

蚁群算法中系统初始化及系统参数的研究

吴春明¹, 陈 治¹, 姜 明²

(1. 浙江大学计算机科学与技术学院, 浙江杭州 310027; 2 杭州电子科技大学, 浙江杭州 310018)

摘 要: 蚁群算法作为近年来一种新的模拟进化算法具有较强的发现解的能力, 但同时也有收敛慢、耗时间的缺点. 本文针对各种不同规模的 TSP 问题, 通过实验对各参数的设置做了研究, 并对蚂蚁初始化提出了新的算法, 并进行了实验验证.

关键词: 蚁群算法; 蚂蚁系统的初始化; 参数设置; 旅行商问题

中图分类号: TP301.6 文献标识码: A 文章编号: 0372-2112 (2006) 08-1530-04

The Research on Initialization of Ants System and Configuration of Parameters for Different TSP Problems in Ant Algorithm

WU Chunming¹, CHEN Zhi¹, JIANG Ming²

(1. Department of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310027, China;

2. Hangzhou Dianzi University, Hangzhou, Zhejiang 310018, China)

Abstract As a novel simulated evolutionary algorithm which was proposed in recent years, ant colony optimization (ACO) algorithm has great capability in searching better solutions, but ACO also has the shortcoming of slow converging. For different dimensions of TSP problems, the paper studies the settings of parameters with experiments and proposes a new method of the initialization of ants system. Experimental results indicate that the enhancement is practical.

Key words ant algorithm; the initialization of ants system; optimum configurations; Traveling Salesman Problem (TSP)

1 引言

蚁群算法 (Ant Colony Algorithm, ACA) 是由意大利学者 M Dorigo, V Maniezzo, A Colomi 等人在 20 世纪 90 年代初首先提出来的^[1], 它是继模拟退火算法、遗传算法、禁忌搜索 (Tabu Search) 算法、人工神经网络算法后的又一种应用于组合优化问题的启发式搜索算法. 最近几年, 各国研究者在组合优化、通信网络及机器人等领域已研究出大量优秀的算法如蚁群优化算法 ACO (Ant Colony Optimization)、蚁群路由选择算法 (Ant Colony Routing) 等^[2], 这些算法已经用于求解各种组合优化问题, 如旅行商问题 (traveling salesman problem, TSP)、车辆路径规划问题 (vehicle routing problem, VRP)、二次分配问题 (quadratic assignment problem, QAP) 等^[3-5].

2 基本蚁群算法的简介

由于关于蚁群算法原理 M Dorigo 的著作已有详细介绍^[1], 在此不做赘述. 有关蚁群算法的研究都是围绕组合

优化问题的, 最初它应用于作为组合优化之一的 TSP 问题, n 个城市的 TSP 问题就是求经过 n 个城市一次最后回到出发点的最短路径. 由于蚂蚁的寻路与 TSP 问题十分相似, 本文就以平面上 n 个城市的 TSP 问题说明基本蚁群算法的模型.

设 m 是蚁群中蚂蚁的数量, d_{ij} ($i, j = 1, 2, \dots, n$) 表示城市 i 和城市 j 之间的距离.

2.1 蚁群系统的初始化

将 m 只蚂蚁分别随机放在 n 个城市之中, 初始时刻各条路径上的信息素相同, 设 $\tau_{ij}(0) = C$ (C 为常数).

2.2 路径选择

蚂蚁 k ($k = 1, 2, \dots, m$) 在运动中, 根据各条路径上信息素的浓度来决定转移方向, $P_{ij}^k(t)$ 表示 t 时刻蚂蚁 k 由城市 i 转移到城市 j 的概率, 其计算公式如下:

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \in \text{allow}_{ik}} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)}, & j \in \text{tabu}_k \\ 0, & \text{其他} \end{cases} \quad (1)$$

式中, $tabu_k (k = 0, 1, 2, \dots, m)$ 为蚂蚁 k 走过的城市集合, $allow_k = \{0, 1, 2, \dots, n-1\} - tabu_k$, 为蚂蚁下一步允许选择的城市. 开始时 $tabu_k$ 中只包含一个蚂蚁出发的城市, 随着算法的进行, $tabu_k$ 中不断增加, 直到所有 n 个城市全在其中. $\tau_{ij}(t)$ 为 t 时刻城市 i 和城市 j 之间的信息素浓度, $\eta_{ij}(t)$ 为 t 时刻 i 和 j 之间的能见度, 一般取 d_{ij} 的倒数, α, β 为调节信息素与能见度的相对重要程度.

2.3 信息素更新原则

随着时间的推移, 路径上的信息素逐渐消失, 称为挥发. 取 $1 - \rho (0 < \rho < 1)$ 为挥发率, 挥发是为了防止信息素聚集过多而湮没蚂蚁行走时留下的启发信息. 经过 m 个时刻, 蚂蚁完成一次循环, 各路径上的信息素要按式 (2) 更新:

$$\tau_{ij}(t+m) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}^k$$

$$\rho \in (0, 1), \Delta \tau_{ij}^k = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (2)$$

式中, $\Delta \tau_{ij}^k$ 为第 k 只蚂蚁在本次循环中留在路径 (i, j) 上的信息素, $\Delta \tau_{ij}$ 为本次循环所有蚂蚁留在路径 (i, j) 上的信息素.

根据信息素更新策略的不同, Dorigo M 曾给出 3 种不同模型, 分别称之为 ant cycle system (ACS), Ant quantity system (AQS), Ant density system (ADS).

(1) ACS 模型

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{如第 } k \text{ 只蚂蚁经过 } (i, j) \text{ 边} \\ 0 & \text{其他} \end{cases} \quad (3)$$

(2) AQS 模型

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{d_{ij}}, & \text{如第 } k \text{ 只蚂蚁经过 } (i, j) \text{ 边} \\ 0 & \text{其他} \end{cases} \quad (4)$$

(3) ADS 模型

$$\Delta \tau_{ij}^k = \begin{cases} Q, & \text{如第 } k \text{ 只蚂蚁经过 } (i, j) \text{ 边} \\ 0 & \text{其他} \end{cases} \quad (5)$$

式 (3) 中, L_k 为蚂蚁 k 走过的路径总长; Q 为常数.

它们的差别在于: (a) 信息素的更新时刻不同: ACS 模型是在蚂蚁走完全程, 回到起点时按式 (2) 更新信息素, 而 AQS、ADS 模型则是在蚂蚁每到达一个城市就更新它所走过的边上的信息素; (b) 每次信息素更新的量 $\Delta \tau_{ij}^k$ 不同, 它们的区别在于后两种模型中利用的是局部信息, 而前者利用的是整体信息. 在 ACS 和 AQS 中分别用 L_k 和 d_{ij} 来调节信息素的释放量.

3 蚁群算法中蚂蚁个数及初始化的研究

在对蚂蚁系统初始化时, 需要确定蚂蚁的数目和每只蚂蚁的出发城市. 蚂蚁数目的确定对算法起着比较大的影响, 一般来说, 蚂蚁数目越多, 算法的全局搜索能力越强, 但随着蚂蚁的数目增多, 计算速度将成指数级增长.

现在蚂蚁初始位置都是以随机方式决定^[9], 但是在仿

真实验中, 经常会出现由于随机化初始蚂蚁时部分蚂蚁的初始位置过于紧密, 这些蚂蚁选择路径时都倾向于选择附近较短的路, 从而使蚂蚁比较密集的地方局部收敛过快, 缩小了解空间, 影响最优解的生成.

鉴于这种情况容易引起早熟停滞现象, 本文提出确定蚂蚁初始位置时使用“蚂蚁边缘化”的原则, 所谓边缘化是指将蚂蚁的初始位置尽量选择在城市网中边界的城市上, 分散蚂蚁的出发点, 这样蚂蚁在出发选择路径时不会过早地互相影响, 因而增大解的空间, 提高解的质量. 但在实际应用中, 如何判断一个城市在城市网络中所处的位置是相对困难的, 因为数据的提供往往都是抽象的, 如同 TSPLIB (专门研究 TSP 问题的国际机构) 上的数据, 一般来说只提供任意两点间的距离 d_{ij} . 这样我们就必须寻找其他办法来确定哪些点处在边缘. 本文使用一种近似方法来确定:

For i from 1 to n (n 为城市数)

首先对 n 个城市分别求出 $S_i, S_i = \sum_{j=1}^n d_{ij} (i = 1, \dots, n)$;

对 S_i 按照从大到小的顺序排列, S_i 中第一个元素即为城市网中离其他城市距离最远的那个城市, 第二个元素为次远, 而在实际中, 往往在城市网络边缘的和分散城市是距离其他城市比较远的^[7]. 这样, 我们把蚂蚁位置分别放在排序后的 S_i 所代表的城市上.

以下为其他参数都不变, 改进蚂蚁初始位置和随机放置蚂蚁位置的结果的比较:

(所用数据为 bays 29 即城市数为 29 蚂蚁数设为 14 共循环 1000 次, 从中挑出最好解算一次实验, 实验 10 次求其平均值, 此数据最好解为 2020 由 TSPLIB 提供)

在这次实验中, 我们使用 α, β, ρ 分别为 7, 10, 0.5

表 1(a) 两种初始化蚂蚁方法的第一组比较

	均值	最优解	最差解	最差与最优之差
随机初始化	2077	2040	2094	54
边缘法初始化	2032	2020	2060	40

表 1(b) 两种初始化蚂蚁方法的第二组比较

α	β	ρ	蚂蚁数 n	边缘法均值	随机法均值	所需时间
7	10	0.5	12	5350	5503	50s
7	10	0.5	24	5274	5382.7	70s
7	10	0.5	36	5238	5284.3	95s
7	10	0.5	48	5231	5233.5	140s

实验结果分析:

(1) 通过边缘法初始化蚂蚁, 实验得到了最优解 2020 而随机初始化法得到的最优解只有 2040, 随机法的均值与边缘法均值之差为 45 相当与 bays 29 数据中两个城市之间的平均距离, 即边缘法得到的解比起随机法得到的解近似缩短了一个城市的行程;

(2) 从最差解与最优解之差来看, 边缘法也强于随机初始化, 说明边缘法初始化的鲁棒性要强于随机法.

为增加实验的可靠性,我们用其他数据对两种方法再进行了实验,表 1(b)为实验结果(所用数据为 TSPLIB 提供的 gr48 蚂蚁数分别为 12 24 36 48 最优解为 5203)

通过实验结果可以看到,在数据量增大时,用边缘初始化得到的解仍然比随机初始化得到的解要好,其中蚂蚁数为 12 时,所缩短的距离为三个城市间的平均距离.而当蚂蚁数逐渐增多时,两种方法得到的解差别逐渐变小,这是因为,当蚂蚁数增多时,两种方法所放置的蚂蚁所在城市逐渐相似,到蚂蚁数等于城市数时,两种方法最终都使每个城市分配到一只蚂蚁.同时,随着蚂蚁数增多,程序所需要时间也大大增多.

从以上两组实验结果的对比,可以看到在性能方面,边缘法初始化确实强于随机初始化.当城市数比较多,而对速度的要求较高时,可以使用边缘法初始化,将蚂蚁数设为城市数的 3/4 或者 1/2 就可以找到较优解,同时可以大大减少计算时间,提高效率.

4 关于蚁群算法中参数 α β ρ 的研究

由于目前还没有一种有效的数学分析方法使不同情况下的蚂蚁系统都能生成最优的参数设置,而国内外有关蚂蚁系统的文献大都只笼统地提出一种参数配置,但在实验中我们发现对于不同规模的问题同样的参数所得的结果优劣差别很大.因此,本文通过大量的仿真实验有针对性地针对不同规模问题进行实验,获得参数的统计数据,研究参数对算法的影响.

本文通过仿真程序分别对 TSPLIB 上不同规模的三组数据(分别为 17 48 120 城市)进行了参数的组合实验,现将实验过程和分析详述如下.

4.1 实验说明

仿真程序使用 1000 次循环进行一次实验,实验中采用只改变一个参数的策略来探索参数的设置对算法效率的影响.对于城市数小于 50 的问题,蚂蚁的总数目总是设置为城市的总数目,即初始时刻每个城市放置一只蚂蚁.而对于城市数大于 50 的,我们使用上文提到的蚂蚁边缘化的算法,蚂蚁数目为 3/4 城市数.

每组数据试验 10 次取平均作比较,试验中所用的 3 组 TSP 问题数据分别来源于 TSPLIB 中的 gr17(17 城市规模), gr48(48 城市规模), gr120(120 城市规模).表中黑体表示得到最优解.

4.2 实验结果

结果见表 2~ 表 4 其中, gr120 由于城市数目太多,程序运行时间太长,只对其中一些具有代表的参数进行了实验,取运行的一次解.

4.3 实验结果分析

α , β 的取值越大,计算量也越大,计算时间就越长,所以在能获得满意解的情况下,建议 α β 取相对较小的值.表中平均值表示将 10 次运行中每次得到的最短线路长的

平均值;最优解表示 10 次运行中得到的 10 条最短线路中的最小值;最差解表示 10 次运行中每次得到的最短线路中的最大值;

分析表 2 至表 4 的结果,可以得到如下的结论:

表 2 gr17 实验结果

α	β	ρ	均值	最优解	最差解
1	1	0.5	2283.1	2172	2354
1	5	0.5	2088.7	2085	2097
5	1	0.5	2305	2219	2384
5	5	0.5	2086.4	2085	2096
10	1	0.5	2255.2	2185	2315
10	10	0.5	2085	2085	2085

表 3 gr48 实验结果

α	β	ρ	均值	最优解	最差解
1	1	0.5	9030	8949	9115
1	5	0.5	5398.8	5276	5463
1	10	0.5	5349	5261	5393
5	1	0.5	8629.2	8182	8821
5	5	0.5	5371.8	5242	5498
5	10	0.5	5338	5202	5395
7	5	0.5	5433.1	5299	5495
7	10	0.2	5394	5352	5432
7	10	0.6	5203	5046	5323
10	5	0.5	5430.3	5280	5519
10	10	0.5	5414	5321	5785

表 4 gr120 实验结果

α	β	ρ	均值
1	10	0.5	7684
1	20	0.6	7653
5	5	0.5	7973
5	10	0.5	7899
7	10	0.5	7709
10	10	0.5	7970
15	20	0.5	8057

(1) 对于 gr17 城市问题,在 $\beta > 5$ $\alpha > 1$ $\rho = 0.5$ 时很快就能找到最优解 2085 而在 $\alpha = 10$ $\beta = 10$ $\rho = 0.5$ 的时候,10 次实验均得到了最优解.而由于取值越大,计算量越大,所以在对于城市规模较小(< 30)的情况下,通过对表 1 结果的分析,建议 α β ρ 的取值分别为 10 10 0.5 通过对其他类似规模的 TSP 问题的验证,使用这个取值结果也很快找到最优解.

(2) 对于 gr48 城市问题, α β ρ 的取值相对要求就比较精确,在大量实验之后,最终在 $\alpha = 7$ $\beta = 10$ $\rho = 0.6$ 时出现了最优值 5046,实验也表明,在固定 $\beta = 10$ $\rho = 0.5$ 时,让 α 在 1 到 10 之间变化,那么可以得到一条开口向上的抛物线轨迹,而抛物线的最低点是在 $\alpha = 7$ 左右.而固定 $\alpha = 7$ $\rho = 0.5$ 当 β 取值为 10 左右时,所得到的解的平均值也是近似优的.而 ρ 的取值相对来说浮动范围比较模糊,实验表明在 0.5~0.8 之间的解均值都比较相近.对于城市规模

中等的问题 ($30 < \text{城市数} < 100$), 建议 α 、 β 、 ρ 的取值在 7、10、0.6 左右浮动。

(3) 对于 gr120 城市问题, 由于城市数目太多, 完成一次实验所需时间较长, 只对其中部分有代表性的参数进行实验, 最终所得的最好解为 7559, TSPLIB 所提供的最优解为 6942。实验表明, 在 α 较小, β 相对较大的情况下, 解的质量要好一些, 这表明在城市数目比较多的情况下, 蚂蚁在选择路径时备选路径的长度对解的影响较大。对于城市规模大的问题 (> 100), 建议 α 的值设相对小, β 的值相对大, ρ 的值在 0.5 左右。

以上分析表明, AQS 系统在参数方面鲁棒性较差。其中 ρ 的作用是使蚂蚁忘掉过去获得的一部分经验, 避免蚂蚁过早收敛于一个局部最优解; 如果 α 的值很高, 则意味着信息素轨迹非常重要, 使蚂蚁倾向于选择其他蚂蚁已经过的边; 而若 β 很高, 则尽管一条边上的信息素量很大, 但蚂蚁总会以很高的概率选择更近的城市。如果 $\alpha = 0$ 则相当于经典贪婪算法。如果 $\beta = 0$ 则只是一个正反馈过程。

不同规模的 TSP 问题, 城市数目和城市距离两者间的制约关系也是不同的, 对于上述实验所得到的结果, 我们可以将之做为以后解不同规模问题时的出发点, 在此基础上进行更精确的参数匹配, 从而减少工作量。

5 结论

本文对使用蚁群算法解决较大规模的 TSP 问题时, 对 AQS 系统的初始化提出了新的方法, 并通过实验数据进行了验证; 针对不同规模的 TSP 问题, 通过实验对 α 、 β 、 ρ 各值的设置提供比较精确的组合方式, 为以后蚁群算法的应用打下基础。

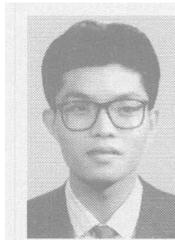
对于蚁群算法收敛较慢的缺点, 以及如何能建立一个通用的算法来确定对不同问题确定最优的参数设置, 将是以后研究中的重点问题。

参考文献:

[1] Dorigo M, Maniczo V, Colmi A. Ant system: Optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems Man and Cybernetics Part B, 1996, 26 (1): 29-41

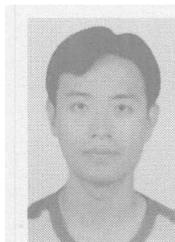
- [2] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem [J]. IEEE Trans on Evolutionary Computation, 1997, 1 (1): 53-66
- [3] Dorigo M, Gambardella L M, Middendorf M, et al. Guest editorial: Special section on ant colony optimization [J]. IEEE Trans On Evolutionary computation, 2002, 6 (4): 317-319
- [4] Dorigo M, Gambardella L M. Solving symmetric and asymmetric TSPs by ant colonies [A]. Proceeding of the IEEE conference on Evolutionary Computation (ICEC 96) [C]. Piscataway, NJ USA: IEEE Press, 1996, 622-627.
- [5] Vittorio Maniczo Antonella carbonaro. Ant colony optimization: an overview [J]. Knowledge and Data Engineering, 1999, 11 (5): 769-778
- [6] 温文波, 社维. 蚁群算法概述 [J]. 石油自动化, 2002, (1): 19-22
- [7] 李军. 非对称距离的旅行商问题的构造算法 [J]. 运筹与管理, 2000, (1): 5-6
- [8] 陈岐, 沈洁, 秦玲. 蚁群算法进行连续参数优化的新途径 [J]. 系统工程理论与实践, 2003, (3): 48-53

作者简介:



吴春明 男, 1967年生, 汉族, 浙江萧山人, 博士, 浙江大学计算机学院教授。主要从事人工智能、智能机器人技术及计算机网络等领域方向的科学研究工作。

E-mail: wuchunming@cs.zju.edu.cn



陈治 男, 1977年生, 汉族, 甘肃省天水市人, 浙江大学计算机科学与技术学院硕士研究生。主要研究方向为人工智能应用与算法研究。