

基于动态调节信息素增量的蚁群算法*

刘瑞杰, 胡小兵
(重庆大学 数学与统计学院, 重庆 401331)

摘 要: 为了避免蚁群算法陷入停滞状态,研究了信息素的更新规则,并在信息素增量更新式中加入动态调节因子,使得次优路径上的信息素增量较大,其他路径则没有明显的变化,从而有利于蚂蚁在较短的时间内找到更好的解。仿真实验结果及收敛过程表明,改进后的算法解决旅行商问题具有更好的全局搜索能力。
关键词: 蚁群算法; 动态调节因子; 信息素; 信息素增量; 旅行商问题
中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2012)01-0135-02
doi:10.3969/j.issn.1001-3695.2012.01.037

Ant colony algorithm based on dynamic adjustment of incremental of pheromone

LIU Rui-jie, HU Xiao-bing
(School of Mathematics & Statistics, Chongqing University, Chongqing 401331, China)

Abstract: This paper studied the updating rule of pheromone and employed a new regulatory factor in this paper to avoid the stagnation, so the pheromone on the routes close to the shortest increased, others didn't change obviously, which encouraged ants to find better solution in less time. Simulation results and convergence process show that the improved ant colony system has better performance in solving the traveling salesman problems (TSP).
Key words: ant colony algorithm(ACA); dynamic adjustment factor; pheromone; incremental of pheromone; TSP

0 引言

蚁群算法是一种源于自然界中蚂蚁觅食行为的一种启发式算法,该算法自意大利学者 Dorigo 等人^[1]提出后,已经成功地解决了规划^[2]、调度^[3]等多个问题,发展成为一种可与遗传算法相媲美的仿生优化算法。

尽管单个的蚂蚁行为极其简单,但整个蚂蚁群体的能力,尤其是搜索食物源的能力却是非常强大的,这种能力来源于蚂蚁群体中很好的个体协作行为。自然界中的蚂蚁会分泌一种化学刺激物——信息素,由于蚁群算法中正反馈的存在,使得蚂蚁都倾向于选择信息素较多的路径,因此在倾向于找到最优路径的同时也产生了副作用,即易陷入局部最优。基于此,许多学者对其进行了改进^[4-8]。本文针对蚂蚁易陷入停滞状态的缺陷,适时地调节信息素更新公式,并在信息素增量的更新式中加入动态调节因子,使得接近最优解的路径上的信息素明显增多,而和最优解相差较大的路径上的信息素没什么大的变化,突出了较短路径的优势。

1 基本蚁群算法数学模型及基本原理^[9]

蚂蚁 $k(k=1,2,\cdots,m)$ 在运动过程中,根据路径上信息素 τ 和能见度 η 独立地选择下一个城市。在 t 时刻位于城市 i 的蚂蚁 k 根据伪随机比例规则选择城市 j 作为下一个要访问的城市。该规则由式(1)给出:

$$j = \begin{cases} \arg \max_{l \in \text{allowed}_k} \{ \tau_{il} [\eta_{il}]^\beta \} & \text{如果 } q \leq q_0 \\ J & \text{否则} \end{cases} \quad (1)$$

其中: q 是均匀分布在 $[0,1]$ 上的随机变量; $q_0(0 \leq q_0 \leq 1)$ 是一个参数; J 根据式(2)给出。

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}(t)^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in \text{allowed}_k} \tau_{is}(t)^\alpha [\eta_{is}(t)]^\beta} & \text{如果 } j \in \text{allowed}_k \\ 0 & \text{否则} \end{cases} \quad (2)$$

其中: $\text{allowed}_k = \{C - \text{tabu}_k\}$ 表示蚂蚁 k 下一步允许选择的城市; α 为信息启发式因子,表示信息素轨迹的相对重要性; β 为期望启发式因子,表示能见度的相对重要性; $\eta_{ij}(t)$ 为启发函数,其表达式为

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \quad (3)$$

在每只蚂蚁走完一步或完成对所有 n 个城市的遍历后,要对信息素进行更新处理, $t+n$ 时刻在路径 (i,j) 上的信息量可按如下规则进行调整。

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (4)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (5)$$

其中: $\rho \in (0,1)$ 表示信息素挥发因子,则 $1-\rho$ 表示信息素残留因子; $\Delta\tau_{ij}(t)$ 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,表达式为:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{l_k} & \text{若第 } k \text{ 只蚂蚁经过路径 } (i,j) \\ 0 & \text{否则} \end{cases} \quad (6)$$

其中: Q 表示信息素强度,它在一定程度上影响算法的收敛速度, l_k 表示第 k 只蚂蚁在本次循环中所走路径的总长度。

2 动态调节信息素增量的蚁群算法

蚁群算法实际上是一种结合了正反馈和启发式的进化算法。在该算法中,进化算法的关键是在“利用”和“探索”之间找到一个平衡点以利于全局搜索。“利用”即是利用群体的历史经验信息使得个体产生的解保持一定的稳定性,“探索”即是个体使搜索空间尽可能地扩大以得到全局最优解。在搜索的前期,由于正反馈的作用,算法还能搜索到较好的解,但随着迭代的不断进行,某些路径上的信息素积累得越来越多,使得更多的蚂蚁选择这些路径,但往往这些路径并不是最好的,即使迭代更多次,也无济于事。“利用”和“探索”之间的平衡被打破。

2.1 信息素更新规则

为了使得蚂蚁群体在搜索时能够更好地平衡两者,在此把信息素更新规则式(4)改为如式(7)的形式。因为在凸组合中,通过调整其中一个系数就可以很好地调节两个量之间的平衡,并用时变函数 $\rho(t)$ 代替其中的参数 ρ 。

$$\tau_{ij}(t+n) = (1-\rho(t)) \cdot \tau_{ij}(t) + \rho(t) \cdot \Delta\tau_{ij}(t) \tag{7}$$

其中: $\rho(t) = \begin{cases} \rho_1, & \text{若 } t \leq T_1 \\ \rho_2, & \text{若 } NC_{\max} \geq t > T_1 \end{cases}$,在最初的一段时间里, ρ 值

较小,信息素的“利用”占主导地位,随着迭代的进行,蚂蚁在所经过的路径上产生的信息素越来越多,在易于找到最优路径的同时,也增加了干扰。如果从第 $T_1 - N$ 次迭代开始,连续迭代 N 次获得的最优解没有变化,即第 $T_1 - N$ 次到第 T 次之间搜索到的最优路径刚好是第 $T_1 - N$ 次搜索到的路径,说明搜索陷入停滞,此时采用强制机制刺激蚂蚁,使蚂蚁的“探索”能力增强,即增大式(7)中的 $\rho(t)$,力图使搜索跳出局部最优,继续搜索更好的解。

2.2 信息素增量的更新规则

同时,在式(6)中加入自适应调节因子,使得在短时间内可以通过信息素增量上的差别来区分开次优和其他路径,在促使蚂蚁找到最优路径的同时,也大大地提高了收敛速度,缩短了搜索时间,因此将式(6)改为

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{e^{l_k - \text{BestTour}}} & \text{若第 } k \text{ 只蚂蚁经过路径 } (i, j) \\ 0 & \text{否则} \end{cases} \tag{8}$$

其中:BestTour 为目前找到的最优路径。新的信息素增量更新规则可以使蚂蚁开始关注次优路径,并加快信息素的积累,而其他路径上的信息素变化不大,以此吸引更多蚂蚁选择较优路径。

改进后的算法步骤如下:

- a) 参数初始化,令循环次数 $nc = 0$,挥发因子 $\rho = \rho_1$,设定蚂蚁个数、最大循环次数 $NC_{\max}$ 、初始化信息素等。
- b) 将 m 只蚂蚁随机地放置到 n 个城市上,把该城市放入 tabu_k 。
- c) 蚂蚁个体根据式(2)计算的选择概率来选择下一个城市 j ,并把城市 j 加入禁忌表中。
- d) 记录本次迭代的最优解,如果本次搜索到的解优于初始最优路径,更新最优路径,即用本次迭代的最优解代替初始最优路径,然后继续迭代。
- e) 按式(7)对路径上的信息素进行全局更新。
- f) 令迭代次数 $nc = nc + 1$,若 $nc < NC_{\max}$,并且如果从第 $T_1 - N$ 次开始,连续迭代 N 次,最优解没有发生变化,则增大参数 ρ ,即 $\rho = \rho_2$,继续迭代;否则 ρ 不变,继续迭代,并输出结果;

若 $nc > NC_{\max}$,则结束迭代。

3 仿真结果及分析

通过仿真比较改进后算法和基本蚁群算法及文献[10]算法的性能,本文各参数选取 $\beta = 5, \alpha = 1, \rho_1 = 0.1, \rho_2 = 0.8, Q = 0.9, m = 100, N = 500$,最大迭代次数设置为 $NC_{\max} = 100 \times n$ 。重复运行 10 次并与 ACS(基本蚁群算法)比较,实验结果如表 1 所示。其中 best、avg、iteration 分别代表 10 次得到的最优值、平均值、平均迭代次数。文献[10]的参数设置为: $m = 10, \beta = 2, \alpha = 1, \rho = 0.1, \xi = 0.9, Q = 0.9, NC_{\max} = 100 \times n$ 。

表 1 本文改进算法与 ACS 的实验结果比较

TSP	最优解	算法	best	avg	iteration
20cities	24	ACS	24	24	167.2
		本文	24	24	2.9
Oliver30	420	ACS	420	420.7	1199.4
		本文	420	420.4	32.6
Eil51	427	ACS	427	428.8	2008.9
		本文	427	428	568.7
Eil76	539	ACS	543	544.7	2912.5
		本文	539	542.1	454.7
Eil101	629	ACS	634	640	5059.9
		本文	629	638	524.6

表 2 是本文改进算法与文献[10]实验结果的比较。通过表 1、2 可以直观地看出,改进算法的优势相当明显,唯一不足的是 Eil51 的最优解不及文献[10]中的最优解。本文改进算法的结果较其他两种算法搜索到的解的平均值有明显改进,说明本文算法有着更强的搜索能力。更重要的是,搜索到最优解的迭代次数明显减少,收敛速度大大提高。

表 2 本文改进算法与文献[10]的实验结果比较

TSP	最优解	算法	best	avg	iteration
20cities	24	本文	24	24	2.9
		文献[10]	24.54	24.91	190
Oliver30	420	本文	420	420.4	32.6
		文献[10]	420	420.3	555
Eil51	427	本文	427	428	568.7
		文献[10]	426	429	1 786.1
Eil76	539	本文	539	542.1	454.7
		文献[10]	541	545.5	3 496.7
Eil101	629	本文	629	638	524.6
		文献[10]	632	642.5	6 453.7

在表 3 中,通过本文算法与文献[11]实验结果的比较,发现本文算法搜索到的结果均比文献[11]要好。其中文献[11]的参数分别为: $\beta = 5, \alpha = 1, \rho = 0.9, NC_{\max} = 100 \times n, Q = 0.9, m = 10$ 。在主要参数一致的情况下,本文数据和文献[10,11]中数据的基准并不相同。在此考虑平均改变量——即分别比较本文算法和文献[10,11]中平均值改进量,找到最优解的迭代次数的平均改变量,结果如表 4 所示。

表 3 本文改进算法与文献[11]的实验结果比较

TSP	算法	best	avg	iteration
20cities	本文	24	24	2.9
	文献[11]	24.53	24.68	323
Oliver30	本文	420	420.4	32.6
	文献[11]	423.74	424.19	521
Eil51	本文	427	428	568.7
	文献[11]	430.2	442.06	1 023
Eil101	本文	629	638	524.6
	文献[11]	635.97	641.24	3 195

$A_1A_2A_3A_4$,如图 3 和 4 所示。

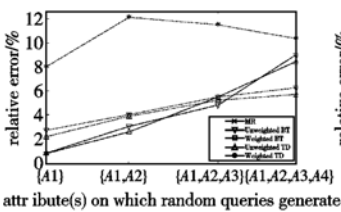


图3 聚集查询时的数据可用性

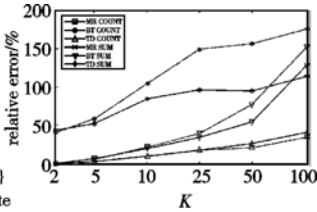


图4 在高斯分布数据上的聚集查询误差

实验结果表明,在考虑不同属性的权重时,BT 和 TD 两种匿名化算法得到的数据查询精确度比不加权的情况要高;又因为局部聚类最小化了信息损失,其查询精确度都比 MR 要高,在匿名化质量和数据可用性方面都显示出较大的优势。当 K 值较小时,BT 和 TD 算法比 MR 算法有明显的优势;而当 K 值变大时,TD 算法依然能保证较低的错误率,虽然 BT 算法导致的错误率增大,但还是比 MR 表现好。

5 结束语

隐私保护在各种数据应用中受到越来越多的关注,人们在享受信息技术带来的各种便利的同时,希望自己的隐私信息得到保护。本文着眼于基于可用性的数据匿名化研究,对如何度量数据可用性,如何在匿名化过程中提高数据可用性问题进行了系统研究,给出了加权确定性惩罚模型作为数据可用性的度量模型,该模型可以定量描述在数值属性和分类属性上由于数据匿名化带来的信息损失。基于这个模型,提出两种局部聚类匿名化算法。实验证明了两种算法在匿名化质量和数据查

询精确度方面具有明显优势,TD 相比 BT 算法更加稳定。

参考文献:

[1] BAYARDO R J,AGRAWAL R. Data privacy through optimal K -anonymization[C]//Proc of ICDE. Washington DC:IEEE Computer Society,2005:217-228.

[2] LEFEVRE K,DEWITT D J,RAMAKRISHNAN R. Mondrian multidimensional K -anonymity [C]//Proc of ICDE. Washington DC: IEEE Computer Society,2006:25.

[3] MEYERSON A,WILLIAMS R. On the complexity of optimal K -anonymity[C]//Proc of PODS. New York:ACM Press,2004:223-228.

[4] 王智慧,许俭,汪卫,等. 一种基于聚类的数据匿名方法[J]. 软件学报,2010,21(4):680-683.

[5] 许俭. 基于可用性的数据匿名化研究[D]. 上海:复旦大学,2008.

[6] MACHANAVAJJHALA A,GEHRKE J,KIFER D,et al. L -diversity: privacy beyond K -anonymity [C]//Proc of ICDE. New York: ACM Press,2006.

[7] 刘坚. K -匿名隐私保护问题的研究[D]. 上海:东华大学,2010.

[8] WINKLER W E. Using simulated annealing for K -anonymity, Technical Report Statistics 2002-7 [R]. Washington DC:U. S Census Bureau,2002.

[9] AGGARWAL G,FEDER T,KENTHPADI K,et al. Achieving anonymity via clustering[C]//Proc of PODS. New York:ACM Press, 2006: 153-162.

[10] SWEENEY L. K -anonymity: a model for protecting privacy [J]. Fuzziness, and Knowledge-based Systems,2002,10(5):509-517.

(上接第 136 页)

表 4 本文改进算法与文献[10,11]改变量的比较

改变量	算法	20cities	Oliver30	Eil50
平均值改变量	本文	0.3	0.8	2.6
	文献[10]	0.1	1.6	3.3
	文献[11]	0.28	2.62	3.65
迭代次数改变量	本文	164.3	1166.8	1440.2
	文献[10]	72.9	-303.9	1427.5
	文献[11]	26	738	562

图 1 和 2 分别给出了本文改进算法在求解 Eil76 和 Eil101 这两个问题时的收敛曲线。从图 1 可以直观地看出,在迭代几乎陷入停滞时,改变参数 ρ 对最优解的寻找起到了很大的作用,这是因为增大 ρ 刺激了蚂蚁对次优路径的注意,然后迭代继续,直到找到最优解。从图 2 也可以看出,改变参数 ρ 确实有利于最优解的寻找,收敛速度明显加快。

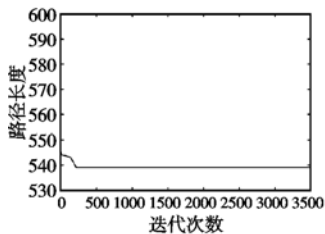


图1 Eil76的收敛过程

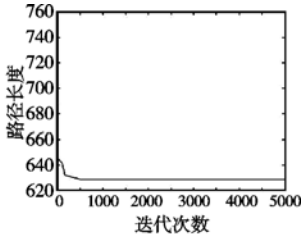


图2 Eil101的收敛过程

4 结束语

本文的基于动态调节信息素增量的蚁群算法在搜索陷入局部最优时,使得次优路径上的信息素明显增多,而其他路径

上的信息素则没太大变化,以此引起蚂蚁的注意,使更多的蚂蚁选择次优路径,力图跳出局部最优,找到更好的解。实验结果也验证了这一思想。虽然关于信息素更新规则的改进已有很多,但相信一些细节问题还有待进一步的探讨。

参考文献:

[1] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating agents [J]. IEEE Trans on Systems, Man, and Cybernetics, 1996,26(1):29-41.

[2] 史士财,李荣,付宜利,等. 基于改进蚁群算法的装配序列规划 [J]. 计算机集成制造系统, 2010,16(6):1189-1194.

[3] 张金标,陈科. 并行设计任务调度的自适应蚁群算法[J]. 计算机辅助设计与图形学学报,2010,22(6):1070-1074.

[4] 覃刚力,杨家本. 自适应调整信息素的蚁群算法[J]. 信息与控制, 2002,31(6):198-201.

[5] 黄国锐,曹先彬,王煦法. 基于信息素扩散的蚁群算法[J]. 电子学报,2004,32(5):865-868.

[6] 刘喜恩. 用于连续空间寻优的一种蚁群算法[J]. 计算机应用, 2009,29(10):2744-2747.

[7] 梁普明,肖金红,龙文,等. 基于记忆表的连续蚁群优化算法[J]. 计算机工程,2010,36(16):183-185.

[8] 孙泽宇,魏巍. 一种改进蚁群算法组合优化问题的研究[J]. 计算机仿真,2010,27(8):194-197.

[9] 段海滨. 蚁群算法原理及其应用[M]. 北京:科学出版社,2005.

[10] 何雪海,胡小兵,赵吉东,等. 基于自适应转移概率的蚁群优化算法[J]. 计算机工程,2010,36(23):165-167.

[11] 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.