

蚂蚁算法在 TSP 问题求解的应用

崔之熠^{1,2}, 王茂芝^{1,2}, 刘国涛^{1,2}, 朱怀朝^{1,2}

(1. 成都理工大学信息管理学院, 成都 610059; 2. 数学地质四川省高校重点实验室, 成都 610059)

摘 要: 蚂蚁算法是目前解决大规模复杂问题比较有效的算法。同时 TSP 问题是经典的 NP - C 问题, 已被广泛应用于在 VLSI 芯片设计、网络路由和车辆选路等领域, 对 TSP 问题的求解的突破意味着大量 NPC 问题的求解可以迎刃而解, 因而有着重要的实际价值和理论意义。文章系统地介绍了 TSP 问题, 并在此基础上对蚂蚁算法求解 TSP 问题做了相关探讨。实验结果表明, 蚂蚁算法对参数的初始值也具有敏感性, 对于一个好的初始值的确定, 需要建立在大量试验的基础上。

关键词: 蚂蚁算法; TSP 问题; 参数

中图分类号: TP75

文献标识码: A

1 TSP 问题及蚂蚁算法

1.1 TSP 问题的背景及研究价值

TSP(Traveling salesman problem) 问题是十九世纪初爱尔兰的数学家 William Rowan Hamilton 和英国数学家 ThomaSPenyngtonkirkman 提出的, 通常被描述为: 对给定城市交通网络, 如何为一个推销商选择一条路线, 从网络的某一点(驻地)出发, 经过每一个结点后回到出发点, 使得总行程最短^[1-4]。TSP 问题是著名的 NP 完全难题, 也是组合优化、计算机科学界经典的问题之一。

由于 TSP 问题是许多领域内复杂工程优化问题的抽象形式, 在 VLSI 芯片设计、网络路由和车辆选路^[5]等领域有着广泛应用, 还为离散优化中各类算法提供了思想方法平台, 同时 TSP 问题是 NPC 中有代表性的问题, 与其他 NPC 问题具有等价性, 若在 TSP 的求解当中取得突破, 则大量 NPC 问题的求解方法就可以迎刃而解, 因而对旅行商问题(TSP)求解方法的研究具有重要的实际价值和理论意义。虽然关于 TSP 问题的完全有效的算法目前尚未找到, 但长期以来人们不断地探索并积累了大量的算法, 归纳起来目前主要算法可分成传统优化算法和现代智能优化算法。传统优化算法主要分为完全算法和近似算法两个方向, 现代智能优化算法主要

有遗传算法、模拟退火算法、蚁群算法、禁忌搜索算法、贪婪算法和神经网络方法等^[5]。

1.2 蚁群算法

1.2.1 蚁群算法的起源及发展

蚁群算法(Ant Colony Algorithm, ACA)是一种最新发展的模拟真实世界蚂蚁觅食行为的仿生优化算法^[6]。1991年, 意大利学者 Dorigo M 在巴黎召开的第一届欧洲人工生命会议(European Conference on Artificial Life, ECAL)最早提出了蚁群算法的基本思想; 1992年, Dorigo M 又在其博士论文中进一步阐述了蚁群算法的核心思想。

由于算法具有分布计算、信息正反馈、启发式搜索且易于与其它方法结合的特点, 在解决许多复杂的优化问题方面展示出优异的性能和巨大的发展潜力, 越来越受到人们的重视。蚁群算法最早成功应用于解决著名的 TSP 问题, 而后渗透到多个应用领域, 由解决一维静态优化问题到解决多维动态组合优化问题, 由离散范围内的研究逐渐扩展到连续范围内的研究, 如在系统控制、人工智能、模式识别、生产调度等方面, 取得了惊人的进展, 成功的运用于解决 TSP 问题、背包问题、二次分配等问题^[7-10]。

1.2.2 蚂蚁的行为特性

为了研究在受约束条件下蚂蚁的觅食行为, Deneu-

bourg 等人提出双桥试验。在现实生活中,通过观察可知,蚂蚁在蚁巢食物源之间的路线几乎是一条直线,而不是曲线,如图 1 所示。



图 1 蚂蚁直线路径示意图

如果在蚁巢和食物源之间放置一个障碍,开始时蚂蚁的分布是均匀的,按同等概率选择路径。如图 2 所示。



图 2 蚂蚁等概率选择路径示意图

蚂蚁在运动过程中,能够在其经过的路径上留下信息(图 2 中以散列的点表示),而且能感知这种物质的存在及其强度并以此指导自己移动的方向。蚂蚁倾向于选择信息素强度较高的方向。相等时间内较短路径上的信息素遗留的比较多,则选择较短径的蚂蚁也随之增多,如图 3 所示。不难看出,由于大量蚂蚁的集体行为表现出了一种信息正反馈现象,蚂蚁个体就是过这种信息交流机制来搜索食物,并最终沿着最短路径行进。



图 3 蚂蚁选择较短路径示意图

应用蚂蚁的上述行为特性求解组合优化问题的关键有两点:一是如何把实际问题的优化目标和信息素之间建立联系,并应用上述正反馈原理确定信息素的调整策略(从而实现对目标的优化);二是信息素浓度调整策略的制定如何和搜索空间的调整建立联系,并和某些启发式规则相结合以确定选择策略作为指导下一步搜索的依据。

2 蚂蚁算法求解 TSP 问题

2.1 求解 TSP 问题的人工蚂蚁行为规则

应用蚂蚁的行为特性求解 TSP 问题时,每只人工蚂

蚁的行为还必须符合下列规律^[6]:

(1) 根据路径上的信息素浓度,以某种概率来选取下一步的路径。

(2) 不再选取自己本次循环已经走过的路径作为下一步的路径(可以通过一个数据结构来控制这一点)。

(3) 当完成一次循环后,根据整个路径长度来释放相应浓度的信息素,并更新走过路径上的信息素浓度。

2.2 蚂蚁算法求解 TSP 问题蚂蚁行为量化

用蚂蚁算法求解 TSP 问题时需要解决三个问题:

(1) 信息素浓度如何刻画;(2) 信息素浓度如何调整;(3) 蚂蚁的选择策略如何确定。而且这三个问题都必须予以量化,并且在量化的过程中必须和所优化的目标以及已知条件建立某种联系。为此,引入变量 $\tau_{ij}(t)$,

$\Delta\tau_{ij}$, $\Delta\tau_{ij}^k$, p_{ij}^k , 其中 $1 \leq i, j \leq n$, $1 \leq k \leq m$, n 表示 TSP 问题的规模,即城市数,而 m 则表示人工蚂蚁的个数。 $\tau_{ij}(t)$ 表示在 t 时刻边 (i, j) 上的信息素浓度。当蚂蚁完成一次循环后,相应边上的信息素浓度为 $\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}$,即等于上一时间段残留下的信息素浓度加上当前时间段新增加的信息素浓度。其中 ρ 为一个取值范围在 0 到 1 之间的常数,显然, $1 - \rho$ 表示在时间段 t 到 $t+1$ 之间信息素浓度的挥发强度因子, $\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$, 其中 $\Delta\tau_{ij}^k$ 是第 k 只蚂蚁在时间 t 到 $t+1$ 之间,在边 (i, j) 上增加的信息素浓度。

对于 M. Dorigo 提出的 $\Delta\tau_{ij}$ 的三种计算模型 ant-cycle system、ant-quantity system 和 ant-density system,后两种模型中利用的是局部信息,而前者利用的是全局信息,经过实验对比,在求解 TSP 问题时 ant-cycle system 性能较好,因而本文采用它作为基本模型:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{如果第 } k \text{ 只蚂蚁在时间 } (t+1) \text{ 之间} \\ & \text{经过边 } (i, j) \\ 0 & \text{否则} \end{cases}$$

其中 Q 是一个常量,用来表示蚂蚁完成一次完整的路径搜索后,所释放的信息素总量, L_k 表示第 k 只蚂蚁的路径总费用,它等于第 k 只蚂蚁经过的各段路径所需费用的总和。显然,蚂蚁不会在其没有经历过的路径上释放信息素。

定义第 k 只蚂蚁当前时刻在顶点 V_i , 下一步选择顶点 V_j , 即选择路径 (i, j) 的概率为:

$$\Delta p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta}{\sum_{j \in \text{allowed}_k} [\tau_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta} & \text{if } j \in \text{allowed}_k \\ 0 & \text{otherwise} \end{cases}$$

其中 $\eta_{ij} = \frac{1}{C_{ij}}$, C_{ij} 为路径 (i, j) 所需费用, α, β 为两个参数,分别用来控制信息素浓度和路径长度的相对重要程

度, $allowed_k$ 是第 k 只蚂蚁在满足人工蚂蚁行为规律 3 个条件的前提下, 下一步可以选择的路径集合。

2.3 蚂蚁算法求解 TSP 问题的算法描述

根据上面的分析, 再引入一个控制“蚂蚁不再选取自己本次循环已经走过的路径作为下一步路径”这一行为特性的数据结构 $visited - city_k$, 表示第 k 只蚂蚁访问过的城市集合, 其中 $1 \leq k \leq m$ 。在初始化的时候, m 只人工蚂蚁被放置在不同的城市上, 赋予每条边的信息素浓度初值为一常数 $\tau_{ij}(0) = C$ 。每只蚂蚁的 $visited - city_k$ 的第一个元素赋值为它所在的城市。当蚂蚁完成一次完整的寻径过程后(即从某个城市出发走完所有城市一次并回到出发点), 计算 $\Delta\tau_{ij}^k$, 并更新每条边上的信息素浓度, 然后开始新一轮循环。直到迭代次数达到事先定义好的最大循环次数 NC 或所有的蚂蚁都选择了同一路径时, 程序终止。

Ant - Cycle System 算法描述如下:

(1) 初始化: $t = 0$, $nc = 0$, $\tau_{ij}(t) = C$, $\Delta\tau_{ij} = 0$, 并置 m 只蚂蚁于 n 个城市上。其中 nc 为迭代循环的次数, t 是一个“伪”时间控制变量。

(2) 将各蚂蚁的初始出发点置于当前解集 $visited - city_k$ 中; 对每只蚂蚁 $k(k = 1, 2, \dots, m)$ 按概率 p_{ij}^k 移至下一顶点 j ; 并将 j 置于当前解集 $visited - city_k$ 中。

(3) 计算各蚂蚁的目标函数值 $L_k(k = 1, 2, \dots, m)$; 记录当前的最好解。

(4) 按信息素浓度调整规则更新每条边的信息素浓度。

(5) 对各边 (i, j) , 置 $\Delta\tau_{ij} = 0$, $nc = nc + 1$ 。

(6) 若 $nc < NC$ 且无退化行为(即找到的都是相同解) 则转步骤(2)。否则结束并输出当前最好解。

2.4 蚂蚁算法求解 TSP 问题的参数选择

蚂蚁算法中涉及的主要参数有 $Q, C, \alpha, \beta, \rho$ 五个, 参数对算法性能有很大的影响。一般取 $0 \leq \alpha \leq 5$, $0 \leq \beta \leq 5$, $0.1 \leq \rho \leq 0.99$, $1 \leq Q \leq 10000$ 。 α 值的大小表明留在每个结点上的信息量受重视的程度, α 值越大, 蚂蚁选择以前经过的路线的可能性越大, 但过大会使搜索过早陷于局部最小解; β 的大小表明启发式信息受重视的程度; β 值越大, 蚂蚁选择离它近的城市的可能性也越大; ρ 表示信息素的保留率, 如果它的值取得不恰当, 得到的结果会很差。

根据以上分析可以知道, 研究参数 α, β, ρ 的最佳配置, 对发挥蚂蚁算法在实际问题中的作用有很重要的意义。除这三个关键参数外, 蚂蚁数量 M 和总信息量 Q 的选择对算法也有一定的影响。

关于蚂蚁数量 m 的大小对蚂蚁算法循环次数(收

敛性能)的影响, 当蚂蚁数量过大(如接近问题的模)时, 虽然搜索的稳定性和全局性得到提高, 但算法的收敛速度减慢。

确定蚂蚁数目: 蚂蚁数目大约为问题规模 n 的 $1/3 - 2/3$ 之间。

参数粗调: 调整 α, β 的取值。

参数细调: 调整 ρ 的取值。

3 结果分析

3.1 蚂蚁算法求解 TSP 问题的结果及分析

通过实验应用蚂蚁算法对 TSPLIB 中城市数为 29 的 bays29.tsp 问题进行求解。数据来源于标准的 TSP 数据库。本文通过选择不同的参数, 并经程序计算, 最后选取效果最优的一组参数。

(1) 城市数目 $N = 29$, 蚂蚁数量 $M = 20$, $C = 1.0$, $Q = 1000.0$, 迭代次数为 $NC = 1000$, 考察参数 α, β, ρ 对结果的影响(表 1)。

表 1 参数 α, β, ρ 对结果影响的分析

参数 α, β		参数 ρ	最优解
$\alpha = 2$	$\beta = 2$	$\rho = 0.2$	opt_dist = 2258.0000
$\alpha = 2$	$\beta = 2$	$\rho = 0.5$	opt_dist = 2293.0000
$\alpha = 2$	$\beta = 2$	$\rho = 0.7$	opt_dist = 2227.0000
$\alpha = 2$	$\beta = 1$	$\rho = 0.2$	opt_dist = 2293.0000
$\alpha = 2$	$\beta = 5$	$\rho = 0.2$	opt_dist = 2146.0000
$\alpha = 1$	$\beta = 2$	$\rho = 0.7$	opt_dist = 2258.0000
$\alpha = 4$	$\beta = 2$	$\rho = 0.7$	opt_dist = 2231.0000

从表 1 可看出, 在 M, C, Q 一定的情况下, 对于影响蚂蚁下一步路径选择概率大小的参数 α 和 β , 其中 β 参数的取值对算法性能的影响是主要的, α 参数对算法收敛结果的影响次之; 信息素浓度的残留因子 ρ 的取值对算法性能影响不大。

(2) 城市数目 $N = 29$, 蚂蚁数量 $M = 20$, $Q = 1000.0$, 迭代次数为 $NC = 1000$, 参数 α, β, ρ 一定, 考察参数 C 对结果的影响(表 2)。

表 2 参数 C 对结果的影响分析

参数 α, β, ρ			参数 C	最优解
$\alpha = 2$	$\beta = 2$	$\rho = 0.2$	$C = 1.0$	opt_dist = 2258.0000
$\alpha = 2$	$\beta = 2$	$\rho = 0.2$	$C = 3.0$	opt_dist = 2227.0000
$\alpha = 2$	$\beta = 5$	$\rho = 0.2$	$C = 1.0$	opt_dist = 2146.0000
$\alpha = 2$	$\beta = 5$	$\rho = 0.2$	$C = 3.0$	opt_dist = 2258.0000

在其它参数确定的情况下, 信息素浓度初值 C 的设置对算法性能具有较大影响, 这可以解释为虽然蚂蚁算法具有较强的并行优化计算特性, 但它的优化结果依然无法摆脱对初值的敏感。

(3) 城市数目 $N = 29$, 蚂蚁数量 $M = 20$, $C = 1.0$, 迭代次数为 $NC = 1000$, 参数 α, β, ρ 一定, 考察参数 Q 对结果的影响(表 3)。

表 3 参数 Q 对结果的影响分析

参数 α, β, ρ			参数 Q	最优解
$\alpha = 2$	$\beta = 2$	$\rho = 0.2$	$Q = 1000$	opt_dist = 2258.0000
$\alpha = 2$	$\beta = 2$	$\rho = 0.2$	$Q = 5000$	opt_dist = 2293.0000
$\alpha = 2$	$\beta = 5$	$\rho = 0.2$	$Q = 1000$	opt_dist = 2146.0000
$\alpha = 2$	$\beta = 5$	$\rho = 0.2$	$Q = 5000$	opt_dist = 2146.0000

一般来说,总信息量 Q 越大,则在蚂蚁已经走过的路径上信息激素的累积越快,可以加强蚂蚁搜索时的正反馈作用,有助于算法的快速收敛。由于在蚂蚁算法中各个参数的作用实际上是紧密联系的,其中对算法性能起着主要作用的是信息激素启发因子 α 自启发量因子 β 和信息激素挥发系数 ρ 这三个参数。总信息量 Q 对算法性能的影响有赖于上述三个参数的选取。

(4) 城市数目 $N = 29$, $Q = 1000$, $C = 1.0$, 迭代次数为 $NC = 1000$, 参数 α, β, ρ 一定,考察参数 M 对结果的影响(表 4)。

表 4 参数 M 对结果的影响分析

参数 α, β, ρ			参数 M	最优解
$\alpha = 2$	$\beta = 2$	$\rho = 0.2$	$M = 15$	opt_dist = 2258.0000
$\alpha = 2$	$\beta = 2$	$\rho = 0.2$	$M = 25$	opt_dist = 2146.0000
$\alpha = 2$	$\beta = 5$	$\rho = 0.2$	$M = 15$	opt_dist = 2258.0000
$\alpha = 2$	$\beta = 5$	$\rho = 0.2$	$M = 25$	opt_dist = 2134.0000

显然,当蚂蚁数量比较多时,其收敛的结果比较好。需要指出的是,上述分析是在确定其它参数的前提下,调整其中某一参数来考察算法的收敛效果和性能的。如何针对参数的组合变化情况对算法的性能进行

分析,目前仍没有明确的结论和理论指导。

参考文献:

- [1] 陈文兰,戴树贵. 旅行商问题算法研究综述[J]. 滁州学院学报, 2006, 8(3): 1-6.
- [2] 邹鹏,周智,江贺,等. 求解旅行商问题的循环局部搜索算法的运行时间和性能分布分析[J]. 计算机学报, 2006, 29(1): 92-99.
- [3] 王茂芝,郭科,徐文哲,等. 蚂蚁算法求解 TSP 问题的性能分析及改进[J]. 成都理工大学学报: 自然科学版, 2009, 21(3): 98-102.
- [4] 廖兴新. 蚂蚁算法在 TSP 问题中的应用与研究[D]. 成都: 四川大学, 2006.
- [5] 张煜东,吴乐南,韦耿. 智能算法求解 TSP 问题的比较[J]. 计算机工程与应用, 2009, 45(11): 11-15.
- [6] 马良,朱刚,宁爱兵. 蚁群优化算法[M]. 北京: 科学出版社, 2008.
- [7] 张良红. 一种改进的蚁群算法—RBFACO 算法求解 TSP 问题[D]. 济南: 山东大学, 2006.
- [8] 夏腾. 改进的蚂蚁算法在 TSP 问题中的研究[D]. 武汉: 中南民族大学, 2008.
- [9] 周康,强小利,同小军,等. 求解 TSP 算法[J]. 计算机工程与应用, 2007, 43(29): 43-49.
- [10] 汪林林. 对“货郎担问题”的研究[J]. 重庆邮电学院学报, 1999(2): 5-8.

Application of Ant Colony Algorithm in TSP Problem Solving

CUI Zhi-yi^{1,2}, WANG Mao-zhi^{1,2}, LIU Guo-tao^{1,2}, ZHU Huai-zhao^{1,2}

(1. Key Lab of Geomathematics of Sichuan Province, College of Information Management, Chengdu University of Technology, Chengdu 610059, China; 2. Key Lab of Geomathematics of Sichuan Province, College of Information Management, Chengdu University of Technology, Chengdu 610059, China)

Abstract: Ant Colony Algorithm is the more effective algorithm in solving the complex problems of large-scale. At the same time, TSP is a classic problem of NP-C and has been widely used in VLSI chip design, network routing, vehicle routing and other fields, so a breakthrough in solving the problem means a large number of NPC problem can be solved, which has important practical value and theoretical significance. The experimental results indicate that Ant Colony Algorithm has the sensitivity for the initial value of parameters, a good initial value of parameters need to build on the basis of a large number of test.

Key words: Ant Colony Algorithm; TSP problem; parameters