

文章编号: 1001-0920(2007)06-0685-04

动态分阶段蚁群算法及其收敛性分析

邢桂华^{1,2}, 于盛林¹

(1. 南京航空航天大学 自动化学院, 南京 210016; 2. 南京师范大学 计算机系, 南京 210097)

摘要: 为了提高蚁群算法的收敛速度和求解精度, 根据仿生优化算法在不同阶段的特点, 提出一种改进的蚁群算法. 该算法对参数和选择策略进行了分阶段设计, 而且参数的分阶段是根据寻优状态动态划分的. 通过对蚁群系统马尔科夫过程进行分析, 证明了该算法的全局收敛性. 针对典型的 TSP 问题进行仿真对比实验, 验证了该算法在速度和精度方面优于传统蚁群算法.

关键词: 蚁群算法; 收敛性; 动态分阶段

中图分类号: TP278 **文献标识码:** A

Dynamic stage ant colony algorithm and its convergence

XING Gui-hua^{1,2}, YU Sheng-lin¹

(1. Department of Automatic Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; 2. Department of Computer Science, Nanjing Normal University, Nanjing 210097, China. Correspondent: XING Gui-hua, E-mail: xingguihua@njnu.edu.cn)

Abstract: According to the stage properties of bionic optimization algorithms, an improved ant colony algorithm is proposed to enhance the speed and accuracy of the original algorithm. In the algorithm, parameters and selection strategy are specially designed in different optimizing stages which are marked referring to the current optimizing states. Its global convergence is proved by analyzing the Markov process of the ant colony system. The contrasting experiments to the typical traveling sales problem prove that the proposed algorithm is advantageous over the traditional ant colony algorithms in speed and accuracy.

Key words: Ant colony algorithm; Convergence; Dynamic stage

1 引言

蚁群优化算法^[1,2]是模拟自然界中真实蚁群的觅食行为而形成的一种模拟进化算法,并在实际优化应用中取得了一定的成功.但蚁群算法本身也存在一些缺陷,如搜索时间较长、容易陷入局部最优解等.

目前对蚁群算法的研究绝大多数着重算法改进和应用,一般仅从实验的角度说明所提出的算法的有效性,很少从理论上研究蚁群算法的优化性能,但在理论研究方面也有成果出现. Gutjahr^[3]分析了一种基于图的蚁群算法的收敛性; Stutzle 等^[4]分析了最大最小蚁群算法的收敛性; 石立宝等^[5]分析了随机摄动蚁群算法的收敛性.

本文在对蚁群算法中参数意义、选择策略及信息素更新等进行研究的基础上,提出了一种动态分

阶段蚁群算法(DSACA),并在文献[4]的基础上,证明了 DSACA 的全局收敛性.

2 DSACA 算法及其收敛性分析

2.1 问题和符号约定

考虑最小优化问题 (S, f, \cdot) , S 为候选解集, f 为目标函数, \cdot 为可行候选解的约束集, 则最小优化问题的目标是找到最优解 s^* 使得目标函数最小. 基于该优化问题, 人工蚂蚁在全连接图 G 上可通过随机的方式构造候选解 $G = (C, L, T)$, 其中 C 为顶点集, L 为 C 的全部连接集, T 为信息素集. 当人工蚂蚁在图 G 上逐顶点行走时, 要满足约束集 \cdot .

为了使算法描述和收敛性分析更清晰, 现将其中所用符号作统一说明: s^* 为所得最优解, s 为任意可行解, s^* 为全局最优解, X 为满足约束条件的可行解集, d 为阶段变量, a_d 为 d 阶段残余信息素重

收稿日期: 2006-03-03; 修回日期: 2006-07-25.

基金项目: 国家自然科学基金项目(10371055, 60673102).

作者简介: 邢桂华(1971—), 女, 黑龙江大庆人, 博士生, 从事智能算法、图像处理的研究; 于盛林(1941—), 男, 南京人, 教授, 博士生导师, 从事信号处理、计算机测控系统等研究.

要度参数, a 为 d 阶段挥发参数, D 为总阶段数, m 为蚂蚁数, N_d 为 d 阶段的总迭代次数, x_k 为已行走 k 步后的顶点集, J_{c_k} 为第 k 步后的可行域, q_0 为选择参数, c_k 为第 k 个顶点, $f_d(s^*)$ 为阶段 d 的最优解的适应函数, $f(s^*)$ 为全局最优估计解的适应函数, $g(s)$ 为 s 对应的信息素增量函数, $g(s^*)$ 为 s^* 对应的信息素增量函数, N_c 为顶点数, n 为序列最大长度, $\tau_{ij}(t)$ 为迭代次数是 t 时边 (i, j) 上的信息素, $\tau_{ij}^*(t^*)$ 为迭代次数是 t^* 发现最优解时属于 s^* 的各连接上的信息素, $\tau_{ij}^{\neq}(t^*)$ 为迭代次数是 t^* 发现最优解时不属于 s^* 的各连接上的信息素, $p^*(t)$ 为迭代次数是 t 时发现最优解的概率, $p_{ij}(t)$ 为迭代次数是 t 时从顶点 i 转移到顶点 j 的概率, $p^{\neq}(t)$ 为迭代次数是 t 时发现最优解的概率下限, p 为找到可行解的概率.

2.2 DSACA 算法描述

Step1 初始化

随机生成 s , 并且设 $\hat{s} = s, f_0(s^*) = f(s)$.
 $\forall (i, j), \tau_{ij} = 0$, 并初始化变量 $d = 1, N_d = 0$.

For $i = 1 : m$

 随机选择出发点 c_1

 则 $x_k = c_1$

当所有的蚂蚁选择完初始位置之后, 初始化完成.

Step2 选择策略

对于每一只蚂蚁, 进行如下过程的线路选择:

While ($x_k \notin \bar{X}$ 且 $x_k \notin S$)

 设目前已完成了第 k 步选择, 故 x_k 更新为 $x_k = c_1, c_2, \dots, c_k$.

If $d < D/2$, 则选择下一个节点 c 的策略为

$$p(c_{k+1} = c | T, x_k) = \begin{cases} \frac{\{\tau_{c_k, c}\}^d}{\sum_{y \in C_{c_k, y}} \{\tau_{c_k, y}\}^d}, & (c_k, c) \in J_{c_k}; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Else

 节点 c 的选择策略为

$$c = \begin{cases} \underset{(c_k, c) \in J_{c_k}}{\operatorname{argmax}} \{\tau_{c_k, c}\}^d, & q < q_0; \\ \text{using (1),} & \text{otherwise.} \end{cases} \quad (2)$$

End

若 $x_{k+1} = c_1, c_2, \dots, c_k, y \in \bar{X}$, 则 c_k, y 属可行域 J_{c_k} ; 若 $\forall y, x_{k+1} \notin \bar{X}$, 则 J_{c_k} 为空, 此时迭代结束.

Step3 信息素更新策略

$$\forall (i, j), \tau_{ij} = (1 - a) \tau_{ij}$$

If $f(s_i) < f(\hat{s})$ then $\hat{s} = s_i$,

$$\forall (i, j) \tau_{ij} = \tau_{ij} + g(\hat{s}),$$

$$(i, j) = \max((i, j) + a g(\hat{s}), \min)$$

$g(s)$ 满足

$$g: S \rightarrow R^+, f(s) < f(s^*) \Rightarrow g(s) = g(s^*).$$

Step4 参数更新策略

$$N_d = N_d + 1.$$

If $N_d = t$ or

$$|f_d(s^*) - f_{d-1}(s^*)| \\ (f(\hat{s}^*) - f_0(s^*)) / D,$$

则 $d = d + 1, a = a_{d+1}, \tau = \tau_{d+1}$, 其中 $t = t_{\max} / D$. 更新时, 按

$$\begin{cases} a = a_{d+1} + (a_d - a_{d+1}) / (D - 1), \\ \tau = \tau_{d+1} + (\tau_d - \tau_{d+1}) / (D - 1) \end{cases}$$

进行.

If $d > D$ or $t > t_{\max}$ 则结束 Else 转 Step2.

从算法描述可以看出, 本文算法是在 $\max\text{-min}^{[6]}$ 算法基础上进行的改进, 改进策略主要在于: 1) 信息素增量: $g(s) = a g(\hat{s})$, 即由固定增量改为与对应阶段挥发度成比例的增量. 2) 参数由恒定到分阶段设计: a, τ , 当前阶段根据算法的求解情况和迭代次数共同决定. 根据参数意义, 随着迭代次数的增加和阶段的刷新, 逐渐增大而逐渐减小. 3) 选择策略因计算阶段而不同, 前面阶段加大搜索的随机性, 有利于搜寻更优解, 后面阶段在已经搜索到较优解的情况下, 减小随机程度, 使其逐步收敛到全局最优解. 这 3 方面的改进策略兼顾了搜寻全局最优解和收敛速度.

2.3 DSACA 收敛性分析

引理 1^[3] 蚁群系统 $\{s(t), s(t), f^*(t)\}$ 为有限齐次马尔科夫过程, 其中 $t = 1, 2, \dots$

定理 1 对于任意 (i, j) , 有 $\lim_t \tau_{ij}(t) = \tau_{ij}^{\max}$.

证明 对于边 (i, j) , 设 $\tau_{ij}(0) = 0$, 每次迭代都能取到最大值 $a g(\hat{s})$, 则经过 t 次迭代后可计算出当前能达到的最大值 $\tau_{ij}^{\max}(t)$. 设当前迭代次数 t 处于第 d_n 个阶段, 则有

$$\tau_{ij}^{\max}(t) = \sum_{d=1}^{d_n} (1 - a)^{d-1} a g(\hat{s}). \quad (3)$$

当 $d = 1$ 时, 有

$$\tau_{ij}^{\max}(1) = (1 - a)^0 a g(\hat{s}). \quad (4)$$

可推知

$$\tau_{ij}^{\max}(N_1) = \sum_{i=0}^{N_1-1} (1 - a)^i a g(\hat{s}). \quad (5)$$

当处于第 d_n 个阶段的迭代次数 t 时,有

$$\begin{aligned} \max_{ij} (t) &= \max_{ij} (N_1 + \dots + N_{d_n}) = \\ &(1 - \alpha_1)^{N_1} \cdot \dots \cdot (1 - \alpha_{d_n})^{N_{d_n}} \cdot 0 + \\ &1(1 - \alpha_2)^{N_2} \cdot \dots \cdot (1 - \alpha_{d_n})^{N_{d_n}} \cdot g(s^*) \\ &\quad \cdot (1 - \alpha_1)^{i-1} + \dots + \\ &\quad \dots + (1 - \alpha_{d_n})^{N_{d_n}} \cdot g(s^*) \cdot (1 - \alpha_{d_n-1})^{i-1} + \\ &\quad \dots + (1 - \alpha_{d_n})^{N_{d_n}} \cdot g(s^*) \cdot (1 - \alpha_{d_n})^{i-1}. \end{aligned} \quad (6)$$

其极限为

$$\lim_t \max_{ij} (t) = \max_{d=2, i=d}^{d_n, d_n} g(s^*) \left(1 + (1 - \alpha_i)^{N_i}\right). \quad (7)$$

故 $\lim_t \max_{ij} (t)$ 得证.

定理 2 对于 $\{f(t), s(t), f^*(t)\} (t = 1, 2, \dots)$, 存在任意小的 $\epsilon > 0$ 和足够大的 t , 使得 $p^*(t) > 1 - \epsilon$ 成立, 且收敛到 $\lim_t p^*(t) = 1, \forall (i, j) \in s^*$, p_{ij}^* 收敛于 \max_{ij} .

证明 因为 $\min_{ij} p_{ij}(t) > 0$, 再由式(1)和(2)可知, $\forall (i, j, t), p_{ij}(t) > 0$. 考虑其最差情况, 有

$$p_{ij}(t) = \frac{p_{\min}}{(N_c - 1)^d_{\max} + \frac{d}{N_c} \frac{d}{\max}} \quad (8)$$

找到任一可行解 s 或最优解 s^* 的概率 $\hat{p} > p_{\min}^n$, 则找到最优解的概率下限为^[4]

$$\hat{p}^*(t) = 1 - (1 - \hat{p})^t. \quad (9)$$

由式(3)有

$$\hat{p}^*(t) = 1 - \prod_{d=1}^{d_n} \left(1 - \frac{1}{N_c} \left(\frac{\min}{\max}\right)^{N_d}\right)^{N_d} = 1 - \prod_{d=1}^{d_n} q_d. \quad (10)$$

从而有 $p^*(t) > \hat{p}^*(t) > 1 - \epsilon$, 则 $\lim_t p^*(t) = 1$ 成立. 当最优解找到后, 由式(2)可知, $p_{ij}^*(t)$ 会收敛到定理 1 中的 \max_{ij} .

定理 3 若 $t = t^*$ 时首次找到最优解, 则存在 t_0 , 满足

$$\begin{aligned} p_{ij}(t) &> \mu(t), \forall (i, j) \in s^*, \\ \forall (k, l) \in L \setminus s^*, \forall t > t^* + t_0. \end{aligned}$$

证明 假设发现最优解时为最差情况, 即有

$$p_{ij}^*(t^*) = \min_{ij}, \mu(t^*) = \max_{kl}.$$

设当 $t = t^*$ 时, $d = d_1$; 当 $t = t^* + t$ 时, $d = d_n$. 此时可推导出

$$p_{ij}(t^* + t)$$

$$g(s^*) \left(1 + \prod_{d=2}^{N_{d_n} - d_1 + 1} (1 - \alpha_i)^{N_i}\right) > g(s^*), \quad (11)$$

$$\mu(t^* + t) = \max_{d=d_1}^{d_n} (1 - \alpha_d)^{N_d}. \quad (12)$$

随着 t 的增大, 存在 d_n 使得

$$(1 - \alpha_{d_1})^{N_{d_1}} < \frac{\min_{ij}}{\max_{kl}}.$$

设此时 $t_0 = t$, 则 $\mu(t^* + t) < \min_{ij}$. 因为 $g(s^*) > \min_{ij}$, 故存在 t_0 , 使 $p_{ij}(t^* + t) > \mu(t^* + t)$ 成立, 从而定理 3 得证.

推论 1 当发现最优解时, 对于 $p_{ij}(t), (i, j) \in s^*$, 有 $\lim_t p_{ij}(t) = \min_{ij}$, 且 t 为有限值 $t = t^* + t_0$ 时, $p_{ij}(t) = \min_{ij}$ 成立.

证明

$$p_{ij}(t) = \max_{d=d_1}^{d_n} \left(\min_{ij}, \max_{d=d_1}^{d_n} (1 - \alpha_d)^{N_d}\right),$$

当 $t \rightarrow \infty, \max_{d=d_1}^{d_n} (1 - \alpha_d)^{N_d} \rightarrow 0$, 则有 $\lim_t p_{ij}(t) = \min_{ij}$. 由于 $p_{ij}(t)$ 单调递减, 设达到 $p_{ij}(t) = \min_{ij}$ 时 $t = t^* + t_0$, 从而推论 1 得证.

3 实验结果

选择 Oliver30, 在 Matlab6.5 中进行实验. 实验中最优解如图 1 所示, 其最佳路径长度随迭代次数的变化过程如图 2 中实线所示.

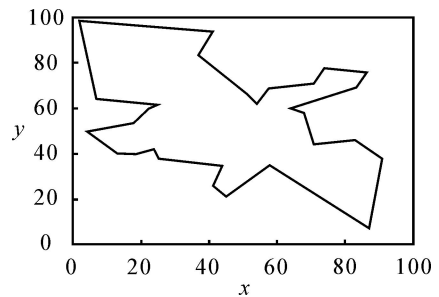


图 1 DSACA 找到的最优解

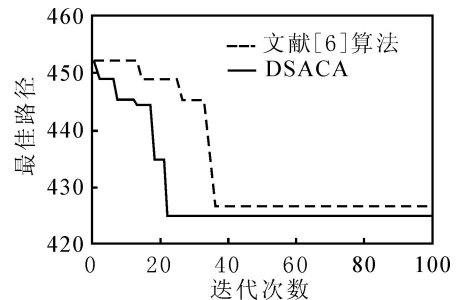


图 2 DSACA 与文献[6]算法的最佳路径变化过程比较

DSACA 在不同的参数下的结果如表 1 所示. 为了与改进前的算法相比较, 在同样的参数条件下

使用文献[6]算法对 Oliver30 进行实验,其最优解对应的最佳路径长度的变化过程如图2中虚线所示.可以看出 DSACA 提高了求解的速度和精度. DSACA 与基本蚁群算法的比较结果如表2所示,可以看出,相对基本蚁群算法而言,本文算法的优势更加明显.

表1 DSACA 在不同参数下的结果比较

α	β	ρ	δ	最优解	20次平均值
1.0	1.50	0.9	0.5	425.910 2	433.513 9
1.0	1.50	0.7	0.3	433.780 3	431.521 8
0.9	0.98	0.7	0.3	426.425 4	431.259 2
0.9	0.98	0.9	0.5	424.869 3	427.655 8

表2 DSACA 与基本蚁群算法的比较

算法	最优解	迭代次数	20次平均值
ACS	427.667 4	316	435.365 7
AQS	435.838 2	367	438.231 7
DSACA	424.869 3	22	427.655 8

4 结 语

仿生优化算法有明显的阶段特点,一般在进化的前期增强其寻找更优解的能力,而在进化的后期使其逐步收敛到最优解.在蚁群算法中,各个参数有明确的意义,所以本文在不同的优化阶段对参数进行分阶段设计,同时也对选择策略进行分阶段设计,形成了动态分阶段的蚁群算法,且对该算法的全局收敛性进行证明.针对典型的 Oliver30 进行实验,

实验中通过数据对比说明了本文算法的有效性.

参考文献(References)

- [1] Dorigo M, Maniezzo V, Colormi A. Ant system: Optimization by a colony cooperating agents [J]. IEEE Trans on Systems, Man and Cybernetics, 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] Walter J, Gutjahr. A graph-based ant system and its convergence[J]. Future Generation Computer Systems, 2000, 16(8): 873-888.
- [4] Stutzle T, Dorigo M. A short convergence proof for a class of ant colony optimization algorithm [J]. IEEE Trans on Evolutionary Computations, 2002, 6(4): 358-365.
- [5] 石立宝,郝晋. 随机摄动蚁群算法的收敛性及其数值特性分析[J]. 系统仿真学报, 2004, 16(11): 2421-2424.
(Shi L B, Hao J. The numerical characteristics analysis and convergence proof for ant colony optimization algorithm with random perturbation behavior [J]. J of System Simulation, 2004, 16(11): 2421-2424.)
- [6] Stutzle T, Hoos H. Max-min ant system [J]. Future Generation Computer Systems, 2000, 16(8): 889-914.

(上接第 684 页)

6 结 语

针对 CAN 网下的 NCS,利用 CAN 网的广播特性,设计了一种基于网络用户层的动态闭环调度策略,采用此调度策略的 NCS 为一个变采样周期系统.文中提出了变采样周期系统的建模与鲁棒控制器的设计方法,最后通过仿真对所提出的调度策略和控制方法进行了验证.仿真结果表明,动态闭环调度策略具有良好的性能,所提出的变采样周期系统的建模与控制方法是有效的,从而实现了 NCS 中控制与调度的综合设计,为 NCS 的实际工程应用提供了理论技术支持.

参考文献(References)

- [1] Branicky M S, Philips S M, Zhang W. Scheduling and feedback co-design for networked control systems[C]. IEEE Conf on Decision and Control. Las Vegas: IEEE Press, 2002: 10-13.

- [2] Yong Ho Kim, Hong Seong Park, Wook Hyun Kwon. A scheduling method for network-based control systems [C]. American Control Conf. Washington DC: IEEE Press, 1998: 718-722.
- [3] Otanez P, Moyne J, Tilbury D. Using deadbands to reduce communication in networked control systems [C]. American Control Conf. Alaska: IEEE Press, 2002: 3015-3020.
- [4] Manel Velasco, Josep M Fuertes, Caixue Lin. A control approach to bandwidth management in networked control systems[C]. The 30th Annual Conf of the IEEE Industrial Electronics Society. Busan, 2004: 782-786.
- [5] 俞立. 鲁棒控制——线性矩阵不等式处理方法[M]. 北京:清华大学出版社, 2002: 96-119.
(Yu L. Robust control — Linear matrix inequality methods [M]. Beijing: Tsinghua University Press, 2002: 96-119.)