

基于改进型蚁群算法的最优路径问题求解^①

张志协, 曹 阳

(华南师范大学 计算机学院, 广州 510631)

摘 要: 如何高效的向用户提供最优路径是蚁群算法大规模应用于导航系统的关键问题, 针对现有最优路径问题研究中蚁群算法收敛速度慢及容易发生停滞的缺点, 利用 A^* 算法的启发式信息改进蚁群算法的路径选择策略, 加快算法收敛速度. 同时引入遗传算法的双种群策略和蚁群系统信息素更新策略, 增加全局搜索能力, 避免算法出现停滞现象. 仿真实验结果表明, 该改进算法具有较好的稳定性和全局优化性, 且收敛速度较快.

关键词: 最优路径; 蚁群算法; A^* 算法; 双种群策略

Solving Optimal Path Problem Based on Improved Ant Colony Algorithm

ZHANG Zhi-Xie, CAO Yang

(School of Computer, South China Normal University, Guangzhou 510631, China)

Abstract: Efficient optimal path is the key issue of the Ant Colony Algorithm used in road traffic navigation system. Aiming at the problem of slow convergence and stagnation phenomenon of Ant Colony Algorithm, this paper introduce path selection strategy which is based on the heuristic factor of A^* algorithm to speed up the convergence. Meanwhile the Double-population strategy of the Genetic algorithm and the pheromone update rules of the Ant Colony System (ACS) are introduced in the algorithm, which avoid stagnation of the algorithm of the algorithm and speed up convergence. The results of the simulation experiment show that the improved algorithm has good stability, global optimization and fast convergence.

Key words: optimal path; Ant Colony Algorithm; A^* algorithm; Double-population strategy

最优路径问题是网络优化技术的核心, 自 20 世纪 50 年代开始就陆续有各个领域的学者对网络最优路径问题进行研究, 其中最经典的是基于 Bellman 优化原理的 Dijkstra 算法, Floyd 算法和 A^* 算法. 这些经典算法以静态的确定性网络为研究对象, 因计算复杂度较大通常仅适用于静态中小规模网络. 近年来, 为了适应动态网络问题发展的需要, 许多研究人员将智能仿生算法应用到最优路径问题上并取得了良好的效果, 如: 遗传算法、模拟退火算法、蚁群算法等^[1].

蚁群算法^[2]是继模拟退火算法、遗传算法和人工神经网络算法等启发式算法后的又一种应用于优化问题的算法. 该算法先后成功应用于调度问题、旅行商问题和网络路由问题, 实验结果表明蚁群算法具有较

强的鲁棒性和发现较好解的能力, 但与此同时蚁群算法也存在一些缺点, 如收敛速度慢、易出现停滞现象等^[1]. 针对蚁群算法的缺点, 学者们提出了一些改进方法, 其内容主要集中在路径选择、信息素局部更新和信息素全局更新三个方面. 其中 L.M.Gambardella, M.Dorigo^[3]提出了 Ant-Q 算法, 该算法用伪随机比例状态转移规则替换蚁群算法中的随机比例规则, 从而使 Ant-Q 算法在构造解的过程中能够更好的保持知识探索与利用之间的平衡. 除此之外, 该算法还引用了局部信息素更新机制和全局信息素更新中的精英策略. Stutzle 和 Hoss^[4]提出了最大-最小蚂蚁系统, 通过为信息素设置上下限来避免算法过早出现停滞现象. Bullnheimer 等^[5]提出了基于排序的蚂蚁系统, 该算法

① 基金项目: 广东省教育部产学研结合项目(2009B090300326); 广东省科技计划(2010A040306003)

收稿时间: 2012-02-10; 收到修改稿时间: 2012-03-12

在完成一次迭代之后, 根据路径的长度赋予不同的权重进行信息素更新。

在应用于两点间最优路径问题求解上, 刘伟^[6]采用双种群策略扩大蚁群算法的搜索范围, 提高算法的全局搜索能力降低停滞现象发生概率, 但算法的收敛速度较慢总体效率不高; 刘好斌等^[7]利用信息素对比度增强规则改进算法的路径选择策略, 在一定程度上加快算法的收敛速度, 但运行过程中容易陷入局部搜索, 导致停滞现象的发生; 胡耀民等^[8]在蚁群算法基础上重新设计了信息素更新规则, 引入信息素更新算法, 通过动态调整、增加最优路径上信息素的规则加速算法的收敛速度。本文在文献[8]的基础上, 根据蚁群算法的改进策略融合 A^* 算法、遗传算法的思想和蚁群系统的信息素更新策略, 提出一种改进型蚁群算法。该改进算法在使用 A^* 算法启发式信息为蚁群算法加入方向启发式因子, 启发蚂蚁向目标方向前进的同时, 利用遗传算法的双种群优势增强蚁群算法的全局搜索能力, 从而有效的提高蚁群算法求解最优路径问题的效率。

1 传统蚁群算法

蚁群算法是一种智能仿生算法, 于 1991 年由 Dorigo M 提出^[1]。它模拟了自然界中蚂蚁觅食的行为, 采取具有智能的蚂蚁, 通过个体间的信息交流与协作来寻找蚁穴到食物源的最优路径(最短路径)。其数学模型如下^[9]:

蚂蚁 $k(k = 1, 2, \dots, m)$ 在运动过程中, 根据各条路径上的信息素浓度决定其转移方向。这里用禁忌表 $tabu_k(k = 1, 2, \dots, m)$ 来记录蚂蚁 k 当前所走过的城市, 集合随着进化过程作动态调整。在搜索过程中, 蚂蚁根据各条路径上的信息浓度及路径的启发信息来计算状态转移概率。 $p_{ij}^k(t)$ 表示在 t 时刻蚂蚁 k 由城市 i 转移到城市 j 的状态转移概率:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in allowed_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & \text{若 } j \in allowed_k \\ 0, & \text{否则} \end{cases} \quad (1)$$

其中 $allowed_k = \{C - tabu_k\}$, 表示蚂蚁 k 下一步允许选择的城市; α 为信息素启发因子, 表示轨迹的相对重要性, 反映了蚂蚁在运动过程中所积累的信息在蚁群运动时所起的作用, 其值越大, 则该蚂蚁越倾向

于选择其他蚂蚁经过的路径, 蚂蚁之间的协作越强; β 为期望启发式因子, 表示能见度的相对重要性, 反映了蚂蚁在运动过程中启发信息在蚂蚁选择路径中的受重视程度, 其值越大, 则该状态转移概率越接近于贪心规则; $\eta_{ij}(t)$ 为能见度启发因子, 表示目标的能见度, 其表达式如下:

$$\eta_{ij}(t) = 1/d_{ij} \quad (2)$$

其中, d_{ij} 表示城市 i 与城市 j 之间的欧式距离。对蚂蚁 k 而言, d_{ij} 越小, 则 $\eta_{ij}(t)$ 越大, $p_{ij}^k(t)$ 越大。显然, 该启发式函数表示蚂蚁从城市 i 转移到城市 j 的期望程度。

为了避免残留信息素过多引起残留信息淹没启发式信息, 在每只蚂蚁走完一步或者完成对所有 n 个城市的遍历后, 要对残留信息进行更新处理。其中, $t+n$ 时刻在路径 (i, j) 上的信息素可按如下规则进行调整:

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (4)$$

其中, ρ 表示信息素挥发系数, 则 $1-\rho$ 表示信息素残留因子, 为了防止信息素的无限量积累, ρ 的取值范围为: $\rho \in [0, 1]$; $\Delta\tau_{ij}^k(t)$ 表示本次循环中路径 (i, j) 上的信息素增量, 初始时刻 $\Delta\tau_{ij}(0) = 0$, $\Delta\tau_{ij}^k(t)$ 表示第 k 只蚂蚁在本次循环中留在路径 (i, j) 上的信息量。

2 蚁群算法的改进策略

传统蚁群算法在实现过程中容易出现停滞和收敛速度慢的现象, 具体表现为: ①蚂蚁在行进过程中常常选择信息素浓度较大的路径, 当搜索进行到一定程度后, 致使所有个体搜索的解相同, 不能对解空间进一步搜索, 不利于发现更好的解; ②蚂蚁在行进过程中, 所有路径上的信息素浓度相差不大, 缺乏一定的启发式信息, 致使蚁群的解空间增大, 不能快速收敛。

在最优路径问题实际的应用过程中, 许多研究人员针对蚁群算法的停滞现象和收敛速度问题提出了许多改进方法, 如: 文献[6]利用双种群策略扩大蚁群算法的搜索范围, 降低停滞现象发生概率; 文献[7]的通过信息素对比增强规则提高算法的收敛速度。虽然这

些改进算法在一定程度上加快了算法的收敛速度或者减少了停滞现象发生的概率,但总的来说,减少停滞现象和加快收敛速度是该算法矛盾的两个方面.收敛速度的加快必然会增加出现停滞现象的概率;反之,减少停滞现象,提高全局搜索能力则会导致收敛速度下降.因此,如何在提高收敛速度和减少停滞现象之间选取一个平衡点就成为了算法改进的关键.本文根据蚁群算法的改进策略融合算法、遗传算法的思想和蚁群系统的信息素更新策略,提出一种改进型蚁群算法,从而有效的解决收敛速度与停滞现象的平衡问题,提高蚁群算法求解最优路径问题的效率.

2.1 路径选择策略

式(1)是蚁群算法的状态转移公式,其最重要部分是给每只蚂蚁添加启发式信息,这也是蚁群算法中最重要的参数之一.启发式信息的好坏影响着整个算法的优劣程度,好的启发式信息可以使蚁群算法很快的收敛于最优解,反之亦然.

在蚁群算法模型中,能见度启发因子 $\eta_{ij}(t)$,表示目标点的能见度,其值为当前节点到下一个节点距离的倒数,即 $1/d_{ij}$.从 $\eta_{ij}(t)$ 的取值可以看出对蚂蚁 k 而言, d_{ij} 越小,则 $\eta_{ij}(t)$ 越大, $p_{ij}^k(t)$ 越大,其在算法运行过程中会趋于选择单步距离较小的节点.然而,针对两点间最优路径问题,全局的最优并不是简单的单步最小可以确定的,片面的选择单步最小只会增加算法的停滞现象.因此为了提高算法的效率,本文引入传统最优路径问题中 A^* 算法的启发式策略改进蚁群算法的启发式因子,使蚁群在搜索过程中选择最优路径概率较大的关键节点,增强蚁群算法的目标性和方向性.具体规则如下:

$$\eta_{ij}(t) = 1/\tilde{f}(i) = 1/(\tilde{g}(i) + \tilde{h}(i)) \quad (5)$$

$\tilde{f}(i)$: 起始节点到目标节点的最优路径代价的估计值;

$\tilde{g}(i)$: 深度因子,启发式搜索算法发现的从起始节点到当前节点 i 的最优路径代价;

$\tilde{h}(i)$: 启发式因子,当前节点 i 到目标节点的最优路径代价.

具体描述如下: S 为源节点, D 为目标节点, i 为当前节点, j 为下一个待选节点,源节点到当前节点的路径花费为 E ,当前节点 i 到待选节点 j 的欧氏距离为 d_{ij} ,与之对应的单位距离时间花费为 k_1 ,待选

节点 j 到目标点 D 的欧氏距离为 d_{jD} ,与之对应的单位距离时间花费为 k_2 ,则经过待选节点 j 到达目标点的估计值为

$$\tilde{f}(i) = \tilde{g}(i) + \tilde{h}(i) = E + k_1 d_{ij} + k_2 d_{jD} \quad (6)$$

$$\eta_{ij} = 1/(E + k_1 d_{ij} + k_2 d_{jD}) \quad (7)$$

2.2 信息素更新策略

基本蚁群算法模型加入方向启发式信息,在提高算法收敛速度的同时,也加剧了蚁群算法趋向于局部最优产生停滞现象的概率,因此本文在信息素更新策略上加以一定的平衡.其策略是算法在进行全局信息素更新时,采用优化排序蚁群系统和蚁群系统相结合的策略,仅对前 k 则最优路径集进行相应的信息素更新.同时在局部信息素更新上,为了扩大搜索范围,本文引入蚁群系统的局部更新策略,即当每一只蚂蚁走完边 (i, j) 时,都要在边 (i, j) 进行信息素的更新,而这种更新不是对信息素的增加而是相应的减少.通过减弱已选择道路的信息素浓度来减弱该道路对后来的蚂蚁选择的影响,从而使蚂蚁对没有选中的路径具有更强的探索能力,其具体方案如下:

全局信息素的更新:算法中信息素的更新动作只是发生在当前的 K 则最优路径中,那么信息素更新规则如下面的式子给出:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \Delta\tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}^{bs}(t) \quad (8)$$

$$\Delta\tau_{ij}^{bs}(t) = 1/L^{bs} \quad (9)$$

其中,对于全局信息素的更新动作或者挥发动作只是发生在最优路径上.

局部信息素的更新:在蚂蚁巡游过程中,每经过一条边 (i, j) ,都将立即调用局部信息素更新规则更新该边上的信息素:

$$\tau_{ij} \leftarrow (1-\xi) \cdot \tau_{ij} + \xi\tau_0 \quad (10)$$

其中, τ_0 为常数是信息素的初始值, $\xi \in (0,1)$ 为可调参数.

2.3 双种群策略

为了更好地兼顾收敛速度和全局搜索能力,许多研究人员通过使用自适应策略改进蚁群算法.通过在算法运行中动态的改变 α , β , ρ , Q 的值,以实现在运行的不同阶段对算法的控制.在实践中自适应的方法的确提高了算法的寻优能力,但解的总体质量不

稳定,主要表现为算法找到最优解所需的迭代次数难以确定,多次试验的结果偏差较大.为了既能利用自适应方法的优点,又避免自适应可能产生的不良影响,本文引入了双种群策略,利用两个种群设置的不同参数获得自适应算法的效果,即其中一个种群的参数设置趋向于快速收敛能力,另一个种群的参数设置趋向于全局搜索能力,使得在算法运行的不同阶段都有一个种群起主要作用,两个种群交替协作从而获得良好的性能.

蚁群算法的搜索是以信息素的正反馈为基础,本文在算法的实际运行过程中通过交换两个种群间的信息素分布表来实现双种群策略的,具体规则如下:

$$\tau_{ij}^A(t+1) = k\tau_{ij}^A(t) + (1-k)\tau_{ij}^B(t) \quad (11)$$

$$\tau_{ij}^B(t+1) = \tau_{ij}^A(t+1) \quad (12)$$

其中

$$k = \text{best_A} / (\text{best_A} + \text{best_B}) \quad (13)$$

其中 $\tau_{ij}^A(t)$ 为种群 A 在 t 时刻路径 (i, j) 上的信息素浓度, $\tau_{ij}^B(t)$ 为种群 B 在 t 时刻路径 (i, j) 上的信息素浓度, best_A 为种群 A 的最优路径时间, best_B 为种群 B 的最优路径时间, k 为种群间的权重因子,其取值范围为: $k \in (0,1)$.

2.4 算法描述

改进算法的具体步骤如下:

Step 1: 参数初始化. 令时间 $t = 0$ 和循环次数 $N_c = 0$, 设置最大循环次数 N_c . 令每条道路 (i, j) 的初始化信息量为 $\tau_{ij}(t) = \text{cons}$ 且初始时刻 $\Delta\tau_{ij}(t) = 0$, 初始化启发因子表为 $\eta_{ij} = 1/(E + k_1 d_{ij} + k_2 d_{jd})$.

Step 2: 循环次数 $N_c \leftarrow N_c + 1$.

Step3: 每只蚂蚁根据状态转移概率公式选择下一顶点 j , $j \in \{C - \text{tabu}_k\}$, 同时按照局部信息素更新规则(2-6)对边 (i, j) 进行信息素更新.

Step 4: 若所有蚂蚁的当前解集包含了终点或死亡, 转到步骤 5, 否则跳转步骤 3.

Step 5: 计算两个种群中每只蚂蚁所走的时间, 分别记录耗时最短的路径 route_A 和 route_B .

Step 6: 根据式(2-4)分别对种群 A、B 的当前耗时最优的 k 则路径按照进行全局信息素更新.

Step 7: 若 $N_c \% 10 = 0$, 则根据式(2-7)和(2-8)进行种群间通信, 否则转到 8.

Step 8: 蚁群收敛于同一条路径则算法结束.

Step 9: 若循环次数 $N_c \geq N_{c\max}$, 则算法结束, 否则转到步骤 2.

3 仿真实验

3.1 实验 1

为了验证改进算法中 A^* 算法启发式因子和双种群策略的收敛性和寻优速度, 本文以凯立德科技有限公司的广州市 1:50000 交通道路网络有向图为例, 其中包含 15944 节点、30502 边, 分别将本文改进算法与蚁群优化算法(ACO)、文献[8]的改进算法进行对比. 其中本文改进蚁群算法的参数设置依照参照文献[1]的分析和自身的选择, 设置为: 启发式因子 $\alpha = 1$, 期望启发式因子 $\beta = 3$, 信息素残留因子 $\rho = 0.7$. 实验通过随机在广州市道路网络中选取多组不同规模的源、目的节点进行测试分析. 其中 4 组结果如下表 1 所示:

表 1 本文改进算法与其他算法对比

算例	算法	最优路径节点数	平均迭代次数
(1)	ACO	102	264
	文献[8]	102	198
	本文算法	102	216
(2)	ACO	236	547
	文献[8]	236	425
	本文算法	236	338
(3)	ACO	531	894
	文献[8]	531	677
	本文算法	531	523
(4)	ACO	1037	2676
	文献[8]	1037	1885
	本文算法	1037	1023

3.2 实验 2

为了验证本文算法在实时交通环境中的优势, 利用现有的广州市交通道路网络数据加入一定的权值变化频率和权值变化系数, 用程序随机修改道路网中部分路径的权值模拟实时交通情况的变化, 并在此基础上对比路径权值后本文改进算法和改进的 Dijkstra 算法的求解最优路径平均求解时间. 实验通过随机选取 4 组包含不同规模节点数的最优路径, 其结果如下表所示:

表 2 本文改进算法与改进 Dijkstra 算法对比

算例	算法	最优路径节点数	平均求解时间
(1)	本文算法	50	94ms
	改进Dijkstra	50	118ms
(2)	本文算法	128	178ms

	改进Dijkstra	128	267ms
(3)	本文算法	277	423ms
	改进Dijkstra	277	648ms
(4)	本文算法	583	933ms
	改进Dijkstra	583	1479ms

由表 3 可见当交通网络路径权值发生变化时, 本文的改进算法较 Dijkstra 算法的搜索速度更快, 其原因蚁群算法利用算法在前一阶段的搜索信息, 路网信息变化后, 算法根据路径权值变化的程度做相应的信息素对比更新操作, 致使蚁群可以利用原有的搜索信息, 避免再次重新搜索, 从而使搜索性能优于 Dijkstra 算法。

仿真实验表明: 改进的蚁群算法在全局搜索能力和收敛速度两方面都显示出了良好的性能, 特别是当问题的规模较大时, 由于本文改进算法在运行开始时就加入了方向启发式信息, 加速了目标解空间的构造, 收敛速度明显提升。同时双种群策略和信息素更新策略的改进增加了算法全局搜索能力, 使算法在最优路径问题的求解上优于本文的其他算法。

4 结语

本文给出的算法信息素启发式因子, 使蚂蚁在搜索过程中具有了较好的目标性, 在加快了算法的收敛速度。同时通过构建两个种群利用种群间信息素分布的相互影响和蚁群系统的信息素更新策略较好的平衡了加快收敛速度带来的停滞现象的矛盾。仿真实验结果表明, 在最优路径涉及节点较多, 规模较大的情况下本文的改进算法较 ACO 算法和文献[8]的算法效率更优。

参考文献

- 1 杨剑峰. 蚁群算法及其应用研究. 杭州: 浙江大学, 2007.
- 2 Dorigo M, Maniezzo V, Colomi A. The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Trans. on Systems, Man, and Cybernetics: Part B, 1996, 26(1): 29–41.
- 3 Gambardella LM, Dorigo M. Ant-Q: A reinforcement learning approach to the traveling salesman problem. In: Frieditis A, Russell S. ed. Proc. of the Twelfth International Conference on Machine Learning. Morgan Kaufmann Publishers, Palo Alto, CA, 1995. 252–260.
- 4 Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. on Evolutionary Computation, 1997, 1(1): 53–66.
- 5 Stutzle T. MAX-MIN Ant System for Quadratic Assignment Problems. Technical Report AIDA-97, Intellectics Group, Department of Computer Science. Darmstadt University of Technology, Germany, 1997.
- 6 刘伟. 基于蚁群算法的动态路径研究. 成都: 西南交通大学, 2005.
- 7 刘好斌, 胡小兵, 赵吉东. 动态调整路径选择的蚁群优化算法. 计算机工程, 2010, 36(17): 201–203.
- 8 胡耀民, 刘伟铭. 基于改进型蚁群算法的最优路径问题求解. 华南理工大学学报, 2010, 38(10): 105–109.
- 9 段海滨. 蚁群算法原理及其应用. 北京: 科学出版社, 2005.

(上接第 56 页)

- 2 van Lamsweerde A. Goal-Oriented Requirements Engineering: A Guided Tour. Proc. of the 5th IEEE International Symposium on Requirement Engineering. 2001. 249–260.
- 3 Yue K. What Does It Mean to Say that a Specification is Complete. Proc. of the 4th IEEE International Workshop on Software Specification and Design. 1987. 1–14.
- 4 Robinson WN. Integrating multiple specifications using domain goals. Proc. of the 5th IEEE International Workshop on Software Specification and Design. 1989. 219–255.
- 5 米歇尔. 贝克特尔. 目标管理. 沈阳: 辽宁教育出版社, 2002.

- 6 Giorgini P, Mylopoulos J, Nicchiarelli E, Sebastiani R. Reasoning with goal models. Technical Report of University of Trento, 2002. 56–70.
- 7 Chung L, Nixon BA, Yu E, Mylopoulos J. Non-functional Requirements in Software Engineering: Kluwer Academic Publisher, 2000.
- 8 CEDITI. A KAOS Tutorial, 2003.
- 9 段新生. 人工智能—证据理论与决策. 北京: 中国人民大学出版社, 1990.