

基于NSGA-II的无线传感网络簇首选择算法

陈作汉[†], 曹 洁, 赵付青

(兰州理工大学 计算机与通信学院, 兰州 730050)

摘 要: 延长网络生命周期是无线传感网络需要解决的主要问题之一, 拓扑控制对于延长网络生命周期具有重要意义. 针对分簇结构无线传感网络的簇首选择问题, 提出一种基于 NSGA-II 的多目标簇首选择算法. 同时考虑网络通信距离、能量消耗、负载均衡以及节点生存时间等多个优化目标, 通过理论计算确定最优簇首数量指导种群初始化, 引入正交实验机制降低搜索次数, 提高寻优效率. 实验结果表明, 所提出的算法与低功耗自适应层次分簇 (LEACH) 算法相比, 簇首分布均匀、负载均衡, 可明显延长网络的生命周期, 与标准 NSGA-II 算法相比, 可更好地提高搜索寻优效率.

关键词: 无线传感网络; 簇首选择; NSGA-II; 正交实验设计; 多目标优化

中图分类号: TP393

文献标志码: A

A NSGA-II-based algorithm for WSN cluster head selection

CHEN Zuo-han[†], CAO Jie, ZHAO Fu-qing

(College of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China)

Abstract: Prolonging the network life cycle is one of the major goals to be achieved in the wireless sensor network (WSN), and topology control is of great significance for it. To solve the problem of cluster head selection in WSN, a multi-objective cluster head selection algorithm, which is based on NSGA-II, is proposed. Many factors such as network communication distance, energy consumption, load balancing and node survival time are considered in the algorithm. The optimal number of cluster heads is determined by theoretical calculation to guide population initialization, and the orthogonal experiment mechanism is introduced to reduce the times of search and improve the efficiency of optimizing. Experiments show that the proposed algorithm, compared with the low-energy adaptive clustering hierarchy (LEACH) algorithm, has evenly distributed cluster head, balanced load and extended network life cycle. In addition, it is of higher searching efficiency than the standard NSGA-II algorithm.

Keywords: WSN; head selection; NSGA-II; orthogonal experimental design; multi-objective optimization

0 引 言

拓扑控制是无线传感网络 (WSN) 的关键问题之一, 是路由协议、MAC 协议、数据融合、时间同步和目标定位等研究的基础. 基于层次结构的分簇拓扑控制方法结构简单、容错性和可扩展性强, 能有效提升网络生命周期, 是一种有效优化能量消耗的拓扑控制技术. 该方法将网络划分为若干个簇, 采用特定方法为每个簇选择一个簇首节点, 通过簇首节点将簇内其他普通节点的数据传输到上一层节点. 簇首节点相对普通节点承担更多的工作, 其选举算法、数量及分布是分簇拓扑控制方法的关键, 直接影响节点能量消耗和网络的生命周期.

Heinzelman 等^[1]提出了经典的 LEACH 层次分簇

算法, 节点以一定概率周期性地随机与某一阈值比较, 决定自己是否被选为簇首节点, 其他节点根据自己离簇首的距离选择加入相应的簇. 虽然 LEACH 算法采用簇首轮换策略在一定程度上降低了节点的能量消耗, 延长了网络生命周期, 但随机性的簇首选择方式无法保证选出的簇首较为均匀地分布在整个网络, 容易造成簇首之间负载不均衡, 导致部分节点过早死亡. 另一方面, 簇首选举概率缺乏理论分析, 簇首过少会导致单个簇内节点数过多, 普通节点到簇首的平均通信距离增大, 增加网络通信能量消耗. 簇首过多则会增加基站的管理难度, 也影响网络的性能和生命周期. 许多学者提出了基于 LEACH 的改进算法^[2], 如集中式低功耗自适应层次分簇算法 (LEACH-C)^[3],

收稿日期: 2019-04-30; 修回日期: 2019-08-20.

基金项目: 国家自然科学基金项目 (61763028, 61663023).

责任编辑: 赵春晖.

[†]通讯作者. E-mail: chenzh@lut.cn.

分布式混合节能分簇算法(HEED)^[4]、传感信息系统的节能采集算法(PEGASIS)^[5]、阈值敏感的高效传感网络协议(TEEN)^[6]等。这些算法大多基于随机概率的方法选取簇首,也没有考虑簇首数量对网络的性能影响,依然不能很好地解决分簇算法的负载均衡、最优分簇数量等问题。

如何确定WSN的最优簇首数量,并选出均匀分布的最优簇首集合是一个受多个因素影响的NP-hard问题^[7],进化算法(evolutionary algorithms, EAs)为解决该问题提供了新的思路。如文献[7-9]等采用粒子群优化算法(particle swarm optimization, PSO)解决了WSN的簇首选择及拓扑控制问题。PSO算法思想得益于鸟群觅食行为,算法主要利用种群个体之间的协作求解复杂问题。在PSO算法求解过程中,粒子被抽象成解空间中具有位置和速度属性的点,该点代表算法的一个可行解,粒子根据自身飞行经验和群体飞行经验动态更新自己的速度和位置,最终通过粒子之间的相互学习寻找最优解。文献[7]采用自适应粒子群优化算法选择簇首,并给出簇首间的连通算法;文献[8]采用模糊聚类算法对传感器节点进行聚类,然后基于粒子群优化算法确定簇首节点;文献[9]基于粒子群优化算法增加中继节点,降低簇首节点的能量消耗。

虽然以上算法考虑了影响簇首选择的多种因素,但都属于单目标进化算法,受限于解决单目标问题这一算法根本机制,只能采用仅考虑某一参数、主要考虑某一参数兼顾另一参数、利用简单加权法考虑多种参数等3种模式解决问题,从本质上而言,并不能真正意义上同时优化影响簇首选择的负载均衡、通信距离、网络能耗和生命周期等多个目标,因此采用多目标优化算法解决该问题是更好的选择。

目前,已有的多目标优化算法大致分为3类^[10]:基于Pareto的多目标优化算法^[11-13]、基于指标的优化算法^[14]和基于分解的优化算法^[15]。其中基于Pareto的多目标优化算法是较为主流的方法,通过Pareto支配机制,得到种群个体之间的支配关系,淘汰较差种群个体,选择占优个体。文献[16]采用多目标粒子群优化算法结合模糊决策理论选择最优簇首方案;文献[17]采用NSGA-II^[13]算法解决了WSN的簇首选择和覆盖优化问题。

NSGA-II是目前最流行的一种基于Pareto的多目标优化算法,具有运行速度快、收敛性好等优点。该算法具有如下优势:所提出的快速非支配排序方法降低了算法的复杂度,同时使下一代个体从包含父代种群和子代种群的个体中选取,从而保留较为优秀的个体;采用精英保留策略,保证得到的最优解不被丢

弃,提高了搜索性能;引入拥挤度计算,使得解集中的个体能够均匀地扩展到Pareto解空间中,保证最优解的多样性。NSGA-II因其良好的性能被广泛应用并作为各类多目标算法的对比算法。

虽然多目标优化算法在WSN的分簇问题上能够考虑多个优化目标,但首先需要进行编码设计。已有的研究,如PSO的粒子或NSGA-II的染色体,其编码长度都和网络节点个数一致,当网络规模较大时,编码方案会导致解空间规模巨大,存在搜索寻优效率低的问题。在设计更优的编码方案降低解空间规模相对困难的情况下,通过改进内在搜索机制提高算法的寻优效率是较为可行的解决方法。近年来,有学者将正交实验^[18-20]引入进化算法,利用正交性从全面实验中选取部分均匀分散、具有代表性的点进行实验。通过正交表指导种群个体交叉,降低搜索次数,提高寻优效率,取得了不错的效果。

另一方面,初始解的质量对进化算法的寻优效率有重要影响,良好的初始解能够加快进化算法的收敛速度。目前解决WSN簇首选择问题的进化算法普遍随机生成初始解,会大概率产生不符合实际情况、质量较差的初始解,例如一个粒子或染色体中簇首节点比例超过50%,则会影响算法的搜索效率。如果通过理论分析和计算获得最优簇首数量,通过最优簇首数量指导粒子或染色体的初始化,则初始解的质量相对更优,有助于提高算法的搜索效率。

针对传统WSN分簇算法未全面考虑影响簇首选择的多种因素、簇间负载不均衡和簇首分布不均匀,进化算法初始解质量较差、解空间规模巨大导致的搜索寻优效率低等问题,本文提出基于NSGA-II的多目标WSN簇首选择算法。同时考虑负载均衡、通信距离、能量消耗、簇首剩余能量等多个影响WSN簇首选择的优化目标,延长网络生命周期。通过理论分析确定最优簇首数量,生成质量较优的初始解,加快算法收敛速度。引入正交实验机制减少试验次数,提高搜索寻优效率。

1 网络模型及能量消耗模型

1.1 网络模型

本文假设所有传感器节点随机分布,一旦节点部署完成就固定位置,并具有以下特性:1)有一个固定位置的基站节点作为汇聚节点,能量不受限;2)所有节点可以在簇首节点和普通节点两种角色之间切换;3)所有节点的能量受限,拥有自身位置信息,能在一定范围内传输数据;4)每个传感器节点具有数据处理、发送和接收能力。

传感器节点之间的数据传输消耗一定能量,随着

数据的传输,节点能量逐渐耗尽,如果某个节点能量耗尽,则被视为死亡,当整个网络死亡节点比例上升到一定阈值时,认为整个网络死亡。

1.2 能量消耗模型

本文采用经典LEACH算法的能量消耗模型,根据发射与接收节点之间的距离,设置距离阈值 d_0 ,将能量衰减模型分为自由空间模型和多径衰减模型.当传输距离小于 d_0 时,能耗与距离平方成正比,否则能耗与距离4次方成正比.假设传输距离为 d ,发送和接收 n bit数据的能量消耗由下式计算:

$$E_{Tx}(n, d) = \begin{cases} nE_{elec} + n\varepsilon_{fs}d^2, & d < d_0; \\ nE_{elec} + n\varepsilon_{mp}d^4, & d \geq d_0. \end{cases} \quad (1)$$

$$E_{Rx}(n) = nE_{elec}. \quad (2)$$

其中: E_{elec} 是发送或接受单位比特所消耗的能量, ε_{fs} 是自由空间模式下传感器节点发送每比特消耗的能量, ε_{mp} 是多径衰减模式下发送每比特消耗的能量,距离阈值 $d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}$.

2 基于正交机制的NSGA-II簇首选择算法

本节基于NSGA-II算法,提出一种基于正交设计机制的多目标算法,在簇首选择时,通过5个适应度函数全面考虑影响簇首选择的通信距离、负载均衡、能量消耗和网络生存时间等因素.根据最优簇首数量生成质量较优的初始解,并设计基于正交实验机制的多目标个体交叉策略,提高NSGA-II算法的搜索寻优效率。

2.1 种群初始化

假设网络有 m 个节点,对网络中每个传感器节点进行顺序编号,依次为 $1, 2, \dots, m$,一条染色体代表一个簇首选择问题的可行解.每条染色体由 m 位基因组成,编码“0”表示该节点为普通节点,编码“1”表示该节点为簇首节点,具有5个节点的网络的染色体初始化编码如表1所示.染色体1代表节点1、节点4为簇首节点,节点2、节点3和节点5为普通节点。

表1 种群个体染色体编码表

节点编号	1	2	3	4	5
节点坐标	(x_1, y_1)	(x_2, y_2)	(x_3, y_3)	(x_4, y_4)	(x_5, y_5)
染色体1	1	0	0	1	0
染色体2	0	1	1	0	0
染色体3	1	1	0	0	0
染色体4	0	1	0	0	1
染色体5	0	0	1	1	0

2.2 适应度函数

WSN的簇首选择与节点间的通信距离、簇间负载情况、簇首剩余能量、节点总能耗等有关,因此本

文考虑选取剩余能量较大的节点作为簇首,平衡簇间负载,减少节点间的总通信距离,降低能量消耗,延长节点生命周期,设置以下5个适应度函数。

适应度函数1: 全网通信距离之和最小。

由式(1)可知,节点传输数据的能量消耗与距离有关,因此计算成簇后全网普通节点到对应簇首节点的通信距离之和,距离之和越小,能量消耗越少。

$$f_1 = \min \left(\sum d_{CH-mem} \right), \quad (3)$$

其中 d_{CH-mem} 为簇首节点到非簇首节点的通信距离。

适应度函数2: 全网能耗最小。

假设网络分为 K 个簇,在一轮数据传输过程中,全网能耗包括簇首节点和非簇首节点的能耗。

$$E_{total} = \sum_{i=1}^K \left(E_{CH}^i + \sum_{j=1}^{N_i} E_{mem}^{ij} \right), \quad (4)$$

$$f_2 = \min(E_{total}). \quad (5)$$

其中: E_{CH}^i 是簇首节点的能量消耗, $\sum_{j=1}^{N_i} E_{mem}^{ij}$ 是非簇首节点的能量消耗。

适应度函数3: 簇首节点间能耗标准差最小。

分簇方案需要考虑每个簇的负载情况,避免出现不同簇首间能耗相差过大的现象.为此,计算簇首之间负载的标准差,以标准差最小为目标。

$$\delta_E = \sqrt{\frac{1}{K} \sum_{i=1}^K (E_{CH}^i - \overline{E_{CH}})^2}, \quad (6)$$

$$f_3 = \min(\delta_E), \quad (7)$$

其中 $\overline{E_{CH}}$ 是簇首节点的平均能量消耗。

适应度函数4: 簇首能量最大化。

在分簇结构WSN中,簇首节点承担更多的功能,比普通节点消耗更多的能量.因此,应该选择剩余能量更高的节点担任簇首,计算簇首能量与全网能量的比值,以最大化该值为优化目标。

$$f_4 = \max \left(\frac{\sum E_{CH}}{\sum E_{CH} + \sum E_{mem}} \right). \quad (8)$$

适应度函数5: 网络生存时间。

良好的分簇方案能有效延长网络的生命周期,不同应用背景的WSN对节点的死亡比例有不同的要求,本文综合考虑网络第一个节点、20%节点、50%节点和全部节点死亡时间,以最大网络生存时间为优化目标。

$$D_{all} = dead_1 + dead_{20\%} + dead_{50\%} + dead_{100\%}, \quad (9)$$

$$f_5 = \max \left(\frac{D_{all}}{4} \right). \quad (10)$$

2.3 正交实验设计

正交实验设计是解决多因素、多水平实验问题的一种有效方法,它设计了具有正交特性的正交表 $L_M(Q^F)$,从全面实验中挑选部分有代表性的点进行实验. 假设实际应用环境是一个具有1000个节点的无线传感网络,采用表1的方式编码后,每条染色体为1000维,即是一个1000因素2水平的问题,如果进行全面组合搜索,则需要 2^{1000} 次实验,而采用正交表设计则只需 $L_M(2^{1000})$,总共1001次实验,大大减少了实验次数. 下面以 5×5 区域部署5个节点为例说明正交实验设计过程.

假设有2个父代个体 $p_1 = (10\ 100)$, $p_2 = (00\ 011)$,为简化说明,此处只列出2个适应度函数值 f_1 和 f_2 ,其值为(7.32, 0.707)和(9.32, 0.212),如表2所示.

表2 WSN簇首选择问题的因素及水平

染色体	因素					多目标适应度值	
	1	2	3	4	5	f_1	f_2
染色体1	1	0	1	0	0	7.32	0.707
染色体2	0	0	0	1	1	9.32	0.212

表3 WSN簇首选择问题的正交交叉实验方案及结果

实验号	因素					多目标适应度值	
	1	2	3	4	5	f_1	f_2
1	1	0	1	0	0	7.32	0.707
2	1	0	1	1	1	1.00	0
3	1	0	0	0	0	10.76	0
4	1	0	0	1	1	4.00	0.173
5	0	0	0	0	1	10.76	0
6	0	0	0	1	0	10.82	0
7	0	0	1	0	1	6.00	0.283
8	0	0	1	1	0	6.00	0.283

因素分析										
	1		2		3		4		5	
L_1	5.77	0.22	7.48	0.177	5.08	0.318	8.71	0.248	8.73	0.248
L_2	8.4	0.141	6.69	0.185	9.09	0.043	5.46	0.114	5.44	0.114
结果	L_1 or L_2		L_1 or L_2		L_1 or L_2		L_2		L_2	

标准遗传算法的交叉机制可能出现“两步前进,一步后退(two steps forward, one step back)”的现象^[18]. 例如表2中,父代个体染色体1和染色体2采用非正交方式交叉后得到子代个体 $p_3 = (00\ 101)$,其适应度值 $(f_1, f_2) = (9.36, 0.83)$ 都被父代个体适应度值(7.32, 0.707)和(9.32, 0.212)支配,即父代个体优于子代个体,交叉结果并未向更优解靠近. 如果采用正交实验机制,则实验结果去除重复后染色体组合共有4种,如表4所示. 与父代个体相比:子代个体1和子代个体2都支配(即优于)父代个体;子代个体3支配父代个体1,与父代个体2互不支配;子代个体4与父

构造正交表 $L_8(2^5)$ 如下:

$$L_8(2^5) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 \\ 2 & 1 & 2 & 1 & 2 \\ 2 & 1 & 2 & 2 & 1 \\ 2 & 2 & 1 & 1 & 2 \\ 2 & 2 & 1 & 2 & 1 \end{bmatrix} \quad (11)$$

根据正交表进行的实验过程如表3所示. 不同于单目标问题正交表,多目标正交表中的实验结果由多个适应度函数值反映,无法和单目标问题一样直接根据单个数值的大小决定优劣. 本文采用支配关系的方法评价多目标实验结果并进行因素分析. 例如本问题的因素分析希望 f_1 和 f_2 都越小越好, $L_2 = (5.46, 0.114)$ 显然优于 $L_1 = (8.71, 0.248)$,所以选择 L_2 作为最终结果. 如果两个结果之间没有支配关系,则将两者都作为实验结果进行组合,例如表3中,在因素分析的前3列, L_1 和 L_2 都不存在支配关系,则将 L_1 和 L_2 的取值作为结果,而第4列和第5列将 L_2 作为结果.

代个体2相同,与父代个体1互不支配. 总体而言,多目标正交交叉后产生的子代个体优于(至少不劣于)父代个体.

表4 WSN簇首选择问题的因素及水平

实验结果	因素					多目标适应度值	
	1	2	3	4	5	f_1	f_2
染色体1	1	0	1	1	1	1	0
染色体2	1	0	0	1	1	4	0.173
染色体3	0	0	1	1	1	5	0.23
染色体4	0	0	0	1	1	9.32	0.212

2.4 最优簇首数量分析

簇首数量决定簇的大小和节点间的通信距离,簇首数过多或过少都会引起网络性能的下降,影响网络的能量消耗和生命周期.

从遗传算法的角度而言,优良的初始解有助于加快算法收敛,提高寻优效率. NSGA-II算法种群个体初始化时需要确定染色体的每一位基因,如表1所示.目前的算法随机生成基因的“0”、“1”数值组合(即簇首数量)完全随机,这有可能产生不符合实际网络应用的较差染色体,如(1, 1, 1, 1, 1)、(0, 0, 0, 0, 0)等簇首节点和普通节点比例失常的染色体,严重影响算法寻优效率.因此,本文对最优簇首数进行分析,采用类似文献[7]中的方法,通过理论计算求解最优簇首数.

假设 N 个节点均匀地分布在边长为 X 的正方形区域内,在第 i 轮的簇首选举中,选出 k 个簇首,则平均每个簇的节点数为 N/k ,其中簇首节点1个,非簇首节点 $N/k - 1$ 个.网络的能量消耗 E_{total} 包括簇建立阶段和簇稳定运行两个阶段,分别用 E_{cre} 和 E_{run} 表示.每个阶段的能量消耗又分为簇首节点和非簇首节点两部分,分别用 E_{CH} 和 E_{mem} 表示.

簇建立阶段:簇首节点发送 $2n$ bit数据,非簇首节点接收.

$$E_{T_{\text{CH}}}(2n, d) = \begin{cases} 2nE_{\text{elec}} + 2n\varepsilon_{fs}d^2, & d < d_0; \\ 2nE_{\text{elec}} + 2n\varepsilon_{mp}d^4, & d \geq d_0. \end{cases} \quad (12)$$

$$E_{R_{\text{mem}}}(2n, d) = 2nE_{\text{elec}}. \quad (13)$$

非簇首节点发送 n bit数据,簇首节点接收.

$$E_{R_{\text{CH}}}(n, d) = nE_{\text{elec}}. \quad (14)$$

$$E_{T_{\text{mem}}}(n, d) = \begin{cases} nE_{\text{elec}} + n\varepsilon_{fs}d^2, & d < d_0; \\ nE_{\text{elec}} + n\varepsilon_{mp}d^4, & d \geq d_0. \end{cases} \quad (15)$$

$$E_{\text{cre}} = kE_{T_{\text{CH}}}(2n, d) + (N - k)E_{R_{\text{mem}}}(2n, d) + kE_{R_{\text{CH}}}(n, d) + (N - k)E_{T_{\text{mem}}}(n, d). \quad (16)$$

稳定传输阶段:一般按轮次进行,每个簇首节点接收 $N/k - 1$ 个非簇首节点发送的 p 帧数据,每帧包含 n bit数据,并发送至sink节点, d_{sink} 为簇首节点到sink节点的距离,假定 $d_{\text{sink}} \geq d_0$.

$$E_{R_{\text{CH}}}(n, d) = \left(\frac{N}{k} - 1\right)pnE_{\text{elec}}. \quad (17)$$

$$E_{T_{\text{mem}}}(n, d) = \begin{cases} pnE_{\text{elec}} + pn\varepsilon_{fs}d^2, & d < d_0; \\ pnE_{\text{elec}} + pn\varepsilon_{mp}d^4, & d \geq d_0. \end{cases} \quad (18)$$

$$E_{R_{\text{CH}}}(n, d_{\text{sink}}) = \left(\frac{N}{k} - 1\right)pnE_{\text{elec}} + \left(\frac{N}{k} - 1\right)pn\varepsilon_{mp}d_{\text{sink}}^4. \quad (19)$$

$$E_{\text{run}} = kE_{R_{\text{CH}}}(n, d) + (N - k)E_{T_{\text{mem}}}(n, d) + kE_{T_{\text{CH}}}(n, d_{\text{sink}}). \quad (20)$$

最终簇从建立到运行一轮的总能量消耗为

$$E_{\text{total}} = E_{\text{cre}} + E_{\text{run}}. \quad (21)$$

在 $X \times X$ 面积内,假设每个簇是一个圆,其半径 $r = \frac{X^2}{\sqrt{\pi k}}$,节点的分布密度函数 $\rho = 1/\frac{x^2}{k}$,则节点到簇首的距离期望为

$$E(d^2) = \iint d^2 \rho(x, y) d\sigma = \iint (x^2 + y^2) \rho(x, y) dx dy = \frac{X^2}{2\pi k}. \quad (22)$$

将式(22)代入(21),得到 E_{total} 并对 k 求导,得到最优簇首数

$$k_{\text{opt}} = X \sqrt{\frac{\varepsilon_{fs} p N}{2\pi(p\varepsilon_{mp}d_{\text{sink}}^4 - pE_{\text{elec}} - 2E_{\text{elec}})}}. \quad (23)$$

由式(23)可见,分簇结构WSN最优簇首数与网络覆盖范围 $X \times X$ 、节点总数 N 、簇首到sink节点的距离、每轮传输数据的帧数 p 和节点硬件能耗参数有关.虽然式(23)给出了网络的最优簇首数求解公式,但这是在假设节点均匀分布下得出的结果,如果网络节点随机分布,则最优簇首数量会有一定范围的变化.

2.5 算法步骤

本文算法除种群初始化和交叉环节以外,其余过程都与标准NSGA-II算法一致,主要步骤如下.

Step 1:采用式(23)计算最优簇首个数.

Step 2:种群初始化.在取值范围内,根据 k_{opt} 初始化种群 $P_I = X^1, X^2, \dots, X^N$.其中: $X^i = (x_1, x_2, \dots, x_n)$, $i = 1, 2, \dots, N$.每条染色体 X^i 的值为1的位数取值范围为 $[k_{\text{opt}} - \gamma, k_{\text{opt}} + \gamma]$, γ 为浮动范围.

Step 3:计算 P_I 中所有个体的适应度值,进行快速非支配排序.

Step 4:选择个体 $X^i = (x_1, x_2, \dots, x_n)$ 和 $X^j = (x_1, x_2, \dots, x_n)$,采用正交实验机制进行交叉,如果出现如表4所示的多个交叉实验结果,则从中随机选择一个结果进行变异,生成子代种群 P_M .

Step 5:合并父、子代种群 $P_T = \{P_I \cup P_M\} = \{X^1, X^2, \dots, X^{2N}\}$.

Step 6:对合并后的新种群 P_T 进行快速非支配排序,生成具有支配关系的支配解集 F_1, F_2, \dots, F_L ,其中 F_i 支配 F_{i+1} .为了维持种群规模保持 N 不变,根

据排序等级从非支配最优解 F_i 开始, 依次从 $F_1, F_2, \dots, F_i (i = 1, 2, \dots, L)$ 中选择 N 个个体进入下一代种群. 如果解集 F_1, F_2, \dots, F_i 的种群大小小于 N , 而解集 F_1, F_2, \dots, F_{i+1} 的种群大小大于 N , 则对 F_{i+1} 采用拥挤距离计算, 删除部分拥挤度高的解, 保证种群大小为 N .

Step 7: 进行和 Step 3 相同的正交交叉, 经过变异产生新种群.

Step 8: 若满足终止条件, 则输出最优解, 否则返回 Step 4.

2.6 算法复杂性分析

设目标函数的维数为 M , 种群个数为 N , 本文算法流程可分为 4 个主要部分, 分别是快速非支配排序、正交交叉、拥挤度计算和根据拥挤度排序. 由文献 [1,18] 可知, 其复杂度分别为: 快速非支配排序 $O(M(2n)^2)$, 正交交叉 $O(N \times 2^{\lfloor \log_2(M+1) \rfloor})$, 拥挤度计算 $O(M(2N) \log(2N))$, 根据拥挤度排序 $O(2N \log(2N))$. 因此, 本文算法的复杂度为 $O(MN^2)$, 与标准 NSGA-II 算法复杂度相同, 其中非支配排序在总复杂度中占据主导地位.

3 仿真实验与分析

为了验证和分析本文所提出算法的性能, 采用 Matlab 平台进行网络分簇效果、生存周期、能量消耗、算法效率等 4 个方面的仿真实验, 实验参数设置如表 5 所示.

表 5 仿真实验参数设置

参数	设置
节点覆盖区域 / m^2	100 × 100
节点数量	100
sink 节点位置	(50, 50)
$E_{fs} / (pJ / bit / m^2)$	10
$E_{mp} / (pJ / bit / m^4)$	0.001 3
$E_{elec} / (nJ / bit)$	50
节点初始化能量 / J	0.5

3.1 分簇效果实验

本实验对比 LEACH 算法与本文算法所选的簇首的位置分布. 图 1 所示的 LEACH 算法随机选择簇首, 没有考虑通信距离及负载均衡等问题, 所选簇首在网络中分布不均匀, 部分簇首相距太近, 各簇之间节点数量相差较大. 图 2 所示的本文算法所选簇首在整个网络中分布较为均匀, 基本位于每个簇的中心位置, 并且各簇的规模大小相互接近.

表 6 是本文算法和 LEACH 分簇结构的统计数据. 可以看出, LEACH 算法最小的簇只有 3 个节点, 而最大的簇有 21 个节点, 在不考虑通信距离的情况下,

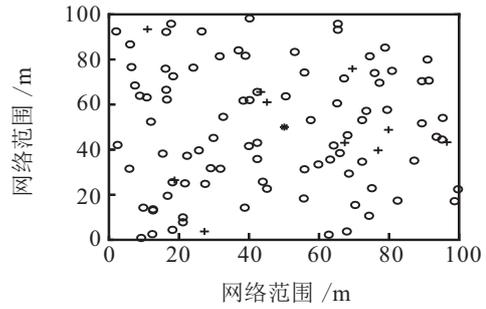


图 1 LEACH 算法簇首分布

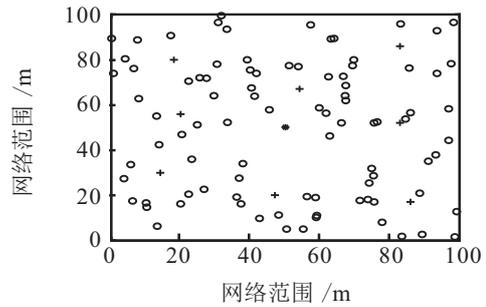


图 2 本文算法簇首分布

假定各簇普通节点发送相同大小的数据给簇首, 由式 (2) 可知, 不同簇首仅接收数据这一项的能耗相差就高达 7 倍左右. 本文算法得出的簇大小比较接近, 最小的簇有 10 个节点, 最大的簇有 14 个节点. 本文算法簇内平均距离 8.54 也小于 LEACH 算法的 10.15. 普通节点到簇首的平均距离标准差也说明本文算法选出的簇首分布更为均匀.

表 6 簇平均距离及均衡程度统计 (与 LEACH 算法对比)

算法	簇内节点数量范围	簇内节点数标准差	簇内平均距离	各簇节点到簇首平均距离标准差
LEACH	[3 ~ 21]	6.16	10.15	14.06
本文算法	[10 ~ 14]	1.53	8.54	1.03

3.2 网络生存周期实验

网络生存周期最大化是能量受限条件下 WSN 追求的目标. 图 3 表示节点存活数量与运行周期之间的关系, 可以看出, LEACH 算法在第 1 500 轮之后全网仅有个别节点存活, 本文算法在第 2 000 轮依然有接近一半节点存活, 并且直到第 2 500 轮还有少量节点存活, 明显延长了网络的生命周期.

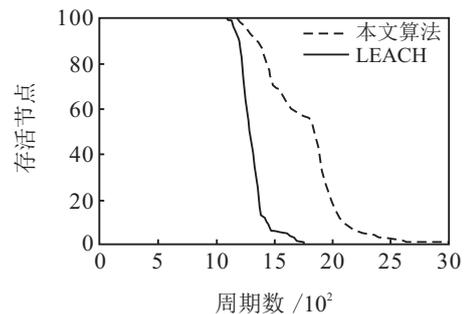


图 3 节点存活数量对比

网络中节点的死亡比例是WSN在实际应用中需考虑的一个指标.图4为网络中1%(第1个)、10%、20%和50%节点死亡的周期统计.可知,LEACH算法在第928轮第1个节点死亡,本文算法在第1156轮第1个节点死亡,推迟了228轮.LEACH算法在第1301轮网络中50%的节点死亡,而本文算法则推迟到第1845轮,相比LEACH算法明显延后死亡时间.

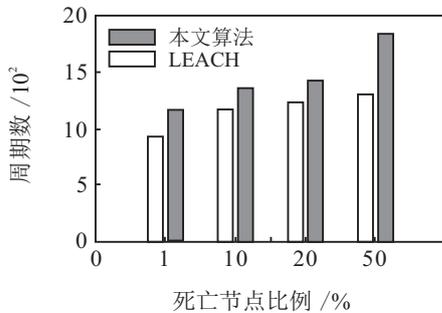


图4 死亡节点比例对比

3.3 网络剩余能量对比

令全网能量消耗最小是本文所设计的适应度函数之一,实验最初全网除sink节点外能量共计50J.从图5可以看出,本文算法与LEACH的剩余能量在运行初期相差较小,随着运行周期的增大,剩余能量相差增大.当运行500轮时,LEACH算法剩余28.9J,本文算法剩余40.2J;当到达第1500轮时,LEACH算法剩余能量0.036J,意味着节点基本全部死亡,而本文算法还剩余20.63J的能量.因此,本文方法中,网络具有更长的生命周期.

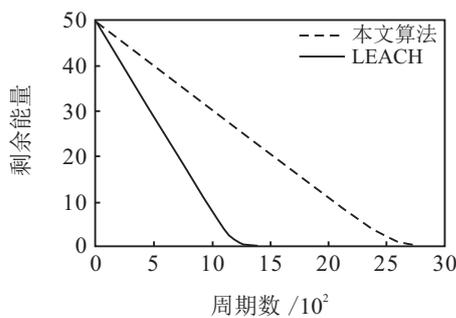


图5 网络剩余能量对比

图6反映网络80%、50%、20%和10%剩余能量随周期变化的情况.

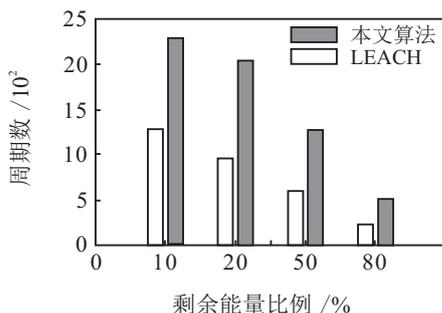


图6 剩余能量比例与周期关系对比

由图6可以看出,LEACH算法在第236轮全网剩余80%能量,本文算法则推迟到第510轮.LEACH算法在第947轮处全网剩余20%能量,本文算法则推迟到第2043轮,明显降低了相同周期的能量消耗比例.

3.4 搜索寻优效率实验

本组实验验证了本文引入的正交实验交叉机制和最优簇首数量指导种群初始化方法对搜索寻优效率的影响.由于WSN簇首选择问题缺乏标准测试集,没有理论最优解可供对比,并且本文算法和NSGA-II算法运行结束后都能找到比较接近的最优解.因此,实验对比相同迭代次数条件下,本文算法和标准NSGA-II算法找到簇首节点集合对应的适应度函数值,通过适应度函数值评价两种算法的寻优效率.由于算法没有完全运行结束,部分适应度函数(如节点死亡时间等数据)可能无法准确反映算法性能差距或不具有可比性,因此本文选取网络通信距离、簇内成员数量范围两个指标,设定算法迭代200次进行比较,结果如表7所示.

表7 簇平均距离及均衡程度(与标准NSGA-II算法对比)

算法	簇内节点数量范围	簇内节点数标准差	簇内平均距离	普通节点到簇首平均距离标准差
标准NSGA-II	[7~20]	4.75	9.22	3.15
本文算法	[9~16]	2.87	8.66	1.42

从表7可以看出,在相同迭代次数条件下,本文算法不论是簇内节点数量范围还是簇内平均距离指标,都比标准NSGA-II算法找到的解更优,算法寻优效率更高.主要原因有:1)标准NSGA-II在解空间复杂、目标数量较多的高维问题上收敛速度较慢,向较优解靠拢缓慢;2)正交实验方法是一种搜索能力较强的算法,不仅能够降低试验次数,而且能够避免出现遗传算法的“两步前进,一步后退”现象,提高算法寻优性能;3)采用最优簇首数量指导种群初始化,选取更优的初始解作为算法输入,也能进一步加快算法的寻优效率.

4 结论

为了有效延长WSN的生存时间,提高多目标优化算法解决WSN簇首选择问题的搜索寻优效率,本文提出了一种基于正交实验设计机制的NSGA-II无线传感网络簇首选择算法.该算法具有以下优点:1)同时考虑多个影响WSN簇首选择的因素,避免簇首选择的随机性,减少全网络总通信距离,有效均衡簇首之间的负载,延长网络生命周期;2)根据最优簇首数初始化种群,提高初始解的质量,加快算法收敛;3)引入正交实验机制减少实验次数,提高了算法

的搜索寻优效率.

参考文献(References)

- [1] Heinzelman W R, Chandrakasan A P, Balakrishnan H. Energy-efficient communication protocol for wireless sensor networks[C]. Proceedings of the 33rd Hawaii International Conference on System Sciences. Hawaii: IEEE, 2000: 1-10.
- [2] Singh S, Kumar P, Singh J. A survey on successors of LEACH protocol[J]. IEEE Access, 2017, 99(5): 4298-4328.
- [3] Heinzelman W B, Chandrakasan A P, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks[J]. IEEE Transactions on Wireless Communications, 2002, 1(4): 660-670.
- [4] Younis O, Fahmy S. HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks[J]. IEEE Transactions on Mobile Computing, 2004, 3(4): 366-379.
- [5] Lindsey S. PEGASIS: Power efficient gathering in sensor information systems[C]. Proceedings of IEEE Aerospace and Electronic Systems Society. Montana: IEEE, 2002: 1125-1130.
- [6] Manjeshwar A, Agrawal D P. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks[C]. Proceeding of the 15th Parallel and Distributed Processing Symposium. San Francisco: IEEE Computer Society, 2001: 2009-2015.
- [7] 苏金树, 郭文忠, 余朝龙, 等. 负载均衡感知的无线传感器网络容错分簇算法[J]. 计算机学报, 2014, 37(2): 445-456.
(Su J S, Guo W Z, Yu C L, et al. Fault-tolerance clustering algorithm with load-balance aware in wireless sensor network[J]. Chinese Journal of Computers, 2014, 37(2): 445-456.)
- [8] Ni Q, Pan Q, Du H, et al. A Novel cluster head selection algorithm based on fuzzy clustering and particle swarm optimization[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2017, 14(1): 76-84.
- [9] Zhou Y, Wang N, Xiang W. Clustering hierarchy protocol in wireless sensor networks using an improved PSO algorithm[J]. IEEE Access, 2017, 99(5): 2241-2253.
- [10] 焦李成. 认知计算与多目标优化[M]. 北京: 科学出版社, 2017: 33-34.
(Jiao L C. Cognitive computing and multi-objective optimization[M]. Beijing: Science Press, 2017: 33-34.)
- [11] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Transactions on Evolutionary Computations, 2004, 8(3): 256-279.
- [12] Kim M, Hiroyasu T, Miki M, et al. SPEA2+: Improving the performance of the strength pareto evolutionary algorithm 2[C]. Proceedings of the 8th International Conference on Parallel Problem Solving from Nature. Birmingham: Springer, 2004: 742-751.
- [13] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [14] Phan D H, Suzuki J. R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization[C]. IEEE Congress on Evolutionary Computation. Cancun: IEEE, 2013: 1836-1845.
- [15] Zhang Q, Hui L. MOEA/D: A multi-objective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [16] Randhawa S, Jain S. MLBC: Multi-objective load balancing clustering technique in wireless sensor networks[J]. Applied Soft Computing, 2019, 74(1): 66-89.
- [17] Hacıoglu G, Kand V F A, Sesli E. Multi objective clustering for wireless sensor networks[J]. Expert Systems with Applications, 2016, 59(10): 86-100.
- [18] Zhan Z H, Zhang J, Li Y, et al. Orthogonal learning particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(6): 832-847.
- [19] 张屹, 万兴余, 郑小东, 等. 基于正交设计的元胞多目标遗传算法[J]. 电子学报, 2016, 44(1): 87-94.
(Zhang Y, Wan X Y, Zheng X D, et al. Cellular genetic algorithm for multiobjective optimization based on orthogonal design[J]. Acta Electronica Sinica, 2016, 44(1): 87-94.)
- [20] 周新宇, 吴志健, 王明文. 基于正交实验设计的人工蜂群算法[J]. 软件学报, 2015, 26(9): 2167-2190.
(Zhou X Y, Wu Z J, Wang M W. Artificial bee colony algorithm based on orthogonal experimental design[J]. Journal of Software, 2015, 26(9): 2167-2190.)

作者简介

陈作汉(1979—), 男, 讲师, 博士生, 从事无线传感网络、智能优化算法等研究, E-mail: chenzh@lut.cn;

曹洁(1966—), 女, 教授, 博士生导师, 从事智能信息处理、非线性理论及应用等研究, E-mail: caoj@lut.cn;

赵付青(1977—), 男, 教授, 博士生导师, 从事人工智能、制造系统建模及优化等研究, E-mail: Fzhao2000@hotmail.com.

(责任编辑: 闫妍)