

文章编号:1001-9081(2009)02-0496-04

基于 ZigBee 网络的 ZiCL 改进算法

王 芳, 柴乔林, 班艳丽

(山东大学 计算机科学与技术学院, 济南 250101)

(58396534@163.com)

摘 要:针对 ZigBee 网络中使用的 ZiCL 算法可能造成节点过度耗能和网络能耗不均的问题,提出一种改进算法。该算法根据网络运行过程中簇成员能量状态来动态调整路由选择策略。仿真结果表明,改进算法提高了网络传输的可靠性,有效节约了能量。

关键词:ZigBee 网络; ZiCL 算法; 剩余能量; OMNET ++ 仿真

中图分类号: TP393 **文献标志码:** A

Improved ZiCL algorithm based on ZigBee network

WANG Fang, CHAI Qiao-lin, BAN Yan-li

(College of Computer Science and Technology, Shandong University, Jinan Shandong 250061, China)

Abstract: ZiCL algorithm may cause excess energy consumption of nodes and unbalanced energy consumption of the whole network in ZigBee network, therefore an improved algorithm was put forward. The improved algorithm adjusted routing-select strategy dynamically according to the energy state of cluster members. The simulation indicates that the improved algorithm can improve the transport reliability and save energy effectively.

Key words: ZigBee network; ZiCL algorithm; residual energy; OMNET ++ simulation

0 引言

ZiCL^[1]是在 ZigBee 网络基础上提出的一种路由算法。ZigBee 技术^[2]基于 IEEE 802.15.4 无线个人局域网标准,人们希望用它来连接简单、低速率、电池供电的无线设备。ZigBee 网络中节点异构^[3],这和传统的 MANET 网络中节点同构的情况不同;同时 ZigBee 节点之间采用短地址进行数据通信。这些特点使得 ZigBee 网络和传统的无线网络有较大的区别。

1 ZigBee 网络

1.1 ZigBee 中节点的类型和角色

IEEE 802.15.4 根据节点的计算能力、存储容量定义了全功能设备(Full Function Device, FFD)和精简功能设备(Reduced Function Device, RFD)两种类型的设备^[4]。FFD 设备存储容量大,计算能力强,它可以参与路由发现,保存并维护路由表,转发数据。而 RFD 只参与最少量的网络信息,并不参与路由过程。FFD 可以和 FFD、RFD 类型的设备通信,而 RFD 只能和 FFD 类型设备进行通信, RFD 之间不能够通信。

在 ZigBee 网络中节点有中心协调器、路由器以及终端设备三种角色之分^[5]。中心协调器负责创建网络,管理网络节点加入以及维持网络拓扑;路由器发现并维护路由,同时在路由路径间转发数据;而 ZigBee 终端设备不能参与路由。由于中心协调器和路由器功能比较强大,它们需要 FFD 来担当以便完成比较复杂的工作,而终端设备由于功能较单一使用 RFD 即可。

1.2 ZigBee 节点的短地址和节点的深度

每个 ZigBee 节点都有两个地址,一个 64 b 的 MAC 地址,还有一个 16 b 的 IEEE 短地址。短地址是 ZigBee 网络中节点进行路由发现和数据传输使用的唯一地址。中心协调器首先创建网络,当一个节点申请加入该网络时,它选择一个有路由功能的父节点加入,与其建立起父子关系,父节点按树状地址分配机制^[6]为该子节点分配一个该网络中唯一的一个 16 b 短地址,若子节点也是 FFD 类型的节点,父节点还会拿出自己可以分配的地址块中的一部分分配给这个子节点,供子路由分配给相应的后代。这个父节点必须是 FFD 类型的设备,并且它和子节点之间需要有较强的通信能力。当新节点成功加入后,网络中节点之间的父子关系可以用树形结构来表示,图 1 给出了一棵父子关系树。

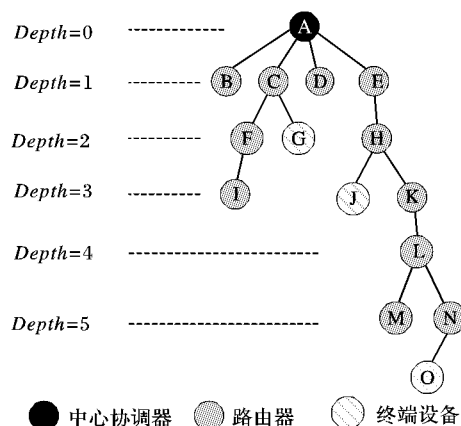


图 1 ZigBee 父子关系树

ZigBee 中用深度 Depth 来描绘一个节点与中心协调器的相对距离。子节点的 Depth 等于其直接父节点的 Depth 值加 1。

收稿日期:2008-08-11;修回日期:2008-09-23。

作者简介:王芳(1984-),女,山东枣庄人,硕士研究生,主要研究方向:网络与分布式技术;柴乔林(1956-),男,山东青岛人,教授,主要研究方向:计算机网络、无线传感器网络、数字化校园;班艳丽(1985-),女,山东省菏泽人,硕士研究生,主要研究方向:网络与分布式技术。

2 ZiCL 算法描述

ZiCL 协议将分簇的思想应用到 ZigBee MESH^[5] 网络中, 基于路由节点的深度将网络划分为一个或者多个逻辑的簇, 允许簇内成员共享路由信息。相比起传统的 AODV, 该算法减少了潜在的路由发现和洪泛带来的信息冗余, 提高了网络的可扩展性。

在簇划分阶段, 中心协调器首先当选为簇头, 同时深度值 *Depth* 为偶数的路由功能节点也被指定作为簇头。*Depth* 为奇数的路由器被划分到其父节点所在的簇中, 同时网络中的终端设备也加入其父节点所在的簇中。通过簇划分, 路由功能节点中的一部分被指定为簇头, 一部分被指定为网关节点, 剩下的一部分作为普通的路由节点。图 2 是图 1 经过簇划分完以后的结构图, 节点 C 由于连接簇标签为 0X0001 和 0X001C 的两个簇而成为网关节点。

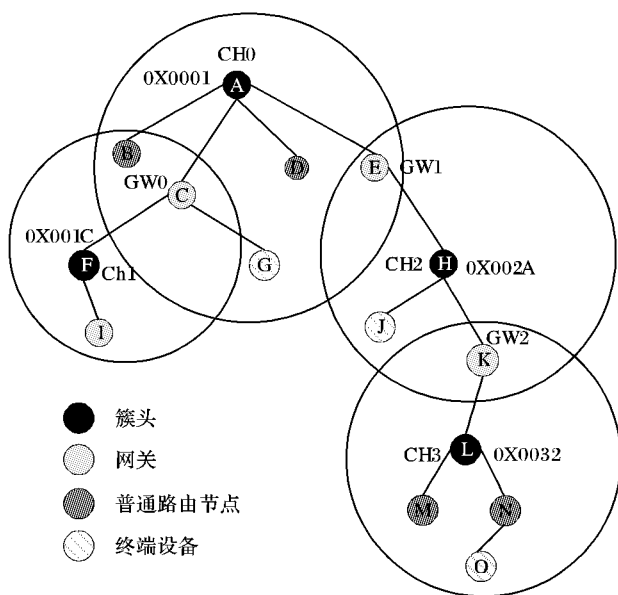


图2 ZigBee 网络的分簇结构

每个簇用簇标签来唯一标识, 簇标签在簇创建时由簇头来分配。文献[1]给出了簇标签的计算方法, 同时给定一个节点地址, 也可以通过计算得到它所在簇的簇标签。在基于深度划分的簇中, 同一簇的簇成员之间最多只有 4 跳。

ZiCL 路由包括内部路由和外部路由。源节点可以根据目的节点的地址计算得到其所在簇的簇标签来决定是选择内部路由还是外部路由。当源和目的节点的簇标签相同时, 说明源和目的节点在同一个簇, 于是选择内部路由。内部路由基于 ZigBee 的分层路由机制。在分层路由中, 转发节点通过计算目的节点的 16 b 短地址来决定数据包的转发方向, 基于树形结构向上转发给父节点或向下转发给某个子节点^[6]。

对于外部路由来说, 源节点依靠路由表中的路由信息来进行数据的传输。当源节点没有目的节点相关的路由信息时, 它使用 AODV 算法洪泛一个路由请求, 目的地是目的节点所在的簇。

ZiCL 鼓励同一个簇中的节点共享路由信息, 为此 ZiCL 提供了一种叫做簇标签广播的机制。当簇标签中的一个簇成员获得一个新的路由信息时, 它会发送一个 RNOT 消息通知给它的簇头。当簇头获得这个消息后, 会广播一个 RUPT 包给它的所有簇成员来实现路由信息的共享。

3 改进算法

ZiCL 分簇算法减少了潜在的路由发现, 但在网络运行过程中没有考虑到节点能耗问题, 而能耗问题对 ZigBee 网络的运行是十分重要的。ZigBee 网络节点基本上都是靠电量有限的电池供电, 在数据传输量大、信息发送频繁的情况下, 簇头、网关节点可能会由于能量消耗过多而成为失效节点, 造成路径失效甚至整个网络的瘫痪, 因此在改进算法中, 要充分考虑到节点能量不足的情况, 通过对节点能量的保护以保证网络正常有效的运行。图 3 是利用新算法在 ZiCL 的基础上做的一些改进。

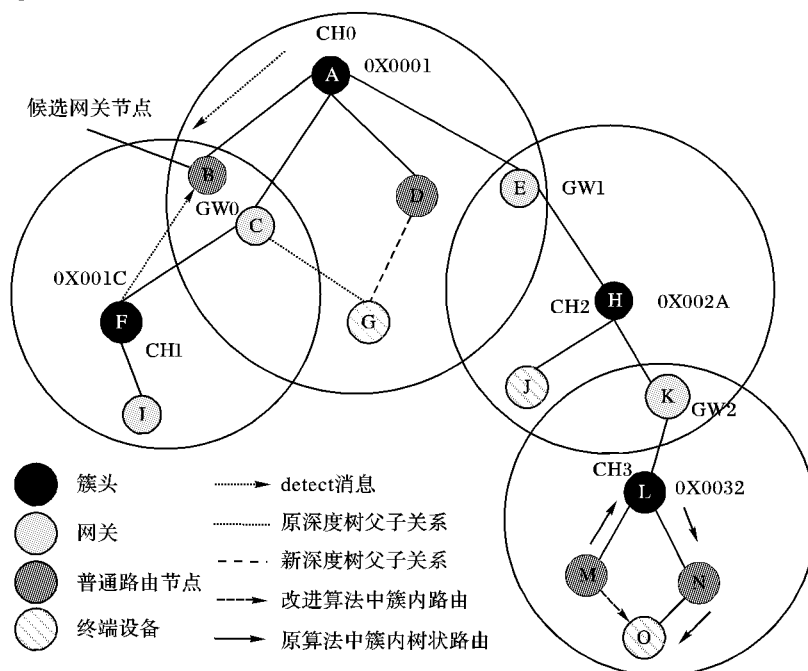


图3 改进算法示意图

3.1 定义候选网关节点

改进算法在簇划分阶段为相邻两个簇指定候选网关节

点。在 ZiCL 原算法中, 一部分路由节点通过分簇直接成为网关节点 (例如图 3 中节点 C), 而改进算法中, 在簇划分完以

后,让簇头节点发送 detect 查询信号给邻居节点,通过这种方式来寻找潜在的网关节点。对于图 3 中的节点 B,虽然它不是网关节点,但是它可以接收到簇头 A、F 发送来的 detect 消息,把 B 标识为候选网关节点。

3.2 簇内路由和簇间路由

在簇内路由阶段,我们不是直接使用分层路由,而是在改进算法中选择性的使用。源节点或中间转发节点根据信息包的目的地地址检查自己的邻居列表,如果发现目的地是自己的邻居则直接传送,不再通过对目的地地址的计算来决定包的转发方向,这样就减少了传送时延和转发包的跳数,相应地减少了网络能耗。若邻居列表中没发现目的节点,则按照原分层算法将包向上或向下传送。在图 3 中, M 是源节点, O 是目的节点,使用 ZiCL 算法按照树结构分层转发经过 M-L-N-O 这条路径需要 3 跳,而在改进算法中 M 通过检查自己邻居列表发现 O 是自己的邻居,直接将消息发送给 O,仅需 1 跳。

在改进算法中,我们为能量偏低节点引入延迟因子。在使用 AODV 算法的簇间路由阶段,当节点收到一个 RREQ 包时,并不是立即对包进行发送,而是判断自己是否能量偏低。若是,则延迟 T_{delay} 时间来发送包,这样其他节点在收到该能量偏低节点发送过来的延时包之前,有可能已经收到了从别的节点送来的发往同一个目的地的相同 RREQ 包,当其他节点再次收到该包时则丢弃它,从而在路由选择时会绕过这个能量偏低的节点。

在整个路由过程中,设置节点的最初能量均为 1000 J。设置阈值 $\varepsilon = \frac{1}{4} \times 1000 \text{ J} = 250 \text{ J}$,当路由节点当前剩余能量 repower 小于 ε 时,认为该节点能量偏低。

每个路由节点的 T_{delay} 值初始为 0,路由节点每收到一个新的 RREQ 包首先检测自身能量状态。当发现 $\text{repower} < \varepsilon$ 时,确定自身处于能量偏低状态,于是修改延迟因子 T_{delay} ,使延迟 T_{delay} 时间来发送 RREQ 包。

$$T_{\text{delay}} = \left(\frac{3}{2} \right)^{\frac{250}{\text{repower}}} + T_{\text{delay}} \quad (1)$$

若该能量不足节点为网关节点,查询该网关节点对应的两个簇是否存在候选网关节点,如果存在,就用剩余能量较大的候选网关节点替代能量较低的原网关节点,同时修改经过源节点和目的节点所在簇的有效路径所对应的路由表的相应表项,将下一跳由原网关节点改为候选网关节点。这样原网关节点就减少了在两个簇之间转发的数据包数量,减轻了工作量。

3.3 重新选择父节点

在 ZigBee 网络中,路由功能节点除了进行路由发现和数据转发外,还要充当终端设备与外界通信的桥梁,其终端设备都要通过它来发送和接收数据,这也要消耗一部分能量。

在利用 3.2 节中的方法对节点路由做出改进以后,划定一个 ΔT 时间段,称为效果评定时间段,根据节能效果对能量偏低节点做进一步改进。 $\Delta \text{repower}$ 表示 ΔT 时间段内节点的能量消耗, λ 是设定的一个能耗上限,在 ΔT 时间内,改进算法还对能量不足节点的每一个终端设备都启动一个计数器。在这个时间段内,终端节点每发送一个数据包给其父节点,让该终端节点对应的计数器加 1。 ΔT 时间过后,如果 $\Delta \text{repower} > \lambda$,说明调整以后节点的节能效果并不是很理想,为了进一步降低能耗,我们希望通过减少其子节点的数量来减少能量偏低并且节能效果不理想节点的负担。考虑到为一个路由节点重新选择父节点可能会修改该节点的深度值,这

就有可能造成基于深度划分的网络簇结构发生很大变化,同时该路由节点可能是其他节点的父节点,这就会造成大规模的短地址的重新分配,增大网络开销,因此不采用断开两个路由节点之间的父子关系来减少负担的方法。改进算法中通过为相应计数器最大(也就是业务量最大)的终端节点重新选择父节点来降低原父节点的能耗。

终端节点在寻找新的父节点的过程中,综合考虑候选节点的剩余能量以及该节点和候选节点之间的相对稳定性。相对稳定性包括两节点之间距离的相对稳定性以及通信状态的相对稳定性。终端节点发出一个探测信号给所有邻居路由节点,每一个邻居收到这个信号以后,都返回一个回复信号,其中携带了自己的当前剩余能量 repower 以及地址标识 flag ,当该节点还有空余地址来供新节点加入时 $\text{flag} = 1$;否则 $\text{flag} = 0$ 。用 F_s 表示回复信号的发送功率,用 F_r 表示回复信号的接收功率。两节点间位置的相对移动以及通信信道的不稳定都会影响到接收点信息接收的功率,用 $M = \frac{F_r}{F_s}$ 表示两者的比值

($0 < M \leq 1$)。 M 可以相对准确地反映两节点间的稳定程度: M 较大时两节点的稳定性较好; M 较小时稳定性较差。使用式 (2) 计算该终端节点的所有邻居节点的 η 值,取 η 最大的候选邻居节点作为该终端设备的父节点。 α, β 表示候选节点的剩余能量以及两节点间的相对稳定性在父节点选择时所占得的比重:

$$\eta = \text{flag} \times (\alpha \times \text{repower} + \beta \times M) \quad (2)$$

其中 $0 < \alpha \leq 1, 0 < \beta \leq 1, \alpha + \beta = 1$ 。在仿真实验中,取 $\alpha = 0.7, \beta = 0.3$ 。

对于图 3 中的节点 C,当在效果评定时间内发现它的节能效果不理想,则选取它的终端设备 G (如果 C 有多个终端设备,就选择 ΔT 时间内计数值最大的那个设备),通过式 (2) 为 G 选择一个新的父节点 D,断开 CG 之间的父子连接,建立新的 DG 父子连接,D 同时为 G 分配新的短地址。

4 算法分析和仿真实验结果

4.1 算法分析

本文的改进算法是在 ZiCL 算法的基础之上,考虑到节点能量状态来决定路由。本算法中所有消息的广播均限制在一跳范围之内。改进算法过程简单,由各节点独立计算。算法的时间复杂度为 $O(N)$,可以在常量时间内结束。

4.2 仿真与结果

改进算法通过仿真实验和原 ZiCL 算法进行了比较,重点比较两者在数据传送过程中网络总体能耗及网络失效节点个数。仿真结果证明了改进算法的有效性。

仿真工具采用 Omnet + + 3.2p1。网络覆盖面积为 $300 \text{ m} \times 500 \text{ m}$,网络节点数目设置为 100 个,节点在网络中随机分布,初始能量均为 1000 J。采用的传输信道数据传输率为 250 Kbps,数据包长度为 128 b。仿真实验结果如图 4.5 所示。

由图 4 可以知道,改进算法在寻找候选网关节点和终端节点重新寻找父节点的过程中广播探测信息需要消耗一定的能量,但通过探测信息的发送可以尽量避免把包传给一些能量偏低节点,减少节点失效的情况,从而大大减少了丢包和重传的数量,这在很大程度上节约了能耗。仿真结果证明新算法比原算法具有更好的节能性。

由图 5 可以知道,改进算法对能量偏低的节点采取了一系列的保护措施,尽量避免这些节点由于能量继续过度消耗而成为失效节点,减少了失效节点的个数。

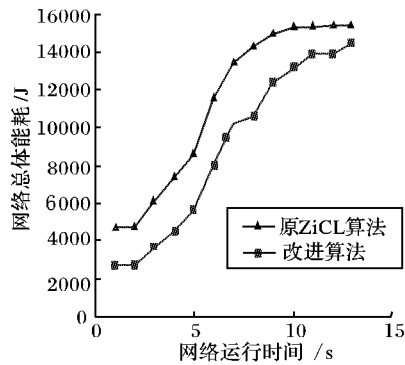


图4 网络总体能量消耗

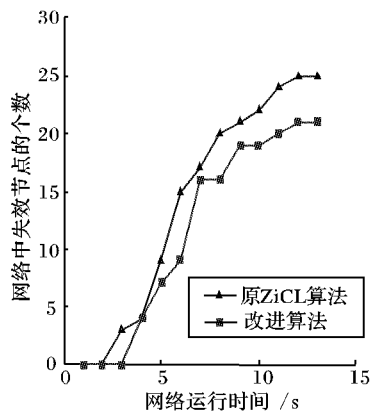


图5 网络总体失效节点的个数

5 结语

本文对基于 ZigBee MESH 网络结构的 ZiCL 分簇算法做

了一些改进,在路由过程中根据节点能耗和 ZigBee 节点特性来选择路由,从而实现了节能的效果。但是该算法在路由过程中有可能为了避开能量偏低的节点而增加了数据传输的延迟。未来的工作对该算法进一步优化,使网络路由在时效上有所提高。

参考文献:

- [1] LEE K K, KIM S H, CHOI Y S, *et al.* A mesh routing protocol using cluster label in the ZigBee network[C]// 2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems. New York: IEEE, 2006: 801 - 806.
- [2] 王智军. 短距离无线通信 ZigBee 技术及其应用[J]. 赤峰学院学报: 自然科学版, 2008(1): 78 - 81.
- [3] LIANG N C, CHEN P C, SUN T, *et al.* Impact of node heterogeneity in ZigBee mesh network routing [C]// Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics. New York: IEEE, 2006: 187 - 191.
- [4] ONDREJ S, ZDENEK B, SUN F, *et al.* ZigBee technology and device design[C]// Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies. Washington, DC: IEEE Computer Society, 2006: 129.
- [5] BARONTI P, PILLAI P, CHOOK V, *et al.* Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards[EB/OL]. [2008 - 07 - 23]. <http://ee.psu.ac.th/~wiklom/PhDSeminar/zigbee.pdf>.
- [6] RAN PENG, SUN MAO-HENG, ZOU YOU-MIN. ZigBee routing selection strategy based on data services and energy-balanced ZigBee routing[C]// 2006 IEEE Asia-Pacific Conference on Services Computing. Washington, DC: IEEE Computer Society, 2006: 400 - 404.

(上接第 495 页)

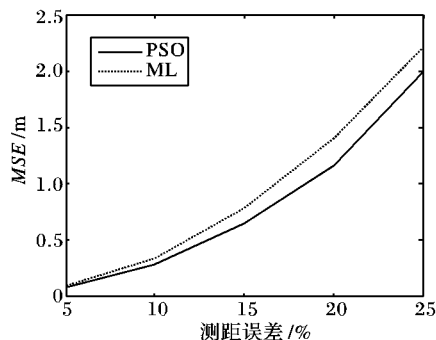


图3 定位均方差

由以上实验结果可知,在无线传感器网络节点定位中, PSO 算法性能要优于 ML 算法,尤其在测距误差较大情况下, PSO 算法与 ML 算法相比有更高的精度和稳定度。这主要因为, ML 算法采用 LS 算法,在测距误差很小情况下,引入 LS 算法的噪声影响可以忽略, ML 算法性能接近最优解。当测距误差增大后,引入 LS 算法的噪声影响变大且不可忽略,致使 ML 算法性能恶化。而 PSO 算法无论在测距误差较小还是较大情况下,都能通过多次迭代寻找最优解,从而减弱了测距误差引入的噪声影响,其性能较 ML 算法优越。而且 PSO 算法主要是一些加、减、乘运算,不涉及复杂的矩阵求逆运算,其算法复杂度也较 ML 算法优越。但 PSO 算法需要多次的迭代寻优,其运行速度较 ML 算法慢。

4 结语

本文将粒子群优化算法应用于无线传感器网络节点定位

中,该算法首先初始化一个随机粒子群,然后根据适应度值更新粒子速度和位置,通过迭代搜索最佳坐标。仿真实验表明: PSO 算法性能优于 ML 算法,尤其在测距误差较大情况下,与 ML 算法相比有更高的精度和稳定度,在 WSN 应用中具有可行性。但这只是对粒子群优化算法在无线传感器网络节点定位应用中的初步探索,对于这两种算法的计算量和能耗大小还有待进一步研究。

参考文献:

- [1] 孙利民, 李建中, 陈渝, 等. 无线传感器网络[M]. 北京: 清华大学出版社, 2005: 135 - 155.
- [2] 于海斌, 曾鹏. 智能无线传感器[M]. 北京: 科学出版社, 2006: 165 - 170.
- [3] 王建刚. 加权最小二乘估计在无线传感器网络定位中的应用[J]. 计算机应用研究, 2006, 23(9): 40 - 43.
- [4] 张贤达. 现代信号处理[M]. 北京: 清华大学出版社, 2002: 49 - 59.
- [5] 杨维, 李歧强. 粒子群优化算法综述[J]. 中国工程科学, 2004, 6(5): 87 - 93.
- [6] 李俊峰. 粒子群优化算法在 TDOA 定位中的应用[J]. 应用科技, 2005, 32(10): 7 - 9.
- [7] EBERHART R C, SHI Y H. Particle swarm optimization: developments, applications and resources[C]// Proceedings of the IEEE Congress on Evolutionary Computation. Piscataway, NJ: IEEE Service Center, 2001: 81 - 86.
- [8] PARASPOULOS K E, PLAGIANAKOS V P, MAGOULAS G D, *et al.* Improving particle swarm optimizer by function stretching[M]. The Netherlands: Kluwer Academic Publishers, 2001: 445 - 457.
- [9] 侯志荣, 吕振肃. 基于 Matlab 的粒子群优化算法及其应用[J]. 计算机仿真, 2004, 20(10): 68 - 70.