

感应半径可调的无线传感器网络生命周期的研究^{*}

成瑜娟[†], 张敏洪

(中国科学院研究生院数学科学学院, 北京 100049)

(2010 年 9 月 30 日收稿; 2011 年 1 月 4 日收修改稿)

Cheng Y J, Zhang M H. Study on lifetime of wireless sensor network with adjustable sensing ranges [J]. Journal of Graduate University of Chinese Academy of Sciences, 2011, 28(6): 768-775.

摘要 结合调整感应半径和采用睡眠唤醒调度机制 2 种重要的最大化网络生命周期方法, 研究了无线传感器网络的最大覆盖子集 MSC-ASR 问题, 提出了相应的混合整数规划模型, 证明了所给 MSC-ASR 问题是一个 NPC 模型. 此外, 提出了所探讨问题的 2 种算法: 集中式和分布式. 最后通过实验表明了所给算法的有效性.

关键词 无线传感器网络, 生命周期, 感应半径可调

中图分类号 O29

由于体积和成本要求的限制, 传感器节点通常使用能量有限的电池. 在环境复杂、无人可达的地方部署传感器, 更换电池或为节点充电都将不太现实. 在保证服务质量的条件下, 如何高效使用有限能量, 使得网络生命周期最大化, 成为无线传感器网络设计面临的挑战性课题^[1].

研究者分别从无线传感器网络的各个层面出发, 提出了降低能耗并延长网络生命周期的策略. 这些策略基本可以分为 4 类: 采用睡眠唤醒调度机制、调整感应半径、选择最佳路由、高效的数据融合机制^[2]. 本文将调整感应半径和睡眠唤醒调度机制相结合, 研究网络生命周期最大化问题.

无线传感器网络经常需要对被监测区域中位置确定的某些重要目标进行连续数据采集和监测, 即必须保证这些重要目标可以被完全覆盖, 这类覆盖称为点覆盖^[3-7]. 点覆盖问题因为其应用的必要性, 得到了广泛关注. 文献[3]研究各传感器感应半径固定时无线传感器网络的点覆盖问题, 提出了 the maximum set covers (MSC) problem, 证明 MSC 问题是 NPC 问题. 基于调整感应半径可以提高网络生命周期, 文献[4]研究传感器感应半径可离散变化时 WSN 的点覆盖问题, 提出了 the Adjustable Range Set Covers (AR-SC) problem, 算法的基本思想是根据贡献度的定义, 优先选择贡献度大的传感器及对应的感应半径. 算法简单, 但是没有综合考虑各目标点的覆盖时间. 文献[5]研究了传感器感应半径可连续变化时 WSN 的目标点覆盖问题, 但是其实际应用性不强. 文献[6]针对 AR-SC 模型, 提出了基于能量均衡和覆盖半径自适应调整的目标覆盖的分布式启发算法 (AR-EBHDA). 算法的基本思想是覆盖时间长的目标点处的传感器感应半径尽量大且等待时间短、优先决定工作状态; 覆盖时间短的目标点处的传感器感应半径尽量小、等待时间长, 后决定工作状态. 此算法在文献[4]的基础上有所改进, 但是仍无法确保各目标点覆盖时间的平衡性. 文献[7]在文献[4] AR-SC 问题基础上加入了连通性的限制条件.

本文做了以下几方面的工作. 首先, 将传感器感应半径可离散变化的点覆盖问题转化成最大覆盖子集 (the maximum set covers problem with adjustable sensing ranges, MSC-ASR) 问题, 建立了混合整数规划

^{*} 中国科学院研究生院院长基金 (085102AN00) 资助

[†]E-mail: chyj1126@126.com

模型,证明 MSC-ASR 问题是 NPC 问题;其次,基于能量充裕度低的目标点优先被覆盖且周围的传感器感应半径尽量小的原则,分别提出了集中式和分布式算法;最后对提出的算法进行了大量实验模拟,验证了算法的有效性.

本文的文章结构如下:第1部分对问题进行描述,提出数学模型;第2部分基于模型给出集中式算法;第3部分提出了分布式算法;第4部分进行实验模拟及结果分析;第5部分总结.

1 问题及模型

假设 N 个传感器随机散布在有 M 个目标点的监测区域内,负责收集、处理监测区域中所有目标点的信息,并将这些信息发送给 BS. 所有传感器感应半径的调整范围及计算处理信息的能力都相同. BS 在每个传感器的通信范围之内,不需要考虑传感器之间的连通性及传感器到 BS 的路由. 传感器的感知模型为圆形,即当传感器的感应半径为 r 时,可以覆盖以自己为圆心、以 r 为半径的圆形区域内的所有目标点. 传感器网络的生命周期定义为所有目标点可以被连续覆盖的时间.

1.1 MSC-ASR 问题

本文研究的问题是给定一个目标集和一个感应半径可离散变化的传感器集合,在满足覆盖要求和能量约束的条件下,选择传感器及相对应的感应半径,使得目标集中所有目标点被连续覆盖的时间最长.

在应用中,常常将大量传感器节点随机部署在若干个离散目标点附近. 将网络节点划分为若干个覆盖子集,且每个覆盖子集能够完全覆盖这些目标点,相继调度这些覆盖子集,通过最大化各覆盖子集的工作时间之和,可以有效延长整个网络的生命周期.

变量说明.

a_{ihj} :布尔变量, $i = 1, \dots, N, h = 1, \dots, H, j = 1, \dots, M$. 当传感器 s_i 以感应范围半径 r_h 可以覆盖目标 u_j 时 $a_{ihj} = 1$, 否则 $a_{ihj} = 0$.

C_k 是由若干个传感器及其对应的感应半径组成的集合. 当对每个目标点都可以在集合 C_k 中找到一个传感器及其对应的感应半径来覆盖它时,就说 C_k 是一个覆盖子集.

c_k :布尔变量, $k = 1, \dots, K$; 当 C_k 是覆盖子集时 $c_k = 1$, 否则 $c_k = 0$.

t_k :第 k 个覆盖子集的工作时间. 当 C_k 不是覆盖子集时, $t_k = 0$.

K :是一个很大的数,表示覆盖子集个数的上限.

x_{ikh} :布尔变量, $i = 1, \dots, N, k = 1, \dots, K, h = 1, \dots, H$; 当 $(i, h) \in C_k$ 时 $x_{ikh} = 1$, 否则 $x_{ikh} = 0$.

M_0 :是一个很大的正数.

MSC-ASR 问题模型:

$$\max \sum_{k=1}^K t_k, \quad (1)$$

$$\text{s. t.} \quad \sum_{k=1}^K \left(\sum_{h=1}^H x_{ikh} e_h \right) t_k \leq E_i, \quad i = 1, \dots, N, \quad (2)$$

$$\sum_{h=1}^H x_{ikh} \leq c_k, \quad i = 1, \dots, N, k = 1, \dots, K, \quad (3)$$

$$\sum_{i=1}^N \left(\sum_{h=1}^H x_{ikh} a_{ihj} \right) \geq c_k, \quad k = 1, \dots, K, j = 1, \dots, M, \quad (4)$$

$$t_k \leq c_k M_0, \quad k = 1, \dots, K, \quad (5)$$

$$t_k \geq 0, x_{ikh} \in \{0, 1\}, c_k \in \{0, 1\}. \quad (6)$$

目标函数(1)是各个覆盖子集的工作时间之和. 模型的目标是最大化目标集被连续覆盖的时间. 约束条件(2)是指每个传感器的能耗不能超过初始能量. 约束条件(3)是指每个覆盖子集中每个传感器至多可以选择一个感应半径. 约束条件(4)指 C_k 是覆盖子集当且仅当每个目标点至少可以被一个传感器

覆盖. 约束条件(5)是当 C_k 不是覆盖子集时, 目标集被覆盖的时间是 0. 约束(6)是对各个变量取值的限制条件.

1.2 问题分析

定理 1 MSC-ASR 问题是 NPC 问题.

证明 首先, MSC-ASR 问题是一个 NP 问题.

给定时间 t 、一族集合 $\{C_1, C_2, \dots, C_K\}$ 以及各集合对应的工作时间 $\{t_1, t_2, \dots, t_K\}$, 可以在多项式时间内验证每个集合 $C_k (k=1, \dots, K)$ 是否可以覆盖所有目标点, 每个传感器消耗的能量是否超过初始能量, $\sum_{k=1}^K t_k > t$ 是否成立.

其次, MSC-ASR 问题是一个 NP-hard 问题.

文献[3]已经证明 MSC 问题是 NPC 问题. 由于 MSC 问题是 MSC-ASR 问题在感应半径只有一种选择时的特殊情况, MSC 问题可以在多项式时间内规约为 MSC-ASR 问题. 因此 MSC-ASR 问题是 NP-hard 问题.

综合上述 2 点, MSC-ASR 问题是 NPC 问题(证毕).

2 集中式算法

由于 MSC-ASR 问题是 NPC 问题, 本文给出了近似的集中式算法. 下面先介绍算法中用到的定义.

定义 1 能量充裕度

由于传感器分布的随机性、非均匀性, 各个目标点可以被连续覆盖的时间不同. 为了对各个目标点可以被连续覆盖的时间做确切描述, 文中提出了能量充裕度的概念. 目标点可以被连续覆盖的时间越长, 就说这个目标点能量充裕度越高. 各目标点能量充裕度的计算公式如下:

$$P_j = \sum_{h=1}^H \sum_{i \in D_{jh}} \frac{E_i}{e_h}, \quad j = 1, \dots, M, \quad (7)$$

其中, $D_{jh} = \{i | a_{ihj} = 1, a_{ih-j} = 0\}, j = 1, \dots, M, h = 1, \dots, H$, 表示以感应半径 r_h 刚好可以覆盖目标点 j 的传感器的集合.

定义 2 传感器的贡献度

Cardei^[4] 传感器贡献度的定义为单位能量可以新增被覆盖目标点的个数. 计算公式为: $B_{ih} = \frac{|U_{ih}|}{e_h}$,

其中 U_{ih} 表示目前未被覆盖的目标点中可以被传感器 s_i 以感应半径 r_h 覆盖的目标点的集合. $|U_{ih}|$ 表示 U_{ih} 集中的目标点数.

集中式算法的基本思想: 优先选择能量充裕度低的目标点周围的传感器, 使得这些传感器感应半径尽量小, 只负责覆盖能量充裕度低的目标点, 节省能量.

下面是集中式算法的变量.

t_0 : 每个覆盖子集工作时间的下限.

U_k : 表示 C_k 目前还没覆盖的目标点的集合, $k = 1, \dots, K$.

集中式算法的说明.

初始阶段: 对所有传感器节点的剩余能量初始化, 对各个目标点计算能量充裕度和在各个感应半径下的贡献度.

覆盖子集构造阶段: for $k = 1, \dots, K$, 构造覆盖集 C_k . 覆盖集初始化为 $C_k = \emptyset$, 未被 C_k 覆盖的目标点集为 $U_k = \{u_1, u_2, \dots, u_M\}$. 当 $U_k \neq \emptyset$, C_k 没有覆盖所有目标点, 找出未被 C_k 覆盖的能量充裕度最小的目标点 u_j . D_j^1 表示可以覆盖 u_j , 但不可以覆盖能量充裕度比 u_j 小的目标点的传感器集合. 在 D_j^1 中找到距离 u_j 最近、剩余能量可以支持其工作时间超过阈值 t_0 的所有传感器, 设这些传感器到 u_j 的距离是 r_h . 在这些传感器中选择贡献度最大的 (i, h) , 把 (i, h) 加入 C_k . 若 D_j^1 为空集, 在其他可以覆盖 u_j 的传感器中选择贡献度最大的、剩余能量可以支持其工作时间超过阈值 t_0 的传感器 s_i 及感应范围 r_h , 把 (i, h) 加

入 C_k . 当 $U_k = \emptyset$ 时, C_k 覆盖所有目标点, C_k 的工作时间是 C_k 中所有传感器工作时间的最小值. 计算每个传感器在 C_k 工作后的剩余能量 $\forall (i, h) \in C_k, E_i = E_i - e_h t_k$. 根据公式重新计算各目标点的能量充裕度, 开始新一轮覆盖集的选择. 当存在某个目标点无法被覆盖时, 算法结束.

下面是集中式算法步骤:

- 1) 初始化各个传感器剩余能量;
- 2) 计算各个目标点的能量充裕度;
- 3) 构造覆盖集 for $k = 1, \dots, K$;
- 4) $U_k = \{u_1, u_2, \dots, u_M\}, C_k = \emptyset, t_k = \frac{E_0}{e_1}$;
- 5) while $U_k \neq \emptyset$ do
- 6) 找到未被覆盖的能量充裕度最小的目标点 u_j ;
- 7) if $D_j^1 = \emptyset$,
- 8) 在 D_j^1 中选择距离 u_j 最近、剩余能量充足且贡献度最大的传感器 s_j, s_j 到 u_j 的距离是 r_h ;
- 9) else if 除 D_j^1 中的传感器之外, 还有其他传感器可以覆盖 u_j ;
- 10) 在其中选择贡献度最大、剩余能量充足的传感器及对应的感应半径 (i, h) ;
- 11) else go to step 22;
- 12) end;
- 13) $t_{ih} = \frac{E_i}{c_h}, t_k = \min\{t_k, t_{ih}\}$;
- 14) $C_k = C_k \cup (i, h)$;
- 15) 更新未覆盖目标点集;
- 16) 更新各传感器及感应半径对应的贡献度;
- 17) end;
- 18) 计算各传感器的剩余能量, $\forall (i, h) \in C_k, E_i = E_i - e_h t_k, t = t + t_k$;
- 19) 重新计算各目标点的能量充裕度;
- 20) $k = k + 1$;
- 21) end;
- 22) Output t (算法结束).

集中式算法中, 计算能量充裕度的时间复杂性是 $O(MNH)$. 每构造一个覆盖子集都会造成至少一个传感器节点的死亡, 因此至多有 N 个覆盖子集. 覆盖子集构造过程中每选择一个传感器至少减少一个未被覆盖的目标点, 因此至多需要选择 M 次传感器及其对应的感应半径. 每选择一个传感器及对应感应半径, 更新未被覆盖目标点的时间复杂度为 $O(M)$, 更新各传感器及感应半径对应的贡献度的时间复杂度为 $O(NH)$. 覆盖子集构造完成后计算传感器剩余能量的时间复杂度至多为 $O(M)$. 整个算法的时间复杂度为 $O(NM(M + NH))$.

从时间复杂度的表达式上, 可以看出算法的复杂性与传感器的个数、目标点的个数以及传感器感应半径可调整的级数呈正相关关系.

3 分布式算法

由于集中式算法不适用于大型的无线传感器网络, 本文提出了具有可扩充性的分布式算法(见图1). 分布式算法按轮次进行, 一轮包括覆盖子集的构造阶段和工作阶段2个部分. 每轮中用于覆盖子集构造的时间 w 远远小于其工作时间. 假设 BS 和各传感器有严格的时钟同步技术. 由 BS 根据新一轮覆盖子集的信息, 发出开始新一轮的命令.

在每轮的开始阶段,由于相邻的传感器工作状态彼此影响,传感器不能同时决定各自的工作状态;否则会造成大量的能量浪费.各个传感器工作状态的决策应该有一个先后顺序.因此,本文对每个传感器设定等待时间.

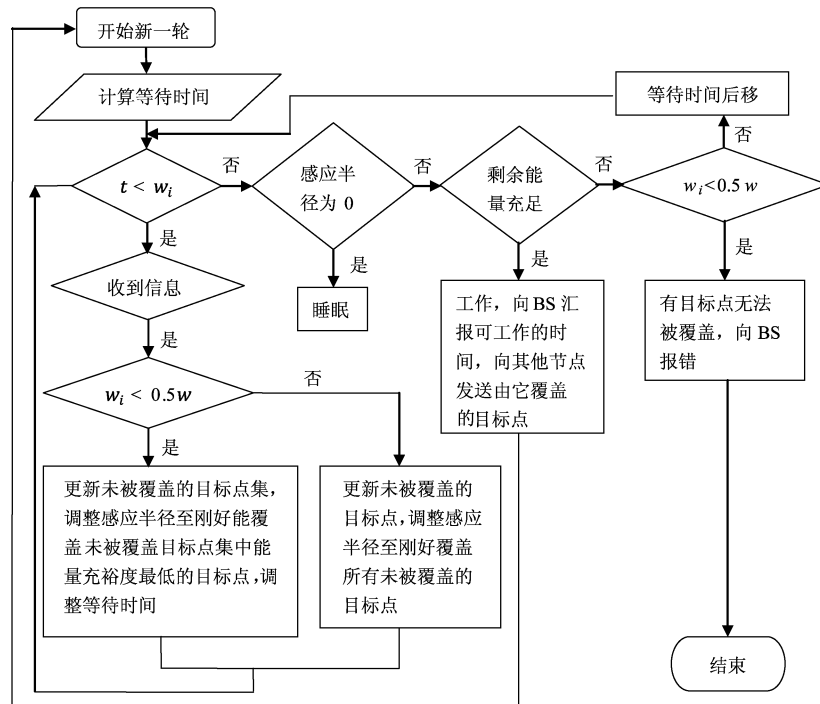


图 1 分布式算法流程图

3.1 等待时间

定理 2 每个传感器可以通过与 $\left[\frac{2r_H}{r_i}\right]$ 跳范围内的其他传感器交换信息, 获得其最大感应范围内所有目标点的能量充裕度, 其中 r_i 表示传感器的通信半径.

证明 首先, 每个传感器可以通过从小到大调节感应半径探测出周围有哪些目标点, 以及覆盖这些目标点需要的最小感应半径. 再者, s_i 可探测到的目标点 u_j 到 s_i 的距离至多是 r_H , 而要了解有哪些传感器覆盖 u_j , 以什么样的感应半径覆盖 u_j , 最多只需要了解以 s_i 为中心, 以 $2r_H$ 为半径的周围区域的情况.

这就是说 s_i 要想获得最大感应范围内各个目标点的能量充裕度的情况, 只需要与 $\left[\frac{2r_H}{r_i}\right]$ 跳的传感器进行通信(证毕).

每个传感器 s_i 的等待时间 w_i , 由 s_i 可以覆盖的目前未被覆盖的目标点中能量充裕度最低的 u_j 和 s_i 覆盖 u_j 需要的最小感应半径 r_h 决定. 离能量充裕度越低的目标点越近的传感器等待时间越短. w_i 的计算公式如下:

$$w_i = \min_{j \in U_{iH}, a_{ij}=1} 0.5w \left\{ \frac{r_h P_j}{r_H P_0} + \varepsilon_i \right\}, i = 1, \dots, N, \quad (8)$$

$P_j (j = 1, \dots, M)$ 是目标点 u_j 的能量充裕度, $P_0 = \sum_{i=1}^N \frac{E_i}{e_i}$ 是各个目标点能量充裕度的上限, $\varepsilon_i \sim$

$U\left(0, \frac{1}{e_H}\right), (i = 1, \dots, N)$. 引入 $\varepsilon_i (i = 1, \dots, N)$ 是为了避免 2 个传感器等待时间一样的情况下同时做出

状态决策, 从而造成可能的能量浪费.

3.2 分布式算法

算法的基本思想:由 BS 向传感器发布开始覆盖子集构造的命令.各传感器根据公式计算等待时间.传感器尽量只负责覆盖最大感应范围内未被覆盖的能量充裕度最低的目标点.若传感器在等待时间内收到来自其他传感器的信息,更新最大感应范围内未被覆盖的目标点集、感应半径及等待时间.若等待时间到达,传感器的状态分2种情况处理:①若等待时间小于 $0.5w$,当最大感应范围内所有目标点都被覆盖,传感器进入睡眠状态;否则,当剩余能量充足,传感器调整感应半径至刚好覆盖最大感应范围内未被覆盖的能量充裕度最低的目标点,进入工作状态;当剩余能量不足,调整等待时间至最大等待时间减去一个介于0到 $0.5w$ 之间的随机数,看看是否有其他传感器覆盖未被覆盖的能量充裕度最低的目标点.②若等待时间大于 $0.5w$,当最大感应范围内所有目标点都被覆盖,传感器进入睡眠状态;否则若剩余能量可以支持传感器覆盖未被覆盖目标点的时间大于时间下限 t_0 ,进入工作状态,否则传感器向 BS 报错,算法结束.

算法步骤:

- 1) 计算 U_{ih} 中各目标点的能量充裕度、等待时间 w_i 、开始时间 t (t 表示从新一轮开始已经经过的时间);
- 2) while $t < w_i$ and $U_{ih} \neq \emptyset$, do
- 3) if 收到从其他传感器发来的信息,更新 U_{ih} 和等待时间 w_i ,调整感应半径至 r_u 刚好能覆盖 U_{ih} 中能量充裕度最低的目标点;
- 4) if $U_{ih} = \emptyset$,
- 5) s_i 没必要工作,感应半径 r_u 设为 0.
- 6) break
- 7) end
- 8) end
- 9) end
- 10) if $w_i \leq 0.5w$,
- 11) if $r_u = 0$,
- 12) s_i 进入睡眠状态;
- 13) else
- 14) if $E_i \geq e_u t_0$,
- 15) s_i 以感应半径 r_u 工作,把这轮中覆盖的目标点集发送给邻点,可以工作的时间 t_i 报告给 BS;
- 16) else
- 17) $w_i = w - \sigma_i, \sigma_i \sim U(0, 0.5w)$, go to step 2.
- 18) end
- 19) end
- 20) else
- 21) if $U_{ih} = \emptyset$.
- 22) s_i 睡眠;
- 23) else
- 24) s_i 调整感应半径至 r_u 刚好能覆盖 U_{ih} 中所有的目标点;
- 25) if $E_i \geq e_u t_0$,
- 26) s_i 以感应半径 r_u 工作,把这轮中覆盖的目标点集发送给邻点,把可以工作的时间 t_i 报告给 BS;
- 27) else
- 28) s_i 向 BS 报错,能量不足,无法覆盖所有目标点.
- 29) end

30) end

31) end.

在分布式算法中,各传感器计算等待时间的复杂度是 $O(MNH)$,在决定状态之前至多收到来自 M 个传感器节点的消息,至多更新 M 次等待时间,因此总的复杂度是 $O(M^2NH)$.

4 实验及结果分析

为了评估算法的性能,本节将文献[4]中的集中式和分布式算法分别记作算法 1 和算法 3,本文提出的集中式和分布式算法分别记作算法 2 和算法 4,对这 4 种算法进行仿真,比较在相同实验条件下的

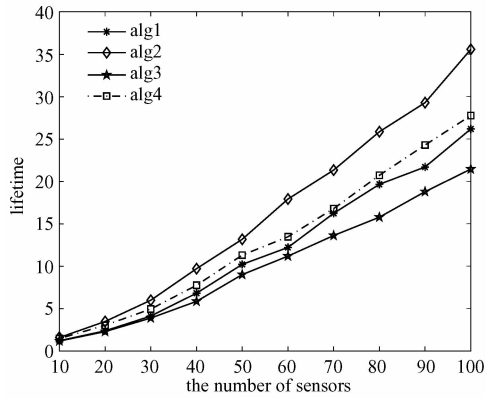


图 2 采用算法 1、2、3、4 时,网络生命周期与传感器个数的关系

在图 2 中,研究传感器个数从 10 开始,以 10 为步长,逐渐增加到 100,最大感应半径为 30m,感应半径可调级数为 5,能耗模型为 $e_h = ar_h^2$,其中 $a = 0.003$,各传感器初始能量都是 3,目标点个数为 10, $t_0 = 0.001$ 时,由算法 1、算法 2、算法 3 和算法 4 得到的网络生命周期图.

在图 3 中,最大感应半径为 20m,感应半径可调级数分别是 1、3、5,其他参数不变,比较由算法 3 和算法 4 得到的网络生命周期.

在图 4 中,最大感应半径为 20m,目标点个数分别为 3、6、9,其他变量与图 1 相同,比较由算法 3 和算法 4 得到的网络生命周期.

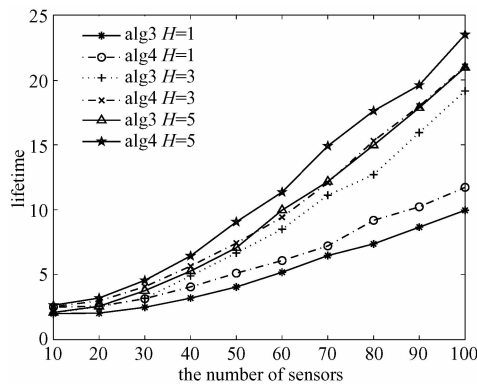


图 3 采用算法 3、算法 4 时网络生命周期与感应半径可调级数的关系

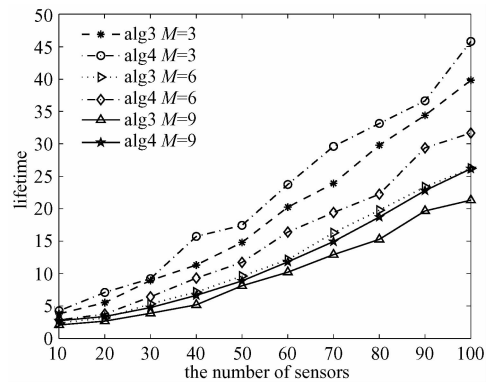


图 4 采用算法 3、算法 4 时网络生命周期与目标点个数的关系

实验结论:

1) 算法 2 与算法 1 相比更有效,算法 4 与算法 3 相比更有效;

生命周期.在图 2 对比图中,alg1、alg2、alg3、alg4 分别表示算法 1、算法 2、算法 3、算法 4.实验数据参考文献[4].在 WindowsXP 上采用 Matlab7.1 作为仿真软件,在 100m × 100m 的设定区域内随机生成目标点的二维坐标和传感器的二维坐标.为简化实验,假设每个传感器的初始能量相同.

实验中,考虑以下可调参数

N:传感器的个数从 10 开始逐次增加 10 到 100.

H:最大感应半径 R 固定的情况下,感应半径可调的级数 H 分别为 1、3、5.可调的感应半径分别为 $\{r_1, r_2, \dots, r_H\}, r_h = \frac{Rh}{H}, h = 1, \dots, H$.

M:目标点个数从 3 开始以 3 为步长,逐渐变为 9.

- 2) 最大感应半径固定的情况下,增加感应半径可调级数,可以延长网络生命周期;
- 3) 在其他参数不变的情况下,目标点个数越多,网络生命周期越短.

5 结论

本文研究了传感器感应半径可调时无线传感器网络的目标点覆盖问题,将它转化为 MSC-ASR 问题,建立了混合整数规划模型,并证明 MSC-ASR 问题是 NPC 问题. 基于能量充裕度低的目标点优先被覆盖,且周围的传感器感应半径尽量小的原则,本文分别提出了集中式算法和分布式算法,并通过实验模拟证明了所提算法的有效性.

参考文献

- [1] Anastasi G, Conti M, Francesco M D, et al. Energy conservation in wireless sensor networks: a survey[J]. *Ad Hoc Networks*, 2009,7(7): 537-568.
- [2] Ahuja S K, Kini S, Ramasubramanian S. Bounds on coverage time and node density for multi-modality sensing[J]. *Ad Hoc Networks*, 2009,7(7): 1302-1314.
- [3] Cardei M, My T Thai, Li Y S, et al. Energy-efficient target coverage in wireless sensor networks[J]. *IEEE Infocom*, 2005,3:1976-1984.
- [4] Cardei M, Wu J, Lu M M. Improving network lifetime using sensors with ad-justable sensing ranges[J]. *International Journal of Sensor Networks (IJSNet)*, 2006, 1(1/2): 41-49.
- [5] Dhawan A, Vu C T, Zelikovsky A, et al. Maximum lifetime of sensor networks with adjustable sensing range[C]//7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06). 2006: 285-289.
- [6] Zhang H W, Wang H Y, Feng H C. Energy-balance heuristic distributed algorithm with adjustable sensing ranges for target coverage in wireless sensor networks[J]. *Journal of Chinese Computer Systems*, 2010,4(4):620-625 (in Chinese).
张红武,王宏远,丰洪才. 无线传感器网络中能量均衡和覆盖半径自适应调整的分布式目标覆盖算法[J]. *小型微型计算机系统*, 2010,4(4):620-625.
- [7] Lu M M, Wu J, Cardei M, et al. Energy-efficient connected coverage of discrete targets in wireless sensor networks[J]. *International Journal of Ad Hoc and Ubiquitous Computing*, 2009,4(3/4):137-147.

Study on lifetime of wireless sensor network with adjustable sensing ranges

CHENG Yu-Juan, ZHANG Min-Hong

(Graduate University, Chinese Academy of Sciences, Beijing 100049, China)

Abstract Based on efficient scheduling and adapting sensing ranges which are the two important methods for maximizing wireless sensor network lifetime, we study the maximum set covers problem with adjustable sensing ranges (MSC-ASR), propose a mixed integer programming model, and demonstrate that MSC-ASR is a NPC problem. For solving MSC-ASR, we present two methods: centralized algorithm and distributed algorithm. Simulations show that the algorithms are effective.

Key words WSN, lifetime, adjustable sensing ranges