

基于改进蚁群的无线传感器网络路由*

郑慧君, 张巍, 滕少华

(广东工业大学 计算机学院, 广州 510006)

摘要: 在结合无线传感器网络的情况下, 利用蚁群算法对无线传感器网络路由在 MATLAB 中进行了仿真, 并对基本的蚁群算法作了改进, 取得了较好的结果。

关键词: 蚁群算法; 无线传感器网络; 网络路由

中图分类号: TP393.04

文献标志码: A

文章编号: 1001-3695(2010)01-0099-02

doi:10.3969/j.issn.1001-3695.2010.01.029

Improved ACO-based wireless sensor networks routing

ZHENG Hui-jun, ZHANG Wei, TENG Shao-hua

(Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: This paper put this algorithm to wireless sensor network, using ACO to simulate WSN routing in MATLAB, then improved this algorithm and got a good result.

Key words: ant colony optimization (ACO); wireless sensor network (WSN); networks routing

蚁群算法 (ACO) 是由 M. Dorigo 等人^[1]提出的解决组合优化问题的一种多 agent 方法。他们用蚁群算法解决旅行商问题、二次分配问题等, 并得出了理想的结论。蚁群算法因其具有正反馈、分布式计算和富于建设性的贪婪启发式搜索的特点而越来越受到人们的关注。

与传统无线网络相比, 无线传感器网络 (WSN) 一般节点数量巨大, 网络中的节点分布稠密, 各传感节点不可能分配一个全局性的 ID, 也不可能维护任何全局信息; 同时, WSN 节点因能量消耗完而成为无效节点, 使得网络的拓扑结构会经常变动^[2]。因此, 传统无线网络中路由算法不适合 WSN, 必须针对 WSN 的特性来研究新的路由算法。由于传统无线网络的路由算法都是集中式的算法, 要求各传感节点维护全局性的网络状态信息, 对于 WSN 来说, 开销太大, 不具备良好的可伸缩性。蚁群算法只是基于本地信息, 适用于 WSN。

1 相关工作

基本 ACO 路由的思想可以描述如下^[3]:

a) 一群相互协作的蚂蚁寻找从起始节点到目标节点的路径。

b) 蚂蚁通过寻找下一跳节点的方式寻找通往目的节点的路径, 寻找下一跳节点是按照一定的概率, 首次搜索时下一跳节点被选中的概率是相同的。

c) 每条路径上都有一定的信息素, 起始状态所有路径的信息素浓度都是相同的。蚂蚁到达目的节点后得到一个路径评价, 然后根据评价更新路径上的信息素浓度。

d) 反复执行此算法, 直到所有蚂蚁都走上同一条路径或到达算法规定的迭代次数, 算法终止。

目前主要的蚁群算法有 AS (ant system)、MMAS (max-min ant system) 和 ACS (ant colony system)^[1]。AS 中蚂蚁找到目的节点后立即更新该路径的信息素, 该路径的信息素会影响其他蚂蚁在本次迭代过程中路径的选择; MMAS 中只更新最佳路径上的信息素; 与 AS 不同, ACS 是在所有的蚂蚁都找到了目的节点之后才更新信息素浓度, 信息素的更新只对下次迭代过程有影响。苏森等人^[4]研究了基于蚁群的无线传感器网络双簇头算法, 依据信息素浓度在每一簇中选择具有分工特征的主簇头和副簇头, 分别进行数据收集、融合、传输, 较好地平衡了网络的能量消耗, 延长网络的生命周期。Gao^[5]研究了免疫的蚁群算法, 引入了自适应转变操作和免疫的选择操作使蚂蚁移动更有目的性。C. Rajendran 等人^[6]将蚁群算法应用于数据报网络, 通过实验取得了较好的结果。耶刚强等人^[7]对蚂蚁规定了一定的寿命, 当蚂蚁在其寿命内没有到达汇聚节点, 就会死亡。这样就避免了在较差路径上继续探索下去而消耗能量, 但蚂蚁的寿命难以确定, 寿命过小容易使蚂蚁还没有到达目的节点就死亡; 寿命过大则不能避免蚂蚁在较差的路径上探索下去。裴志利等人^[8]对信息素的更新速度作了研究, 但是容易陷入局部收敛。以上的算法都没有考虑算法收敛到局部最优还是全局最优解的问题, 本文重点对这方面作了研究, 对基本 ACO 和改进 ACO 通过仿真实验进行了比较。

2 算法改进与设计

2.1 距离计算和确定邻节点

为了使蚂蚁的移动更具有目的性, 即朝着目标节点的方向移动。在蚂蚁寻找下一跳节点时, 计算出当前节点到目标节点的距离 d_1 和邻节点到目标节点的距离 d_2 , 距离 d 按欧氏距离

收稿日期: 2009-04-07; 修回日期: 2009-06-03 基金项目: 广东省科技计划资助项目 (2008A060201011)

作者简介: 郑慧君 (1985-), 男, 湖北孝感人, 硕士研究生, 主要研究方向为智能算法、图形图像 (jxk0301zhj@163.com); 张巍 (1964-), 女, 江西南昌人, 副教授, 硕士, 主要研究方向为协同工作、网络安全、数据挖掘; 滕少华 (1962-), 男, 江西南昌人, 教授, 博士, CCF 高级会员, 主要研究方向为协同工作、网络安全、数据挖掘。

公式计算。删掉邻节点中满足 $d_1 < d_2$ 的节点,保证了蚂蚁能向着目标节点的方向前进。

2.2 信息素的自适应性

为了使算法能搜索全局的最优路径,防止陷入局部最优,调节信息素的更新速度,使信息素的更新速度具有自适应性,能够随着迭代次数的增加而逐渐减慢。引入了自适应因子 μ (式(5))对信息素的更新速度进行控制,随着迭代次数的增加,蚂蚁所寻找的路径逐渐接近最优解,所以应减小信息素的增加速度,以免陷入局部最优。算法描述如下:

a) 初始化各参数和信息素。初始化网络图邻接矩阵 A 、各邻接边信息素 p 、信息素挥发系数 ρ 、信息素强度系数 Q 、蚁群大小 M 、算法迭代次数 GEN 、路径的延迟矩阵 lc 、各节点的能量值 $energy$ 、控制相关信息素痕迹的可见性参数 α 或是控制路径时延的参数 β 。

b) 转移概率计算。在每个节点 r ,若是第一次迭代则蚂蚁的转移概率是随机的,否则按照 ACO 算法中提到的概率规则式(1)来选择下一跳节点。蚂蚁在寻径的过程中,若到达节点的邻节点集为空,则该蚂蚁死亡。

$$P_i = p(r, i)^\alpha \times \frac{1}{lc(r, i)^\beta} / \sum p(r, i) \quad (1)$$

转移概率直接受信息素强度和路径延迟的影响。

c) 路径评价计算。当所有蚂蚁找到了源节点到目的节点的路径之后利用式(2)计算路径的适应度值。

$$fit = 1 / \sum delay(i) + 1 / cost + \sum energy(i) / 10 \quad (2)$$

其中: $cost$ 是路径的费用,用路径长度表示; $delay$ 是路径的延迟,路径的适应度值与路径的长度路径的延迟以及路径的能量有关,能量越大则适应度值越大,能够更好地平衡网络的能量消耗。

d) 信息素更新。根据计算得到的适应度值利用式(3)更新所有被访问过的路径信息素,对未选中的边则不增加信息素。

$$p_{i,j} = p_{j,i} = (1 - \rho) \times p_{i,j} + \nabla \tau_{i,j} \quad (3)$$

$$\nabla \tau_{i,j} = \begin{cases} \rho \times Q \times fit \times \mu & \text{对于选中的边} \\ 0 & \text{其他} \end{cases} \quad (4)$$

$$\mu = (gen - N) / gen \quad (5)$$

其中: μ 为自适应因子,控制信息素的生长速度; N 为当前的迭代次数。

e) 转到 b) 反复执行,直到执行完规定的迭代次数。

3 实验结果与分析

3.1 环境设置

仿真工具采用 MATLAB 7.0,网络的节点数为 100,蚂蚁的个数为 10,迭代次数为 500 次。信息素挥发系数 $MODULUS = 0.05$,信息素强度系数 $Q = 10$,信息素系数 $alpha = 1$,启发因子数 $beta = 1$,目的节点为 1,目标节点为 58。为了不使当前的最佳路径被覆盖掉,设定一个变量存储整个迭代过程的最佳路径,并根据每次迭代的最佳路径进行更新。最终蚁群所寻找到的就是所有迭代过程中的最佳路径。路径的好坏根据式(7)计算的适应度值的大小来评价。

3.2 算法性能比较

实验时,将本文的改进算法同基本 ACO 算法从搜索路径上、收敛速度和算法稳定性方面进行比较,按照上面的网络环

境,让蚂蚁寻找节点 1~58 的路由。实验结果如图 1 所示。

图 1 为 ACO 所找的路径 [1, 77, 5, 69, 80, 62, 72, 22, 26, 46, 76, 58], 适应度值为 0.141 4。

图 2 为改进 ACO 算法所找的路径 [1, 62, 72, 78, 46, 76, 58], 适应度值为 0.273 8。由图 1 和 2 以及计算出来的路径的适应度值可以看出,改进的 ACO 算法得到了较好的路径。

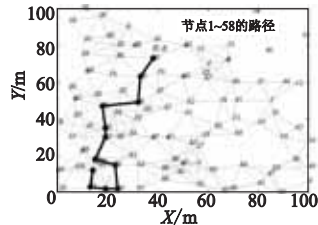


图1 基本ACO路径

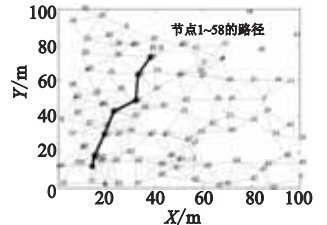
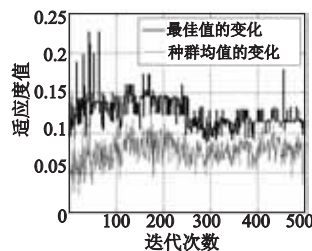
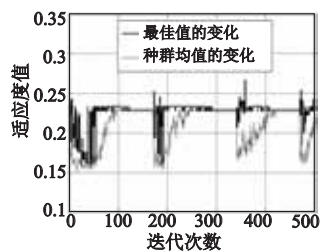


图2 改进ACO算法路径

当所有蚂蚁所找的路径的适应度均值等于最佳路径的适应度值则表示算法收敛,从图上表示就是两个曲线重合。由图 3 可以看出,ACO 算法在进行了 500 次迭代之后还没有收敛;而改进 ACO 算法在运行 110 次就收敛了,为了使算法的搜索有效(因为算法收敛后所有的蚂蚁在以后的迭代中一直会走同一条路径,使算法的迭代无效),在改进后的算法收敛之后防止陷入局部最优,进行收敛判断,收敛后又重新初始化信息素,重新开始搜索。



(a) ACO算法



(b) 改进ACO算法

图3 收敛性比较

分别对基本 ACO 和改进的 ACO 运行了 10 次,图 4 画出了两种算法每次运行过程中的最终路径的适应度值。

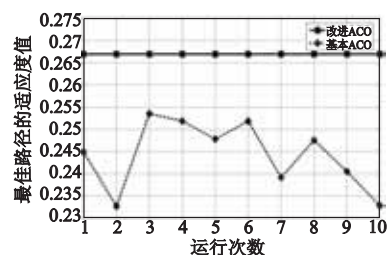


图4 稳定性比较

从图 4 可以看出,ACO 算法 10 次运行后得到的路径都不一样,算法不稳定,改进的 ACO 算法 10 次运行都得到同样的结果,算法比较稳定。根据图上的适应度值可以看出,改进的 ACO 算法得到的路径比 ACO 算法好。

4 结束语

本文对 WSN 的路由进行了研究,对 ACO 算法进行了改进,给出了一种改进的 ACO。在 MATLAB 7.0 中建立了网络模型,对 ACO 算法和改进 ACO 进行了路由仿真实验和对比。实验结果表明,改进的 ACO 算法中蚂蚁能够朝着目标节点的方向搜索,算法在寻找路径、收敛速度和稳定性方面都优于 ACO 算法,能够搜索到全局较好的路径。 (下转第 119 页)

h) 从 P 中移除 P' 和 q 点, 重复步骤 a) ~ g), 直到 P 为空为止。

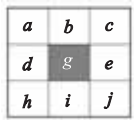


图3 邻接网格

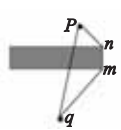


图4 障碍距离



图5 局部可视图

3 实验结果及算法分析

3.1 实验结果分析

不考虑空间实体约束下的情况, 实验对二维空间 1 300 个数据对象进行了聚类, 得到了七个簇, 如图 6 所示。

在考虑空间实体约束的情况下, 对障碍物和便利体进行网格化处理, 重复进行了实验, 如图 7 所示, 得到了九个簇, 算法对被障碍物阻挡的簇进行了重新划分。

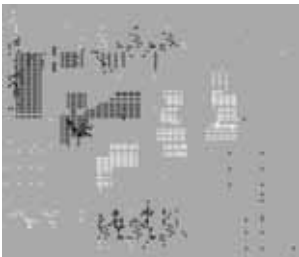


图6 不考虑空间实体约束

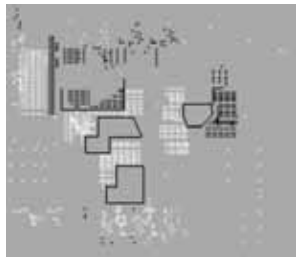


图7 考虑约束的情况

从实验数据中发现, 在第一步微簇产生后, 某些噪声数据 (即稀疏网格中的数据对象) 在第二步 r 障碍距离内微簇合并时, 处在某中心点的 r 障碍距离内, 实际上是被合并到较大的簇中, 而不再是噪声数据, 本算法避免了直接计算数据对象之间的障碍距离, 而是在粗粒度上求微簇的障碍距离, 大大减少了计算量, 特别适用于空间实体约束下数据相对集中的情况。

3.2 算法复杂度分析

a) 微簇产生算法的复杂度受空间数据元素数量、障碍物数量、便利体数量以及所占空间区域大小的影响。设数据规模为 n , 障碍物的数量为 m , 便利体的数量为 f , 二维空间区域大小为 $a \times b$, 网格步长参数为 w , 则划分后的网格数量为 $(a \times b / w^2)$ 。算法的时间复杂度: 预处理阶段包括网格划分、计算每个元素存在的网格、确定每个网格的密度以及障碍网格和便利网格的标记处理, 时间复杂度为 $O(n + m + f + a \times b / w^2)$; 遍历各个网格并进行标记, 每个网格只进行一次标记, 该过程的复杂度只与网格数目有关^[5], 即为 $O(a \times b / w^2)$; 最后对每个元素进行标记的时间复杂度等同于数据集元素的个数, 即为 $O(n)$ 。综合起来该步骤时间复杂度为 $O(n + m + f + a \times b / w^2)$ 。

b) r 障碍距离内微簇合并的时间复杂度受 a) 产生的微簇、障碍物和便利体数目的影响, 因为要在 r 半径内计算障碍距离。

(上接第 100 页)

参考文献:

- [1] DORIGO M, BIRATTARI M, STUZLE T. Ant colony optimization [J]. IEEE Computational Intelligence Magazine, 2006, 1(4): 28-39.
- [2] OKDEM S, KARABOĞA D. Routing in wireless sensor networks using ant colony optimization [C]//Proc of the 1st NASA/ESA Conference on Adaptive Hardware and Systems. 2006:401-404.
- [3] ZHANG Ying, KUHN L D, FROMHERZ M P J. Improvements on ant routing for sensor networks [C]//Proc of the 4th International Workshop on ant Colony Optimization and Swarm Intelligence. Ber-

lin: Springer, 2004:154-165.

设微簇数目为 c , 即有 c 个代表点, 障碍物数目为 m , 便利体数目为 f 。这里 c 远远小于 a) 中的网格数目 $(a \times b / w^2)$ 和数据规模 n , r 半径内局部可视图的建立代价远远小于 COD-CLARANS 和 AutoClust + 算法建立 Delaunay 图, 建立可视图之后求最短路径的时间复杂度为 $O(c \times m \times f)$ 。

4 结束语

约束条件下的空间聚类问题具有很强的使用价值, 也是近几年空间数据挖掘的热点。本文通过对便利体和障碍物约束下的研究, 在 CLIQUE 算法和文献[5]所提出算法的基础上, 引入障碍网格和便利网格的概念, 考虑了障碍物的阻碍以及便利体的连通作用, 同时对便利体和障碍物的网格化处理, 解决了任意形状约束的处理问题。实验证明 GBSCFO 算法处理效果良好, 时间复杂度较低, 在处理海量数据时较其他算法具有优势, 并能够方便地扩展到高维空间。GBSCFO 算法的不足之处在于: 需要输入参数; 当网格 g 是稠密网格且邻接网格中不存在稠密网格而 g 距离其他稠密网格又较远时, g 形成了很小的微簇, 在 r 半径内微簇合并的过程中, g 可能单独形成了一个簇, 当密度阈值 MinPts 取得太小时, g 形成的簇就没多大实际意义; 参数选取需要凭借经验, 聚类数据需要预处理。

参考文献:

- [1] HAN Jia-wei, KAMBER M. 数据挖掘概念与技术 [M]. 范明, 孟小峰, 译. 北京: 机械工业出版社, 2007.
- [2] 高飞, 樊明辉, 陈崇成, 等. 基于空间实体约束的空间聚类算法研究 [J]. 测绘科学, 2008, 33(1): 45-47.
- [3] 杨杨, 孙志伟, 赵政. 一种处理障碍约束的基于密度的空间聚类算法 [J]. 计算机应用, 2007, 27(3): 1689-1691.
- [4] ZHANG J, PAPANIAS D, MOURATIDIS K, et al. Spatial queries in the presence of obstacles [C]//Proc of International Conference on Extending Database Technology (EDBT). Berlin: Springer-Verlag, 2004:366-384.
- [5] 孙宇清, 赵锐, 姚青, 等. 一种基于网格的障碍约束下空间聚类算法 [J]. 山东大学学报: 工学版, 2006, 36(3): 87-90.
- [6] 演馨, 周丽华, 陈克平, 等. 一种改进的带有障碍的基于密度和网格的聚类算法 [J]. 计算机应用, 2005, 25(8): 1819-1820, 1823.
- [7] 陈克平, 周丽华, 王丽珍, 等. Dcell0-网格弥散聚类算法 [J]. 计算机研究与发展, 2004, 41(增刊): 205-212.
- [8] ZHANG J, PAPANIAS D, MOURATIDIS K, et al. Query processing in spatial databases containing obstacles [J]. International Journal of Geographic Information Science, 2005, 19(10): 1091-1111.
- [9] WANG X, ROSTOKER C, HAMILTON H J. Density-based spatial clustering in the presence of obstacles and facilitators [C]//Proc of the 8th European International Conference on Principles and Practice of Knowledge Discovery in Databases. Berlin: Springer-Verlag, 2004:446-458.

lin: Springer, 2004:154-165.

- [4] 苏森, 钱海, 王煦法. 基于蚁群的无线传感器网络双簇头算法 [J]. 计算机工程, 2008, 34(13): 174-176.
- [5] GAO Wei. Study on immunized ant colony optimization [C]//Proc of the 3rd International Conference on Natural Computation. 2007:759-763.
- [6] RAJENDRAN C, ZIEGLER H. Two ant-colony algorithms for minimizing total flowtime in permutation flowshops [J]. Computers & Industrial Engineering, 2005, 48(4): 789-797.
- [7] 耶刚强, 梁彦, 孙世, 等. 基于蚁群的无线传感器网络路由算法 [J]. 计算机应用研究, 2008, 25(3): 715-717.
- [8] 裴志利, 王建, 时小虎, 等. 带个体差异的蚁群算法的应用 [J]. 计算机应用研究, 2008, 25(4): 1036-1038.