

面向 WSN 的移动锚节点路径规划算法*

黄冰倩, 杜庆治, 龙 华

(昆明理工大学 信息工程与自动化学院, 云南 昆明 650500)

摘要:针对无线传感器网络中使用移动锚节点辅助未知节点定位问题,提出了一种动态路径规划算法.该算法首先引用相关图论知识,把无线传感器网络看成一个连通的无向图,将传感器节点转化为图的顶点选取虚拟信标节点,通过蚁群算法遍历所选取的节点获得移动路径.并对传统宽度优先搜索算法中节点选取冗余和经典蚁群算法中存在的收敛速度慢等问题提出改进.仿真实验结果表明,改进算法能在保证一定通信覆盖率的情况下减少节点重复遍历以及锚节点的移动距离.

关键词:移动锚节点;无线传感器网络;宽度优先搜索算法;蚁群算法

中图分类号: TP 391 **文献标志码:** A **文章编号:** 0258-7971(2018)01-0029-07

无线传感器网络(WSN)由大量传感器节点组成,通过节点传递信息,可应用在军事、医疗、环境科学以及自然灾害监测等诸多领域.对于大多数应用而言,位置信息意义重大,因此节点定位技术成为 WSN 的关键技术之一,未知节点可以通过与已知位置信息的锚节点进行通信来辅助自身定位^[1].

目前根据节点是否移动主要分为固定锚节点和移动锚节点,采用固定锚节点辅助定位时需要锚节点数量达到一定程度来满足节点之间的连通性^[2],通常未知节点的定位精度随锚节点数量的增多而增高,这也就增加了网络的成本.而利用可以获取自身位置信息的锚节点移动来产生虚拟的信标节点,可以很大程度上减少锚节点数量,同时可以提高定位精度^[3-4].

近年来,有大量文献在关于无线传感器网络节点定位技术中提出了移动锚节点的定位算法.文献[5]中为了解决重复扫描未知节点问题,提出了一种基于 3 个移动信标和 TDOA 技术测距为未知节点提供距离信息的定位算法.文献[6]提出一种多个信标移动的定位算法 MMB-DV-Hop,利用多个移动信标遍历整个网络,并使用 RSSI 测距获得节点之间的距离信息进行未知节点的定位,将测距和

非测距 2 种算法的特点有效地结合在一起.

使用移动锚节点求解未知节点位置信息时,锚节点沿着设计好的路径遍历整个网络区域,为此,许多文献对锚节点的移动轨迹规划问题进行了研究^[7-9].文献[10]提出利用刚性理论将网络进行子图划分,再遍历各子图以获取最短路径,锚节点沿着该路径移动从而定位各单元.文献[11]研究了移动信标在障碍物情况下的路径规划方法,在 RSS 算法基础上提出一种 Z 字型移动路径定位方法,将网络分成 4 个正方形区域进行路径规划.文献[12]提出将节点的路径规划转化为经典的选址问题和 TSP 问题,将两者结合起来,并利用贪婪算法搜索最优的信标点从而获得最佳路径.

在本文的锚节点路径规划问题研究中,首先将传感器节点转化为图的顶点确定传感器网络区域内虚拟信标节点分布,再通过蚁群算法遍历这些信标节点进行路径规划,并对传统图的生成树遍历方法-广度优先搜索算法(Breadth First Search, BFS)和传统蚁群算法提出一些改进.

1 移动锚节点发射位置相关算法

1.1 传统 BFS 算法 假设移动信标的通信半径

* 收稿日期:2017-04-27

基金项目:云南省科技惠民计划(2014RA051).

作者简介:黄冰倩(1992-),女,安徽人,硕士生,主要研究方向为信息与通信工程.E-mail:632817570@qq.com.

通信作者:杜庆治(1977-),男,云南人,高级实验师,主要研究方向为信息与通信工程.E-mail:57960748@qq.com.

为 R , 在移动信标上安装 GPS 接收器, 由于存在路径损耗等原因, 传感器节点可接受信号的传播范围为 $\frac{2}{3}R$. 根据 RSSI 值, 将其转化为 2 个节点之间的距离 d 和通信半径 R 的比较.

邻居节点: 移动信标访问 WSN 节点时, 如果节点 v_n 到 v_i 的距离 $d < R$, 称节点 v_n 为 v_i 的邻居节点.

内部节点: 移动信标访问 WSN 节点时, 如果节点 v_n 到 v_i 的距离 $d < \frac{3}{4}R$, 称节点 v_n 为 v_i 的内部节点.

边缘节点: 移动信标访问 WSN 节点时, 如果节点 v_n 到 v_i 的距离 $\frac{3}{4}R < d < R$, 称节点 v_n 为 v_i 的边缘节点.

在选取虚拟信标时, 移动锚节点会首先随机选取一个传感器节点 v_i 进行访问, 并在访问过程中定期向邻居节点广播自身位置信息^[13]. 当邻居节点 v_n 与被访问的节点 v_i 互为内部节点时, 则说明该节点 v_n 可以接收到 v_i 的位置信息, 可以利用三边定位原理估算出自身位置. 当判断出节点 v_n 是边缘节点时, 将其直接存入虚拟信标候选集. 依次循环, 直到所有节点都被访问.

1.2 改进的 BFS 算法 本节提出一个权重值 W_i 的概念, 统计每个传感器节点周围的邻居节点数量作为其权重值. 并提出如下几点改进:

(1) 针对传统 BFS 算法中可能出现同层边缘节点互为内部节点, 或者边缘节点与上层不是父节点的其他节点互为内部节点而重复遍历的情况, 选取距离访问节点更近的点加入候选集, 另外一点加入禁忌集, 以此来减少边缘节点的重复遍历.

(2) 与传统 BFS 算法中随机选取节点不同的是, 当一棵树搜索完成后, 如果还有未遍历的节点, 改进算法选取其中权重值最大的节点作为移动信标的下一个访问节点, 直至所有节点都被访问.

算法步骤:

步骤 1 传感器节点随机分布, 网络初始化, 统计未知节点周围邻居节点的数量, 将数量值作为该未知节点的权重值 W_i ;

步骤 2 选取连通无向图权重值 W_i 最大的节点 v_i 为起始节点, 移动锚节点访问 v_i , 遍历 v_i 的邻居节点 v_n ;

步骤 3 当 v_n 是内部节点时, 存入禁忌集 J , 若

v_n 是边缘节点则加入虚拟信标候选集 H ;

步骤 4 如果访问的节点 v_i 不存在边缘节点, 则选取网络中未被搜索的节点中权值 W_i 最大的点作为下一个访问目标;

步骤 5 如果访问的节点 v_i 存在边缘节点, 且候选集 H 中存在节点与其互为内部节点时, 则选择距离父节点较近的边缘节点加入 H , 较远的加入禁忌集 J ;

步骤 6 更新未知节点的权值 W_i ;

步骤 7 如果仍存在未被遍历的节点, 返回步骤 2 重新遍历;

步骤 8 直到遍历完所有节点后, H 中的节点即可被选为虚拟信标节点.

改进的 BFS 算法设计的优点在于可以在保证一定通信覆盖率的情况下选取更少的虚拟信标, 能有效减少传统 BFS 算法选取虚拟信标时的冗余现象.

2 移动锚节点路径规划算法

遍历虚拟信标节点获取移动锚节点的最短移动路径, 这类似于经典的旅行商问题, 蚁群算法被广泛应用其中^[14].

2.1 经典蚁群算法 当蚂蚁在某条路径上移动时, 会释放出一种叫信息素的物质, 且移动的路径越短, 该路径上留下的信息素浓度越高, 吸引的蚂蚁数量越多, 而较长的路径由于移动时间较长遗留的信息素较少被蚂蚁选择的概率就会降低. 本节采用蚁群算法遍历已获取的虚拟信标来完成移动锚节点的最短路径规划.

设有 n 个节点需要遍历, 将 m 只蚂蚁随机部署在 n 个节点位置上, 节点 i 到 j 的距离为 d_{ij} (通过 RSSI 计算得到), $\tau_{ij}(t)$ 表示在 t 时刻路径 ij 上的信息素浓度. $\delta_{ij}(t)$ 表示 t 时刻蚂蚁从节点 i 移动到 j 的期望度, 通常 $\delta_{ij}(t) = 1/d_{ij}$. 则第 k ($k = 1, 2, 3, \dots, m$) 只蚂蚁在 t 时刻由节点 i 移动到节点 j 的概率 $P_{ij}^k(t)$ 可表示为

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\delta_{ij}(t)]^\beta}{\sum_{s \in \omega_k} [\tau_{is}(t)]^\alpha [\delta_{is}(t)]^\beta}, & j \in \omega_k, \\ 0, & \text{others.} \end{cases} \quad (1)$$

α, β 分别为信息素启发因子期望启发因子. $\omega_k = \{n - \varphi_k\}$ 代表第 k 只蚂蚁尚未遍历的信标节点集合, φ_k 是第 k 只蚂蚁已遍历过的信标节点集合.

经过 s 时刻后,当 m 只蚂蚁访问完所有节点后,需要对路径上的信息素浓度进行更新,节点 i 与节点 j 路径之间残留信息素浓度为

$$\tau_{ij}(t+s) = (1-\rho)\tau_{ij}(t) + \rho \sum_{k=1}^m \Delta\tau_{ij}^k, \quad (2)$$

其中, $\rho(0 \leq \rho \leq 1)$ 为信息素挥发系数, $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁移动后路径 ij 上的信息素增量。

2.2 算法改进 蚁群算法在利用正反馈机制获取最短路径的同时也存在一些缺陷,如搜索路径时间过长,造成收敛速度慢,且易造成局部最优,故本文针对这些问题提出以下几点改进:

2.2.1 初始信息素浓度的改进 在蚁群算法中,由于刚开始各局部路径信息素浓度相同,蚂蚁搜索时没有明确的指向,使得搜索时间过长,且增加许多无关路径的信息素浓度.因此,提出对初始信息素浓度进行改进:

$$\tau_{ij}(0) = 1 - \frac{d_{ij}}{L_i}, \quad (3)$$

$$L_i = \sum_{i \neq j}^m d_{ij}, \quad (4)$$

其中, d_{ij} 表示信标节点 i 和 j 之间的距离, L_i 表示信标节点 i 到其他信标节点的距离之和。

通过公式(3)对信息素浓度进行初始化设置,相当于在搜索起始时对节点引入不同权重,路径短浓度高,反之浓度低,为蚂蚁初始状态下的搜索进行合理的方向引导,加快了收敛速度。

2.2.2 全局信息素更新规则的改进 传统算法中,蚂蚁在单次迭代完成后,如果获取的最短路径为全局最优路径时,会加强路径上信息素浓度,这样也会造成部分较短路径信息素浓度累积过多,进而影响获取全局最优路径^[15].为了避免这种情况的发生,本文提出对全局信息素的更新规则进行改进,在全局规则中引入动态因子 $\mu(\mu \in (0,1))$,该因子在每次迭代中能根据路径长短自适应地更新最优解中信息素浓度。

其更新规则如下:

$$\tau_{ij}(t+s) = \tau_{ij}(t) + \varphi\mu\Delta\tau_{ij}, \quad (5)$$

$$\mu = \frac{1}{1 + e^{\left(\frac{L_{\min} - L_m}{L - L_{\min}}\right)}}, \quad (6)$$

其中, φ 为固定参数, \bar{L} 为每次迭代后获取的最短路径的平均长度, L_{\min} 为当前全局最短路径长度, L_m 为本次迭代获取的最短路径长度。

由上式可以看出,当 L_m 越大,即本次迭代获取

的最优路径越长时, μ 越接近于 0;反之,即本次迭代获取路径越短时, μ 越接近于 1,使得该路径上的信息素浓度会显著增加.并且, μ 是根据函数 $f(x) = \frac{1}{1 + e^{-x}}$ 而来,由 $f(x) = \frac{1}{1 + e^{-x}}$ 的函数曲线可知,当 $x \in (-2,2)$ 时, $f(x)$ 变化率大,能显著区分每次迭代获取的最短路径上的信息素浓度,当 $x < -2$ 时, $f(x)$ 的值缓慢趋向于 0,使较长路径上的信息素浓度不会因为挥发而减小到零;当 $x > 2$ 时, $f(x)$ 的值缓慢趋向于 1,从而避免一些非全局最优路径的较短路径信息素累积过多,导致陷入局部最优解。

算法步骤:

步骤 1 参数初始化,初始迭代次数 N_d 设为 0,最大迭代次数为 N_{\max} . $t=0$ 时,根据式(3)为信标节点之间的路径设置初始信息素浓度值 $\tau_{ij}(0)$,将 m 只蚂蚁随机放置在 n 个虚拟信标位置上,禁忌表 $\omega_k = 0(k=1,2,\dots,m)$;

步骤 2 将 m 只蚂蚁所在信标节点 i 存入 φ_k ,根据式(1)计算第 k 只蚂蚁的转移概率 $P_{ij}^k(t)$,将下一个要遍历的信标节点 j 存入 φ_k ;

步骤 3 若蚂蚁 k 仍存在未遍历的节点,返回步骤 2 继续遍历;若蚂蚁 k 已遍历完所有节点,则执行步骤 4;

步骤 4 当 m 只蚂蚁遍历完都所有节点后,会产生 m 条路径,根据式(5)更新最短路径上的信息素浓度;

步骤 5 若此时 $N_d \leq N_{\max}$,则返回步骤 1,否则,比较每次迭代产生的最优解,输出全局最优解。

3 仿真结果分析

3.1 虚拟信标位置选取 在 Matlab 平台上对上述路径规划算法进行仿真,首先在 $200\text{m} \times 200\text{m}$ 区域随机部署 50 个节点,可以根据 RSSI 信号传播损耗模型^[16]计算得到节点之间的距离,通信半径设为 30 m 时,如图 1 和图 2 所示,星号表示 2 种算法选取的信标节点。

在图 1 中移动锚节点所选取的虚拟信标数为 31 个,图 2 中所选取的信标数为 23 个,改进 BFS 算法相对于传统算法而言,在一定程度上减少了边缘节点的重复遍历,因此选取的虚拟信标数量有所减少。

接下来对不同通信半径下的 2 种算法所选取

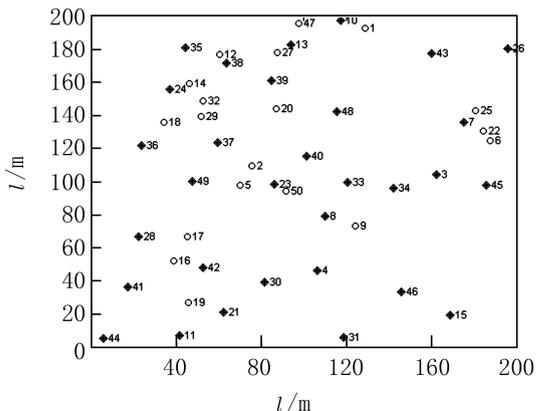


图 1 经典 BFS 选取信标个数

Fig.1 The number of beacons selected by classical BFS

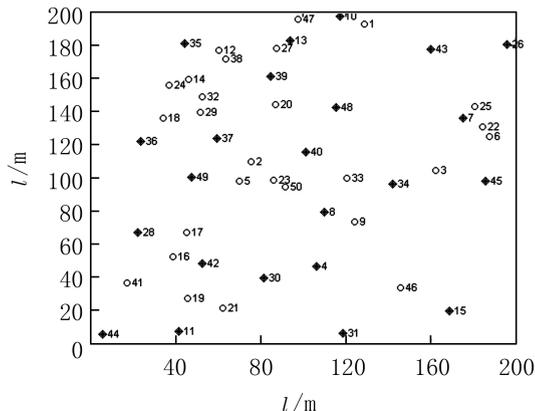


图 2 改进 BFS 选取信标个数

Fig.2 The number of beacons selected by improved BFS

的信标节点个数进行比较.同样将在区域内随机放置 50 个节点,通信半径从 20 m 到 60 m 变化.2 种算法选取的虚拟信标数目变化趋势如图 3 所示.

由图 3 可以看出,在通信半径逐渐增大时,由于 BFS 算法可选取的内部节点越来越多,被存入禁忌表,所以选取的虚拟信标数量也随之减少.且在不同通信半径下,改进的 BFS 算法选取的信标节点个数均少于传统的 BFS 算法.

除了通信半径外,传感器节点密度的变化也会影响虚拟信标节点的数量.因此针对传感器密度对选取信标节点的影响进行了仿真:设置仿真区域为,由于传感器节点增多,因此通信半径选为 40 m,传感器节点个数从 100 到 500 变化,比较 2 种算法所选取的虚拟信标数量,如图 4 所示.

仿真结果表明,当仿真区域内节点密度增大

时,选取的虚拟信标也随之增多,由图 4 可以看出改进后的 BFS 算法选取的信标节点数量明显减少,降低了网络成本.

3.2 信标移动路径仿真 同样,为了验证改进蚁群算法的有效性,通过仿真实验对两种算法进行了分析比较.在仿真区域设置 50 个节点,通信半径为 30 m,在改进的 BFS 算法所选取的 23 个信标节点中寻找最佳路径.设蚂蚁个数 $m = 50$,迭代次数 N_{max} ,信息素 $Q = 500$, $\alpha = 1$, $\beta = 5$, $\rho = 0.1$, 2 种算法得到的最优路径如图 5、图 6 所示.

由图 5、图 6 可以看出,在参数相同的条件下,由于搜索方式的不同,2 种算法所获取的最短路径也不一样,其中经典蚁群算法最短路径约为 977 m,改进算法后最短路径约为 933 m,减少了锚节点移动距离.

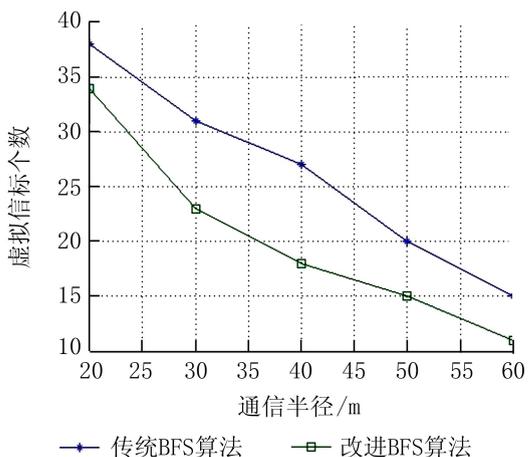


图 3 2 种算法虚拟信标节点选取对比图

Fig.3 Comparison of two algorithms for virtual beacon node selection

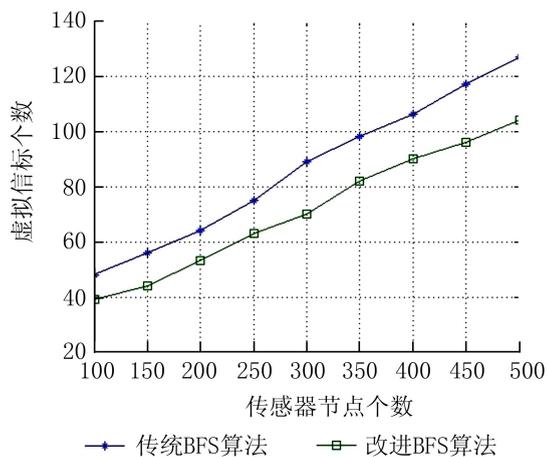


图 4 不同节点密度下的虚拟信标数对比图

Fig.4 Comparison of virtual beacon numbers at different node densities

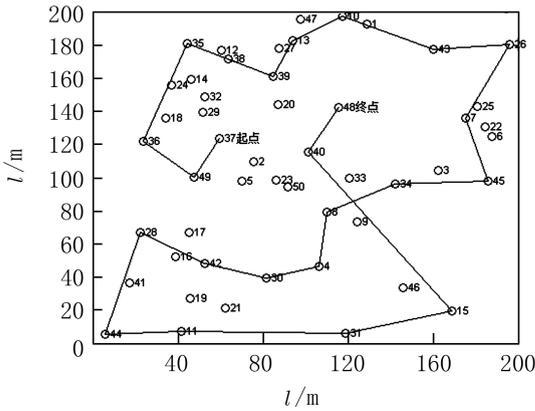


图 5 经典蚁群算法最优路径

Fig.5 Optimal path of classical ant colony algorithm

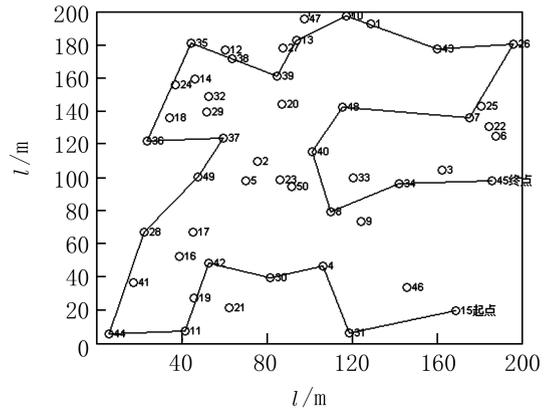


图 6 改进的蚁群算法最优路径

Fig.6 Optimal path of improved ant colony algorithm

图 7 分析了迭代 100 次的情况下,2 种算法的最短路径对比,由图 7 可以看出,改进算法由于加入了方向引导,收敛速度显著提高,并且每次迭代获得的最短路径均小于经典蚁群算法。

3.3 定位性能评价 由上述实验可以看出改进算法选取的节点数和锚节点移动路径都优于传统算法,但在此基础上若能保证一定的定位覆盖率才足以说明改进算法的优越性.为此,利用改进的算法对不同节点密度和通信半径下的未知节点可定位比例进行了仿真分析.设仿真区域为 200 m×200 m,通信半径为分别为 20, 40, 60 m,传感器节点个数从 100 到 300 变化,其仿真结果如图 8 所示。

由图 8 可知,在通信半径和节点密度不同的情况下,未知节点的可定位比例随着通信半径的增大而增高,且均可达到 70%以上,保证了改进算法的有效性。

4 结 论

本文主要研究了 WSN 中移动锚节点路径规划问题,提出改进的宽度优先搜索算法选取虚拟信标位置,结合改进的蚁群算法动态获取最优移动路径.通过理论分析和仿真实验表明,改进的算法能用更少的虚拟信标点有效地覆盖传感器网络,移动路径更短且高效,节约了移动锚节点的能耗。

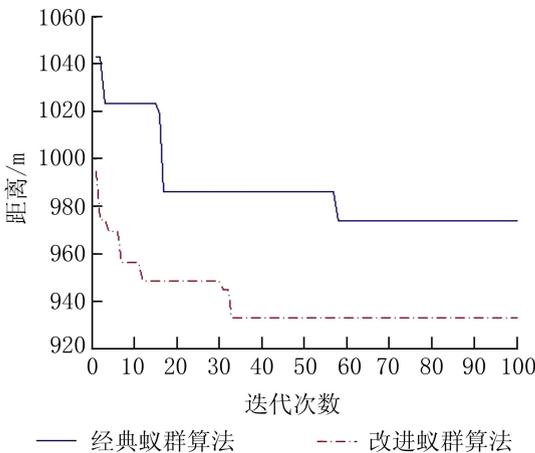


图 7 2 种算法仿真结果对比

Fig.7 Comparison of simulation results of two algorithms

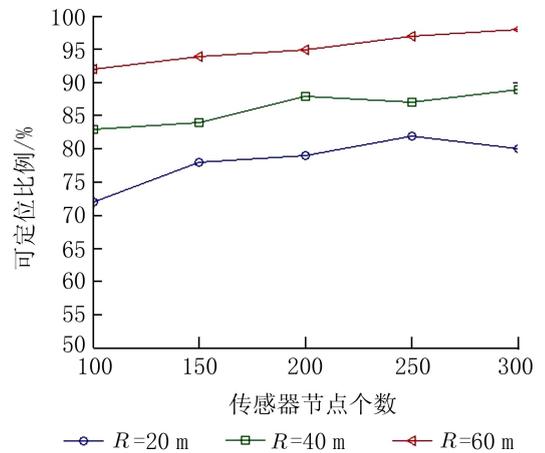


图 8 定位覆盖率对比图

Fig.8 Comparison of location coverage

参考文献:

- [1] 程海军. 基于 RSSI 的无线传感器网络定位算法改进研究[D]. 重庆: 重庆理工大学, 2013.
CHENG H J. Research on localization algorithm improvement of wireless sensor network based on RSSI [D]. Chongqing: Chongqing University of Technology, 2013.
- [2] 杨铮, 吴陈沐, 刘云浩. 位置计算: 无线网络定位与可定位性[M]. 北京: 清华大学出版社, 2014.
YANG Z, WU C S, LIU Y H. Location-based computing: Localization and localizability of wireless network [M]. Beijing: Tsinghua University Press, 2014.
- [3] HALDER S, GHOSAL A. A survey on mobile anchor assisted localization techniques in wireless sensor networks[J]. *Wireless Networks*, 2016, 22(7): 2317-2336.
- [4] KARTHIGA G, PREETHI C, DELSHI H D R. Localization in wireless sensor network based on mobile anchor and chord selection[C]//International Conference on Communication and Network Technologies, IEEE, 2014: 124-128.
- [5] 崔焕庆, 王英龙, 郭强, 等. 多移动信标辅助的分布式节点定位方法[J]. *通信学报*, 2012, 33(3): 103-111.
CUI H Q, WANG Y L, GUO Q. Distributed node localization method assisted by multiple mobile beacons[J]. *Journal of Communications*, 2012, 33(3): 103-111.
- [6] 任明明, 谢志军, 金光, 等. 基于多移动信标的 DV-Hop 定位算法[J]. *计算机工程*, 2014, 40(10): 92-97.
REN M M, XIE Z J, JIN G. DV-Hop location algorithm based on multiple mobile beacons[J]. *Computer Engineering*, 2014, 40(10): 92-97.
- [7] MONDAL K, KARMAKAR A, MANDAL P S. Path planning algorithm for mobile anchor in connected sensor networks[M]//Distributed Computing and Internet Technology. Springer International Publishing, 2015: 302-308.
- [8] SACHIN G V, BINU G S. Mobile anchor path planning in wireless sensor networks[C]//International Symposium on Advanced Computing and Communication, IEEE, 2016: 315-319.
- [9] 曹晓清. WSN 激励节点移动定位路径规划研究[D]. 成都: 西南交通大学, 2013.
CAO X Q. Research on path planning of WSN excitation node moving positioning [D]. Chengdu: Southwest Jiaotong University, 2013.
- [10] 汪晗, 齐望东, 王坤. 无线传感器网络中基于刚性的移动锚节点路径规划[J]. *电子与信息学报*, 2011, 33(10): 2353-2357.
WANG H, QI W D, WANG K. Path planning for mobile anchor nodes based on rigidity in Wireless Sensor Networks [J]. *Journal of Electronics and Information*, 2011, 33(10): 2353-2357.
- [11] JAVAD R, MARJAN M, ABDUL S I, et al. Superior path planning mechanism for mobile beacon-assisted localization in wireless sensor networks[J]. *IEEE Sensors Journal*, 2014, 14(9): 3052-3064.
- [12] 张玉强, 何泾沙, 徐晶, 等. 移动参考节点动态路径最优规划[J]. *北京工业大学学报*, 2016, 42(6): 851-855.
ZHANG Y Q, HE J S, XU J, et al. Dynamic path optimal planning for mobile reference nodes[J]. *Journal of Beijing University of Technology*, 2016, 42(6): 851-855.
- [13] 张健. 无线传感与激励网络中激励节点移动定位及路径优化算法[D]. 石家庄: 河北师范大学, 2015.
ZHANG J. Mobile node localization and path optimization algorithm for wireless sensor and incentive networks [D]. Shijiazhuang: Hebei Normal University, 2015.
- [14] LI C B, JIA H F, CHEN W D, et al. Application of a new improved ant-algorithm in TSP[J]. *Applied Mechanics & Materials*, 2013, 291-294(3): 2905-2908.
- [15] 袁亚博, 刘羿, 吴斌. 改进蚁群算法求解最短路径问题[J]. *计算机工程与应用*, 2016, 52(6): 8-12.
YUAN Y B, LIU Y, WU B. Improved ant colony algorithm for solving the shortest path problem [J]. *Computer Engineering and Application*, 2016, 52(6): 8-12.
- [16] XU J, LIU W, LANG F, et al. Distance measurement model based on RSSI in WSN [J]. *Wireless Sensor Network*, 2010, 2(8): 606-611.

Path planning algorithm of mobile anchor node for WSN

HUANG Bing-qian, DU Qing-zhi, LONG Hua

(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

Abstract: A dynamic path planning method is proposed in order to solve the problem of localization in Wireless Sensor Networks with mobile beacon. Firstly, considering the graph theory, WSN is regarded as a connected undirected graph, the sensor node is transformed into the vertex of the graph and the virtual beacon node is selected, then using the ant colony algorithm to obtain the optimal path through these virtual beacon points. This method also improves the problem of node selection redundancy in traditional breadth first search algorithm and accelerates the convergence speed in the classical ant colony algorithm. Simulation results show that the improved algorithm can reduce the repeated traversal of nodes and shorten the moving distance of anchor nodes under the condition of ensuring a certain coverage rate.

Key words: mobile anchor; Wireless Sensor Networks; breadth first search algorithm; ant colony algorithm