

信息素增量动态更新的改进蚁群算法

贾瑞玉, 张新建, 冯伦阔, 李永顺

(安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039;

安徽大学 计算机科学与技术学院, 安徽 合肥 230039)

摘 要: 蚁群算法是一种优秀的拟生态启发式算法, 具有较强的鲁棒性, 易与其它拟生态算法结合等特点。不过, 它也存在容易陷入局部最优、收敛速度慢等问题。通过分析蚁群算法的运行机制, 得出了蚁群算法易陷入局部最优、收敛速度慢等不足产生的原因, 针对这些不足, 通过修改基本蚁群算法中信息素的更新规则, 使得每轮搜索后信息素的增量能更好地反映解的质量, 以加快收敛; 同时引进了路径信息素平滑机制以平衡路径上的信息量。对 TSP 问题的仿真实验结果表明, 改进后的算法加快了收敛速度, 提高了全局搜索能力。

关键词: 蚁群算法; 信息素更新规则; 全局收敛

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2009)09-0032-03

Ant Colony Algorithm with Dynamic Pheromones Increment Updating

JIA Rui-yu, ZHANG Xin-jian, FENG Lun-kuo, LI Yong-shun

(Ministry of Education Key Laboratory of Intelligent Computing & Signal Processing,

Anhui University, Hefei 230039, China;

School of Computer Science and Technology of Anhui University, Hefei 230039, China)

Abstract: Basic ant colony is an excellent meta-heuristic algorithm and has strong robustness, easy to mix together with other meta-heuristic algorithm, but it also has slow convergence and easily be trapped in a local optimum. Could get the reasons that the ant colony has slow convergence and easily be trapped in a local optimum from analyzing the principle of the algorithm and aiming at these disadvantages, present a new solution that modifies the rule of updating pheromones in ACA, so that after every round of search, the increment of pheromones can better reflect the quality of a solution to quicken the convergence; also use the solution that pheromones smoothing to balance the pheromones on the route. Experimental results for solving TSP problems indicate that the improved algorithm increases the speed of convergence and enhances the ability of searching the whole best solution.

Key words: ant colony algorithm; rule of updating pheromones; global convergence

0 引言

蚁群算法是由意大利学者 Dorigo. M^[1]等首先提出的一种新的启发式优化算法, 它通过采用正反馈原理, 加快了进化进程, 同时它本质上是一种并行算法, 个体之间不断进行信息交流和传递, 从而能够相互合作, 有利于发现较好解。蚁群算法是目前国内外启发式算法研究的热点和前沿课题, 它已经成功地被应用于车间作业调度问题、控制参数优化问题、气象图像处理问题、圆排列问题^[2]等实际应用领域, 同时蚁群算法也可应用于聚类问题研究^[3], 并且已经取得了很好的

效果。自 Dorigo. M 提出蚁群算法后, 不断有学者对其完善, 提出了许多改进算法, 如 Gambardella 等学者结合 Q-Leaving 提出的 Ant-Q 算法^[4], Stutzle 等学者提出的 MMAS 算法^[5], 黄国锐等提出的基于信息素扩散模型的 PDACO 算法^[6]以及带非均匀小窗口的蚁群算法^[7]和免疫算法的融合^[8]等。蚁群算法和其它进化算法同样存在易于陷入局部最优值^[9]的缺点, 针对蚁群算法的这个缺点, 通过改进信息素的更新规则对算法作出了改进。

1 基本蚁群算法原理

蚁群算法是最先用于求解 TSP 问题, 下面以 TSP 为例来说明蚁群算法^[9]。设有 n 个城市组成的集合 C , m 只蚂蚁, 用 $d_{ij}(i, j = 1, 2, 3, \dots, n)$ 表示城市 i 和城市 j 之间的距离, $\tau_{ij}(t)$ 表示在时刻 t 城市 i 和城市 j

收稿日期: 2009-01-13; 修回日期: 2009-03-17

基金项目: 安徽省自然科学基金项目(kj2008B092)

作者简介: 贾瑞玉(1965-), 女, 副教授, 研究方向为计算机图形学、数据挖掘、人工智能。

之间的路径上的残留信息素强度,以此来模拟实际蚂蚁之间用于进行通信的特殊分泌物。蚂蚁 $k(k = 1, 2, 3, \dots, m)$ 在运动过程中,根据各条路径上的信息量决定其转移方向,同时用禁忌表 $\text{tabu}_k(k = 1, 2, 3, \dots, m)$ 来记录蚂蚁 k 当前所走过的城市,集合随着 tabu_k 进化过程作出动态调整。在搜索过程中,蚂蚁根据各条路径上的信息量及路径上启发信息素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 和城市 j 的状态转移概率,其表达式为

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta}{\sum_{i \in \text{allowed}_k} [\tau_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta}, & \text{如果 } j \in \text{allowed}_k \\ 0, & \text{否则} \end{cases} \quad (1)$$

式中: $\text{allowed}_k = \{C - \text{tabu}_k\}$ 表示蚂蚁 k 下一步允许选择的城市; α 为信息启发式因子, β 为期望启发式因子, $\eta_{ij}(t)$ 为启发函数,是路径长度的倒数,表达式为 $\eta_{ij}(t) = 1/d_{ij}$ 。为了避免残留信息素过多引起残留信息淹没启发信息,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。其表达式为

$$\tau_{ij}(t+n) = (1-\rho) \times \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (3)$$

式中: $\Delta\tau_{ij}^k(t)$ 表示经过了城市 i 和城市 j 之间路径的蚂蚁 k 在这条路径上留下的信息素增量,它的表达式为

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k, & \text{如果蚂蚁 } k \text{ 经过了路径}(i,j) \\ 0, & \text{否则} \end{cases} \quad (4)$$

Q 为信息素总量, L_k 为蚂蚁 k 在本次循环中所走的总长度。

2 基本蚁群算法的不足

基本蚁群算法虽然对解决 TSP 问题有着很好的效果,但是它也存在着收敛速度慢,易陷入局部最优值等问题^[9]。之所以会有这些问题是因为在算法开始阶段,每条路径上的起始信息素是一样的,这时蚂蚁创建第一条路径的引导信息主要是城市间的距离信息,这样蚂蚁在所经过的路径上留下的信息就不一定能反映出最优路径的方向,不能保证蚂蚁创建的第一条路径能引导蚁群走向全局最优解。因此,在第一次循环,蚂蚁创建的本次循环最优解可能离全局最优解较远,但这条路径的信息素会因正反馈而得到增强,随着算法的重复执行,信息素会积累在这条不是最优的路径上,而其它原本是比较优秀的路径,因为在起始阶段没有

被蚂蚁选中,导致随着算法的执行,经过它的蚂蚁越来越少,导致它的信息素越来越低,直到被认为是比较差的路径而被蚂蚁所放弃,这样就会导致算法产生收敛于局部最优解,而无法找到真正的全局最优解。并且,在基本蚁群算法中,每次循环结束之后,所有路径上的信息素都会进行更新,而这种更新方法并没有区分出优秀的路径和比较差的路径对算法寻找最优路径贡献程度的不同。优秀路径上信息素量的增加会促进算法更快更好的找到最优路径,而比较差路径上的信息素量的增加则会影响算法的收敛速度甚至使得算法收敛于局部最优值。

3 对基本蚁群算法的改进

蚁群算法中选择下一个城市节点用的是轮盘赌方法,它带有一定的不可预知性,所以算法收敛于局部最优值在很大程度上是不可避免的,因此文中要做的并不是完全避免算法于局部最优值处收敛,而是当算法收敛于局部最优值时如何使算法跳出来,避免停滞过早的产生。通过分析可以知道算法之所以会收敛于局部最优值,是因为一些非全局最优路径上面的信息素大大地高于其它路径包括真正的全局最优路径,当这种情况出现时,算法过早停滞即收敛于局部最优值就无可避免地发生了。对此,文中做出的改进是:建立路径上的信息素量的平滑机制。即当算法连续发生 N 代最优值没有变化,就可以认为算法已经陷入了停滞,这就意味着一些路径上面的信息素量已经远远大于其它上面的信息素量,这时靠算法自身的调节已经无法跳出停滞,这时通过强制性地削减那些信息素量过高的路径上的信息素量,同时适当提高那些信息素量偏低的路径上的信息素量,通过适当地平衡各条路径上的信息素含量,使得算法可以跳出停滞,即

$$\tau_{ij}(t+1) = \tau_{ij}(t) \times \delta \quad (5)$$

其中 δ 是可变的,当削减信息素量时, $\delta < 1$, 增加信息素量时, $\delta > 1$, δ 和判定信息素过大或过小的具体阈值可以根据 TSP 问题的具体情况来确定。

在基本蚁群算法中,所有路径的更新方法都是一样的,没有体现出优秀路径与包括较差路径在内的其它路径在信息素量增加上的区别,尤其是当路径长度较长时,每次迭代过后各条路径上的信息素量的增量的差别并不明显,因此各条路径上的信息素量在一定的时间内并没有明显的差距,这会降低信息素对蚂蚁寻优的引导作用,导致蚂蚁对下个节点的选择主要受到路径长度的影响,这种情况下,蚂蚁移动的盲目性大大增加,这在很大程度上降低了算法的收敛速度,同时各条路径上信息素量差别的不明显,使得优秀路径很

难脱颖而出,而较差的路径却很容易干扰蚂蚁的寻优,导致算法收敛于局部最优,过早地陷入停滞。

为了改进这个不足,通过改进公式(4),在短时间内通过信息素的增量上的差别来增大优秀路径和其它路径之间的信息素量的差距,引导算法收敛到最优路径,同时加快算法的收敛速度。因此根据路径的优秀程度的不同,将式(4)改为

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{2L_k - \text{BestSolution}}, & \text{如果蚂蚁 } k \text{ 经过路径}(i, j) \\ 0, & \text{否则} \end{cases} \quad (6)$$

其中, BestSolution 为从开始至当前代所找出的最优路径值, L_k 为当前代中蚂蚁 k 所找出的路径值。

在式(4)中将原先的分母 L_k 换为 $2L_k - \text{BestSolution}$, 新的信息素更新规则可以快速增加优秀路径上的信息素量,而对非优秀路径上的信息素增加则不明显,每次迭代后,每条路径都会根据本身值的优劣来获得相应的信息素增量,这样,越优秀的路径每次迭代后都会获得比较多的信息素增量,而比较差的路径的信息素增量则相应的较少,多次迭代后优秀路径和较差路径上的信息素量就会有会一定差别,这有利于排除较差路径的干扰,大大加快了算法的寻优速度,有利于算法快速收敛到最优值。同时,采用 MMAS 算法的思想,对各条路径上信息素量的最大值和最小值作出限制,再结合路径上信息素的平滑方法,有效地避免了算法收敛于局部最优值。

改进蚁群算法的步骤为:

(1) 参数初始化,令时间 $t = 0$, 循环次数 $nc = 0$, 设定最大循环次数 N_{\max} , 设定自适应循环次数 N , 信息素 Q , 各路径上信息素的起始值 $\tau_{ij}(0) = \text{const}$, 蚂蚁数 m 的初始值,并将 m 只蚂蚁随机放到 n 个城市中。

(2) 循环次数 $nc = nc + 1$;

(3) 蚂蚁的禁忌表号 $k = 1$;

(4) 蚂蚁数目 $k = k + 1$;

(5) 蚂蚁个体根据状态转移式(1) 计算的概率选择城市 j 并前进, $j \in \{C - \text{tabu}_k\}$;

(6) 修改禁忌表指针,将蚂蚁移动到新的城市;

(7) 若 $k < m$, 则跳转到(4) 继续执行, 否则执行(8);

(8) 根据式(6) 更新每条路径上的信息量;

(9) 比较当前最优解在经过 N 代后有没有变化, 如果没有变化, 则对路径上的信息素 $\tau_{ij}(t)$ 执行式(5)。

(10) 若满足结束条件, 即循环次 $nc > N_{\max}$, 则

循环结束并输出程序计算结果, 否则清空禁忌表并跳转到(2)。

4 仿真实验结果

为了验证改进算法的有效性, 通过对 TSPLIB 中的 Oliver30、20cities 和 Eil51 问题进行仿真研究, 参数设置为 $\alpha = 1, \beta = 5, \rho = 0.5, Q = 500, N_{\max} = 400, N = 100, m = 100$ 。平均运行 100 次作为仿真结果, 并将文中算法和基本蚁群算法 AC 和蚁群系统算法 ACS 的实验结果进行了比较。仿真实验比较结果见表 1 和表 2(其它算法数据来源于文献[10])。

表 1 算法仿真对比结果

TSP	基本蚁群算法(AS)		蚁群算法算法(ACS)		文中改进的算法	
	最优值	平均值	最优值	平均值	最优值	平均值
20cities	25.31	26.34	24.54	24.96	24.54	24.91
Oliver30	425.74	431.9	424.46	426.81	423.74	424.67
Eil51	445.58	459.12	441.9	445.71	428.982	437.74

表 2 ACS 算法与文中蚁群算法对比详单

TSP	算法名称	最短路径长度	平均路径长度	最差路径长度	平均迭代次数
20cities	ACS 算法	24.54	24.96	26.47	349
	文中算法	24.54	24.91	25.31	190
Oliver30	ACS 算法	424.46	426.81	434.63	1259
	改进算法	423.74	424.67	425.14	275
Eil51	ACS 算法	431.72	445.71	458.5	1585
	文中算法	428.982	437.74	447.15	274

5 结束语

针对基本蚁群算法的一些不足之处进行了改进, 从实验结果来看, 较好地解决了基本蚁群算法在解决 TSP 问题是易陷入局部最优值的问题, 提高了算法的收敛速度, 增强了算法的寻优能力。蚁群算法是一种全新的分布式智能模拟算法。它从提出到现在只有十多年时间, 但其诸多的优点, 如鲁棒性、正反馈等优点已经使它显示出了巨大的潜力和发展前景。

参考文献:

- [1] Dorigo M, Maniezzo V, Colomi A. The ant System: Optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems Man and Cybernetics, 1996, 26(1): 29-41.
- [2] 章义刚, 贾瑞玉, 张燕平, 等. 快速蚁群算法求解圆排列问题[J]. 计算机技术与发展, 2007, 17(8): 48-50.
- [3] 刘念涛, 刘希玉. 基于改进的启发式蚁群算法的聚类问题的研究[J]. 计算机技术与发展, 2007, 17(8): 37-39.
- [4] Gambardella L M, Dorigo M. Ant - Q: a reinforcement learning approach to the traveling salesman problem[C]// Proceedings of the 12th International Conference on Machine Learning. Palo Alto, CA: Morgan Kaufman, 1995: 252-260.

(下转第 37 页)

参数 α 是 0~1 之间的系数因子,它的入度在后面介绍。

link 用来计算链入到网页 P_j 的链接数量,公式:

$$\text{link}(p_j) = \beta \sum_{p_k \in s_i} \frac{\text{OIL}(p_j)}{\text{OIL}(p_k)} + (1 - \beta) \sum_{p_k \in s_i} \frac{\text{IIL}(p_j)}{\text{IIL}(p_k)} \quad (4)$$

其中 β 是给予链出和链入不同权值的系数因子, $\text{IIL}(P_j)$ 是网页 P_j 的内部超链接数量, $\text{OIL}(P_j)$ 是网页 P_j 的外部链接数量。

3.3.2 用 DHC 计算网页入度

基于以上分层权值结构,一个网页在超级结点中的入度可以用 DHC^[9] 算法从根网页到底层网页进行计算。每个网页 P_j 有一个权值 w_{ij} , 它表示在超级结点中网页 P_j 的入度。

$$w_{ij} = \prod_{p_k \in \{\text{nodes from } n_j \text{ to root}\}} \gamma \times \omega_k \quad (5)$$

其中参数 γ 是分散热量系数因子。

最后,一个网页 P_j 在整个网络图中的入度定义为 PI_j , 计算方程如下:

$$PI_j = SI_i \times w_{ij} \quad (6)$$

其中网页 P_j 属于超级结点 S_i , SI_i 是超级结点 S_i 的入度。根网页的入度等于超级结点的入度。

4 实验

选择爬虫收集的 50 000 张网页集作为搜索的数据库。在实验中选择了 5 个不同领域的关键词进行搜索,对每个关键词检索出的结果选取前 50 条记录,每隔 10 条记录进行一次查准率分析,然后将搜索结果和链状解析算法和网状解析算法进行查准率的对比,如图 4 所示。

通过对比,不难发现分层排序算法,比已知的链状解析算法和网状解析算法查准率更高。

5 结束语

在 PageRank, BlockRank 等知名算法的基础上,给出了基于网状结构的分层排序算法。考虑到网络既有分层结构又有链状结构,提出了分层自由遍历模型,这种模型能够模拟用户进行网上冲浪的行为。基于该模

型,提出了一个分层排名算法来计算网页的入度。该排名算法可以显著提高网络搜索的效率,并且对于新出现的网页能够分配合理的排名。在今后的工作中,将进行试验的大型网站收集来评价我们的算法。

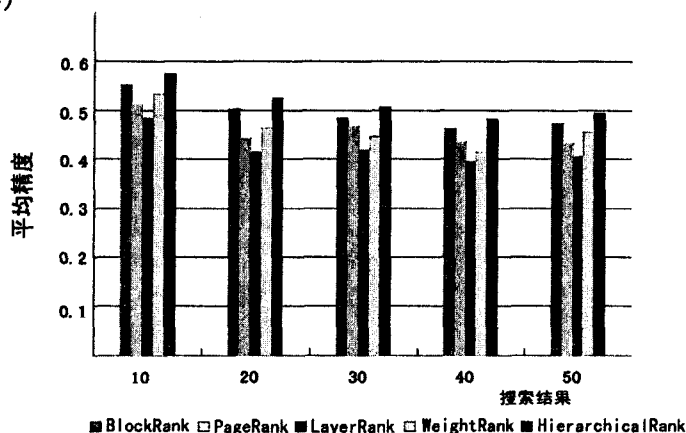


图 4 排序结果对比

参考文献:

- [1] 何晓阳, 吴强, 吴治蓉. HITS 算法与 PageRank 算法比较分析[J]. 情报杂志, 2004, 17(3): 77-80.
- [2] 吴淑燕, 许涛. PageRank 算法的原理简介[J]. 图书情报工作, 2003, 26(2): 364-367.
- [3] 白似雪, 刘华斌. 基于页面分块模型的 PageRank 算法研究[J]. 南昌大学学报: 工科版, 2008, 12(1): 34-36.
- [4] Wu J, Aberer K. Using a Layered Markov Model for Decentralized Web Ranking[R]. [s.l.]: ACM Press, 2004.
- [5] 李绍华, 高文字. 基于层次分类的页面排序算法[J]. 计算机工程, 2007, 19(2): 56-60.
- [6] 钱功伟, 倪林, 曹荣. 基于网页链接和内容分析的改进 PageRank 算法[J]. 计算机工程与应用, 2007, 43(3): 71-74.
- [7] Laura L, Leonardi S, Caldarelli G, et al. A Multi-Layer Model for the Web Graph[C]// In 2nd International Workshop on Web Dynamics. Honolulu: [s.n.], 2002.
- [8] Eiron N, McCurley K S, Tomlin J A. Ranking the Web Frontier[C]// In Proceedings of the 13th International World Wide Web Conference. [s.l.]: ACM Press, 2004: 309-318.
- [9] Zhou D, Weston J, Gretton A, et al. Ranking on Data Manifolds[M]. Cambridge, MA: MIT Press, 2004.

(上接第 34 页)

- [5] Stutzle T, Hoos H H. Max-Min ant system[J]. Future Generation Computer System, 2000, 16(19): 889-914.
- [6] 黄国锐, 曹先彬, 王照法. 基于信息素扩散的蚁群算法[J]. 电子学报, 2004, 32(5): 865-868.
- [7] 张然, 贾瑞玉, 钱光超, 等. 带佳点交叉算子的非均匀窗口蚁群算法[J]. 计算机技术与发展, 2007, 17(12): 68-70.
- [8] 张亮, 孙力娟. 蚁群算法和免疫算法的融合及其应用[J]. 计算机技术与发展, 2006, 16(3): 31-33.
- [9] 段海滨. 蚁群算法原理及其应用[J]. 北京: 科学出版社, 2005.
- [10] 郑松, 侯迪波, 周泽魁. 动态调整选择策略的改进蚁群算法[J]. 控制与决策, 2008, 23(2): 225-228.