}^L f_k$,则将比例 $f_k / \sum_{k=1}^L f_k$

f_k 作为第 k 个个体的选择(复制)概率。

4.2.2 交叉算子

采用两点交叉算子来产生新一代个体,每一个经过交叉的个体要进行单字段变异。典型的交叉概率 P_c 取值范围为 0.6~1.0 之间。

4.2.3 变异算子

鉴于本算法保留了每代的较优个体,并且在算法中增加了自定义约束,本文选择了较大的变异概率 $P_m = 0.34$ 。

4.2.4 选择策略

本算法根据适应度大小从杂交变异后产生的新种群和原种群中选取前 10 个适应度大的个体,组成下一代新种群。

4.2.5 算法的改进:修正机制

考虑到动态联盟过程中的约束条件,本文在选择、交叉和变异过程中增加了一个修正机制,以确保每一个代码串在进行适应度评价之前满足约束。在修正机制中,每次必须进行如下断判。

IF (($R_{max}(C) < G$) OR ($\sum_{i=1}^I H_i \notin [0, 2]$)) THEN 修正

ELSE 适应度评价即一个经过选择、交叉或变异后的代码串不符合约束条件,代码串会被重置,选择、交叉或变异过程会重新进行,直至产生的代码串满足约束条件。

4.2.6 算法终止条件

采用相邻代适应度无显著变化作为终止条件(小于 0.01%),并输出群体中多个具有较优结构的个体作为最终结果输出。

5 算法仿真与结果分析

5.1 仿真条件设定

为了综合验证算法本身的性能及其在无线传感器网络优化问题上的适用性,本文设定一个 $100 \times 100m^2$ 的正方形仿真侦测区域。考虑现有传感器网络 ZigBee 通信协议的有效距离为 35~70m,这里设定传感器节点的感知半径 $r = 15m$ 。这样,在理想情况下,15 个节点可以构成侦测区域的完全覆盖。为了保证充分的冗余性,在该仿真区域上利用均匀分布随机函数生成 40 个节点,且生成的 40 个节点构成侦测区的完全覆盖。侦测区域被化分为 25 个边长 $a = 20m$ 的小正方形,如图 3 所示。

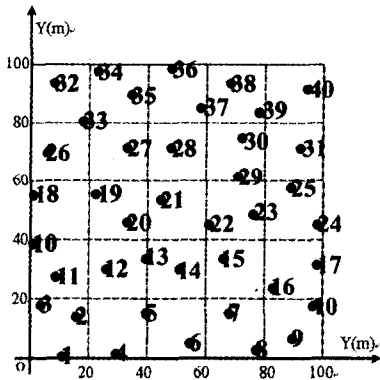


图 3 仿真侦测区域示意图

综合考虑动态联盟算法的复杂度和现有的路由协议,设定节点单次分布式计算平均内在成本为 $0.5\mu J$,每次重新选择路由的运行成本在 $3\mu J$ 到 $7\mu J$ 范围内服从中心点为 $5\mu J$ 的正态分布。由此可得构造适应度函数的正、负理想点为 (30,15)和(3060,40)。另外,设定覆盖率约束门限 $G=98\%$ 。

5.2 仿真结果及分析

在多次试验中发现:当交叉算子 P_c 取值一定时,取较小的变异算子 P_m ,算法收敛速度较快,但其最优结果却是局部最优,而非全局最优;而当取较大 P_m 时,由于检查机制的存在,搜索范围并不会由此增加,反而会导致算法在较低的目标水平下收敛,算法稳定性较差;当 P_m 取值一定时,取较小的 P_c ,算法要搜索到全局最优解很困难,且容易陷入局部最优;而当取较大 P_c 时,最优种群包含的个体种类会减少,不利于大规模传感器网络的分布式决策。最终确定 P_m 的合适取值范围为 $[0.2, 0.4]$,本例中取定 $P_m = 0.3$; P_c 的合适取值范围为 $[0.5, 0.8]$,且取定 $P_c = 0.6$ 。运行结果如表 1。

表 1 遗传算法运行结果中结构不同的 4 组最优解

	1	2	3	4
最大适应度	0.956888	0.956077	0.958172	0.940918
联盟结构	(1, 2, 4, 6, 8, 10, 11, 12, 14, 16, 19, 20, 22, 24, 26, 28, 30, 31, 33, 38, 40)	(1, 2, 4, 5, 6, 8, 10, 12, 15, 18, 19, 21, 25, 26, 29, 32, 35, 37, 39, 40)	(1, 2, 4, 6, 8, 9, 18, 19, 21, 22, 24, 26, 28, 29, 31, 32, 33, 35, 37, 38, 39, 40)	(1, 3, 4, 6, 8, 10, 15, 19, 21, 22, 24, 26, 27, 29, 31, 34, 36, 37, 40)
能耗	160.5	163	156.5	209
节点数	21	20	23	18

从表中可以看出,该遗传算法经过 83 代模拟,最优联盟组合的平均适应度函数从 0.542389 上长到 0.952226,最大适应度最后稳定在 0.956887,最优结构得到了保留,从而大大提高了联盟的整体性能,充分说明了该算法的有效性,且最优种群中包含了多种结构,本次运行结果包含 4 种不同结构,网络可以根据实际因素从中做出选择。如仅根据适应度判断,最后得到的动态联盟的最优组合为(1,2,4,5,6,8,10,12,15,18,19,21,23,25)。

表 2 网络生命周期仿真结果

	原始分块			重新分块		
	1	2	3	4	5	6
最大适应度	0.956888	0.956077	0.958172	0.934756	0.866082	0.864904
联盟结构	(1, 2, 4, 5, 6, 8, 10, 11, 12, 14, 16, 15, 18, 19, 20, 22, 24, 26, 28, 25, 26, 29, 30, 31, 33, 38, 40)	(1, 2, 4, 5, 6, 8, 10, 12, 8, 9, 18, 19, 21, 22, 19, 21, 24, 26, 28, 25, 26, 29, 31, 32, 29, 32, 33, 35, 37, 35, 37, 38, 39, 40)	(1, 2, 4, 6, 8, 9, 18, 15, 17, 18, 19, 21, 23, 24, 26, 28, 25, 26, 29, 31, 32, 30, 32, 33, 29, 32, 33, 35, 37, 35, 36, 38, 39, 40)	(1, 3, 4, 5, 7, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 30, 32, 33, 34, 35, 36, 38, 38, 40)	(1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 30, 32, 33, 32, 33, 34, 35, 36, 38, 40)	(1, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35, 36, 38, 40)
死亡节点	(31)	(10)	(37)	(5)	(18)	(4)
能耗	160.5	163	156.5	227.5	435.5	439
节点数	21	20	23	24	31	33

需要说明的一点是,由于算法增加了修正机制,使得初始种群的平均适应度和最大适应度上升较快。而且由于加入了选择机制,这两种适应度呈上升趋势,并且在经过多次迭代

(下转第 50 页)

5 访问控制服务算法流程

当用户利用 PKC 证书访问某个服务时, Web 服务注册中心调用相应的访问控制组件对服务调用进行相应的控制, 以确保服务访问的安全性。访问控制算法流程如图 4 所示。

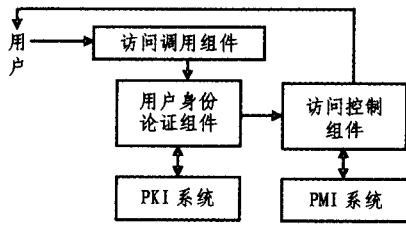


图 4 访问控制算法流程图

其算法步骤描述如下:

(1) 用户调用安全应用支撑层的“访问调用组件”, 请求调用某个服务;

(2) “访问调用组件”提取用户 PKC 证书的属性, 然后发送给安全应用支撑层的“用户身份论证组件”;

(3) “用户身份论证组件”调用 PKI 系统, 检查该用户是否是合法用户和 PKC 证书的有效性, 如是则继续, 否则请求失败;

(4) 调用安全应用支撑层的“访问控制组件”;

(5) “访问控制组件”调用 PMI 系统, 检查是否有与申请相一致的 AC 证书, 以及该证书的有效性;

(6) 如通过检查则提供相应的服务, 否则拒绝提供服务。

结论 基于 PKI/PMI 的 Web 服务安全框架利用相对成熟的安全技术, 来保障 Web 服务的安全, 在目前是可以实现的。该安全框架不需对 Web Services 现有的协议作改动, 仅需对现有 UDDI 和 SOAP 协议传输的内容进行加密即可。该安全框架提供的授权是分布式的, 由服务提供者进行服务授权, 经初步实验表明, 该框架一方面与 Web services 提供分布式计算相一致, 同时可以减少网络的传输量, 相应地也降低了

传输中的安全隐患。

参考文献

- Bartel M, Boyer J, Fox B, et al. XML-Signature Syntax and Processing. W3C Recommendation. <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>
- Imamura T, Dillaway B, Simon E. XML Encryption Syntax and Processing. W3C Recommendation. <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>
- OASIS Security Services TC. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)[S]. OASIS Standard. <http://www.oasis-open.org/committees/security/2002-11>
- OASIS Access Control Markup Language TC. eXtensible Access Control Markup Language (XACML) Version 1.0[S]. OASIS Standard. <http://www.oasis-open.org/committees/xacml/2003-02>
- OASIS Rights Language TC. eXtensible Rights Markup Language (XrML) Core 2.1 Specification. <http://www.oasis-open.org/committees/rights/2002-05>
- XML Key Management Specification (XKMS) Version 2.0. W3C Working Draft. <http://www.w3.org/TR/2003/WD-xkms2-20030418/2003-04>
- IBM, Microsoft, VeriSign. Web Services Security (WS-Security) Version 1.0. <http://www.ibm.com/developerworks/library/ws-secure/2002-04>
- IBM, Microsoft, RSA Security, VeriSign. Web Services Secure Conversation Language (WS-SecureConversation) Version 1.0. <http://www.ibm.com/developerworks/library/ws-secon/2002-12>
- IBM, Microsoft, RSA Security, VeriSign. Web Services Trust Language (WS-Trust) Version 1.0. <http://www.ibm.com/developerworks/library/ws-trust/2002-12>
- IBM, Microsoft, BEA, SAP AG. Web Services Policy Framework (WS-Policy) Version 1.0. <http://www.ibm.com/developerworks/library/ws-polfram/2002-12>
- IBM, Microsoft, RSA Security, VeriSign. Web Services Security Policy Language (WS-SecurityPolicy) Version 1.0. <http://www.ibm.com/developerworks/library/ws-secpol/2002-12>
- ITU-T Recommendation X.509, Information Technology—Open System Interconnection—The Directory: Authentication Framework, 1993
- ITU-T Recommendation X.501, Information Technology—Open Systems Interconnection—The Directory: Models, 2001

(上接第 22 页)

后, 算法最终可以收敛到一个稳定的最优解。

之后, 随机的选择工作节点死亡。再次运行算法, 根据适应度得出最优联盟。算法运行 5 代后无法再次满足约束条件。这时, 将分块起始点分别沿水平和垂直方向移动 10m, 重新分块, 再次运行算法, 得到最优解, 运行 2 代后, 网络最终失效。运行结果如表 2 所示。

结论 在大规模随机散布模式无线传感器网络中在保证全网可靠性的前提下, 以较低的能耗、较少的节点数动态地结盟, 以延长传感器网络的生命周期, 这无疑存在着广阔的研究前景和应用价值。本文针对上述问题, 设计了一种基于遗传算法的动态联盟优化模型, 该方法具有以下特点:

- 遗传算法本身非常适合在传感器网络内进行分布式计算, 即可以在不同的节点上分别进行若干代繁殖, 再综合比较得到最优动态联盟。

- 算法具有全局搜索能力, 因而可以克服局部最优解问题。

- 算法具有快速性、鲁棒性, 并且易于扩展至能解决任何数量级上的传感器网络动态联盟问题。

参考文献

- Bulusu N, Heidemann J, Estrin D. GPS-less low cost outdoor localization for very small devices; [Technical report]. Computer science department, University of Southern California, 2000
- Byrne A J. The virtual corporation. Business Week, 1993(2): 98~103
- Hightower J, Borriella G. Location systems for ubiquitous computing. IEEE Computer, 2001, 34(8): 57~66
- Zhang H, Hou J C. Maintaining scheme coverage and connectivity in large sensor networks; [Technical report]. UIUC, 2003