

引用格式:陶洋,王进,潘蕾娜,等. 能量获取无线传感网能耗均衡分簇路由算法[J]. 微电子学与计算机,2019,36(10):83-90.
[TAO Y, WANG J, PAN L N, et al. Energy balanced clustering routing algorithm for energy harvesting wireless sensor network[J]. Microelectronics & Computer, 2019, 36(10): 83-90.]

能量获取无线传感网能耗均衡分簇路由算法

陶 洋,王 进,潘蕾娜,杨 柳

(重庆邮电大学 通信与信息工程学院,重庆 400065)

摘 要: 针对大规模能量获取无线传感器网络能耗不均衡产生能量空洞现象,进而造成能源利用效率较低的问题,提出一种基于改进 GSA 的能耗均衡多跳分簇路由算法(VPMCR). 通过合理的区域划分策略,控制网络各个区域簇头节点的数量,以均衡簇头节点能耗. 通过最小化簇间数据传输总消耗得到最优的网络单元数. 针对 GSA 算法收敛速度慢和容易陷入局部极值的问题,提出改进策略,利用改进 GSA 算法给出一种能耗均衡簇间路由方法,将能耗均衡性、能量状态等多种条件纳入适应度函数设计中,提高了路由阶段能耗均衡性. 通过仿真验证,算法在平衡能量消耗、提升网络吞吐量方面的性能优于 PHC、AEHAC 和 NEEC 算法.

关键词: 能量获取无线传感器网络;GSA 算法;能耗均衡;分簇路由

中图分类号: TP393

文献标识码: A

文章编号: 1000—7180(2019)10—0083—08

Energy balanced clustering routing algorithm for energy harvesting wireless sensor network

TAO Yang, WANG Jin, PAN Lei-na, YANG Liu

(School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: In order to solve the problem of energy imbalance in large-scale energy harvesting wireless sensor network, resulting in energy hole phenomenon, which leads to low energy utilization efficiency, an improved energy-balanced multi-hop clustering routing algorithm (VPMCR) based on improved GSA is proposed. Through a reasonable regional division strategy, the number of cluster head nodes in each area of the network is controlled to balance the energy consumption of the cluster head nodes. The optimal number of network elements is obtained by minimizing the total consumption of data transmission between clusters. Aiming at the problem that the GSA algorithm is slow in convergence and easy to fall into local extremum, an improved strategy is proposed. An improved energy-balanced inter-cluster routing strategy is proposed by using the improved GSA algorithm, and various conditions such as energy balance and energy state are incorporated into the fitness. In the function design, the energy balance in the routing phase is improved. Through simulation verification, the performance of the algorithm in balancing energy consumption and improving network throughput is better than PHC, EBHAC and NEEC algorithms.

Key words: energy harvesting wireless sensor network; GSA algorithm; energy balance; clustered routing

1 引言

在传统的无线传感器网络(WSN)中,节点最终都会不可避免地耗尽能量而停止运作^[1]. 为了满足未来对传感器节点不断增长的应用需求,开始为节点配备各种环境能量获取装置,构成能量获取无线传感网络(energy harvesting wireless sensor network, EN-WSN)^[2]. 然而自然能源供应不稳定,振动能获取节点受到时间和不同位置的影响^[3-6]. 在这种能量获取场景中,针对传统 WSN 设计的分簇路由协议已经不能满足需求.

文献[7]提出 PHC 算法,在振动能获取下,考虑节点剩余能量和上一轮能量获取状态进行簇头选取,在小规模网络传输时能够有效均衡能耗.

EP-LEACH 协议利用一个能量势能函数来估算每个节点的能量获取能力和剩余能量的大小,并以此计算节点成为簇头的概率,协议最终延长了网络的生存周期,同时增加网络吞吐量.

文献[8]提出 AEHAC 算法,在振动能获取下,不再对节点成为簇头次数进行限制,使得获取能量和剩余能量较多节点可以多次当选簇头. 然而上述算法均采用簇间单跳传输,在大规模传输时能耗效率较低,文献[9]提出一种能量获取下非均匀多跳分簇路由算法 NEEC,将网络划分四层,根据能量状态和获取状态计算节点延迟时间竞选簇头,但是该算法簇间路由阶段仅仅选择剩余能量多的节点作为下一跳,造成基站附近节点能耗不均衡,降低了算法网络吞吐量.

本文提出一种振动能获取下 EH-WSN 能耗均衡多跳分簇路由算法,通过合理的区域划分平衡簇头节点能耗,在簇头选取上提升补给能力较强节点成为簇头的概率,利用改进 GSA 算法提出能耗均衡簇间路由策略,从而达到了平衡网络能耗提升吞吐量的目的.

2 系统模型

2.1 网络和能量补给模型

如图 1 所示,本文网络模型为面积为 S 的矩形区域,网络内节点随机均匀分布,监控区域又被细分为 K 个等尺寸的矩形子单元,根据单元划分簇头节点在下一单元中选择其它簇头节点作为下一跳路由节点,合作地将数据传输到基站节点.

2.2 能量消耗模型

建立传感器节点之间的无线信道传输模型,本

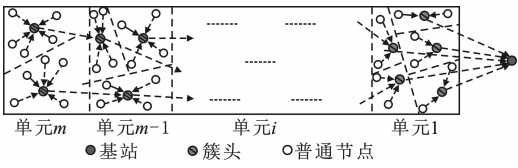


图 1 网络模型图

文采用文献[8]中的能耗模型如下式所示:

$$E_{Tx}(k,d)=\begin{cases} k\xi+k\varepsilon_{fs}d^2 & d\leqslant d_0 \\ k\xi+k\varepsilon_{amp}d^4 & d>d_0 \end{cases} \quad (1)$$

具体请参见符号定义表 1.

本文采用振动能量补给模型,文献[9]提出振动能量补给受地理位置影响较大,即同一个时刻网络中不同位置由于振动强度不同,节点能量采集率也不同,具有位置上的非对称特性. 本文采用文献[10]中正态分布的补给模型 (μ,σ^2) ,其中 $\mu=275\text{ }\mu\text{w}/\text{cm}^3$, $\sigma^2=100$,电源为体积 9 cm^3 的电池.

表 1 符号定义

变量	含义
N	监测区域内节点总个数
S	监测区域的面积
l	最优矩形单元的长度
m	最优单元个数
n	单元 i 中的节点个数
C_i	单元 i 中的簇头个数
$\bar{d}$	单元 i 和单元 $i+1$ 中两个簇头的平均距离
d_0	距离阈值
ρ	传感器的密度
q	每轮数据的传输量
ξ	发送电路或者接受电路发送或接收 1bit 数据消耗的能量
ε_{fs}	自由空间信道模型的功率放大系数
ω	融合 1bit 数据消耗的能量
α	融合率

3 理论分析

本节通过能耗分析对网络进行区域划分得到网络各区域簇头节点数量、网络单元数等信息. 根据不同单元间能耗均衡得到簇头数量的迭代关系,利用最小化簇间数据传输总消耗得到最优的网络单元数.

3.1 不同单元能耗平衡

在上面的分层网络模型下,单元 i 中簇头节点的能耗为:

(1) 接收簇内、簇间和融合簇内数据能耗:

$$E_r(i) = \frac{((m-i)anq + (n-C_i)q)\xi + nq\omega}{C_i} \quad (2)$$

上式分子第一项表示接收前 $m-i$ 个单元和簇内节点数据的平均能耗,第二项表示融合簇内数据的平均能耗。

(2) 中继簇内和簇间多跳传输数据能耗:

$$E_t(i) = \frac{(m-i+1)anq}{C_i} (\xi + \epsilon_{fs} \bar{d}_i^2) \quad (3)$$

单元之间的数据传输模型满足自由空间传输模型,其中, ϵ_{fs} 为自由空间信道模型的功率放大系数, $\bar{d}_i$ 为单元 i 和单元 $i-1$ 中簇头的平均距离:

$$\bar{d}_i = l/m \quad (4)$$

所以单元 i 中簇头节点的平均能耗为:

$$E_{CH}(i) = \frac{((m-i)anq + (n-C_i)q)\xi + nq\omega}{C_i} + \frac{(m-i+1)anq}{C_i} (\xi + \epsilon_{fs} \bar{d}_i^2) \quad (5)$$

为了均衡网络中不同单元内的簇头节点能耗,令 $E_{CH}(i) = E_{CH}(i+1)$, ($i = 1, 2, \dots, m-1$), 根据公式(5)可以得到不同单元间簇头节点之间的关系:

$$c_{i+1} = c_i \left(1 - \frac{2a\xi + a\epsilon_{fs} \bar{d}_i^2}{(2m-2i+1)a\xi + \xi + \omega + (m-i+1)a\epsilon_{fs} \bar{d}_i^2} \right) \quad (6)$$

网络内第一个单元的簇头数量可以用最小化单元内节点一轮传输总能耗计算得到,网络内第一个单元传输总能耗 $E_{total}(1)$ 为:

$$E_{total}(1) = C_1 E_{CH}(1) + (n-C_1) E_{CM}(1) \quad (7)$$

$E_{CM}(1)$ 为第一个单元中普通节点发送 $qbit$ 数据消耗的能量:

$$E_{CM}(1) = q(\xi + \epsilon_{fs} \hat{d}^2) \quad (8)$$

式中, $\hat{d}$ 是第一个单元中节点到簇头的平均距离,每个簇的平均大小为 S/mC_1 ,则式(8)可化简为:

$$E_{CM}(1) = q(\xi + \epsilon_{fs} \frac{S}{2\pi m C_1}) \quad (9)$$

联立公式(5)、(7)、(9)可以得到 $E_{total}(1)$, 对 $E_{total}(1)$ 求解关于 C_1 的一阶导数,如下式:

$$\frac{dE_{total}(1)}{dC_1} = -2q\xi - \frac{nq\epsilon_{fs}S}{C_1^2 2\pi m} \quad (10)$$

显然 $\frac{dE_{total}(1)}{dC_1} < 0$, 函数单调递减,故 C_1 应该取最

大值,故有下式:

$$C_1 = n \quad (11)$$

由公式(6)和(11)可知为了确定网络中每个单元的簇头节点个数,就必须确定网络中的单元数 m 。

3.2 网络最优单元数计算

由于网络中节点在数据传输中消耗的能量远远高于接收数据的能耗,且网络中划分单元数主要影响簇间数据传输能耗,这里用最小化簇间数据传输总消耗得到最优的网络单元数。网络中所有簇头节点转发簇间数据总能耗如下式:

$$E_{CH_total} = \sum_{i=1}^m (m-i+1)anq (\xi + \epsilon_{fs} \bar{d}_i^2) = \frac{(m+1)aSq}{2} (\xi + \epsilon_{fs} \frac{l^2}{m^2}) \quad (12)$$

对 E_{CH_total} 求解关于 m 的一阶导数,令偏导数为0,得到如下式:

$$\frac{1}{m^2} + \frac{2}{m^3} = \frac{\xi}{\epsilon_{fs} l^2} \quad (13)$$

由于本文不同单元间簇间路由阶段采用的是自由空间传输模型,对于单元 i 中的簇头节点从单元 $i-1$ 中选择簇头作为路由节点,最靠近外边界的节点有最长的数据传输距离,最长数据传输距离要小于自由空间传输模型的距离阈值,表示如下:

$$\frac{2l}{m} \leq d_0 \quad (14)$$

综上所述,求解最优单元数问题转化为如下最优化问题:

$$\begin{aligned} \min \varphi(m) &= \left| \frac{1}{m^2} + \frac{2}{m^3} - \frac{\xi}{\epsilon_{fs} l^2} \right| \\ \text{subject to:} \\ \frac{2l}{m} &\leq d_0 \\ m &\in [1, l] \\ m &\in N_+ \end{aligned} \quad (15)$$

4 非均匀分簇算法设计

4.1 簇头生成策略

本文簇头生成策略与 EP-LEACH 算法相似,在 EP-LEACH 算法中依赖于节点获取能力预测,由于振动能具有随机性和位置相关性是不可预测的,这里使用节点上一次成为簇头节点的轮次开始到当前轮总获取的能量来衡量节点的获取能力,如下式所示:

$$P_k(i) = \frac{e^{\lambda_k(i)[R_k(i) - \mu_k(i) + \sum_{r=r-n+1}^r E_{\text{hav}}(i,r)]}}{1 + e^{\lambda_k(i)[R_k(i) - \mu_k(i) + \sum_{r=r-n+1}^r E_{\text{hav}}(i,r)]}} \quad (16)$$

式中, $P_k(i)$ 表示节点 k 在第 i 个时隙的能量潜能; $R_k(i)$ 表示节点 k 第 i 轮剩余能量; $E_{\text{hav}}(i,r)$ 表示节点 i 在第 r 轮获取到的能量补充. $R_k(i) + \sum_{r=r-n+1}^r E_{\text{hav}}(i,r)$ 表示节点 k 在第 i 轮剩余的能量与节点上一次成为簇头开始到当前轮次总共补给的能量之和, 代表节点 i 的能量潜力. $\mu_k(i) = E[R_k(i) + \sum_{r=r-n+1}^r E_{\text{hav}}(i,r)]$ 和 $\lambda_k(i) = E[R_k(i) + \sum_{r=r-n+1}^r E_{\text{hav}}(i,r)]$ 分别表示能量潜力的平均值和标准差的倒数, 计算方式如下:

$$\mu_k(i) = E[R_k(i) + \sum_{r=r-n+1}^r E_{\text{hav}}(i,r)] \quad (17)$$

$$\lambda_k(i) = \frac{1}{\sqrt{\text{Var}[R_k(i) + \sum_{r=r-n+1}^r E_{\text{hav}}(i,r)]}} \quad (18)$$

结合上面的能量潜力函数, 给出簇头选取的阈值如下:

$$T_k(i) = \frac{P_k(i)}{\sum_{r \in \zeta_k} P_k(i)} \times |\zeta_k| \times \frac{c}{n} \quad (19)$$

式中, $\zeta_k = \{r | D(r,k) < D_t\}$, 代表网络中同一单元内与节点 k 之间的距离小于 D_t 的节点集合; $|\zeta_k|$ 代表集合内节点的数量; c 代表当前单元内簇头数量; n 代表单元内的传感器节点总数.

4.2 非均匀成簇

本小节详细介绍算法非均匀成簇流程, 在初始化阶段, 基站广播消息包, 包含网络最优单元数量、不同单元簇头数量, 节点根据接收信号的强度估算自身与基站之间的距离, 根据距离和单元数量确定所处的单元. 每个节点在 0 到 1 之间产生一个随机数, 根据簇头生成策略计算自身阈值. 为了避免选取出来的簇头节点距离过近和能量等级低的节点成为簇头, 本文簇头选取分为两个阶段, 临时簇头选取阶段节点产生随机数如果小于阈值, 则成为临时簇头节点.

之后临时簇头节点开始计算成簇后的簇半径, 在自身所处网络单元簇半径内广播当选临时簇头消息“temporaryCluster”, (节点 ID、所在单元) 簇头节点同时接收其他临时簇头节点发送来的当选消息, 网络内每一个临时簇头节点得到自身簇半径内所有邻居簇头节点, 如果自身簇半径内没有邻居簇首, 则

该临时簇头节点成为真正簇头节点. 否则, 临时簇头节点将会随机地选择一个时间片内广播当选簇头消息, 临时簇头节点如果在自己选择的时间片到达之前接收到其他节点的簇头当选消息, 就放弃竞选成为普通节点.

至此, 所有簇头节点已经确定, 簇头节点在自身单元内广播成簇消息“join”(节点 ID, 竞争半径, 单元数), 普通节点根据接近距离选择距离最近的本单元簇头来加入形成簇.

5 能耗均衡簇间路由

引力搜索算法(GSA)广泛应用于求解静态优化问题, 其本质是模仿物理学当中的万有引力定律, 通过粒子之间的相互吸引作用运动来完成对最优解的搜索. 本节研究利用 GSA 对簇间路由进行优化从而在路由阶段均衡簇头能耗. GSA 原理如下:

假设空间中存在 N 个粒子, 第 i 个粒子如下式表示:

$$A_i = (x_i^1, x_i^2(t), x_i^3(t), \dots, x_i^m), i = 1, 2, \dots, N \quad (20)$$

在 t 时刻两个粒子之间的作用力可以用下式表示:

$$F_{ij}^k(t) = G(t) \frac{M_{pi}(t) \cdot M_{aj}(t)}{R_{ij}(t) + \epsilon} (x_j^k(t) - x_i^k(t)) \quad (21)$$

式中, M_{pi} 和 M_{aj} 分别表示粒子 i 和 j 的质量; ϵ 为一极小常量; R_{ij} 为粒子 i 和 j 之间欧氏距离; $G(t)$ 为 t 时刻万有引力常数值, 计算方法如下:

$$G(t) = G_0 \cdot e^{-\alpha t/T} \quad (22)$$

在 t 时刻粒子 i 在 k 维上面收到其他粒子的合力为:

$$F_i^k(t) = \sum_{j=1, j \neq i}^N \text{rank}_j F_{ij}^k(t) \quad (23)$$

式中, rank_j 为 $[0, 1]$ 之间随机数, 定义 t 时刻粒子 i 在 k 维上面的加速度为:

$$a_i^k(t) = \frac{F_i^k(t)}{M_i(t)} \quad (24)$$

进化过程中, 速度和位置更新如下式所示:

$$\begin{cases} v_i^k(t+1) = \text{rank}_i \cdot v_i^k(t) + a_i^k(t) \\ x_i^k(t+1) = x_i^k(t) + v_i^k(t+1) \end{cases} \quad (25)$$

各个粒子质量与其对应的适应度函数值有关, 使用下式更新粒子质量:

$$\begin{aligned} m_i(t) &= \frac{f_i(t) - \omega(t)}{b(t) - \omega(t)} \\ M_i(t) &= m_i(t) / \sum_{j=1}^N m_j(t) \end{aligned} \quad (26)$$

式中, $b(t)$ 和 $w(t)$ 表示种群最优适应度和最差适应度值. 引力搜索算法直接应用还存在以下问题:

(1) 种群中粒子随着进化次数的增加, 无法保证粒子受到的合力方向仍然是全局最优的, 这样搜索到最优解的速率将会降低.

(2) 迭代次数增加后, 粒子的移动步伐减小, 缺乏跳出局部极值的能力, 容易陷入局部最优解. 针对上面两个问题提出下文改进策略.

5.1 改进的引力搜索算法

5.1.1 优化的粒子进化策略

提出一种优化进化策略, 通过比较两次迭代的过程是否得到了更优解, 如果得到了更优解则计算粒子在各个维度上的变化率, 然后针对此变化率对当前的粒子继续进行相应的变化操作. 这种进化策略下, 对处于较优区域内的粒子可以快速得到最优解, 同时其它的粒子也会受到影响增大向更优解方向的进化速度. 判断粒子 i 是否开启优化进化策略的方法如下:

$$|f_i(t)| > |f_i(t-1)| \cdot (\alpha)^{\frac{t}{T}-1} \quad (27)$$

式中, $f_i(t)$ 和 $f_i(t-1)$ 分别表示粒子 i 在 t 时刻和 $t+1$ 时刻的适应度值; 由于种群在进化的前期具有较好的种群多样性, 在合力的作用下很容易到达更优的位置, 所以在前期应该更多采用 GSA 算法的合力主导粒子进化策略, 在后期由于粒子收敛到较小的区域, 寻优难度增大应该更多采用优化进化策略, 所以在式 (27) 中, 设计随时间递减的系数 $(\alpha)^{\frac{t}{T}-1}$, 其中 α 取值设置为 0.1, 在算法迭代后期有更大的几率采用优化的进化策略. 满足式 (27) 后, 计算当前粒子的进化率, 得到新的粒子, 如下式所示:

$$\bar{X}_i^k(t) = \frac{X_i^k(t)}{X_i^k(t-1)} \cdot X_i^k(t) \quad (28)$$

式中, $\frac{X_i^k(t)}{X_i^k(t-1)}$ 表示粒子 i 在 t 时刻第 k 维上面的进化率, 进而得到新的粒子, $\bar{X}_i^k(t)$ 代表根据进化率得到的解, 将其与 $X_i^k(t)$ 分别对应的目标函数值进行对比, 保留更优的解. 根据改进, 粒子不仅仅依赖于群体之间的合力吸引进化, 粒子获得了寻找更优解的机会, 同时这两种进化方式又能够提升种群进化过程之中的多样性, 降低了陷入局部最优解的几率.

5.1.2 自适应变异策略

算法中的变异操作能够提升种群的多样性和局

部开采能力. 文献[12]将变异策略引入粒子群算法, 达到了优化粒子群算法寻优能力的目的, 本节将算法变异操作引入 GSA 来提升种群的多样性, 跳出局部最优的现象. 首先设置变异触发函数, 考虑到 GSA 算法在初期多样性较好, 全局搜索能力较强, 在算法后期要减小变异几率, 所以这里设计变异触发函数表达式为:

$$TR_i^t = e^{(-\frac{d}{2} \cdot \frac{t}{T})} \cdot (r \cdot e^{(-\frac{t}{N})} + \beta) \quad (29)$$

式中, TR_i^t 表示第 t 代中粒子 i 的变异触发函数; t 为算法进化代数; d 代表算法维度; T 为算法设定的最大迭代次数; r 为取值在 $[0, 1]$ 内的随机数; N 代表种群的规模; β 为 $[0.25, 0.35]$ 之间的常数, 当计算得到的 TR_i^t 大于 0.5 时, 粒子开启变异策略. 在算法运行前期, 由于满足公式 (29) 的粒子数量较大, 采用均匀变异的方法, 粒子以较大的步幅速率移动, 不满足条件的粒子仍按照原来速率移动, 这样在前期粒子更快向最优区域接近, 提高收敛速率, 均匀变异的位置更新公式如下式所示:

$$x_i^k(t) = (1+r)x_i^k(t) \quad (30)$$

在算法的后期, 采用混合变异的方式, 根据高斯分布的概率密度函数构造高斯变异函数如下式所示:

$$gauss_i(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-\frac{(x_{best}^k(t) - x_i^k(t))^2}{2\sigma^2})} \quad (31)$$

式中, $x_{best}^k(t)$ 为 t 时刻粒子在第 k 维上面的最优值; σ 取值为搜索空间内每一维长度的 0.1 倍, 根据拉普拉斯分布构造拉普拉斯变异函数如下式所示:

$$laplace_i(t) = \frac{1}{2\lambda} e^{(-\frac{|x_{best}^k(t) - x_i^k(t)|}{\lambda})} \quad (32)$$

综上所述, 改进算法自适应变异过程粒子根据变异进行移动的位置更新公式如下:

$$x_i^k(t) = \begin{cases} (1+r)x_i^k(t) & rand \leq (1 - \frac{t}{T}) \\ [\omega \cdot gauss_i(t) + (1-\omega) \cdot laplace_i(t)]x_i^k(t) & \text{else} \end{cases} \quad (33)$$

式中, $rand$ 为 $[0, 1]$ 之间的随机数; ω 为混合变异权重, 取值为 0.3. 这种均匀变异与混合变异相结合的方式, 互相配合不仅达到了提高粒子向最优解的区域移动、跳出局部极值的能力, 而且还提升了算法的求解精度和寻优能力.

5.2 能耗均衡簇间路由

基于上述改进引力搜索算法, 提出能耗均衡簇间路由, 距离基站较远的节点选择较近单元内簇头

节点转发数据,处于第一个单元内簇头节点与基站直接通信。

首先初始化种群,每个粒子代表一种路由连接策略,在区间 $[0,1]$ 之间随机产生 N 个粒子,粒子维数与网络内成簇后的簇头节点数量相同,每个簇头根据自身的所处的单元数,请求其前一单元内且在自身通信半径内的簇头节点信息构成通信列表 $NextHops(CH_d)$ 。如下表所示, $X_{i,d}$ 表示第 i 个粒子第 d 个簇头节点选择下一跳节点的概率。

$$CH_k = Index(NextHops(CH_d), j) \quad (34)$$

式中, CH_k 代表簇头 k 从其下一跳通信列表 $NextHops(CH_d)$ 内选择第 j 个簇头节点作为其最终的下一跳节点,其中 j 的计算方法如下:

$$j = Ceil(X_{i,d} \cdot |NextHops(CH_d)|) \quad (35)$$

5.2.1 适应度函数设计

在本文路由算法中,重力质量和惯性质量通过适应度函数来计算。以传输距离、簇头势能、簇头能耗偏差三个参数作为适应度函数。

(1) 势能:能量是选择下一跳 CH 的重要。如果所有选择的 CHs 的剩余能量都很高,会增加网络的生命周期和路由持续时间更长。因此,第一个目标是簇头势能最大化:

$$g_{ep} = \sum_{i=1}^m \max_{j \in 1} [E_{res}(j) + \sum_{r=r-n+1}^r E_{hav}(j, r)] \quad (36)$$

(2) 距离:从当前 CH 到下一跳的距离和下一跳距离基站的距离,如果此距离是最小的,那么将会降低簇间路由中数据传输能耗。可以用下式表示:

$$g_{dis} = \frac{1}{\sum_{i=1}^m \min_{j \in 1} [dis(CH_i, NH_j) + dis(NH_j, BS)]} \quad (37)$$

(3) 能量均衡性:为了达到能耗均衡降低簇失败次数缓解能量空洞问题,这里将簇头节点剩余能量标准差作为适应度函数设计的一个指标,具体如下式所示:

$$g_{balance} = \frac{1}{\sqrt{\frac{1}{m} \sum_{i=1}^m [\frac{1}{m} \sum_{j=1}^m Q(j) - Q(i)]^2}} \quad (38)$$

式中, m 代表网络簇头节点数量; $Q(i)$ 表示簇头节点 i 在每轮通信过后的剩余能量,簇间通信中簇头能耗根据下式计算得到:

$$Q(i) = E(i) - \xi(l_{data} \cdot PacketNum(i)) - (\xi + \epsilon_{fs} d_{next}^2(i))(l_{data} \cdot PacketNum(i) + 1) \quad (39)$$

式中, $E(i)$ 为簇间通信开始之前簇头节点 i 剩余能量; l_{data} 为数据报大小,综上所述,本文能耗均衡簇间路由的适应度函数如下式所示:

$$Maximize fit_i = \alpha_1 \cdot g_{ep} + \alpha_2 \cdot g_{dis} + \alpha_3 \cdot g_{balance} \quad (40)$$

5.2.2 路由算法伪代码

本节给出本文基于改进GSA算法的能耗均衡簇间路由由算法伪代码如下表所示:

表2 簇间路由算法伪代码

算法1:GSA-Based Routing

Inputs:(1)Set of CH nodes $CH = (CH_1, CH_2, \dots, CH_m)$

(2)Predefined number of agents N_p

(3)Energy of CH_i ,and Distance of CH_i

(4)Dimension of agent = Number of CHs m

Result:Routing paths for all CHs

1. Initialize the agent $A_i, \forall i, 1 \leq i \leq N_p, A_i = [x_i^1(t), x_i^2(t), x_i^3(t), \dots, x_i^m(t)]$

2. Each cluster head builds next hop communication list $nextHops(CH_d)$

3. for($t = 1$ to T)

4. for($i = 1$ to N_p) do

5. Calculate Fitness(A_i) $f_i(t)$ Using Eq. 40

6. Update the best and worst fitness of all agents

7. Caculate mass(A_i) Using Eq. 26

8. Calculate force of Agent A_i Using Eq. 23

9. Calculate acceleration $a_i^k(t)$ of Agent A_i Using Eq. 24

10. Calculate the position $x_i^k(t+1)$ of Agent A_i Using Eq. 25

11. if ($|f_i(t)| > |f_i(t-1)| \cdot (\alpha)^{\frac{t}{T-1}}$) do

12. Get predicted values $\bar{X}_i^k(t)$ based on evolution rate Using Eq. 28

13. Compare and keep better agent

14. else

15. The position of Agent A_i is $x_i^k(t+1)$

16. end if

17. if (Eq. 29 > 0.5) do

18. Perform variation Using Eq. 33

19. end if

20. Update the position and velocity of x_i

21. end for

22. end for

23. Caculate the route for each CH using $maximum(fitness(A_i))$

6 仿真设计与结果分析

这一节中,在相同的能量供应规则下,使用

MATLAB 仿真软件来评估本文所提出的 VPMCR 算法性能,使用 PHC, AEHAC, NEEC 三个协议和本文 VPMCR 算法进行对比. 仿真参数设置如下表 3 所示,此外 PHC 协议中簇头节点比例设置为 0.1, AEHAC 协议中簇头所占比例调整系数 ρ 初始值设置为 0.3. 通过模拟在网络活跃节点数、稳定周期数、能量消耗均衡性、网络吞吐量四个指标比较和分析各个算法之间的性能差异.

表 3 参数设置	
参数	值
监控区域宽度	200m
监控区域长度	200—500m
ρ	0.5(node/100m ²)
E_{dec}	5×10^{-8} J
ϵ_{fs}	1×10^{-11} J
ϵ_{amp}	1.3×10^{-15} J
E_{DA}	5×10^{-9} J
数据融合比 α	0.5
数据包长度 q	4000bit
控制数据报长度	400bit
初始能量 E_0	0.5J
每轮时长	20s
N_p	20
α_1	0.4
α_2	0.3
α_3	0.3
迭代次数 T	100

6.1 结果分析

6.1.1 网络稳定周期数对比

在 WSN 中通常将网络初始化到第一个节点陷入休眠状态的轮数称为网络的“稳定周期”,稳定周期对于确保整个网络的正常运行和高效的监测、数据传输非常重要. 如图 2 所示,在不同的监控区域长度下,各个算法稳定周期数对比,在各个监控区域面积下 VPMCR 算法的稳定周期数量最大,这是由于 VPMCR 算法簇头选取时保证了剩余能量和补给能量机会较大的节点成为簇头,同时采用多跳簇间传输模式,降低了簇头节点陷入休眠状态概率. 同时由于 PHC 和 AEHAC 算法均使用单跳的数据传输模式,距离基站较远的节点较快耗尽能量,所以稳定周期数较小. 在监控区域增大后距离基站较远的簇头节点能耗急剧增大,所以稳定周期数降幅明显,这也表明在大规模网络中使用多跳传输模式更加合

适. NEEC 算法虽然也采用多跳传输和非均匀分簇方法,但是该算法分层为固定的四层,在网络面积增大时,远距离传输能耗增大,同时该算法在簇间传输时仅考虑到节点剩余能量,这造成了数据传输压力集中在了距离基站较近单元部分剩余能量较多节点上面,所以稳定周期数小于 VPMCR 算法.

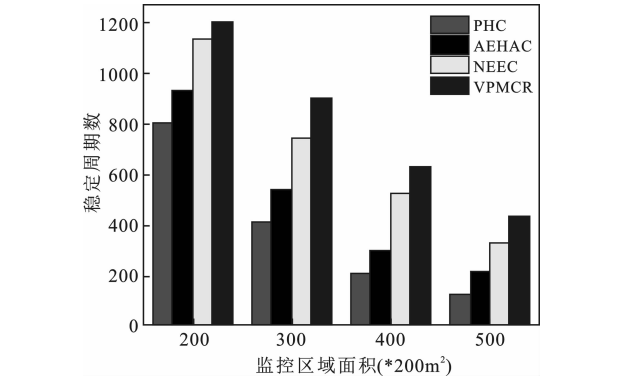


图 2 不同监控区域面积下各个算法稳定周期对比图

6.1.2 节点剩余能量方差

实验结果如图 3,随着网络轮次的增大,VPM-CR 算法能够有效地均衡各个节点之间的能量消耗. PHC 算法和 AEHAC 算法曲线斜率最大,说明节点之间能耗均衡性最差,这是由于单跳的传输模式,边缘节点能耗较大,节点之间能量消耗差异较大.

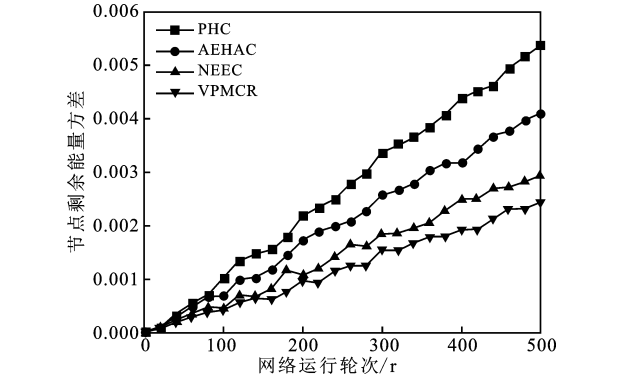


图 3 各个算法网络节点剩余能量方差变化曲线对比图

6.1.3 网络吞吐量对比

图 4 为各个算法成功发送到基站的数据包数量随网络运行轮次对比图,各个算法的吞吐量在前 250 轮中基本相同,并以相对较高的速度稳步增长,在第 250 轮数据传输之后几个算法的吞吐量存在显著差异. PHC 和 HNS 算法由于活跃节点数量最早开始下降,所以吞吐量曲线在 250 轮之后开始放缓,到 700 轮之后吞吐量基本保持不变,NEEC 和 VPMCR 算法网络吞吐量仍然以高速率增长,分别

在 534 轮和 643 轮数据传输之后开始放缓. 从整体上看 VPMCR 吞吐量明显大于其他三种算法, NEEC 算法由于簇间传输基站附近单元簇间数据传输压力集中在部分簇头节点上面, 造成能量空洞造成簇失败, 所以吞吐量低于 VPMCR 算法, VPMCR 基于改进 GSA 的能耗均衡簇间路由和区域划分策略有效降低了能量空洞现象的发生, 提升了网络吞吐量, 增加了能耗效率.

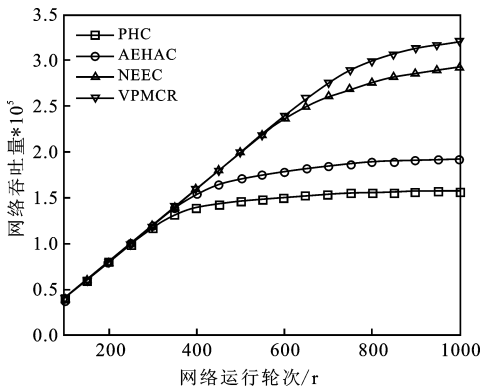


图 4 各个算法网络吞吐量对比图

7 结束语

本文提出振动能获取下能耗均衡多跳分簇路由算法, 通过合理的区域划分进行非均匀分簇, 控制网络各个区域簇头节点的数量, 均衡各个区域簇头节点能耗. 利用改进 GSA 算法提出一种能耗均衡簇间路由策略, 提高了路由阶段能耗均衡性. 通过仿真验证, 算法在平衡能量消耗、提升网络吞吐量方面的性能优于 PHC、AEHAC 和 NEEC 算法.

参考文献:

[1] YANG L, LU Y, ZHONG Y, et al. A Multi-Hop Energy Neutral Clustering Algorithm for Maximizing Network Information Gathering in Energy Harvesting Wireless Sensor Networks. [J]. Sensors, 2015, 16 (1): 26-34.

[2] DEHWAH A H, TAIB S B, SHAMMA J S, et al. Decentralized Energy and Power Estimation in Solar-Powered Wireless Sensor Networks[C]// International Conference on Distributed Computing in Sensor Systems. IEEE, 2015: 199-200.

[3] BAO X, DING G. An Routing Algorithm for Maximizing Network Performace in Energy Harvesting Wireless Sensor Network [C] // 2016 3rd International Conference on Information Science and Control Engi-

neering (ICISCE). IEEE, 2016: 1267-1270.

[4] SHAFIEIRAD H, ADVE R S, SHAHBAZPANAHI S. Opportunistic Routing in Large-Scale Energy Harvesting Sensor Networks[C]// Globecom Workshops. IEEE, 2017: 1-6.

[5] LU T, LIU G, CHANG S. Energy-efficient data sensing and routing in unreliable energy-harvesting wireless sensor network [J]. Wireless Networks, 2016, 24(3):1-15.

[6] ESCOLAR S, CHESSA S, CARRETERO J. Energy-neutral networked wireless sensors [J]. Simulation Modelling Practice & Theory, 2014, 43(2):1-15.

[7] 樊晓平, 杨玺, 刘少强, 等. 具有能量补给的无线传感器网络分簇路由协议[J]. 计算机工程, 2008, 34 (11): 120-122.

[8] MENG J, ZHANG X, DONG Y, et al. Adaptive energy-harvesting aware clustering routing protocol for Wireless Sensor Networks[C]// Communications and Networking in China (CHINACOM), 2012 7th International ICST Conference on. IEEE, 2012: 742-747.

[9] BOZORGI S M, ROSTAMI A S, HOSSEINABADI A A R, et al. A new clustering protocol for energy harvesting-wireless sensor networks[J]. Computers & Electrical Engineering, 2017, 64(2): 233-247.

[10] BAHBAHANI M S, ALSUSA E. DC-LEACH: A duty-cycle based clustering protocol for energy harvesting WSNs[C]// Wireless Communications & Mobile Computing Conference. IEEE, 2017: 974-979.

[11] YANG L, LU Y Z, ZHONG Y C, et al. An unequal cluster-based routing scheme for multi-level heterogeneous wireless sensor networks[J]. Telecommunication Systems, 2018(12): 1-16.

[12] 鲁妹颖. 粒子群优化协议的几种改进协议及应用[D]. 徐州: 中国矿业大学, 2014.

作者简介:

陶 洋 男, (1964-), 博士后, 教授. 研究方向为无线传感器网络, 异构网络, 网络管理研究及应用等.

王 进(通讯作者) 男, (1992-), 硕士研究生. 研究方向为能量获取无线传感器网络, 分簇路由.
E-mail: 1074737586@qq.com.

潘蕾娜 女, (1994-), 硕士研究生. 研究方向为无线传感器网络, 网络安全, 信任机制.

杨 柳 男, (1987-), 博士, 讲师. 研究方向为无线传感器网络、网络安全以及数据分析等.