

# WSN 中考虑负载均衡的贪婪寻优中继节点布局算法的研究

张航<sup>1</sup> 佟晓筠<sup>1</sup> 王翥<sup>2</sup>

(哈尔滨工业大学(威海)计算机科学与技术学院 威海 264209)<sup>1</sup>

(哈尔滨工业大学(威海)信息与电气工程学院 威海 264209)<sup>2</sup>

**摘要** 针对目前 WSN 中的中继节点布局算法中忽略负载均衡的问题,首先引入通信路径不可逆等优化模型来进行节点布局;其次,提出阈值法及平均值法来动态更新网络中各路径的负载;然后针对负载均衡与节点数目难两全的问题,提出一种优化贪婪寻优算法,使得在考虑负载的同时,所需节点的数目能尽量减少。测试结果表明,使用不考虑负载的传统贪婪寻优算法以及使用考虑负载均衡的优化贪婪寻优算法进行节点布局时,两种算法所耗费的节点数目相差无几,然而后者却能够使整个网络的负载更为平均,更适合于实际应用中灵活多变、环境复杂的无线传感器网络,具有更深的实际意义。

**关键词** 无线传感器网络,负载均衡,中继节点布局,阈值法,平均值法,优化贪婪寻优算法

**中图分类号** TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.6.026

## Research on Relay Node Placement Considering Load Balancing Based on Greedy Optimization Algorithm in Wireless Sensor Networks

ZHANG Hang<sup>1</sup> TONG Xiao-jun<sup>1</sup> WANG Zhu<sup>2</sup>

(School of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai 264209, China)<sup>1</sup>

(School of Information and Electrical Engineering, Harbin Institute of Technology at Weihai, Weihai 264209, China)<sup>2</sup>

**Abstract** At present, all the relay node placement algorithms in WSN ignore the factor of load balancing, and we introduced several layout optimization models based on this. Then we proposed threshold value method and mean value method to update each path's load. At last, we put forward an optimal greedy optimization algorithm to try to reduce the number of required nodes when considering the load balancing. The test results show that the optimization greedy optimization algorithm can make the load of the whole network more average and it is more suitable for practical application.

**Keywords** Wireless sensor networks, Load balancing, Relay node placement, Threshold value method, Mean value method, Greedy optimization algorithm

## 1 引言

节点的合理布局是无线传感器网络(Wireless Sensor Networks, WSN)正常运行的前提。在很多场合,网关节点(Gateway Node, GN)及传感器节点(Sensor Node, SN)的位置相对固定,为了保证网络的连通性,就必须在两者之间合理地布局中继节点(Relay Node, RN)。同时 WSN 中的节点能量有限,且其经常布置在人员不可达的户外恶劣环境中,故如何控制节点的能耗成为一个严峻的问题。基于此, Misra 等人提出近似比例为  $O(1)$  的多项式时间近似算法,保证网络中放置的 RN 数目最少<sup>[1]</sup>。对于分层的无线传感器网络, Bari 等通过移动的数据采集器(MDC)来改善网络性能, MDC 沿着固定的轨线移动,采集每个节点的数据<sup>[2]</sup>。Ying Zhu 等设计

了一种基于 Steiner 最小树迭代改进过程的算法,但其忽略了节点在实际布局过程中可能遇到的多限定物理因素<sup>[3-5]</sup>。王翥等给出了贪婪寻优算法在实际应用中的具体模型<sup>[6]</sup>,并认为 RN 的数目和能耗是评价算法好坏的标准。但是 RN 的数目和能耗是一个需要平衡的问题。严格追求尽量减少 RN 的数目必将导致某些通信路径上的节点负载过多,从而使得该路径上的节点由于过量转发数据而过早死亡。

因此,考虑到实际应用中复杂多变的网络环境,如何平衡 RN 数目和节点的负载具有更加实际的意义。

## 2 传统贪婪寻优算法

本节将详细描述传统的不考虑负载均衡的贪婪寻优算法(Greedy Optimization Algorithm, GWLB)以及节点的多约束布局模型。

到稿日期:2014-07-02 返修日期:2014-09-29 本文受国家自然科学基金(60973162),山东省科技攻关项目(2013GGX10129, 2010GGX10132, 2012GGX10110),国家密码发展基金(MMJJ201301006),威海市信息安全工程技术研究中心资助。

张航(1990-),女,硕士生,主要研究方向为无线传感器网络、轻量级密码;佟晓筠(1963-),女,博士,教授,主要研究方向为混沌密码学、网络与信息安全、无线传感器网络安全等, E-mail: tong\_xiaojun@163.com(通信作者);王翥(1963-),男,博士,教授,主要研究方向为无线传感器网络、信号检测与处理、传感器及应用技术等。

## 2.1 网格路由算法布局中继节点

网格路由算法使用确定性部署原则放置 WSN 节点,其将监测区域划分为二维网格,网格的交叉点即为 RN 的候选位置。

对监测区域进行网格划分时,网格间距  $L$  是一个重要参数。间距设置过大,相邻网格内的节点可能无法相互通信;设置过小,SN 到 sink 节点的数据转发跳数增加,选择的 RN 数目增加。吴谋等通过仿真实验,设定网格上限  $L \leq R_n / \sqrt{5}$  ( $R_n$  为最大通信距离),以保证相邻网格内的节点可以通信<sup>[4]</sup>;陈林星等设定网格小于  $(1/\sqrt{8})R_{trans}$  ( $R_{trans}$  为传输范围)<sup>[6]</sup>;王翥等采用接收信号强度指示器(Received Signal Strength Indicator, RSSI)和链路质量指示(Link Quality Indication, LQI)确定网格间距<sup>[6]</sup>,即网格间距需满足式(1)所示的不等式:

$$L^2 + (3 \times L)^2 \leq R^2 \quad (1)$$

综上,本文采用式(1),求得  $L \leq \frac{R}{\sqrt{10}}$ 。

## 2.2 RN 布局多约束模型

WSN 中 RN 的布设在实际应用中受到很多物理因素的限制。本文引入如下多约束性优化模型。

### (1) 节点通信半径

由于每个 SN 的发射功率有限,因此其只能在一定范围内转发数据。我们用以该节点为圆心、 $R$  为半径的圆表示这个范围。 $R$  即为节点的通信半径,如图 1 所示。

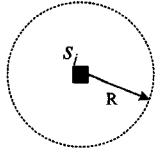
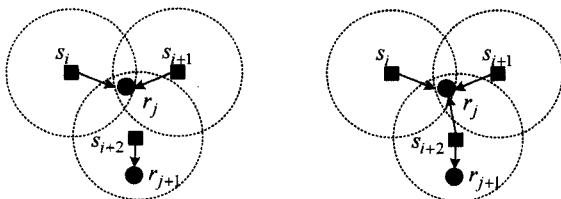


图 1 SN 的通信半径

### (2) 通信容量

严格追求减少 RN 的数目必然会导致某条或某些路径上的 RN 能耗过大,故需要限制每个 RN 的通信容量,只有不超过该 RN 的最大通信容量,才可以进行数据的转发,否则需要重新选择其它 RN 转发数据。

如图 2 中的 3 个 SN ( $s_i, s_{i+1}, s_{i+2}$ ) 和 2 个 RN ( $r_j, r_{j+1}$ ), 两个 RN 在这 3 个 SN 的通信范围内,  $r_{j+1}$  在  $s_{i+2}$  的通信范围内。若 SN 的流量为 6, RN 最大通信流量为 15, 那么  $r_j$  最多与其中两个 SN 进行通信, 即 RN 允许的通信容量为 2, 如图 2(a) 所示。若 RN 最大通信流量为 20, 那么  $r_j$  至多可以与 3 个 SN 进行通信, 即 RN 允许的通信容量为 3, 如图 2(b) 所示。



(a) 超过 RN 通信容量

(b) 未超过 RN 通信容量

图 2 通信容量示意图

### (3) 通信路径不可逆

从减少网络构建成本以及构建后期地理路由角度出发,

我们引入通信路径不可逆的约束条件,即向一个待构建网络中布局 RN 时,保证构建完成的网络从任意一个 SN 出发经过多个 RN 到达 sink 的通信路径的传播方向上不存在逆向的传播。从向量角度出发,即是 RN 数据传输方向与自身和 sink 方向的向量角应小于  $90^\circ$ 。定义判断因子  $Re\_1, Re\_2$  和  $Re\_3$  如下:

$$Re\_1 = v_i v_0 \cdot v_i v_j \quad (1 \leq i \leq n, n+1 \leq j \leq n+m) \quad (2)$$

$$Re\_2 = v_{k-1} v_k \cdot v_i v_0 \quad (n+2 \leq k \leq n+m) \quad (3)$$

$$Re\_3 = v_h v_0 \cdot v_i v_0 \quad (n+1 \leq h \leq n+m) \quad (4)$$

式中,  $\vec{v}_i v_0$  为基准向量;  $\vec{v}_i v_j$  为传感器节点  $v_i$  与其通信范围内中继节点  $v_j$  之间的向量;  $\vec{v}_{k-1} v_k$  为  $v_i$  与  $v_0$  通信路径上,两个相邻中继节点  $v_{k-1}$  与  $v_k$  ( $v_{k-1}$  通信范围内的中继节点)之间的向量;  $v_h$  为  $v_i$  与  $v_0$  通信路径上最末一个中继节点,即  $v_0$  通信范围内的中继节点;  $\vec{v}_h v_0$  为中继节点  $v_h$  与  $v_0$  之间的向量。

在选取路径时需满足式(5)一式(7):

$$Re\_1 > 0 \quad (5)$$

$$Re\_2 > 0 \quad (6)$$

$$Re\_3 > 0 \quad (7)$$

图 3 示出了两条通信路径,其中  $\vec{s}_i s_0$  是基准向量。

针对式(5)一式(7),  $s_i$  最终选择路径  $s_i \rightarrow r_5 \rightarrow r_6 \rightarrow s_0$  传输数据。

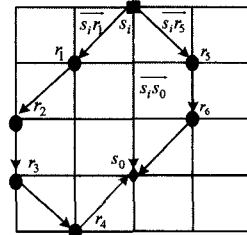


图 3 通信路径的选择

## 2.3 贪婪寻优算法布局 RN

### (1) 贪婪准则

作出贪婪决策的依据是贪婪准则,贪婪准则的好坏直接影响算法的性能,本文采用最近贪婪准则,即在节点的邻居节点中选择离 sink 最近的节点转发数据,以使数据将要经过的路径最短。在节点分布密集且均匀的网络中,这一贪婪准则有利于返回具有最小跳数的路径。

根据上述贪婪准则,节点  $v$  向目的节点  $u$  的下一跳  $v_{father}$  由式(8)确定。

$$v_{father} = \min \{ \| uv_{father} \| \}_{v_{father} \in N(v)} \quad (8)$$

其中,  $\| uv_{father} \|$  为目的节点  $u$  和节点  $v$  下一跳  $v_{father}$  之间的欧氏距离;  $N(v)$  为节点  $v$  的邻居节点集合。

本文在设计 RN 贪婪布局时,不仅考虑了节点通信半径、RN 通信容量和通信路径不可逆等限制条件,还采用最近机制贪婪准则求局部最优解,逐步迭代,直至每个 SN 可以与 sink 节点进行通信。

### (2) 传统贪婪寻优算法布局 RN

基于网格结构的传统贪婪寻优算法布局 RN 的具体步骤如下:

输入:监测区域、SN:  $S = \{s_1, s_2, \dots, s_n\}$ 、候选 RN:  $R = \{r_1, r_2, \dots, r_m\}$ 、sink 节点  $s_0$ 、网格间距  $L$ 、通信半径  $R$  及节点通信容量上限  $\max$  等信息;

输出: RN 数目和位置。

- Step1 将放置 RN 的区域进行网格化处理,确定候选 RN 的坐标。
- Step2 将  $s_i$  到  $s_0$  的欧氏距离  $d(s_i, s_0)$  记为  $d_{i0}$  ( $1 \leq i \leq n$ ), 降序排列为  $d = (d_1, d_2, \dots, d_n)$ , 与  $d$  中元素一一对应的 SN 集合  $S_0 = \{s_{10}, s_{20}, \dots, s_{n0}\}$ 。
- Step3 从距离 sink 节点最远的 SN (即  $s_{10}$ ) 开始, 在其通信范围内搜索 RN, 选择满足最近机制且离  $s_0$  最近的 RN (设为  $r_j$ ) 作为  $s_{10}$  的下一跳。
- Step4 若  $r_j$  此时的通信容量  $(RNCAPACITY[r_j] + 1) < r_j$  的通信容量 (CAPACITY), 则选择  $r_j$  作为下一跳, 且更新选中的 RN 的通信容量  $RNCAPACITY[r_j] + 1$ 。
- Step5 若  $d(r_j, s_0) > R$ ,  $r_j$  重复 Step3, 继续在其通信范围内搜索 RN, 设选中  $r_{j+1}$  作为下一跳, 进入下一步。否则, 从集合  $d$  中删除  $d_{j0}$ , 继续为下一个 SN 生成通信路径。
- Step6  $r_{j+1}$  重复 Step4 至 Step5,  $r_{j+1}$  的通信容量更新为  $r_j$  的最大通信容量 -  $r_j$  目前的通信容量, 即令  $RNCAPACITY[r_{j+1}] = RNCAPACITY[r_j]$ 。
- Step7 若  $d = \emptyset$ , 算法结束; 否则重新从  $d$  中第一个距离对应的 SN 开始, 重复上述步骤。

### 3 负载均衡

#### 3.1 负载均衡及具体实现方法

本文中的负载均衡指每条路径上的 RN 转发的数据流量应近乎平均, 避免某条路径上的任务过重、某条路径上的任务过少, 防止出现两极分化的情况。

在 2.1 节中谈到通信容量上限, 大多数文献都默认其为节点的最大数据转发量。然而考虑到负载均衡的问题, 我们认为节点的通信容量的上限应是一个依据网络状况不断更新变化的节点的平均负载。下面给出两种更新该平均值的具体方法:

##### (1) 阈值法

- Step1 初始化  $capacity\_total[n] = \{0\}$ ,  $k=0$ ,  $capacity\_max = \max$ ,  $capacity\_min = 1$ , 阈值  $level$  (具体取值将会在第 4 节中详细给出), 其中  $\max$  是输入条件中给出的每个节点的实际通信容量上限。
- Step2 根据贪婪寻优算法为  $SN_i$  ( $1 \leq i \leq n$ ) 生成到 sink 的路径, 其中  $n$  为传感器节点总数。
- Step3 计算当前 SN 的通信路径上各节点的通信容量耗费情况, 即若此路径上各节点连接有  $n$  个 SN, 则  $capacity\_total[k++] = n$ 。
- Step4 比较  $capacity\_max$ ,  $capacity\_min$  与  $capacity\_total$  数组中各元素的大小, 将最大的赋给  $capacity\_max$ , 最小的赋给  $capacity\_min$ 。
- Step5 更新节点通信容量的上限  $capacity = (capacity\_max + capacity\_min) * level$ 。
- Step6 重复以上步骤, 直至所有的 SN 均处理完毕。

阈值法核心在于: 每次选择网络中所有通信路径的最大负载和最小负载, 然后根据给定的阈值取一个折中值, 作为整体网络的平均负载, 以后每个节点的通信容量上限即为该平均负载。关于阈值的取值将会在第 4 节中详细给出。

##### (2) 平均值法

- Step1 初始化  $capacity\_initial = capacity$ ,  $capacity\_total[n] = \{0\}$ ,  $k=0$ , 其中  $capacity$  是每个节点的实际通信容量上限。
- Step2 根据贪婪寻优算法为  $SN_i$  ( $1 \leq i \leq n$ ) 生成到 sink 的路径, 其中  $n$  为传感器节点总数。

- Step3 计算当前 SN 的通信路径上各节点的通信容量耗费情况, 即若此路径上各节点连接有  $n$  个 SN, 则  $capacity\_total[k++] = n$ 。
- Step4  $capacity\_initial = capacity\_initial + capacity\_total[j]$  ( $0 \leq j \leq k$ )。
- Step5 更新通信容量的上限, 即  $capacity = capacity\_initial / (k+1)$ 。
- Step6 重复以上步骤, 直至所有的 SN 均处理完毕。

平均值法的核心在于: 每次求网络中所有通信路径的负载平均值, 以该平均值作为下一个 SN 生成通信路径时各节点的通信容量的上限。

#### 3.2 优化贪婪寻优算法

负载均衡与 RN 数目是一个难两全的目标。在追求负载均衡的同时, RN 数目必会有所增加。为了尽量减少 RN 数目, 我们增加一条贪婪寻优的规则, 以此构成一种改进的贪婪寻优算法, 简称 IGWLB (Improved Greedy Optimization Algorithm)。

##### (1) 算法描述

此方法重在修改贪婪寻优算法中的第三步规则, 具体如下:

输入: 监测区域、SN:  $S = \{s_1, s_2, \dots, s_n\}$ 、候选 RN:  $R = \{r_1, r_2, \dots, r_m\}$ 、sink 节点  $s_0$ 、网格间距  $L$  及通信半径  $R$  等信息

输出: RN 数目和位置

- Step1 为每个  $r_i$  ( $1 \leq i \leq m$ ) 选择下一跳时, 在其通信范围内搜索候选中继节点  $r_j$  ( $1 \leq j \leq m, j \neq i$ );
- Step2 去除  $d(r_i, s_0) \leq d(r_j, s_0)$  的候选 RN;
- Step3 在剩余的候选 RN 内查找是否有  $r_j$  在之前已被选择, 即是否  $chosen[r_j] = 1$ 。我们将最初  $chosen$  数组初始化为零;
- Step4 若存在该  $r_j$ , 使  $chosen[r_j] = 1$ , 查看其通信容量是否超过上限。若超过上限, 该  $r_i$  选择满足最近机制且离  $s_0$  最近的 RN (设为  $r_{j'} (1 \leq j' \leq m, j' \neq i, j' \neq j)$ ) 作为下一跳, 并令  $chosen[r_{j'}] = 1$ ; 反之, 该  $r_i$  选择  $r_j$  作为下一跳, 并令  $RNCAPACITY[r_j] + 1, RNCAPACITY$  为节点通信容量。

##### (2) 对比测试

给出如表 1、表 2 的输入数据。表 2 中 SN 的坐标是相对于 sink 的相对坐标。SN 用  $(x, y)$  表示, 若  $x > 0$ , 则该节点在水平方向上处于 sink 的右边;  $x < 0$  则反之。若  $y > 0$ , 表示该节点在垂直方向上处于 sink 节点的下方;  $y < 0$  则反之。

表 1 节点限制条件

监测区域 (m)	节点的通信容量	网格间距 L (m)	通信半径 R (m)	sink 节点坐标
1000 * 1000	6	30	80	80, 70

表 2 SN 坐标位置

节点序号	$x_i, y_i$	节点序号	$x_i, y_i$
SN <sub>1</sub>	40, -800	SN <sub>6</sub>	640, -500
SN <sub>2</sub>	200, 40	SN <sub>7</sub>	700, -640
SN <sub>3</sub>	200, -800	SN <sub>8</sub>	900, -400
SN <sub>4</sub>	300, -400	SN <sub>9</sub>	900, -800
SN <sub>5</sub>	600, -700	SN <sub>10</sub>	910, -700

表 3 GWLB 及 IGWLB 输出对比

算法	所需 RN 数目
GWLB: 不考虑负载均衡的传统贪婪寻优算法 B	63
IGWLB: 不考虑负载均衡的优化贪婪寻优算法	58

现分别用传统贪婪寻优算法及优化贪婪寻优算法予以测

试,得如图 4 的输出(图表中的算法名称我们用英文缩写),对应耗费 RN 的数目如表 3 所列。

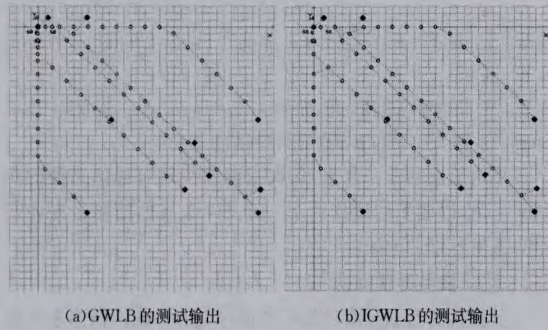


图 4 GWLB 及 IGWLB 的测试输出

图 4 及第 4 节的图 5 中,黑色菱形代表传感器节点,坐标轴交点处的圆圈代表网关节点,其他圆圈代表中继节点。各条从传感器节点到网关的连线,代表传感器节点的通信路径(测试平台为自身设计的中继节点布局系统)。

图 4 及表 3 呈现的对比数据表明,相比于传统贪婪寻优算法,优化贪婪寻优算法减少了 5 个 RN。如果当前 SN 数目众多且通信容量足够大,则运用优化贪婪寻优算法为 SN 生成通信路径时,会节省很多不必要的 RN。

#### 4 多节点下算法综合实验验证

在数学中常用方差来衡量一组数据的波动大小,方差越小整组数据就越稳定,偏离平均值的程度就越小。故我们用方差来评判网络中的负载均衡情况。网络中负载平均值以及对应的方差公式如下:

$$ave = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (9)$$

$$s^2 = \frac{(x_1 - ave)^2 + (x_2 - ave)^2 + \dots + (x_n - ave)^2}{n} \quad (10)$$

其中,  $n$  代表网络中主路径的条数;  $x_1, x_2, \dots, x_n$  对应每条主

路径的负载。

在 3.1 节中提出使用阈值法动态控制网络的平均负载,显然阈值的选取是个十分关键的问题。由 3.1 节阈值法中的具体步骤可知,每一步得出的  $(capacity\_max + capacity\_min) * level$  作为节点新的通信容量上限。分析得到式(11)一式(13):

$$capacity\_min \leq (capacity\_max + capacity\_min) * level \leq max \quad (11)$$

$$capacity\_min \leq capacity\_max \leq max \quad (12)$$

$$1 \leq capacity\_min \leq max \quad (13)$$

综上所述得式(14):

$$\frac{1}{max+1} \leq level \leq \frac{max}{max+1} \quad (14)$$

由式(14)我们求得了阈值  $level$  的合理取值范围,为能取得一个理想的阈值,我们以表 5、表 6 为输入条件,取不同的阈值根据传统贪婪寻优算法 GWLB 进行对比测试,测试结果如表 4 所列。此处,表 5 的输入数据表明  $max=8$ ,结合不等式(14)得:  $0.11 \approx 1/9 \leq level \leq 8/9 \approx 0.89$ 。

表 4 不同阈值的综合比较

level	所需 RN 数目	平均负载: ave	负载方差 $S^2$
0.2	235	—	—
0.3	197	—	—
0.4	154	3.9	2.69
0.5	124	4	2.6
0.6	123	4.5	4.47
0.7	106	5.6	3.39
0.8	103	7	0.75

注:表中的“—”代表存在生成路径失败的节点,原因是网关附近的所有节点负载已达到上限  $max$ 。

表 5 节点限制条件

监测区域(m)	节点通信容量	网格间距 L(m)	通信半径 R(m)	sink 节点坐标
1000 * 1000	8	30	80	80,70

表 6 SN 坐标位置

节点序号	$x_i, y_i$	节点序号	$x_i, y_i$	节点序号	$x_i, y_i$	节点序号	$x_i, y_i$
SN1	350,20	SN11	567,-678	SN21	383,-481	SN31	286,-286
SN2	420,-900	SN12	900,-800	SN22	218,-287	SN32	630,-144
SN3	800,-100	SN13	910,-700	SN23	453,-529	SN33	700,-44
SN4	900,-200	SN14	900,-400	SN24	233,20	SN34	640,-310
SN5	763,-800	SN15	700,-640	SN25	407,-747	SN35	-36,-316
SN6	780,-620	SN16	700,70	SN26	-86,-649	SN36	73,-329
SN7	750,-600	SN17	600,-700	SN27	741,-415	SN37	724,-378
SN8	700,-145	SN18	640,-500	SN28	472,-330	SN38	856,-178
SN9	670,-650	SN19	300,-400	SN29	566,-480	SN39	264,56
SN10	600,-790	SN20	490,-200	SN30	369,-429	SN40	243,-100

从表 4 的对比结果可看出,从所需节点数目来看,当  $level \geq 0.5 \approx 1/2 \times 0.89$  时,最终效果较好。从平均负载和方差角度看,当  $level=0.8$  时,效果最好,然而对应的平均负载几乎与给定的  $max$  相同,使得节点剩余通信容量太少,后期节点布局的可扩展性不好。

经过多次反复试验,我们建议  $level$  最理想的取值范围应满足式(15)所示的结论,其中  $MAX(level)$  表示不等式(14)给出的  $level$  的上限:

$$level \in [MAX(level) - 0.2, MAX(level)] \quad (15)$$

为能更好地体现考虑负载均衡后网络的整体负载情况,

同样以表 5、表 6 为输入,分别用阈值法传统贪婪寻优算法、平均值法传统贪婪寻优算法、阈值法优化贪婪寻优算法、平均值法优化贪婪寻优算法与不考虑负载均衡的传统贪婪寻优算法进行对比测试,如图 5 所示。

基于式(15),我们取  $level=0.75=3/4$ ,以表 4、表 5 为输入,有:图 5(a)中共有 8 条主路径,各路径的负载分别为 8、8、6、5、4、4、2、1;图 5(b)中共有 8 条主路径,各路径的负载分别为 8、6、6、5、4、4、4、3;图 5(c)中共有 8 条主路径,各路径的负载分别为 8、8、6、4、4、4、3、2、1;图 5(d)中共有 8 条主路径,各路径的负载分别为 8、6、6、5、4、4、4、2;图 5(e)中共有 8 条主

路径,各路径的负载分别为 8,8,6,4,4,2,1。

用式(9)、式(10)分别求图 5(a)~图 5(e)对应的各条路径的平均通信容量(即平均负载)以及对应的负载方差,如表 7 所列。

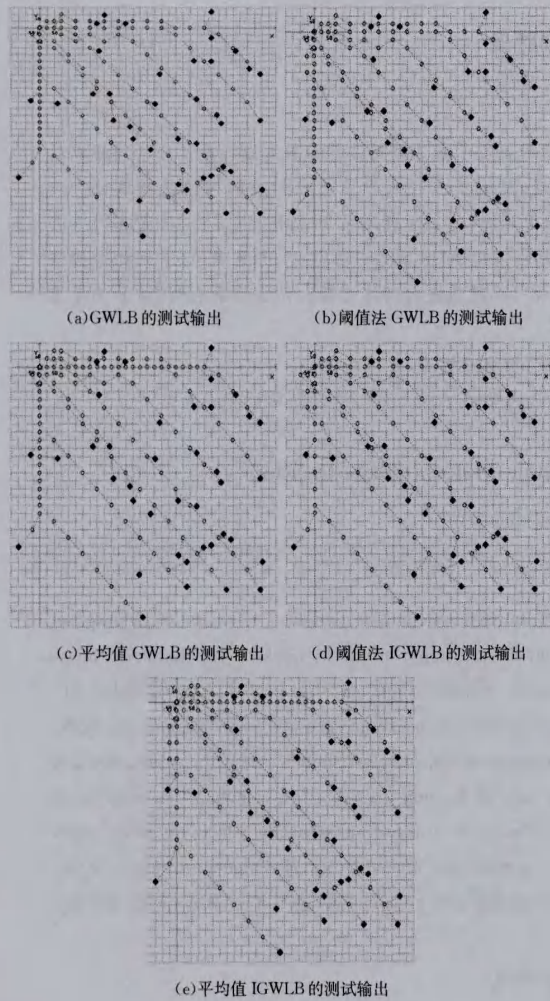


图 5 GWLB 等的测试输出

表 7 5 种算法综合比较

算法	所需 RN 数目	平均负载 ave	负载方差 $S^2$
GWLB	109	4.75	5.61
阈值法 GWLB	113	5	2.25
平均值法 GWLB	121	4.45	5.36
阈值法 IGWL	106	4.86	2.64
平均值法 IGWL	117	4.63	5.73

对比表 7 中的方差数据,发现相较于不考虑负载的传统贪婪寻优算法 GWLB,考虑负载均衡的优化贪婪寻优算法能使整个网络的负载更平均。

对比传统贪婪寻优算法与优化贪婪寻优算法,两者对应的负载方差相差无几,但后者耗费的节点数目更少。

对比阈值法以及平均值法,我们发现在选取一个理想的阈值的情况下,阈值法更新通信容量使路径的负载方差接近于平均值法对应的方差的一半,具有更好的效果。而平均值法不需要事先设置阈值,减少了人机交互的次数,算法简单稳定。

**结束语** 本文提出了阈值法及平均值法来动态更新网络中各路径的负载,同时提出一种优化贪婪寻优算法来尽量减少所需节点的数目。测试结果表明,使用考虑负载均衡的优化贪婪寻优算法进行节点布局,能够使整个网络的负载更为平均,且算法耗费时间与传统的不考虑负载的贪婪寻优算法相差无几,这更适合实际应用中动态复杂的无线传感器网络。

## 参考文献

- [1] Misra S, Hong S D, Xue Guo-liang, et al. Constrained relay node placement in wireless sensor networks; formulation and approximations[J]. IEEE/ACM Trans on Networking, 2010, 18(2): 4447
- [2] Bari A, Da Teng, Jaekel A. Optimal Relay Node Placement in Hierarchical Sensor Networks with Mobile Data Collector[C]// Proceedings of 18th International Conference on Computer Communications and Networks, Canada, 2009: 1-6
- [3] 樊勇,张晓彤,万亚东,等. 实现能量均衡消耗的传感器网络节点摆放策略[J]. 计算机工程, 2007, 33(16): 11-13  
Fan Yong, Zhang Xiao-tong, Wan Ya-dong, et al. The placement strategy of sensor network nodes to achieve balanced energy consumption[J]. Computer engineering, 2007, 33(16): 11-13
- [4] 吴谋,张晴. 自适应的移动 Ad hoc 网络贪婪地理路由协议[J]. 计算机应用研究, 2010, 27(8): 3124-3126  
Wu Mou, Zhang Qing. Adaptive greedy geographic routing protocol in mobile Ad Hoc Networks[J]. Application Research of computers, 2010, 27(8): 3124-3126
- [5] 蒋毅,张若南,史浩山. 一种基于地理位置的无线传感器网络安全路由协议[J]. 西北工业大学学报, 2012, 30(1): 11-16  
Jiang Yi, Zhang Ruo-nan, Shi Hao-shan. A secure routing protocol for wireless sensor networks based on geographical position [J]. Journal of Northwestern Polytechnical University, 2012, 30(1): 11-16
- [6] 王翥,王祁,魏德宝,等. 无线传感器网络中继节点布局算法[J]. 物理学报, 2012, 61(12): 1-8  
Wang Zhu, Wang Qi, Wei De-bao, et al. Relay node placement algorithm in wireless sensor network[J]. Journal of Physics, 2012, 61(12): 1-8
- [7] Lin G H, Xue Guo-liang. Steiner tree problem with minimum number of Steiner points and bounded edge length[J]. Information Processing Letters, 1999, 69(2): 53-57
- [8] Lloyd E L, XUE Guo-liang. Relay node placement in wireless sensor networks[J]. IEEE Trans on Computers, 2007, 56(1): 134-138
- [9] Lu Ke-zhong, Liu Gang, Mao Rui, et al. Relay node placement based on balancing power consumption in wireless sensor networks[J]. IET Wireless Sensor Systems, 2011, 1(1): 1-6
- [10] Misra S, Majd N E, Huang Hong. Constrained relay node placement in energy harvesting wireless sensor networks[C]// Proc of the 8th IEEE International Conference on Mobile Ad hoc and Sensor Systems. Washington DC: IEEE Computer Society, 2011: 25-34