蚁群算法的收敛速度分析

黄 翰[1] 郝志峰[1),2)] 吴春国[3] 秦 勇[4]

[1)](华南理工大学计算机科学与工程学院 广州 510640)

[2)](南京大学软件新技术国家重点实验室 南京 210093)

[3)](吉林大学计算机科学与技术学院符号计算与知识工程教育部重点实验室 长春 130012)

[4)](茂名学院信息与网络中心 广东 茂名 525000)

摘 要 蚁群算法(ACO)作为一类新型的机器学习技术,已经广泛用于组合优化问题的求解,同时也应用于工业工程的优化设计.相对于遗传算法(GA),蚁群算法的理论研究在国内外均起步较晚,特别是收敛速度的分析理论是该领域急待解决的第一大公开问题.文中的研究内容主要是针对这一公开问题而开展的.根据蚁群算法的特性,该研究基于吸收态 Markov 过程的数学模型,提出了蚁群算法的收敛速度分析理论.作者给出了估算蚁群算法期望收敛时间的几个理论方法,以分析蚁群算法的收敛速度,并结合著名的 ACS 算法作了具体的案例研究.基于该文提出的收敛速度分析理论,作者还提出 ACO-难和 ACO-易两类问题的界定方法;最后,利用 ACS 算法求解 TSP 问题的实验数据,验证了文中提出的分析结论,得出了初步的算法设计指导原则.

关键词 蚁群算法;吸收态 Markov 过程;期望收敛时间;ACO-难易问题;优化路径

中图法分类号 TP181

The Convergence Speed of Ant Colony Optimization

HUANG Han[1] HAO Zhi-Feng[1),2)] WU Chun-Guo[3] QIN Yong[4]

[1)](*College of Computer Science and Engineering, South China University of Technology, Guangzhou* 510640)

[2)](*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210093)

[3)](*College of Computer Science and Technology, Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun* 130012)

[4)](*Center of Information and Network, Maoming University, Maoming, Guangdong* 525000)

Abstract Ant colony optimization (ACO) which is one of the popular methods in machine learning is used widely to solve combinatorial optimization problems. However, there are few theoretical studies for ACO, compared with the counterparts for genetic algorithm (GA). How to analyze the convergence speed is the first open problem of ACO research. In this paper the first open problem is studied via modeling ACO algorithm as an absorbing Markov process, based on which the theoretical results of convergence speed are presented. The convergence speed of ACO algorithm is analyzed by estimating the expected convergence time. The authors propose the method to estimating the expected convergence time of ACO algorithm, and the approach to judging whether a TSP problem belongs to ACO-easy class or ACO-hard class. Finally, the convergence speed of ant colony system (ACS) is analyzed as an example to demonstrate the effectiveness of the theory proposed in this paper.

Keywords ant colony optimization; absorbing Markov process; expected convergence time; ACO-hard and ACO-easy problems; optimal path

收稿日期:2007-03-04;修改稿收到日期:2007-05-25.本课题得到国家自然科学基金(60433020,10471045,60673023)、南京大学软件新技术国家重点实验室项目基金(200603)、广东省自然科学基金(970472,000463,04020079,05011896)和吉林省科技发展项目基金(20050705-2)资助.黄 翰,男,1980 年生,博士研究生,主要研究方向为进化计算方法的理论基础、进化计算方法的优化设计及其应用.E-mail:bssthh@163.com;hanhuang_scut@hotmail.com.郝志峰,男,1968 年生,博士,教授,博士生导师,主要研究领域为代数学及其应用、组合优化与算法研究、仿生算法的数学基础研究.吴春国,男,1976 年生,博士,主要研究方向为进化计算与基于核的学习算法.秦 勇,男,1970 年生,博士研究生,副教授,主要研究方向为网络并行分布式处理技术与应用.

1 引　言

蚁群算法也称作蚁群优化(Ant Colony Optimization, ACO)，最早是由 Dorigo 及其研究同伴所提出，用于求解诸如旅行商问题(Traveling Salesman Problem, TSP)之类的组合优化问题[1-2]．自然界蚂蚁在其经过的路径上会留下某种生物信息物质(信息素)，该物质会吸引蚁群中的其它成员再次选择该段路径；食物与巢穴之前较短的路径容易积累较多的信息素，因而使得更多的蚂蚁选择走该段路径，最终几乎所有的蚂蚁都集中在最短路径上完成食物的搬运．Dorigo 等从此现象中抽象出路径选择和信息素积累的数学模型；作为蚁群算法的核心；并通过对蚂蚁寻找最短路径的计算机模拟，实现了对 TSP 问题的求解．自此，蚁群算法越来越多地被用于求解其它的组合优化问题，也被推广到工业工程上应用[3]．近年来随着 ACO 的广泛应用，该算法的结构和执行策略也不断地得到改进和丰富，其中最成功的两个蚁群算法模型分别是：蚁群系统(ACS)[4] 和最大最小蚂蚁系统(MMAS)[5]，它们的核心思想对蚁群算法的设计有着深远的影响．

应用研究的发展促进了学者们对蚁群算法理论的研究，主要内容在于算法数学模型、收敛性和收敛速度的分析．Gutjahr 针对一种特殊的蚁群算法(graph-based ant system)建立了概率转换模型，并分析了该算法收敛的条件[6-8]．受此结果的启发，Stützle 和 Dorigo[9] 进一步给出了保证蚁群算法收敛的一般性条件：最优解路径对应信息素的下确界应大于 0，以确保算法至少有一次找到全局最优解．这个结论对于绝大多数蚁群算法的分析都是合适的，不过结论的证明类似于对非启发式随机搜索算法的分析，对算法性能评价以及设计方面的指导意义不明显．Dorigo 等又将蚁群算法的收敛性分为两种类型[3,10]：(1) 值收敛(convergence in value)，即当迭代时间趋于无穷时，蚁群算法至少一次达到最优解；(2) 解收敛(convergence in solution)，即当迭代时间趋于无穷时，蚁群算法找到最优解的概率趋于 1．两种收敛性的证明[3] 都类似于文献[9] 的分析，结论充实了蚁群算法的收敛性理论．

上面主要是基于概率模型的收敛性分析，国内外学者还从随机过程的角度研究了蚁群算法的收敛性．Badr 和 Fahmy[11] 利用随机游走模型(random branching)分析了一种特殊蚁群算法的收敛性．但是由于约束条件较多，其结论难以进行推广．国内 Ke 和 Yang 等[12-13] 则是利用有限 Markov 链模型(Markov chain)作为研究 ACO 收敛性的工具；但是，分析结论只能适用于信息素矩阵状态有限的特殊蚁群算法．

收敛性分析理论仅仅告诉我们蚁群算法存在最终找到全局最优解的可能性[2]，较难用于实际算法性能的对比评价．只有对蚁群算法收敛速度进行分析，才能知道算法求解最优解的计算消耗时间．然而，这方面的研究目前在国内外几乎没有．Dorigo 曾在 2005 年将蚁群算法的收敛速度研究问题列为 ACO 领域的第一大公开问题[10]，并建议学者们尝试对一些简单蚁群算法进行收敛速度的分析，以填补此项研究空白．2006 年的文献[14] 对此公开问题作了首次回应，但是，结论仅仅局限于求解线性函数的二进制单蚁 ACO 算法．

本文主要针对这一公开问题，建立了比以往研究更具通用性的吸收态 Markov 链模型，提出了蚁群的收敛性和收敛速度分析理论．以期望收敛时间作为研究蚁群算法收敛速度的主要指标，我们给出了其估算分析方法，并首次提出了 ACO-难和 ACO-易问题的界定方法．本文又以经典的蚁群算法 ACS 作为案例研究，分析了其收敛速度，并提出了 ACS 能否在多项式时间内求解具体问题的判定条件，最后用 ACS 求解 TSP 问题的试验数据，验证了这一判断条件的正确性．

2 符号说明

先说明本文用到主要符号．

C：s_i 可取点的集合；

$E\gamma$：蚁群算法的期望收敛时间；

$E\mu$：首达最优状态的期望时间；

$J(x_i)$：s_i 与 s_i 邻接点边集；

K：蚁群算法的虚拟蚂蚁数目；

n：TSP 问题的城市数；

$p_{\min}, p_{\max}$：$q^k(t, i^*, j^*)$ 最小、最大值；

$q^k(t, i^*, j^*)$：第 k 只蚂蚁找到 s^* 中边 (i^*, j^*) 的概率；

q_0：不采用模拟退火选择的概率；

R^+：正实数集；

S：优化问题的解空间；

s^*：S 中目标函数值最小的解；

s_i：S 中解 s 的第 i 个点；

s_{bs}: ACO 中的当前最优解;
S_{iter}: ACO 一次迭代找到的解集;
$\boldsymbol{T}(t)$: 第 t 次迭代的信息素矩阵;
x_i: 解 s 的前 i 个点;
$X(t)$: 第 t 次迭代的$\{s_{bs}\}\cup S_{iter}$;
Y_X: $X(t)$所属的状态空间;
Y_T: $\boldsymbol{T}(t)$ 所属的状态空间;
Y: $\xi(t)$所属的状态空间;
Y_X^*: $\forall X^*\in Y_X^*$都包含 s^*;
Y^*: $\forall(X^*, T^*)\in Y^*$有 $X^*\in Y_X^*$;
(i, j): TSP 中点 i 到点 j 的边;
α: 蚁群算法中信息素的权重;
β: 蚁群算法中启发式值的权重;
$\eta_{i,j}$: 边(i, j)对应的启发式值;
η_{min}, η_{max}: 启发式值最小、最大值;
$\lambda(t)$: 蚁群算法在 t 时刻达到最优状态的概率;
θ: 信息素局部更新的挥发系数;
ρ: 信息素全局更新的挥发系数;
τ_{ij}: 边(i, j)对应的信息素;
τ_{min}, τ_{max}: 信息素最小、最大值;
$\xi(t)$: $(X(t), \boldsymbol{T}(t))$;
$\{\xi(t)\}_{t=0}^{+\infty}$: ACO 对应的随机过程.

3 蚁群算法简介

本节将对蚁群算法的框架和基本内容作简介, 首先参考 Dorigo 给出的基本蚁群算法的相关定义[2].

定义 **1**(组合优化模型). 一个组合优化问题模型(S, Ω, f)包含有三方面内容: 离散解空间 S、约束集合 Ω和目标函数 $f: S\to R^+$.

运用蚁群算法求解组合优化问题可以视为: 搜索 $s^*\in S$, 满足 $f(s^*)\leq f(s)$, $\forall s\in S$(以极小化问题为例). 记 s 中的组成元素来源于一个离散集 $C=\{c_1, c_2, \cdots, c_N\}$, $s=\{s_1, s_2, \cdots, s_n\}$, $\forall s_i\in C$, $i=1, 2, \cdots, n$. 蚁群算法的执行过程可以被看作蚂蚁在图 $G=(C, L, \boldsymbol{T})$ 上的随机游走. 离散集 C 为图 G 的结点集; L 为边集刻画了图的连通状况; $\boldsymbol{T}$ 为信息素向量, 其中 τ_{ij} 对应结点 c_i 到 c_j 边(i, j) 上的信息素. 下面是基本蚁群算法的框架[2].

算法 **1**. 基本 ACO 算法.
输入: 一个组合优化问题(S, Ω, f)
输出: 当前最优解 s_{bs}
1. 初始化信息素矩阵 $\boldsymbol{T}$
2. $s_{bs}\leftarrow$NULL(s_{bs} 为当前最优解)
while 算法终止条件未满足 do
3. $S_{iter}\leftarrow\varnothing$
for $j=1, 2, \cdots, K$ do
4. 每只虚拟蚂蚁求解 s
5. if $(f(s)\leq f(s_{bs}))$ or $(s_{bs}=$ NULL$)$ then $s_{bs}\leftarrow s$
6. $S_{iter}\leftarrow S_{iter}\cup\{s\}$
end for
7. 根据 S_{iter}, s_{bs}更新信息素矩阵 $\boldsymbol{T}$
end while

第 1 步初始化信息素 $\tau_{ij}=\tau_0\geq\tau_{min}>0$, 可以设置 $\tau_0=f(s^r)/2$, s^r 是一个随机搜索得到的解. 第 2 步将当前最优解 s_{bs} 变量置空, 求解过程中用于记录当前的最优解. 第 3 步将候选解集 S_{iter} 清空, 用于记录每次迭代虚拟蚂蚁寻得的解. 第 4 步首先为每只虚拟蚂蚁(共 K 只)任意选择一个结点作为起点, 再按式(1)选择下一个目标结点, 直至完成一个可行解的搜索:

$$P(s_{i+1}=c_j\mid T, x_i)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\,\eta_{ij}^{\beta}}{\sum\limits_{c_y\in C\wedge(i,y)\in J(x_i)}\tau_{iy}^{\alpha}(t)\,\eta_{iy}^{\beta}}, & (i, j)\in J(x_i)\\ 0, & \text{其它}\end{cases}\tag{1}$$

其中, $x_i=\{s_1, s_2, \cdots, s_i\}$和 $x_{i+1}=\{s_1, s_2, \cdots, s_i, y\}$, $\forall s_i\in C$, $i=1, 2, \cdots, n$. $J(x_i)$表示 s_i与 s_i邻接点构成的边集, 有些问题如 TSP, 不允许有重复边, 则会将已选中的边排除在外. $\eta_{i,j}$ 为边(i, j)对应的启发式值, $\alpha, \beta>0$为权重参数. 当 $s=x_n$时, 虚拟蚂蚁求解完成. 第 5 步主要记录当前求得的最优解, 第 6 步把新求得的解纳入 S_{iter}. 第 7 步是对信息素向量的更新, 一种方式是用当前最优解 s_{bs}进行更新(全局更新):

$$\forall(i, j)\in s_{bs}: \tau_{ij}\leftarrow(1-\rho)\,\tau_{ij}+z_1(s_{bs})\tag{2}$$

另外一种方式是用本次迭代所求得的解 S_{iter} 进行更新(局部更新):

$$\forall(i, j)\in s\in S_{iter}: \tau_{ij}\leftarrow(1-\theta)\,\tau_{ij}+z_2(s)\tag{3}$$

其它 $\forall(i, j)\notin s\in S_{iter}$ 且$(i, j)\notin s_{bs}$, 则进行 $\tau_{ij}\leftarrow(1-\rho)\,\tau_{ij}$. 更新后对所有的边进行调整:

$$\forall(i, j): \tau_{ij}\leftarrow\max\{\tau_{min}, \tau_{ij}\}\tag{4}$$

挥发因子满足 $0<\theta, \rho<1$, 最低信息素满足 $\tau_{min}>0$; 信息素增量 $z_1(s)$, $z_2(s)$: $S\to R^+$, 满足: 若 s_1 优于 s_2, $z_2(s_1)\geq z_2(s_2)$, 即更优的解将带来更大的信息素增量.

大多数蚁群算法均属于以上的框架, 只是具体算法步骤调整和参数设置有所不同而已.

4　基于吸收态 Markov 过程的蚁群算法数学建模

本节将建立蚁群算法的吸收态 Markov 过程模型. 本质上, 蚁群算法的求解过程是蚂蚁根据信息素和启发式向量进行随机搜索. 启发式向量 η 一般与迭代时间 t 无关, 在算法运行中不作更新; 而信息素向量 τ 则是在每次迭代中进行更新. 根据算法中第 7 步的内容, 第 t 次迭代中的信息素向量状态则是由第 $t-1$ 次迭代中 s_{bs} 和 S_{iter} 所决定. 因此, 蚁群算法的数学模型可以由以下定义描述.

定义 **2**.　$\{\xi(t)\}_{t=0}^{+\infty}$ 称为蚁群算法对应的随机过程, 其中 $\xi(t)=(X(t), \boldsymbol{T}(t))$, $X(t)=\{s_{\text{bs}}(t)\}\cup S_{\text{iter}}(t)$ 和 $\boldsymbol{T}(t)$ 是第 t 次迭代的信息素.

考虑 $\xi(t)=(X(t), \boldsymbol{T}(t))$ 的状态空间 Y, 其中 $X(t)\in Y_X$ 和 $\boldsymbol{T}(t)\in Y_T$, 有以下引理.

引理 **1**.　设蚁群算法对应的随机过程为 $\{\xi(t)\}_{t=0}^{+\infty}$, 满足 $\xi(t)=(X(t), \boldsymbol{T}(t))$, 则 $\{\xi(t)\}_t^{+\infty}$ 具有 Markov 性.

证明.　考虑 $\{\xi(t)\}_{t=0}^{+\infty}$ 为离散时间的随机过程, 因为状态 $(X(t), \boldsymbol{T}(t))$ 只是由 $(X(t-1), T(t-1))$ 所决定, $\{X(0), T(0)\}$ 可以任意选择; 所以, $P\{\xi(t)\in Y' \mid \xi(0), \cdots, \xi(t-1)\}=P\{\xi(t)\in Y' \mid \xi(t-1)\}$, $\forall Y'\subseteq Y$, 即 $\{\xi(t)\}_{t=0}^{+\infty}$ 具有 Markov 性. $\{\xi(t)\}_{t=0}^{+\infty}$ 可以视为一个离散时间的 Markov 过程 (以下简称 Markov 过程).　证毕.

当 Y 为离散状态空间时, $\xi(t)$ 符合 Markov 链的性质. 若 $\forall s\in X(t)$, s 来源于离散集 C, Y_X 是个离散状态空间; 所以, Y 是否为离散状态空间取决于 Y_T 是否为离散状态空间, 取决于信息素向量 $\boldsymbol{T}(t)$ 的取值域. 下面的定理 1 给出了蚁群算法能否用 Markov 链建模的条件.

定理 **1**.　设蚁群算法对应的随机过程为 $\{\xi(t)\}_{t=0}^{+\infty}$, 且满足 $\xi(t)=(X(t), \boldsymbol{T}(t))\in Y$, 其中 $X(t)\in Y_X$ 和 $\boldsymbol{T}(t)\in Y_T$; 若 Y_T 为离散的状态空间, 则 $\{\xi(t)\}_{t=0}^{+\infty}$ 是 Markov 链.

证明.　由引理 1, $\{\xi(t)\}_{t=0}^{+\infty}$ 是 Markov 过程; 因为 Y_X 和 Y_T 均为离散状态空间, 所以 Y 为离散状态空间, $\{\xi(t)\}_{t=0}^{+\infty}$ 是 Markov 链.　证毕.

以往文献研究[12-13] 建立的均为有限状态的 Markov 链, 是满足定理 1 的特例. 下面给出吸收态 Markov 过程的定义, 以进一步刻画蚁群算法的本质, 需要引入最优状态空间的定义.

定义 **3**(最优状态空间).　Y^* 称为最优状态空间, 如果 $Y^*\subseteq Y$ 且对于 $\forall \xi^*=\{X^*, T^*\}\in Y^*$, 至少有一个解 $s^*\in X^*$, 满足 $f(s^*)\leqslant f(s)$, $\forall s\in S$.

由定义 3, 如果蚁群算法达到最优状态空间的任意一个状态, 则蚁群算法至少能找到一次全局最优解. 因为蚁群算法用 s_{bs} 记录当前最优解, 所以只要能一次找到全局最优解, 就能永远保留下来. 换而言之, 最优状态空间是蚁群算法求解过程要达到的目标状态空间. 根据此性质, 定义 4 给出了吸收态 Markov 过程的定义.

定义 **4**(吸收态 Markov 过程).　给定一个 Markov 过程 $\{\xi(t)\}_{t=0}^{+\infty}$ ($\forall \xi(t)\in Y$) 和最优状态空间 $Y^*\subseteq Y$, 若 $\{\xi(t)\}_{t=0}^{+\infty}$ 满足 $P\{\xi(t+1)\notin Y^* \mid \xi(t)\in Y^*\}=0$, 则称 $\{\xi(t)\}_{t=0}^{+\infty}$ 为一个吸收态 Markov 过程.

同理, 若 $\{\xi(t)\}_{t=0}^{+\infty}$ 为一个 Markov 链, 可以给出吸收态 Markov 链的定义[15].

引理 **2**.　设蚁群算法对应的随机过程 $\{\xi(t)\}_{t=0}^{+\infty}$, 若满足 $\xi(t)=(X(t), \boldsymbol{T}(t))\in Y$, 则 $\{\xi(t)\}_{t=0}^{+\infty}$ 是一个吸收态 Markov 过程.

证明.　根据引理 1, $\{\xi(t)\}_t^{+\infty}$ 是一个 Markov 过程. 当 $s_{\text{bs}}(t)$ 为全局最优解时, $\xi(t)$ 属于最优状态空间 Y^*; 根据 ACO 算法第 5 步, $s_{\text{bs}}(t+1)=s_{\text{bs}}(t)$ 成立, 即 $\xi(t+1)\in Y^*$. 因此, $P\{\xi(t+1)\notin Y^* \mid \xi(t)\in Y^*\}=0$, $\{\xi(t)\}_{t=0}^{+\infty}$ 是一个吸收态 Markov 过程.

证毕.

第 5 和第 6 节将基于吸收态 Markov 过程模型给出蚁群算法的收敛性和收敛速度分析理论.

5　蚁群算法的收敛性分析

这一节将给出基于吸收态 Markov 过程的蚁群算法收敛性分析, 作为蚁群算法收敛速度研究的理论准备, 首先需要给出收敛性的定义.

定义 **5**(收敛性).　给定一个吸收态 Markov 过程 $\{\xi(t)\}_{t=0}^{+\infty}$ ($\forall \xi(t)=(X(t), \boldsymbol{T}(t))\in Y$) 和最优状态空间 $Y^*\subseteq Y$, 记 $\lambda(t)=P\{\xi(t)\in Y^*\}$ 表示在 t 时刻达到最优状态的概率, 若 $\lim\limits_{t\to+\infty}\lambda(t)=1$, 则称 $\{\xi(t)\}_{t=0}^{+\infty}$ 收敛.

因为 $X(t)$ 是属于离散状态空间 Y_X, 所以有 $\lambda(t)=\sum\limits_{(v,x)\in Y^*}P(X(t)=v)$.

根据定义 5, 基于蚁群算法的 Markov 过程能收敛, 表示该蚁群算法能在无穷次迭代后最终收敛到最优状态空间, 即找到全局最优解的概率趋于 1. 根

据吸收态 Markov 过程的性质,蚁群算法只要在无穷次迭代中找到一次全局最优解,就能将其保持到永远.定理 2 给出了蚁群算法一般的收敛性判断准则.

定理 2. 给定蚁群算法对应的一个吸收态 Markov 过程$\{\xi(t)\}_{t=0}^{+\infty}(\forall\xi(t)=(X(t),\boldsymbol{T}(t))\in Y)$和最优状态空间$Y^*\subset Y$,若$P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}\geq d(t-1)\geq 0$且$\lim\limits_{t\to+\infty}\prod\limits_{i=0}^{t}[1-d(i)]=0$,则$\lim\limits_{t\to+\infty}\lambda(t)=1$,其中$\lambda(t)=P\{\xi(t)\in Y^*\}$.

证明. 根据全概率公式可得,对 $\forall t=1,2,\cdots$有

$$\lambda(t)=[1-\lambda(t-1)]P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}+\lambda(t-1)P\{\xi(t)\in Y^*\mid\xi(t-1)\in Y^*\}.$$

因为$\{\xi(t)\}_{t=0}^{+\infty}$为吸收态 Markov 过程,所以$P\{\xi(t)\in Y^*\mid\xi(t-1)\in Y^*\}=1$,则有

$$\lambda(t)=[1-\lambda(t-1)]P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}+\lambda(t-1)$$

$$\Rightarrow 1-\lambda(t)=[1-P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}]\cdot[1-\lambda(t-1)]$$

$$\Rightarrow 1-\lambda(t)\leq[1-d(t-1)][1-\lambda(t-1)]=[1-\lambda(0)]\prod_{i=0}^{t-1}[1-d(i)]$$

$$\Rightarrow\lim_{t\to+\infty}\lambda(t)\geq 1-[1-\lambda(0)]\lim_{t\to+\infty}\prod_{i=0}^{t-1}[1-d(i)]\geq 1-[1-\lambda(0)]\lim_{t\to+\infty}\prod_{i=0}^{t}[1-d(i)]$$

$$\Rightarrow\lim_{t\to+\infty}\lambda(t)\geq 1\quad(\text{因为}\lim_{t\to+\infty}\prod_{i=0}^{t}[1-d(i)]=0).$$

因为概率$\lambda(t)\leq 1$,所以$\lim\limits_{t\to+\infty}\lambda(t)=1$. 证毕.

定理 2 是基于吸收态 Markov 过程的蚁群算法收敛性分析结论.当最优解路径对应信息素的下界大于 0 时,定理 2 必成立.因此,Gutjahr 与 Dorigo 等[3,6-10]基于概率模型分析的结论本质上是定理 2 的特例.这一理论可以用于分析 ACS、MMAS 等算法的收敛性.由于类似的研究中外学者已经有较多的讨论,这里不详细给出基于吸收态 Markov 过程的各类蚁群算法收敛性分析案例,本文重点在于对蚁群算法收敛速度的分析.

6 蚁群算法的收敛速度分析

本节将基于吸收态 Markov 过程,提出蚁群算法收敛速度分析理论.主要内容包括:提出计算蚁群算法期望收敛时间的方法以分析其收敛速度;给出蚁群算法难求解或易求解问题的判别方法,进而给出蚁群算法设计的指导原则.

6.1 期望收敛时间

以往学者的研究没有给出蚁群算法收敛速度的严格定义,但是提出的本意是希望研究蚁群算法多快能收敛到全局最优解[2,10],或者可以理解为对蚁群算法期望收敛时间的估算.

定义 6(期望收敛时间). 蚁群算法对应的 Markov 过程$\{\xi(t)\}_{t=0}^{+\infty}(\forall\xi(t)=(X(t),\boldsymbol{T}(t))\in Y)$和最优状态空间$Y^*\subset Y$,若$\gamma$是一个随机非负整数变量满足:当$t\geq\gamma$时,$P\{\xi(t)\in Y^*\}=1$,且当$0\leq t<\gamma$时,$P\{\xi(t)\in Y^*\}<1$;则称$\gamma$称为蚁群算法的收敛时间,$\gamma$的期望$E\gamma$称为蚁群算法的期望收敛时间.

期望收敛时间刻画了蚁群算法首次以概率 1 达到全局最优解的期望时间.收敛时间越小表示蚁群算法的收敛速度越快,效率越高;反之,蚁群算法的收敛速度越慢.因为蚁群算法对应的随机过程满足吸收态 Markov 性,由定义 4,我们可使用首达最优解期望时间(Expected First Hitting Time,EFHT)作为研究蚁群算法收敛速度的指标.EFHT 并非本文首次给出定义,文献[15]在研究进化算法期望时间复杂度估算时已经给出该定义,这里给出类似的定义.

定义 7(首达最优解期望时间). 给定一个吸收态 Markov 过程$\{\xi(t)\}_{t=0}^{+\infty}(\forall\xi(t)\in Y)$和最优状态空间$Y^*\subset Y$;若$\mu$是一个随机变量满足:当$t=\mu$时,$\xi(t)\in Y^*$,当$0\leq t<\mu$时,$\xi(t)\notin Y^*$,则称$\mu$的期望$E\mu$为首达最优解期望时间.

引理 3. 蚁群算法的收敛时间γ等于首达最优解时间μ.

证明. 当$t=\mu$时,$\xi(t)\in Y^*$;因为$\{\xi^a\}_{t=0}^{+\infty}$是吸收态过程,则$P\{\xi(\mu+1)\notin Y^*\mid\xi(\mu)\in Y^*\}=0$.又因为$P\{\xi(\mu)\in Y^*\}=1$,所以$P\{\xi(\mu+1)\in Y^*\}=1$;同理可证,当$t>\mu$时,$P\{\xi(t)\in Y^*\}=1$.根据定义 7,当$t<\mu$时,$P\{\xi(t)\in Y^*\}<1$.根据定义 6,有$\gamma=\mu$. 证毕.

因此,可以通过计算$E\mu$来得到蚁群算法的期望收敛时间,定理 3 给出了计算$E\gamma$的最直接方法.

定理 3. 给定蚁群算法对应的吸收态 Markov 过程$\{\xi(t)\}_{t=0}^{+\infty}(\forall\xi(t)\in Y)$和最优状态空间$Y^*\subset Y$,$\lambda(t)=P\{\xi(t)\in Y^*\}$且$\lim\limits_{t\to+\infty}\lambda(t)=1$;其期望收敛时间为$E\gamma=\sum\limits_{i=0}^{+\infty}(1-\lambda(i))$.

证明. 根据定义 4 和定义 5,因为$\xi(t)$是一个

收敛的吸收态 Markov 过程, 假设 μ 为首达最优解的时间, 对于 $\forall t=1, 2, \cdots$有

$$\lambda(t)=P\{\xi(t)\in Y^*\}=P\{\mu\leq t\}$$
$$\Rightarrow\lambda(t)-\lambda(t-1)=P\{\mu\leq t\}-P\{\mu\leq t-1\}$$
$$\Rightarrow P\{\mu=t\}=\lambda(t)-\lambda(t-1),$$

则
$$E\mu=0\cdot P\{\mu=0\}+\sum_{t=1}^{+\infty}t\cdot P\{\mu=t\}$$
$$=\sum_{t=1}^{+\infty}t\cdot(\lambda(t)-\lambda(t-1))$$
$$=\sum_{i=1}^{+\infty}\sum_{t=i}^{+\infty}(\lambda(t)-\lambda(t-1))$$
$$=\sum_{i=0}^{+\infty}[\lim_{t\to+\infty}\lambda(t)-\lambda(i)]$$
$$=\sum_{i=0}^{+\infty}(1-\lambda(i)).$$

根据引理 3, $E\gamma=E\mu=\sum_{i=0}^{+\infty}(1-\lambda(i))$ 成立.　证毕.

6.2　期望收敛时间的估算方法

根据定理 3, 符合基本 ACO 框架的蚁群算法, 如果对于 $\forall t=1, 2, \cdots$, $\lambda(t)$ 或 $P\{\mu=t\}=\lambda(t)-\lambda(t-1)$ 是已知的, 则可以准确地计算出该蚁群算法的期望收敛时间, 从而可以衡量算法的收敛速度. 很可惜, 在实际应用中, 对 $\forall t=1, 2, \cdots$, $\lambda(t)$ 是很难确定的. 下面通过定理 4 给出一种稍微简单的方法, 就是对期望收敛时间上下界的估算, 首先需引入一个引理.

引理 4.　给定两个离散随机非负整数变量 u 和 v, 令 $D_u(\cdot)$ 和 $D_v(\cdot)$ 为 u 和 v 的分布函数, 当 $D_u(t)\geq D_v(t)$ 对 $\forall t=0, 1, 2, \cdots$成立, 则 u 和 v 的期望满足 $Eu\leq Ev$.

证明.　因为 $D_u(t)=P\{u\leq t\}$ 和 $D_v(t)=P\{v\leq t\}$ $(\forall t=0, 1, 2, \cdots)$, 所以,

$$Eu=0\cdot D_u(0)+\sum_{t=1}^{+\infty}t[D_u(t)-D_u(t-1)]=$$
$$\sum_{i=1}^{+\infty}\sum_{t=1}^{+\infty}[D_u(t)-D_u(t-1)]=\sum_{i=0}^{+\infty}[1-D_u(i)],$$

则由已知

$$Eu-Ev=\sum_{i=0}^{+\infty}[1-D_u(i)]-\sum_{i=0}^{+\infty}[1-D_v(i)]=$$
$$\sum_{i=0}^{+\infty}[D_v(i)-D_u(i)]\leq 0\Rightarrow Eu\leq Ev.$$

证毕.

引理 3 反映了一个直观的事实: 在 t 越小时事件发生的概率越大 $(D_u(t)\geq D_v(t))$, 则事件发生的期望时刻越小 $(Eu\leq Ev)$. 类似的引理曾在文献[15]用来辅助分析进化算法的 EFHT. 下面定理 4 给出了蚁群算法期望收敛时间上下界估算的方法.

定理 4.　给定蚁群算法对应的吸收态 Markov 过程 $\{\xi(t)\}_{t=0}^{+\infty}$ $(\forall\xi(t)=(X(t), \boldsymbol{T}(t))\in Y)$ 和最优状态空间 $Y^*\subset Y$, 若 $\lambda(t)=P\{\xi(t)\in Y^*\}$ 满足 $0\leq D_l(t)\leq\lambda(t)\leq D_h(t)\leq 1$ $(\forall t=0, 1, 2, \cdots)$ 且 $\lim_{t\to+\infty}\lambda(t)=1$; 则

$$\sum_{t=0}^{+\infty}(1-D_h(t))\leq E\gamma\leq\sum_{t=0}^{+\infty}(1-D_l(t)).$$

证明.　构造两个离散随机非负整数变量 h 和 l, 其分布函数分别为 $D_h(t)$ 和 $D_l(t)$; 显然, 定义 7 中的 μ 也是一个离散随机非负整数变量, 其分布函数为 $\lambda(t)=P\{\xi(t)\in Y^*\}=P\{\mu\leq t\}$.

因为 $0\leq D_l(t)\leq\lambda(t)\leq D_h(t)\leq 1$, 根据引理 3 和引理 4, 有

$$Eh\leq E\mu\leq El\Leftrightarrow\sum_{t=0}^{+\infty}(1-D_h(t))\leq E\gamma=$$
$$E\mu\leq\sum_{t=0}^{+\infty}(1-D_l(t)).$$　证毕.

定理 4 说明了如果能够知道 $\lambda(t)$ 的上下界, 则期望收敛时间的上下界也能确定. 虽然 $\lambda(t)$ 在实际中很难知道; 但是, $\lambda(t)$ 的上下界相对而言较容易估算得出的, 从而可以得到蚁群算法期望收敛时间至多或者至少的情况. 关键地, 蚁群算法收敛速度分析的精确程度取决于: 能否为 $\lambda(t)$ 找到一个足够小的上界和足够大的下界. 因此, 下面对 $\lambda(t)$ 进行具体化的分析, 以求得更加精确的结果.

考虑 $\lambda(t)=P\{\xi(t)\in Y^*\}=P\{\mu\leq t\}$ 较难直接分析, 下面的定理 5 对 $\lambda(t)$ 进行具体分析, 给出关于期望收敛时间 $E\gamma$ 的实用估算方法.

定理 5.　给定蚁群算法对应的吸收态 Markov 过程 $\{\xi(t)\}_{t}^{+\infty}$ $(\forall\xi(t)=(X(t), \boldsymbol{T}(t))\in Y)$ 和最优状态空间 $Y^*\subset Y$, 若 $\lambda(t)=P\{\xi(t)\in Y^*\}$ 满足 $a(t)\leq P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}\leq b(t)$, 则

$$\sum_{t=0}^{+\infty}[(1-\lambda(0))\prod_{i=1}^{t}(1-b(t))]\leq E\gamma\leq$$
$$\sum_{t=0}^{+\infty}[(1-\lambda(0))\prod_{i=1}^{t}(1-a(t))].$$

证明.　因为 $\lambda(t)=[1-\lambda(t-1)]P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}+\lambda(t-1)P\{\xi(t)\in Y^*\mid\xi(t-1)\in Y^*\}$ $(\forall t=0, 1, 2, \cdots)$, 所以,

$$1-\lambda(t)\leq[1-a(t)][1-\lambda(t-1)]=$$
$$[1-\lambda(0)]\prod_{i=1}^{t}[1-a(i)].$$

根据定理 3,

$$E\gamma=\sum_{i=0}^{+\infty}(1-\lambda(i))\leq\sum_{t=0}^{+\infty}[(1-\lambda(0))\prod_{i=1}^{t}(1-a(t))].$$

同理可得,

$$E\gamma=\sum_{i=0}^{+\infty}(1-\lambda(i))\geq\sum_{t=0}^{+\infty}\left[(1-\lambda(0))\prod_{i=1}^{t}(1-b(t))\right].$$

证毕.

定理 5 的结论与文献[15]的结论有些相似,不过各自分析侧重点不同:文献[15]的结论以 $\sum_{y\notin Y^*}P\{\xi(t)\in Y^*\mid\xi_t=y\}P\{\xi(t-1)=y\}$ 为分析依据;而定理 5 以 $P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}=\sum_{y\notin Y^*}P\{\xi_t\in Y^*\mid\xi_{t-1}=y\}$ 为分析依据.因为 $P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}$ 比 $\sum_{y\notin Y^*}P\{\xi_{t+1}\in Y^*\mid\xi_t=y\}P\{\xi_t=y\}$ 更易于在实际分析中得到,因此,本文提出的定理 5 是将文献[15]结论应用于蚁群算法分析的改进版本.

定理 5 通过 $P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}$ 得到对 $\lambda(t)$ 和 $E\gamma$ 的分析,与定理 3 和定理 4 相比,更容易用于实际分析.因为 $P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}$ 刻画了一次迭代从非最优状态到最优状态的变化概率,根据算法的具体设置,这个概率的取值范围是比较容易估算的.$P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}$ 取值范围的估算精确度将直接影响到对 $E\gamma$ 的精确程度.高精度的取值估算有利于对蚁群算法收敛速度的精确衡量,然而,高精度的取值估算往往都是需要高难度的分析.下面是定理 5 的一个较实用的推论.

推论 **1**. 给定蚁群算法对应的吸收态 Markov 过程 $\{\xi(t)\}_{t=0}^{+\infty}(\forall\xi(t)=(X(t),\boldsymbol{T}(t))\in Y)$,最优状态空间 $Y^*\subset Y$ 和 $\lambda(t)=P\{\xi(t)\in Y^*\}$,如果满足 $a\leq P\{\xi(t)\in Y^*\mid\xi(t-1)\notin Y^*\}\leq b(a,b>0)$ 且 $\lim_{t\to+\infty}\lambda(t)=1$,则蚁群算法的期望收敛时间 $E\gamma$ 满足 $b^{-1}[1-\lambda(0)]\leq E\gamma\leq a^{-1}[1-\lambda(0)]$.

证明. 由定理 5 可得,

$$E\gamma\leq[1-\lambda(0)]\left[a+\sum_{t=2}^{+\infty}ta\prod_{i=0}^{t-2}(1-a)\right]$$

$$\Rightarrow E\gamma\leq[1-\lambda(0)]\left[a+\sum_{t=2}^{+\infty}ta(1-a)^{t-1}\right]$$

$$\Rightarrow E\gamma\leq a[1-\lambda(0)]\left[\sum_{t=0}^{+\infty}t(1-a)^t+\sum_{t=0}^{+\infty}(1-a)^t\right]$$

$$\Rightarrow E\gamma\leq a[1-\lambda(0)]\left(\frac{1-a}{a^2}+\frac{1}{a}\right)=\frac{1}{a}[1-\lambda(0)].$$

同理可得 $E\gamma\geq b^{-1}[1-\lambda(0)]$,则 $b^{-1}[1-\lambda(0)]\leq E\gamma\leq a^{-1}[1-\lambda(0)]$ 成立. 证毕.

推论 1 给出了当概率上下界与时间 t 无关时的一个结论,同时给出了蚁群算法参数优化设计的方法.若下界 a 与蚁群算法的参数有关,则可以通过参数设计,实现 $E\gamma\leq a^{-1}[1-\lambda(0)]\leq P(n)$,其中 $P(n)$ 为关于 n 的多项式.通过参数设计提高蚁群算法收敛速度的研究分析将在以后的工作中进一步开展.

因此,有了估算蚁群算法期望收敛时间的方法,就可以得出相关的算法设计指导原则(推论 1 的结果),同时可以衡量蚁群算法是否能有效求解一个问题(6.3 节).

6.3 ACO-难与 ACO-易问题

原则上,凡是如定义 1 所描述的问题都可以用蚁群算法进行求解;然而,对问题的求解效果如何是我们最关心的.以往研究主要是通过实验数据对蚁群算法求解问题的效果进行评价,至今仍未有理论上的衡量方法.根据定理 4,我们可以给出属于某个蚁群算法易问题和难问题界定的理论方法.蚁群算法的难问题(ACO-难问题)是一类蚁群算法难以求得全局最优解的问题集合;ACO-易问题则是蚁群算法较容易求得全局最优解的问题集合.定义如下.

定义 **8**(ACO-易问题). 给定一个蚁群算法 A 和一个组合优化问题 $\boldsymbol{P}$(规模为 n),如果 A 求解 $\boldsymbol{P}$ 的期望收敛时间至多是关于 n 的多项式,则称 $\boldsymbol{P}$ 为 A 的一个 ACO-易问题.

定义 **9**(ACO-难问题). 给定一个蚁群算法 A 和一个组合优化问题 $\boldsymbol{P}$(规模为 n),如果 A 求解 $\boldsymbol{P}$ 的期望收敛时间至少是关于 n 的指数式,则称 $\boldsymbol{P}$ 为 A 的一个 ACO-难问题.

由定义 8 和定义 9,A 的 ACO-难问题是一类至少要花指数时间去求得最优解的问题,无法求得最优解的问题也属于这一类;A 的 ACO-易问题则是一类至多要花多项式时间去求解的问题,即 A 能有效求解此类问题.可以根据定理 3 直接做判断,但实际分析难度很大.

以上两个定义给出了判断 ACO-难和 ACO-易问题的一种理论方法.设 $P(n)$ 为多项式,$G(n)$ 为指数式,γ 和 μ 分别为蚁群算法的期望收敛时间和 EFHT,如果 $P(n)\leq E\gamma=E\mu\leq G(n)$,则需要继续寻找更精确的上下界,才能作具体的界定.结合定理 5 和推论 1 还可以给出更加具体的判别方法,关于 ACO-难和 ACO-易问题的具体分析将在以后做进一步研究,这里只是给出了一个简单的框架和方法.第 7 节将以 ACS 为例,分析该算法的难解和易解 TSP 问题.

7　研究案例：ACS 算法的收敛速度分析

ACS 全称为 Ant Colony System，有学者译为蚁群系统，由 Dorigo 等[4]所提出．以往研究文献[2-4]的实验数据显示：ACS 比其它蚁群算法有着更强的全局搜索能力．ACS 符合基本 ACO 算法框架，主要特征在 3 个方面：(1) 采用模拟退火式选择，即以 $1-q_0$ 的概率按式(1)选择邻接点，以 q_0 的概率选择 $\tau^{\alpha}(t)\eta^{\beta}$ 值最大的邻接点，q_0 一般取 0.9；(2) 采用全局更新策略如式(2)，采用局部更新策略如式(3)；(3) 不采用式(4)且不对所有的边进行 $\tau\leftarrow(1-\rho)\tau$ 更新．

7.1　ACS 算法的期望收敛时间

本节将用定理 5 和推论 1 对 ACS 算法的收敛速度进行分析，结论见定理 6．

定理 6.　ACS 算法的期望收敛时间 $E\gamma^{acs}$ 满足

$$\frac{1}{1-[1-(p_{max})^n]^K}\leq E\gamma^{acs}\leq\frac{1}{1-[1-(p_{min})^n]^K},$$

其中，$0\leq p_{min}\leq p_{max}\leq 1$．

证明．　假设 s^* 为 ACS 算法求解问题的全局最优解，记第 t 次迭代中第 k 只蚂蚁选择边 $\forall(i^*,j^*)\in s^*$ 的概率为 $q^k(t,i^*,j^*)$．根据 ACS 的特征之一：以 $1-q_0$ 的概率按式(1)选择邻接点，以 q_0 的概率选择 $\tau^{\alpha}(t)\eta^{\beta}$ 值最大的邻接点．$q^k(t,i^*,j^*)$ 满足

$$p_{min}\leq q^k(t,i^*,j^*)\leq p_{max}\qquad(5)$$

其中 p_{min} 和 p_{max} 为选择 $\forall(i^*,j^*)\in s^*$ 的概率下界和上界(根据式(1)可得)：

$$\frac{(1-q_0)\tau_{min}^{\alpha}\eta_{min}^{\beta}}{(n-1)\tau_{max}^{\alpha}\eta_{max}^{\beta}+\tau_{min}^{\alpha}\eta_{min}^{\beta}}=p_{min}\leq p_{max}=$$

$$q_0+\frac{(1-q_0)\tau_{max}^{\alpha}\eta_{max}^{\beta}}{(n-1)\tau_{min}^{\alpha}\eta_{min}^{\beta}+\tau_{max}^{\alpha}\eta_{max}^{\beta}}\qquad(6)$$

根据文献[9]关于信息素上下界的结论，$0<\tau_{min}^{\alpha}\leq\tau_{max}^{\alpha}<+\infty$；根据 ACS 算法的特性，$\eta_{min}$ 和 η_{max} 是最长和最短边的倒数，所以 $0<\eta_{min}^{\alpha}\leq\eta_{max}^{\alpha}<+\infty$．

因为 ACS 算法把当前最优解 s_{bs} 保存，用于全局更新(见式(2))，所以一旦 ACS 找到了全局最优解，将永远保留在 s_{bs} 中．因此，ACS 算法的状态 $\{\xi^{acs}(t)\}_{t=0}^{+\infty}$ 是一个吸收态 Markov 过程．假设 ACS 算法采用了 K 只人工蚂蚁，则

$$P\{\xi^{acs}(t)\in Y^*\mid\xi^{acs}(t-1)\notin Y^*\}=$$

$$1-\prod_{k=1}^{K}[1-(q^k(t,c^*))^n]\quad(\forall t=0,1,2,\cdots).$$

由式(5)和(6)，

$$1-[1-(p_{min})^n]^K\leq P\{\xi^{acs}(t)\in Y^*\mid\xi^{acs}(t-1)\notin Y^*\}\leq 1-[1-(p_{max})^n]^K\qquad(7)$$

取 $a=1-[1-(p_{min})^n]^K$ 和 $b=1-[1-(p_{max})^n]^K$，因为 ACS 收敛[9]，即 $\lim_{t\to\infty}\lambda^{acs}(t)=\lim_{t\to\infty}P\{\xi^{acs}(t)\in Y^*\}=1$，由推论 1 可得，$b^{-1}[1-\lambda(0)]\leq E\gamma^{acs}\leq a^{-1}[1-\lambda(0)]$．

因为在初始化过程中，$X(t)=s_{bs}(t)\cup S_{iter}(t)=\varnothing$；所以，$\lambda(0)=0$．因此，$E\gamma^{acs}\leq\frac{1}{1-[1-(p_{min})^n]^K}$．同理可得 $E\gamma^{acs}\geq\frac{1}{1-[1-(p_{max})^n]^K}$．　证毕．

定理 6 给出了对 ACS 算法的收敛速度的一般性分析，由此可以进一步得到 7.2 节的结论．

7.2　ACS-易问题

根据定理 6，可以给出 ACS-易问题的判定条件，第 8 节将提供相应的实验数据为验证．

推论 2.　若 ACS 算法的期望收敛时间 $E\gamma$ 至多为 $P(n)$，则必有：当 $1-q_0-g_1>0$ 时，$\tau_{min}^{\alpha}\geq A$；当 $1-q_0-g_1<0$ 时，$\tau_{min}^{\alpha}\leq A$($K$ 为蚂蚁数目，n 为 TSP 问题的城市数目，$g_1=(1-(1-P(n)^{-1})^{1/K})^{1/n}$，$A=g_1(n-1)\tau_{max}^{\alpha}\eta_{max}^{\beta}\eta_{min}^{-\beta}(1-q_0-g_1)^{-1}$)．

证明．　当 $E\gamma\leq[1-(1-(p_{min})^n)^K]^{-1}\leq P(n)$，$P(n)$ 为关于 n 的多项式时，有 $[1-(1-(p_{min})^n)^K]^{-1}\leq P(n)\Leftrightarrow p_{min}\geq(1-(1-P(n)^{-1})^{1/K})^{1/n}$．

假设 $g_1=(1-(1-P(n)^{-1})^{1/K})^{1/n}$，根据不等式(6)，可得

当 $1-q_0-g_1>0$ 时，

$$\tau_{min}^{\alpha}\geq g_1(n-1)\tau_{max}^{\alpha}\eta_{max}^{\beta}\eta_{min}^{-\beta}(1-q_0-g_1)^{-1}\qquad(8)$$

当 $1-q_0-g_1<0$ 时，

$$\tau_{min}^{\alpha}\leq g_1(n-1)\tau_{max}^{\alpha}\eta_{max}^{\beta}\eta_{min}^{-\beta}(1-q_0-g_1)^{-1}\qquad(9)$$

因此，若 ACS 可以在多项式时间内求解 TSP 问题，则有：(i) 当 $1-q_0-g_1>0$ 时，式(8)成立；(ii) 当 $1-q_0-g_1<0$ 时，式(9)成立．　证毕．

推论 2 表明：若(i)或(ii)不成立，则意味着 ACS 算法不可能在多项式时间 $P(n)$ 内求解该 TSP 问题．因为参数 α，β，η_{max}^{β}，$\eta_{min}^{-\beta}$ 和 q_0 是可知的，τ_{min}^{α} 和 τ_{max}^{α} 由问题的最优解和最差解决定[9]；所以，我们可以根据(i)和(ii)判断某个 TSP 问题是否属于 ACS-易问题．第 8 节将用实验数据说明这一判断方法的有效性．

8　数值实验

本节将以 ACS 算法求解 TSP 问题的数据，验

证第 7 节所得出理论分析结果. 仿真实验采取的 ACS 算法参数设置[3-4] 如下: $\alpha=1$, $\beta=2$, $q_0=0.9$ 和 $\rho=\theta=0.1$. η_{max}, η_{min} 为 TSP 问题中两点间最小和最大距离的倒数. 根据更新公式(2), 取 $z_1(s^*)=\rho\cdot L(s^*)^{-1}$, 其中 $L(s^*)$ 为最优解 s^* 的长度. 由文献[9]的结果, $\tau_{max}=L(s^*)^{-1}$; 实验考察的已知最优解长度 $L(s^*)$ 的 TSP 问题, 所以 τ_{max} 可以确定. 取 $P(n)=n^2$, n 为城市数, 假设 $A=g_1(n-1)\tau_{max}^{\alpha}\eta_{max}^{\beta}\eta_{min}^{-\beta}(1-q_0-g_1)^{-1}$.

如果 ACS 可以在多项式时间 $P(n)=n^2$ 内求得 TSP 问题的最优解, 根据推论 2, 7.2 节的(i)和(ii)必成立. 在实验中, ACS 算法对每个 TSP 分别计算了 20 次, 每次运行在达到最大迭代次数时停止, 实验数据如表 1 所示.

表 1 ACS 算法求解 TSP 问题的实验数据观察

TSP 问题	$1-q_0-g_1$	A	ACS 最优路径长度	ACS 最大迭代次数	TSP 问题最优路径长度
kroA100	<0	<0	21285.44	100×100	21282
kroE100	<0	<0	22078.66	100×100	22068
berlin52	<0	<0	7544.36	52×52	7542
kroB150	<0	<0	26127.35	150×150	26130
ch150	<0	<0	6530.90	150×150	6528
bier127	<0	<0	118773.00	127×127	118282
d198	<0	<0	15888.00	198×198	15780
ts225	<0	<0	128829.00	225×225	126643

如表 1 数据所示, 对于所有的 TSP 问题, 实验得到的数据都满足 $1-q_0-g_1<0$ 且 $A<0$. 因此, 如果 ACS 能在多项式 $P(n)=n^2$ 时间内求解这些 TSP 问题, 则由推论 2, 当 $1-q_0-g_1<0$ 时, 必有 $\tau_{min}^{\alpha}\leq A$. 因为 $A<0$, 所以 $\tau_{min}^{\alpha}<0$; 这与 ACS 算法的参数 $\tau_{min}>0$ 且 $\alpha=1$ 矛盾. 因此, 假设不成立, ACS 不能在多项式 $P(n)=n^2$ 迭代时间内求解这些 TSP 问题. 通过平均 20 次对 8 个 TSP 问题的求解, ACS 算法在 n^2 次迭代时间内求得最优路径长度均未能达到实际问题的最优解长度. 实验数据验证了定理 6 和推论 2 理论分析的正确性. 文献[4]也正是因为 ACS 未能达到全局最优解的不足, 在算法中加入了局部搜索等改进策略, 最后才得到比单纯 ACS 算法更佳的效果. 关于辅助策略对 ACO 收敛速度的影响将在以后的研究中做进一步讨论.

9 结束语

本文针对机器学习领域中的公开难题"蚁群算法收敛速度分析", 提出了蚁群算法的吸收态 Markov 过程模型、收敛速度分析理论、ACO 难易问题的界定方法、蚁群参数优化设置理论方法、经典 ACS 算法收敛速度分析的案例研究和相应的仿真实验数据.

本文建立的吸收态 Markov 过程数学模型与以往蚁群算法的 Markov 链模型相比, 有着更加广泛的适用性. 基于吸收态 Markov 过程模型, 本文以期望收敛时间作为研究指标提出了蚁群算法收敛速度的分析理论; 根据吸收态的性质给出了期望收敛时间的估算方法, 理论上实现了对蚁群算法收敛速度的衡量, 继而提出了 ACO-易和 ACO-难问题的界定方法. 本文还给出了蚁群算法参数的优化设计理论方法, 以确保算法能在多项式时间内求解 ACO-易问题. 我们又以经典 ACS 算法作为分析案例, 分析了算法参数与迭代时间的关系, 提出了 ACS 算法易求解问题的具体判断规则. 最后, 我们用数值实验数据验证了这一判断规则的有效性.

基于本文对蚁群算法收敛速度的分析结论, 我们在未来的工作中将集中研究提高蚁群算法收敛速度的参数优化设计方法以及对比分析蚁群算法性能的理论与方法. 同时, 我们的研究会考虑一次迭代的时间复杂度和 ACO 算法的辅助策略, 从而完善对收敛速度的分析.

致 谢 伯明翰大学计算机科学学院 CERCIA 中心的姚新教授在 SEAL'06 国际学术会议上对本研究提出过方向性的指导意见,《计算机学报》本期"机器学习与数据挖掘"专辑匿名审稿专家也对本文理论分析部分提出了宝贵的修改意见. 在此, 本文作者一并表示衷心的感谢!

参 考 文 献

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

[2] Dorigo M, Caro G D, Gambardella L M. Ant algorithms for discrete optimization. Artificial Life, 1999, 5(2): 137-172

[3] Dorigo M, Stützle T. Ant Colony Optimization. Cambridge, MA: MIT Press, 2004

[4] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66

[5] Stützle T, Hoos H H. MAX-MIN ant system. Future Generation Computer Systems, 2000, 16(8): 889-914

[6] Gutjahr W J. A generalized convergence result for the graph-based ant system metaheuristic. Department of Statistics and Decision Support Systems, University of Vienna, Austria: Technical Report 99-09, 1999

[7] Gutjahr W J. Agraph-based ant system and its convergence. Future Generation Computer Systems, 2000, 16(9): 873-888

[8] Gutjahr W J. ACO algorithms with guaranteed convergence to the optimal solution. Information Processing Letters, 2002, 82(3): 145-153

[9] Stützle T, Dorigo M. A short convergence proof for a class of ant colony optimization algorithms. IEEE Transactions on Evolutionary Computation, 2002, 6(4): 358-365

[10] Dorigo M, Blum C. Ant colony optimization theory: A survey. Theoretical Computer Science, 2005, 344(2-3): 243-278

[11] Badr A, Fahmy A. A proof of convergence for Ant algorithms. Information Sciences, 2004, 160(1-4): 267-279

[12] Ke Liang-Jun, Feng Zu-Ren, Feng Yuan-Jing. Ant colony optimization algorithm with finite grade pheromone. Acta Automatica Sinica, 2006, 32(2): 296-303(in Chinese)
(柯良军, 冯祖仁, 冯远静. 有限级信息素蚁群算法. 自动化学报, 2006, 32(2): 296-303)

[13] Yang Wen-Guo, Guo Tian-De. An Ant colony optimization algorithm for the minimum Steiner tree problem and its convergence proof. Acta Mathematicae Applicatae Sinica, 2006, 29(2): 352-361(in Chinese)
(杨文国, 郭田德. 求解最小 Steiner 树的蚁群优化算法及其收敛性. 应用数学学报, 2006, 29(2): 352-361)

[14] Hao Z F, Huang H. A time complexity analysis of ACO for linear functions//Wang Tzai-Der et al. Simulated Evolution and Learning. Lecture Notes in Computer Science 4247. Springer-Verlag, 2006: 513-520

[15] Yu Y, Zhou Z H. A new approach to estimating the expected first hitting time of evolutionary algorithms//Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06). Boston, MA, California, USA, 2006: 555-56

HUANG Han, born in 1980, Ph. D. candidate. His research interests include the fields of theoretical foundation of evolutionary computation methods, evolutionary optimization, and application of evolutionary algorithms.

HAO Zhi-Feng, born in 1968, Ph. D., professor, Ph. D. supervisor. His research interests include the fields of algebra group, analysis and design of algorithms, SAT, combinational optimization and data mining.

WU Chun-Guo, born in 1976, Ph. D.. His research interests include the fields of evolutionary computation and kernel methods.

QIN Yong, born in 1970, Ph. D. candidate, associate professor. His research interests include the fields of load balancing, route selection, routing algorithm and multi-objective optimization.

Background

Ant colony optimization (ACO) is one of the popular methods in machine learning. How to analyze the convergence speed is the first open problem of ACO research. This paper introduces a framework to solve this open problem by proposing the absorbing Markov process model, the method to estimating the expected convergence time of ACO algorithm, and the approach to judging whether a TSP problem belongs to ACO-easy class or ACO-hard class. This work is supported by the National Natural Science Foundation of China (60433020, 10471045, 60673023), Natural Science Foundation of Guangdong Province (970472, 000463, 04020079, 05011896), State Key Laboratory for Novel Software Technology, Nanjing University (200603), open research fund of National Mobile Communications Research Laboratory, Southeast University (A200605) and Nature Science Foundation of Education Department of Guangdong Province (Z03080). The work of the projects is mainly about the theoretical foundation and application of evolutionary computation. The results can be used to analyze the performance of evolutionary computation methods, solve the combinatorial optimization problems and optimize the design of engineering.