

文章编号: 1001-0920(2004)10-1187-03

## 利用信息量留存的蚁群遗传算法

邵晓巍, 邵长胜, 赵长安

(哈尔滨工业大学 航天学院, 黑龙江 哈尔滨 150001)

**摘 要:** 提出一种结合蚁群算法中“信息量留存”思想的遗传算法。该算法将问题空间进行均匀分割, 基于这些子空间选取初始种群, 并定义每个子空间的初始信息量, 遗传操作中根据信息量留存情况来控制个体选择。由于初始种群均匀地分散在解空间, 降低了发生过早收敛的可能性; 而采用蚁群算法中“信息量留存”的思想, 可保证算法快速收敛到具有最优(次优)解的子空间, 从而达到提高收敛速度的目的。

**关键词:** 混合遗传算法; 蚁群算法; 信息量留存

**中图分类号:** TP18      **文献标识码:** A

## Ant colony genetic algorithm using pheromone remaining

SHAO Xiao-wei, SHAO Chang-sheng, ZHAO Chang-an

(School of Astronautics, Harbin Institute of Technology, Harbin 150001, China Correspondent: SHAO Xiao-wei, Email: xw-shao@hit.edu.cn)

**Abstract:** A hybrid genetic algorithm called ant colony genetic algorithm is presented. The initial population of the method is generated from each subspace which is divided from the feasible solution space of the optimization problem evenly. And each subspace is marked by initial pheromone. During the genetic operation, selection operator is under the effect of subspace's pheromone remaining. Because of the initial population of points that are scattered uniformly over the feasible solution space, the algorithm can evenly scan the feasible solution space once to locate good points for further exploration in subsequent operator. The pheromone of each subspace in the selection can improve the speed of convergence of the algorithm.

**Key words:** hybrid genetic algorithm; ant colony system; pheromone remaining

### 1 引 言

遗传算法存在过早收敛和局部寻优能力差等问题。结合其他优化算法形成混合遗传算法(HGA)是克服这些缺点的有效手段。文献[1]将SA和模糊集方法融入GA,以解决短期调度问题;[2]利用NEH算法得到的解作为GA,以求解flow shop问题的初始解;[3]将单纯形搜索嵌入GA对Metabolic系统建模,克服了GA对模型评价的复杂性;[4]提出一种结合模拟退火法的GA;[5]将正交设计融入GA初始种群和交叉算子,对解决多维问题具有良好的效果。

蚁群算法(ACA)<sup>[6]</sup>可用于求解TSP问题、分配问题、job-shop调度问题以及着色问题。Wang等将蚁群算法引入连续空间优化问题求解<sup>[7]</sup>,为蚁群算法的应用提供了更广阔的空间。

本文结合蚁群算法“信息量留存”的思想,用信息量标定分解后的问题子空间,使搜索快速定位到含有最优解的子空间,从而提高了收敛速度。由于初始种群取自均匀分解的子空间,使得算法可在解空间中均匀地搜索最优点,具有一定的克服过早收敛的效果。

收稿日期: 2003-11-24; 修回日期: 2004-02-25

作者简介: 邵晓巍(1974—),男,安徽肖山人,博士生,从事智能决策、数据建模等研究;赵长安(1936—),男,黑龙江哈尔滨人,教授,博士生导师,从事鲁棒控制、最优控制等研究。

## 2 蚁群遗传算法

全局优化问题定义如下:

$$\begin{cases} \text{Maximize } f(x), \\ \text{Subject to } l \leq x \leq u. \end{cases} \quad (1)$$

其中:  $x = \{x_1, x_2, \dots, x_n\}$ , 变量  $x_i$  的定义域为  $[l_i, u_i]$ ;  $l = \{l_1, l_2, \dots, l_n\}$ ;  $u = \{u_1, u_2, \dots, u_n\}$ ;  $f(x)$  是目标函数;  $n$  为维数

在解决优化问题之前, 通常不会有全局最优点在解空间位置分布的信息, 因此希望算法的搜索种群能够均匀地分散在解空间, 这有利于算法在后续操作中较均匀地扫描解空间. 蚁群遗传算法首先将解空间均匀地分解为若干个子区域, 分别从这些子区域中产生初始种群, 然后采用信息量标定每一个子区域, 在各个子区域信息量变化的约束下进行遗传操作搜索整个解空间; 最后在结束条件控制下停止搜索

蚁群遗传算法的关键步骤如下:

第 1 步: 将解空间按一定的原则分解成子空间

针对全局最优问题的维数和每一维定义域的大小对解空间加以分解. 为简便起见, 这里以二维优化问题为例. 设  $x = \{x_1, x_2\}$ , 其定义域分别为  $[l_1, u_1]$  和  $[l_2, u_2]$ , 定义域均匀地分解为  $M \times N$  个子空间  $E_{ij}$ ,  $i = 1, 2, \dots, M$ ,  $j = 1, 2, \dots, N$ , 且子区域的区间长度分别为

$$D_{1L} = \frac{u_1 - l_1}{M}, D_{2L} = \frac{u_2 - l_2}{N}. \quad (2)$$

$E_{ij}$  的左右边界分别为  $x_{1L}, x_{2L}$  和  $x_{1R}, x_{2R}$ , 其中

$$\begin{aligned} x_{1L} &= l_1 + (i - 1)D_{1L}, x_{2L} = l_2 + (j - 1)D_{2L}; \\ x_{1R} &= l_1 + iD_{1L}, x_{2R} = l_2 + jD_{2L}. \end{aligned}$$

第 2 步: 确定初始种群并标定各个子空间

1) 初始种群的产生: 在第 1 步确定的每个子空间中随机产生一个个体, 所有个体组成初始种群. 与子空间  $E_{ij}$  相对应的个体定义为  $A_{ij}(1)$ , 括号中的 1 表示是整个算法的第 1 代个体, 即初始种群. 则初始种群可表示为  $\{A_{ij}(1)\}$ ,  $i = 1, 2, \dots, M$ ,  $j = 1, 2, \dots, N$ , 种群规模为  $M \times N$ .

2) 子空间的初始标定: 每个子空间  $E_{ij}$  都由一个信息量  $Ph_{ij}$  标定, 子空间的信息量标定为蚁群遗传算法遗传操作的改进提供了必要条件. 各个子空间信息量的初始值由其中产生的初始个体的适应度值确定. 当  $f(A_{ij}(1)) > 0$  时, 定义  $Ph_{ij}(1) = C_1 f(A_{ij}(1))$ ,  $C_1$  为根据问题而设定的正常数; 当  $f(A_{ij}(1)) < 0$  时, 定义  $Ph_{ij}(1) = \frac{C_3}{C_2 - f(A_{ij}(1))}$ ,

$f(A_{ij})$  为个体  $A_{ij}$  的适应度值.  $C_2$  和  $C_3$  的设定同  $C_1$ .

第 3 步: 对种群进行蚁群遗传操作

1) 选择操作: 选择操作的主要目的是为了避免基因缺失, 提高全局收敛性和计算效率. 蚁群遗传算法选择操作的基本思想是: 第  $k$  代中的个体  $l$  被选中的概率是其个体适应度值和所处子空间的信息量的函数. 群体规模为  $M \times N$ , 第  $k$  代中个体  $l$  的适应度值为  $f(A_l(k))$ , 个体  $l$  所处子空间的上一代中标定的信息量为  $Ph_l(k - 1)$ , 则个体  $l$  被选中的概率为

$$P_l(k) = \frac{Ph_l^\alpha(k - 1) f^\beta(A_l(k))}{\sum_{i=1}^{M \times N} Ph_i^\alpha(k - 1) f^\beta(A_i(k))}. \quad (3)$$

其中:  $\alpha$  为个体所处子区域残留信息量的相对重要程度,  $\beta$  为个体适应度值的相对重要程度

2) 交叉操作: 交叉操作是蚁群遗传算法中产生新个体的主要方法, 可以针对具体问题, 根据编码方法的不同选择各种常用交叉操作和交叉概率

3) 变异操作: 变异操作是产生新个体的辅助方法, 同时也决定了蚁群遗传算法的局部搜索能力. 与交叉算法相似, 可根据具体问题选取具体的变异方法和变异概率

4) 子空间信息量更新: 随着种群一代代进化, 各子空间的信息量也不断积累, 在积累过程中必须对残留的信息量进行更新处理. 本文模仿人类记忆的特点, 对旧的信息进行削弱, 将新的信息量不断加入. 根据进化代数, 各个子空间的信息量积累过程如下:

根据第 2 步确定第 1 代中各子空间的信息量

$$Ph_{ij}(k) = \rho Ph_{ij}(k - 1) + Ph_{ij}(k). \quad (4)$$

其中:  $Ph_{ij}(k)$  为第  $k$  代子空间  $ij$  上的信息量,  $Ph_{ij}(k - 1)$  为第  $k - 1$  代子空间  $ij$  上的信息量,  $Ph_{ij}(k)$  为第  $k$  代遗传操作后子空间  $ij$  上加入的信息量,  $\rho$  为残留信息保留系数. 不妨设在子空间  $ij$  上, 第  $k$  代以前具有最大适应度值的个体为  $A_{ijmax}$ , 第  $k$  代选择、交叉和变异操作后, 具有最大适应度值的个体为  $A_{ijmax}(k)$ . 如果  $f(A_{ijmax}(k)) > f(A_{ijmax})$ , 则  $A_{ijmax} = A_{ijmax}(k)$ ; 否则,  $A_{ijmax}$  不变. 当  $f(A_{ijmax}) > 0$  时,  $Ph_{ij}(k) = C_1 f(A_{ijmax})$ ; 当  $f(A_{ijmax}) < 0$  时, 定义  $Ph_{ij}(k) = \frac{C_3}{C_2 - f(A_{ijmax})}$ ,  $C_1, C_2, C_3$  为根据问题而设定的正常数, 且  $C_2 > C_3$ ,  $f(A_{ijmax})$  为个体  $A_{ijmax}$  的适应度值. 如果  $k$  代中某一子空间没有个体, 则  $A_{ijmax}$  保持不变.

子空间  $ij$  中具有最大适应度值个体  $A_{ijmax}$  是随

着蚁群遗传操作一代代保留下来的, 所以不必每一代都比较  $ij$  中所有个体的适应度值, 只需同该代中交叉和变异产生的新个体的适应度值进行比较即可. 结合各子空间初始信息量值, 利用式(4) 可保证各子空间的信息量随遗传进化不断积累和更新

第 4 步: 当群体中的最优个体满足一定要求或总代数达到一定数量时, 结束进化操作

借鉴遗传算法和蚁群算法中参数的取值范围, 根据具体问题对蚁群遗传算法参数加以选取. 一般情况取值如下: 群体大小: 20 ~ 100; 交叉概率: 0.4 ~ 0.9; 变异概率: 0.0001 ~ 0.1; 中止代数: 100 ~ 500;  $\alpha$ : 0.05 ~ 1;  $\beta$ : 1 ~ 4;  $C_1, C_2, C_3$ : 0.5 ~ 4;  $\rho$ : 0.01 ~ 0.8

### 3 实例仿真

算法的收敛速度是衡量单峰值函数优化方法的最重要指标. 对于多峰值函数优化问题, 不但要求算法应有快速的收敛性, 而且要求能找到全局最优解, 避免陷入局部极值点. 下面用文献[8] 给出的函数和 Schaffer 的  $F_6$  函数, 分别采用蚁群遗传算法和简单遗传算法进行仿真

#### 3.1 Michalewicz 函数

$$f(x_1, x_2) = 21.5 + x_1 \sin(47\pi x_1) + x_2 \sin(20\pi x_2).$$

自变量范围:  $-3.0 < x_1 < 12.1, 4.1 < x_2 < 5.8$ . 函数包含多个局部最优点, 因此容易出现过早收敛现象.

本文采用二进制编码形式, 要求精确到小数点后 5 位,  $M = 10, N = 5$ , 故初始种群为 50; 采用单点交叉, 交叉概率  $P_c = 0.4$ ; 采用简单变异, 变异概率  $P_m = 0.1$ ; 选择操作中,  $\Delta = 0.1, E = 2$ ; 信息量更新时,  $\rho = 0.2, C_1 = 1$ ; 遗传终止代数 500.

为使仿真具有可比性, 简单遗传算法也采用二进制编码形式, 初始种群取 50; 单点交叉,  $P_c = 0.4$ ; 简单变异,  $P_m = 0.1$ ; 终止代数 500.

#### 3.2 Schaffer 的 $F_6$ 函数

$$f(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2}$$

自变量范围:  $-4 < x, y < 4$ . 函数有无数个局部极大点, 其中只有一个  $(0, 0)$  为全局最大, 最大值为 1. 此函数的最大值峰周围有一圈脊, 它们的取值均为 0.99028, 因此很容易停滞在此局部极大点.

同样采用二进制编码形式, 要求精确到小数点后 5 位,  $M = 8, N = 8$ , 故初始种群为 64; 采用单点交叉, 交叉概率  $P_c = 0.4$ ; 采用简单变异, 变异概率  $P_m = 0.08$ ; 选择操作中,  $\Delta = 0.1, E = 2$ ; 信息量更新时,  $\rho = 0.1, C_1 = 1$ ; 终止代数 500.

简单遗传算法也采用二进制编码形式, 初始种群取 64; 单点交叉,  $P_c = 0.5$ ; 简单变异,  $P_m = 0.06$ ; 终止代数 500.

经过 100 次重复仿真, 统计结果如表 1 所示.

表 1 测试函数仿真结果

算 法	Michalewicz 函数		Schaffer 的 $F_6$ 函数	
	平均收敛代数	过早收敛次数	平均收敛代数	过早收敛次数
蚁群遗传算法	141	7	153	11
简单遗传算法	207	21	245	34

仿真结果表明, 与简单遗传算法相比, 本文提出的蚁群遗传算法不论是收敛速度还是抗早熟特性都有明显的优势. 测试的两个函数都包含很多局部最优点, 在搜索过程中很容易出现过早收敛现象. 从仿真结果可以看出, 蚁群遗传算法能较好地处理这类多局部最优解问题.

## 4 结 论

遗传算法作为一种自适应全局优化概率搜索算法, 在最优化问题的求解中取得了良好的效果. 蚁群算法作为一类模拟生物群体突现聚集行为的非经典算法, 已成为近年来研究的热点. 本文将蚁群算法融入遗传算法, 利用“信息量”作为算法收敛过程中的正反馈信息, 并利用交叉和变异算子进行全局和局部搜索. 对 Michalewicz 函数和 Schaffer 的  $F_6$  函数进行仿真和分析, 表明蚁群遗传算法由于初始种群均匀地分散在解空间, 能较好地克服过早收敛现象; 同时由于对子空间引入标定信息量, 并将其融入选择算子, 提高了算法的收敛速度.

### 参考文献(References):

- [1] Wong K P, Wong Y W. Combined genetic algorithm/simulated annealing/fuzzy set approach to short-term generation scheduling with take-or-pay fuel contract [J]. *IEEE Trans on Power Systems*, 1996, 11(1): 128-136.
- [2] Reeves C R. A genetic algorithm for flow shop sequencing [J]. *Computers and Operations Research*, 1995, 22(1): 5-13.

(下转第 1193 页)

识精度有了明显的改善

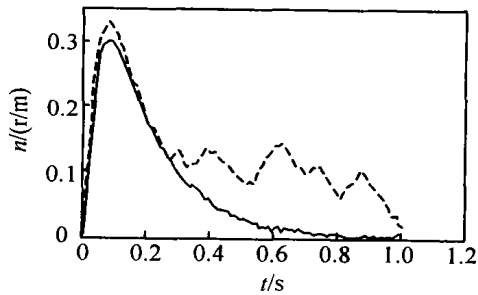


图 2 本文方法与相关分析法的对比

为反映尺度变换对提高估计精度的功效, 尺度变换前后的误差曲线如图 3 所示. 经计算, 尺度变换前平均估计误差  $\bar{e}_2 = 0.0059$ , 尺度变换后平均误差  $\bar{e}_3 = 0.0022$ . 由此可见, 尺度变换能有效地抑制干扰的影响, 提高输出信号的信噪比, 从而进一步降低估计误差, 更好地逼近系统真实的脉冲响应过程

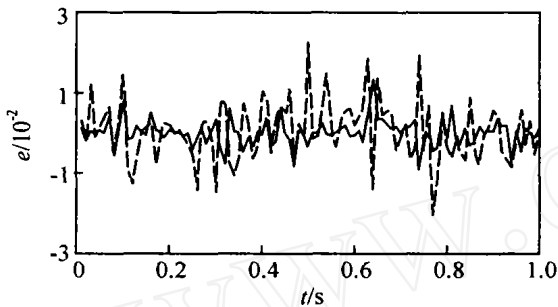


图 3 尺度变换前后辨识误差对比

实验结果验证了该算法的正确性和实用性

## 5 结 语

本文提出用紧支非正交小波作为基函数逼近系统脉冲响应过程, 并证明了经正交尺度变换后系统干扰将得到有效的抑制, 从而可提高系统的辨识精度. 理论分析和实验结果验证了算法的正确性和实用性. 小波基具有良好的时频联合局部性, 其辨识结果具有明确的频域物理意义, 并能有效地跟踪时变系统的参数变化. 该算法如能结合递推算法(如递推最小二乘法、随机梯度算法等), 便可应用于时变系统的过程估计. 因此, 该算法在一些复杂的非参数辨识领域(如工业生产过程)具有良好的应用前景

## 参考文献(References):

- [1] 冯培悌. 系统辨识[M]. 杭州: 浙江大学出版社, 1999. 43-53
- [2] Eykhoff D. *System Identification-parameter and State Estimation* [M]. New York: John Wiley & Sons Inc, 1974. 157-182
- [3] Zhang Qinghua. Using wavelet network in nonparametric estimation [J]. *IEEE Trans on Neural Network*, 1997, 8(2): 227-236
- [4] Zhang J, Walter G, Mial Y. Wavelet neural networks for function learning [J]. *IEEE Trans on Signal Processing*, 1995, 43(6): 1485-1497.
- [5] Daubechies I. The wavelet transform, time-frequency localization and signal analysis [J]. *IEEE Trans on Information Theory*, 1990, 36(5): 961-1005.
- [6] 崔锦泰. 小波分析导论[M]. 西安: 西安交通大学出版社, 1995. 2-29

(上接第 1189 页)

- [3] Yen J, Liao J C, Bogju L, et al. A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method [J]. *IEEE Trans on Systems, Man and Cybernetics*, 1998, 28(2): 173-191.
- [4] Li Shouzhi, Li Minyan, Pan Yongxiang. Genetic annealing algorithms and its convergence analysis [J]. *Control Theory and Applications*, 2002, 19(3): 376-380.
- [5] Yiu-wing Leung, Yiping Wang. An orthogonal genetic algorithm with quantization for global numerical optimization [J]. *IEEE Trans on Evolutionary Computation*, 2001, 5(1): 41-53.
- [6] Coloni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies [A]. *Proc 1st European Conf on Artificial Life[C]*. Paris: Elsevier, 1991. 134-142.
- [7] Wang Lei, Wu Qidi. Further example study on ant system algorithm based continuous space optimization [A]. *Proc of the World Congress on Intelligent Control and Automation[C]*. Shanghai, 2002. 3: 2541-2545.
- [8] Michalewicz Z. *Genetic Algorithm + Data Structure = Evolution Programs* [M]. 2nd ed. New York: Springer-Verlag, 1994.