

# 一种多群竞争进化规划算法

王向军, 嵇 斗, 张 民

(海军工程大学 电气工程系, 湖北武汉 430033)

**摘 要:** 易早熟收敛是传统进化规划算法的致命缺点. 本文在分析了导致进化规划算法早熟原因的基础上, 提出了一种改进的多群进化规划算法. 在该算法中, 进化在多个不同的子群间并行进行, 子群间的竞争决定个体的变异能力, 子群间的信息交流通过子群重组实现. 对典型算例的数值仿真表明, 该算法能够改善传统的进化规划算法易早熟收敛的弱点, 同时具有良好的快速收敛性和参数鲁棒性.

**关键词:** 进化规划; 竞争; 性能

**中图分类号:** TP18      **文献标识码:** A      **文章编号:** 0372-2112 (2004) 12-1824-05

## A Mult2Subgroup Competition Evolutionary Programming Algorithm

WANG Xiang2jun, JI Dou, ZHANG Min

(Dept. of Electrical Engineering, Naval Univ. of Engineering, Wuhan, Hubei 430033, China)

**Abstract:** Premature convergence is the fatal shortcoming of traditional evolutionary programming. Based on the analysis of premature convergence of traditional evolutionary programming, a novel Mult2subgroup Competition Evolutionary Programming (MCEP) algorithm is proposed. In the algorithm, evolutions of subgroups are parallelly performed with the different mutation strategies. The mutation capability of an individual is determined by the competition among subgroups. The individual, together with information, is exchanged while subgroup is reorganized. The simulations based on benchmarks confirm that this algorithm is better than classical evolutionary programming algorithm in the aspects of global optimization, convergence speed and the robustness.

**Key words:** evolutionary programming; competition; performance

## 1 引言

进化计算借鉴生物界自然选择法则、遗传机制, 模拟生物进化过程以解决函数优化、神经网络结构优化、模式识别与系统辨识等问题<sup>[1-3]</sup>. 进化规划(Evolutionary programming, EP)是进化计算中的代表算法之一, 但大量的研究证明, 该算法存在一些亟待克服的弱点<sup>[4-6]</sup>: (1) 进化容易出现过早收敛, 从而陷入局部极值点, 即早熟现象; (2) 进化后期, 个体之间的竞争趋缓导致算法后期的搜索效率降低; (3) 对初始参数敏感. 本文针对传统进化规划算法的弱点, 提出了一种多群竞争进化规划算法(Mult2subgroup competition evolutionary programming, MCEP), 在该算法中, 进化在多个平行的子群间同时进行, 而且进化不仅取决于适应度函数的大小, 同时也考虑了不同的子群间竞争和交流的结果. 子群间的竞争决定了不同子群的变异能力, 竞争获胜的子群使用小的变异率, 来保持自己的优势, 竞争失败子群使用大的变异率来力求改变自己的生存条件. 多个使用或大或小变异的子群, 促使整个种群能够以尽量分散的变异尺度来对解空间进行更加详尽的探索. 子群重组导致个体在子群间的迁移, 而这种迁移则是子群间信息交换的表现形式. 对典型算例的仿真证明, MCEP 算法在全局收

敛性、收敛速度和参数鲁棒性上均具有良好的性能.

## 2 传统的 EP 算法

传统 EP 算法中, 参与进化的只有一个种群, 因此在这种种群中的个体只能遵循唯一的规律进化(相同的高斯变异算子标准差  $R_0$ ). 较大的高斯变异算子标准差可以保证种群具有较好的探索能力, 即种群中的部分个体能够以比较大的概率落在全局最优解所在的邻域, 在每一个局部极值都具有很好的局部逃逸能力, 但此种探索(局部逃逸)能力的取得是以降低解的精度为代价的, 种群能够到达全局最优解的附近, 但无法使用小的变异以足够高的精度逼近最优解. 较小的高斯变异算子标准差能够保证种群在局部具有良好的搜索能力, 即能够以高的精度找到种群所在局部的极值, 但小的变异算子不能(或者需要很多的进化次数才能)产生足够大的变异从而使种群从这个局部进化到具有更大极值(或者全局最优解)的局部, 从而早熟收敛. 由于我们无法预知函数局部极值间的距离, 因此无法选取合适的  $R_0$ , 使其既能大到足够使个体产生大的变异从局部极值点逃逸, 又能小到使个体产生小的变异, 在全局最优解附近以高的精度逼近最优解.

近年来, 一些学者提出了很多方法来改善 EP 算法的性

能<sup>[7~8]</sup>. 文献[7]提出了一种基于自适应变异规则的有效进化规划算法(Adaptive mutation based evolutionary programming, AEP), 在该算法中通过进化的快慢来判断种群是在有效进化还是陷入了局部极值, 当陷入局部极值时使用大的变异算子促进种群从局部极值点附近脱离. 该算法的变异规则具体表述如下:

如果  $d_k - d_{k-1} \geq a$  (1)

则  $x_i^{(k+1)} = d_k + (d_k - d_{k-1}) \# N(0, 1)$  (2)

$age(k+1) = 1$  (3)

如果  $x_{\max}^{(k)} - x_{\min}^{(k)} \geq b$  (4)

则  $x_i^{(k+1)} = x_i^{(k)} + (x_{\max}^{(k)} - x_{\min}^{(k)}) \# N(0, 1)$  (5)

$age(k+1) = 1$  (6)

否则  $x_i^{(k+1)} = x_i^{(k)} + age(j) \# c \# f(x_i^{(k)}) \# N(0, 1)$  (7)

其中:  $d_k = \sum_{i=1}^N x_i^{(k)} / N$ ,  $X^{(k)}$  为第  $K$  代种群,  $N$  为种群包含的个体数目, 并有  $x_i^{(k)} \in X^{(k)}$ ,  $a$  和  $b$  为常数, 值可取 1,  $c$  为修正常数, 一般取 1. 该算法根据种群进化的快慢来调节变异尺度的大小的方法是有效的, 但种群在全局最优解附近的收敛速度同样是比较慢的, 使用大的变异算子将使种群不能以足够高的精度逼近全局最优解. 文献[8]提出了一种改进的进化规划算法(Improved evolutionary programming, IEP), 该算法中使用递减(或适当调小)的变异算子来以高的精度求解, 即算法迭带超过 10 次以后:

$$R_0 = 0.02R_0 \quad (8)$$

但该算法不能保证在进化的前期, 变异算子为合适的大, 能够在进化的前段就使种群从局部极值的邻域变异到全局最优解的邻域. 此两种算法对初始种群及高斯变异算子标准差的初值  $R_0$  敏感, 一旦初始种群的性能不够好(即初始种群中没有任何一个个体落在全局最优解的邻域), 同时使用了过小的的高斯变异算子标准差, 将导致算法以很高的概率收敛于局部极值, 或者需要很多的进化代数才可以收敛于全局最优解.

### 3 MCEP 算法流程

将整个种群按照对环境的适应能力划分为多个子群. 进化在多个子群间并行进行, 子群之间通过竞争决定其变异能力, 对环境适应能力强的子群使用小的变异, 对环境适应能力弱的子群使用大的变异. 子群间的个体交流与信息交流通过子群重组来实现. 就好像有多个镜头的显微镜, 同时使用多个不同粗细的镜头对目标观察, 粗镜头用来在大范围内对目标进行大略的搜索, 而用细镜头对粗镜头发现的感兴趣部位进行详细的观察, 而粗镜头发现的目标转入细镜头的过程就是一个信息传递的过程. 整个算法的流程如下:

(1) 参数初始化. 设置环境容纳的个体数目  $N$ , 设置子群数目  $M$  (其中  $N$  为  $M$  的整数倍, 并  $P = N/M$ ), 随机竞争个体数目  $q$ , 最大进化代数  $K_{\max}$ , 高斯变异算子标准差  $R_0$ , 设置种群进化代数  $K = 0$ , 设置进化终止条件, 其中  $R_0 = [R_1^{(0)} \quad R_2^{(0)} \quad \dots \quad R_M^{(0)}]$ ;

个体作为初始种群  $X^{(0)}$ , 使用随机  $q$  竞争法则对个体排序, 按照排序结果将整个物种划分为  $M$  个子群, 即排在前  $P$  个的个体组成子群  $X_1^{(k)}$ , 排在最后  $P$  个的个体组成子群  $X_M^{(k)}$ ;

(3) 终止进化判断. 满足, 停止进化并输出计算结果, 否则转(4);

(4) 子群竞争. 计算子群  $X_i^{(k)}$  中每个个体的适应度  $\text{Fit}(x_{ij}^{(k)})$ , 其中  $j \in [1, P]$ ,  $i \in [1, M]$ . 则第  $i$  个子群  $X_i^{(k)}$  的适应度  $\text{Fit}X_i^{(k)}$  为:

$$\text{Fit}X_i^{(k)} = \sum_{j=1}^{P_i^{(k)}} \text{Fit}(x_{ij}^{(k)}) \setminus P \quad (9)$$

不同子群之间相互竞争, 由于其适应环境能力的不同而获得不同的变异能力, 因此第  $i$  个子群所使用的变异算子标准差为:

$$R_i^{(k)} = R_i^{(k-1)} \exp\left(-\frac{M \# \text{Fit}X_i^{(k)} - \sum_{i=1}^M \text{Fit}X_i^{(k)}}{\text{Fit}X_{\max} - \text{Fit}X_{\min}}\right) \quad (10)$$

$$\text{Fit}X_{\max} = \max(\text{Fit}X_1^{(k)}, \dots, \text{Fit}X_M^{(k)}) \quad (11)$$

$$\text{Fit}X_{\min} = \min(\text{Fit}X_1^{(k)}, \dots, \text{Fit}X_M^{(k)}) \quad (12)$$

(5) 种群繁殖. 不同的子群使用不同的变异算子来繁殖后代:

$$x_{ij}^{(k)c} = x_{ij}^{(k)} + D_i^{(k)} \quad (13)$$

式中,  $D_i^{(k)}$  为一服从  $N(0, R_i^{(k)2})$  分布的高斯白噪声.  $X^{(k)}$  经变异以后产生的子代称为  $X^{(k)c}$ .

(6) 种群重组. 在  $X^{(k)}$  和  $X^{(k)c}$  组成的临时种群中, 计算每个个体的适应度并排序, 排在前  $N$  个的个体组成下一代种群  $X^{(k)}$ , 使用随机  $q$  竞争法则对个体排序, 按照排序结果将整个种群划分为  $M$  个子群, 排在前  $P$  个的个体组成子群  $X_1^{(k)}$ , 排在最后  $P$  个的个体组成子群  $X_M^{(k)}$ ;

(7)  $K = K + 1$ , 转(3).

在使用 MCEP 对函数进行优化时需注意以下几个问题:

<sup>1</sup> 式(2)中变异算子的进化是一个递归过程, 对于排在后面的子群的变异算子标准差可能很大, 因此对变异算子的标准差作如下规范:

$$\text{如果 } R_i^{(k)} > W/4 \quad \text{则 } R_i^{(k)} = |W/4 - R_i^{(k)}| \quad (14)$$

$W$  为变量空间的宽度, 重复使用上式, 直到  $R_i^{(k)} < W/4$ ;

<sup>2</sup> 即使对变异算子进行规范以后, 变异后的个体依然可能超出问题可行解空间的范围, 本文使用如下的方式进行处理:

$$\text{如果 } x_{ij}^{(k)c} = x_{ij}^{(k)} + D_i^{(k)} > b \quad (15)$$

$$\text{则 } x_{ij}^{(k)c} = 2b - (x_{ij}^{(k)} + D_i^{(k)}) \quad (16)$$

$$\text{如果 } x_{ij}^{(k)c} = x_{ij}^{(k)} + D_i^{(k)} < a \quad (17)$$

$$\text{则 } x_{ij}^{(k)c} = 2a - (x_{ij}^{(k)} + D_i^{(k)}) \quad (18)$$

式中,  $[a, b]$  为变量空间的取值约束条件, 重复使用(15)~(18)直到  $x_{ij}^{(k)c} \in [a, b]$ ;

» (10)~(18)均以一维函数为例给出的计算方法, 如果待优化的函数为多维, 则对每一维单独进行处理, 同时对于每一维的高斯变异算子应独立产生.

## 4 性能比较

用此MCEP算法与标准的EP算法、AEP<sup>[7]</sup>算法和IEP算法<sup>[8]</sup>同时进行函数优化,以考察算法的各种性能。在本文算法中参数选取如下:

$N=80, M=4, q=60, K_{\max}=10\ 000, R_0=0.01$  @待优化函数各维区间长度,进化终止条件设置为,子群中具有最好适应度的前20个个体的平均适应度达到指定精度要求,算法的精度均为 $10^{-4}$ 。AEP算法和IEP算法的参数使用文献[7,8]给出的缺省值。使用下述目标函数用来测试算法的性能:

Schaffer's Function<sup>[9]</sup>:

$$F(x) = 0.5 - \sin^2 \left( \sqrt{\sum_{i=1}^n x_i^2} - 0.5 / (110 + 0.001 \sum_{i=1}^n x_i^2) \right), x_i \in [-100, 100] \quad (19)$$

多峰余弦函数<sup>[10]</sup>:

$$F(x) = (5 - |x|) \# |\cos(10\pi x)|, x \in [-5, 5] \quad (20)$$

Ackley's Path Function:<sup>[11]</sup>

$$F(x) = 20 + e^{-20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right)} - \exp \left( \sum_{i=1}^n \cos(2\pi x_i) \right) / n, x_i \in [-30, 30] \quad (21)$$

De Jong's Function 1<sup>[12]</sup>:

$$F(x) = \sum_{i=1}^n x_i^2, x_i \in [-512, 512] \quad (22)$$

Rastrigin's Function<sup>[13]</sup>:

$$F(x) = nA + \sum_{i=1}^n (x_i^2 - A \cos(2\pi x_i)), x_i \in [-512, 512], A=10 \quad (23)$$

各个函数的维数分别取2、1、4、5、4。各函数的图形如图1所示(均以一维为例画出,各图形依次为Schaffer's Function,多峰余弦函数, Ackley's Path Function, De Jong's Function 1, Rastrigin's Function)。

### 4.1 全局收敛性

同时使用EP算法、AEP算法、IEP算法和MCEP算法对上述函数进行优化,种群规模均为80,每种算法重复计算100次,每种算法到达指定精度的次数如表1所示。

由表1可以看出,相对于传统的EP算法以及一些改进的EP算法,在全局收敛性上, MCEP算法具有良好的性能。这种性能的取得与算法将进化划分为多个子群并行进行,同时不同的子群使用不同大小的变

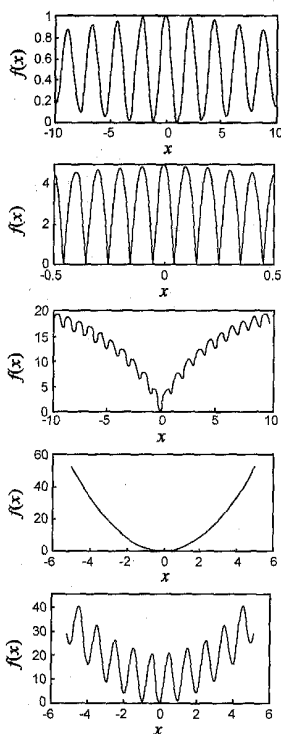


图1 各函数一维图形

表1 不同算法成功收敛次数

算法	Schaffer	多峰余弦	Ackley	De Jong	Rastrigin
EP	12	85	8	50	1
IEP	7	94	23	91	0
AEP	31	51	58	72	14
MCEP	100	100	100	100	67

### 4.2 快速收敛性

同时使用4种算法对Schaffer's Function进行优化,不同算法到达指定精度所需要的平均进化代数如表2所示。每种算法重复计算100次,如果进化10 000代仍然不能到达指定精度,则认为此次进化失败,其进化代数不计入平均进化代数。

表2 不同算法收敛速度比较

算法	01	01.05	01.01	01.005	01.001	01.0005	01.0001	01.00005	01.00001
EP	代数	28	36	83	1049	2816	4372	5843	8303 > 10000
	时间	01.016	01.020	01.045	01.595	11.578	21.530	31.347	41.770
IEP	代数	13	15	29	59	81	179	436	855 1343
	时间	01.007	01.008	01.016	01.034	01.045	01.103	01.243	01.482 01.786
AEP	代数	11	39	245	1753	3932	4377	6291 > 10000	> 10000
	时间	01.006	01.023	01.149	11.040	21.339	21.634	31.787	
MCEP	代数	11	17	106	741	999	1097	1196	1375 1654
	时间	01.008	01.012	01.078	01.550	01.742	01.815	01.890	11.025 11.235

在表2中,各种算法的进化时间是在PIV 210GHz, WIN 2000和Matlab 512的仿真环境下获得,时间的单位为秒。由表2可以看出,在进化前期,MCEP相对于EP和AEP收敛速度上并没有明显的优势,甚至还稍稍落后于AEP算法,但在进化的后期,MCEP算法的收敛速度要明显快于其他两种算法。IEP算法的收敛速度明显快于MCEP算法,但这种快速性是以牺牲算法的收敛成功率为代价的。

### 4.3 参数鲁棒性

在 $R_0$ 取不同值的情况下,分别使用EP、IEP和MCEP算法对多峰余弦函数进行优化,每种算法重复计算100次,每种算法到达指定精度的次数如表3所示(AEP算法中并不使用 $R_0$ 参数,因此未列入比较)。

由表3可以看出当 $R_0$ 参数作大范围的变化时,MCEP算法收敛均具有良好的收敛性,而实际上,除了AEP算法外(AEP算法并不使用参数 $R_0$ ),其他算法均对参数 $R_0$ 敏感。

表3 不同 $R_0$ 参数下算法的收敛成功次数

	1	0.5	0.1	0.05	0.01	0.005	0.001
EP	31	70	95	77	49	37	1
IEP	100	100	94	72	43	17	2
MCEP	100	100	100	97	94	97	93

如果 $R_0$ 参数取的比较小,除AEP算法外的其他算法均对初始种群敏感,即当初始种群中有某些个体落在全局最优解

的邻域的时候,算法能够快速收敛到全局最优解,如果初始种群中没有任何一个个体落在最优解的邻域的时候,算法的收敛速度将极大的减慢,甚至不收敛。表 4 给出了当种群在整个变量空间随机初始化和初始种群偏置(即将初始种群限定在  $[b - \frac{b-a}{10}, b]$ )两种情况下,3 种算法在不同  $R_0$  参数下对多峰余弦函数进行优化成功收敛次数。每种情况测试 100 次。种群规模均为 80,最大进化代数设置为 10000,进化终止条件设置为,种群中具有最好适应度的前 20 个个体的平均适应度达到指定精度要求。

表 4 随机初始与偏置初始算法性能比较

		1	0.5	0.1	0.05	0.01	0.005	0.001
EP	随机初始	31	70	95	77	49	37	1
	偏置初始	33	65	13	1	0	0	0
IEP	随机初始	100	100	94	72	43	17	2
	偏置初始	77	84	5	0	0	0	0
MCEP	随机初始	100	100	100	97	94	97	93
	偏置初始	99	97	92	100	94	99	90

由表 4 可以看出 MCEP 不仅对  $R_0$  参数具有很强的鲁棒性,而且对于初始种群的好坏也不敏感。需要指出的是, AEP 算法不使用  $R_0$  参数,由于其变异的标准差是随进化的快慢(或者说有效进化与无效进化)而改变的,因此该算法对初始种群也不敏感,特别是在求解精度不是很高的情况下。

## 5 理论分析

在进化规划算法中,如果算法未能以指定精度搜索到全局最优解通常称之为不收敛,但经过分析发现,这种不收敛包含两种情况:

(1) 由于算法使用了不合适的变异算子标准差(通常是过小),导致种群陷入了局部极值不能脱离,因此求解精度不能满足需要,对于这种情况的不收敛在本文中定义为第一类不收敛,也就是平时所说的早熟收敛;

(2) 由于算法使用了不合适的变异算子标准差(通常是过大),种群中的某些个体虽然已经到达了全局最优解所在的邻域,但是这些个体的变异强度是如此的大,以至于个体与全局最优解间的距离始终不能足够的小,因此求解精度不能满足需要,对于这种情况的不收敛在本文中定义为第二类不收敛。

对于某一待优化的一维函数为例,假设某一时刻种群中的所有个体到达了函数的某个局部极值点  $x_5$  的附近,但是还没有任何个体进入区域  $[x_1, x_2]$ , 假设算法的参数  $R_0 = 0.1$ ,  $N = 40$ ,  $x_5 - x_2 = 1$ ,  $x_2 - x_1 = 0.15$ , 那么种群中在  $x_5$  附近的个体经过一次变异进入区域  $[x_1, x_2]$  的概率为:

$$P_1 \approx 80\% \int_{x_1}^{x_2} \exp\left(-\frac{x^2}{2\sigma_1^2}\right) dx \approx 6120 \times 10^{-22} \quad (24)$$

也就是大约平均需要  $115 \times 10^{21}$  代的进化,才有一个个体从局部极值进入区域  $[x_1, x_2]$ , 而这个进化代数对于工程应用来讲实在是太长了,这也是理论上证明了能够全局收敛的 EP 算法之所以还会陷入局部极值的原因。在此时,种群陷入早熟收敛。

设种群中的某个个体经过有限次的进化后进入区域  $(x_7,$

$x_2)$ , 并位于  $x_7$  附近,此时收敛精度对应的区域为  $[x_6, x_7]$ , 而且  $R_0 = 0.1$ ,  $x_7 - x_6 = 0.0001$ , 那么个体经过一次变异进入收敛精度对应的区域  $[x_6, x_7]$  的概率为:

$$P_2 = \int_{x_6}^{x_7} \exp\left(-\frac{x^2}{2\sigma_1^2}\right) dx \approx 4 \times 10^{-4} \quad (25)$$

因此大约需要平均 2500 次进化才能有获得一个满足精度要求的个体,这也就是较大的变异算子能够在设定的进化代数内搜索到全局最优解的邻域,但是在设定的进化代数内无法满足精度要求的原因。在此时,种群陷入了第二类不收敛。

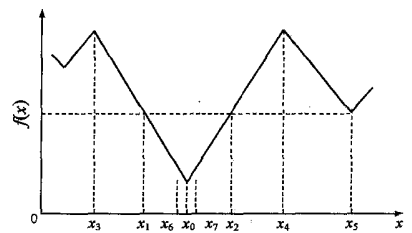


图 2 待优化函数的区域划分

对于 MCEP 算法,虽然对于不同的函数进行优化时的变异算子进化规律并不相同,但是其大致规律如图 3 所示(横坐标为进化代数  $K$ , 纵坐标为变异算子标准差  $R$ )。

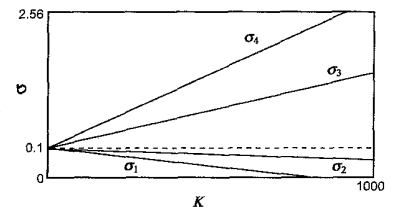


图 3 MCEP 算法变异算子随进化代数演变图

假设某一时刻种群中的所有个体到达了函数的局部极值点  $x_5$  的附近,假设此时子群  $X_4$  的变异算子标准差为  $R_4 = 1$ , 那么子群  $X_4$  中的某个个体经过一次变异进入区域  $[x_1, x_2]$  的概率为:

$$P_1 = \int_{x_1}^{x_2} \exp\left(-\frac{x^2}{2\sigma_1^2}\right) dx \approx 0.092 \quad (26)$$

对于一个包含 20 个个体的子群而言,大约经过 10 余次进化就有一个个体进入区域  $[x_1, x_2]$ 。

设子群  $X_1$  中的某个个体经过有限次的进化后进入区域  $(x_7, x_2)$ , 并位于  $x_7$  附近,且  $R_1 = 0.0001$ , 那么个体经过一次变异进入收敛精度对应的区域  $[x_6, x_7]$  的概率为:

$$P_2 = \int_{x_6}^{x_7} \exp\left(-\frac{x^2}{2\sigma_1^2}\right) dx \approx 0.134 \quad (27)$$

因此大约需要 3 次进化个体就能进入收敛精度对应的区域  $[x_6, x_7]$ 。

正是由于 MCEP 算法同时使用不同大小的变异算子标准差,从而保证种群中的个体既能对变量空间进行分散的探索,又能够对探索到的局部进行充分的搜索。

## 6 结论

针对进化规划算法的弱点,提出了一种多群竞争的进化规划算法。在该算法中,进化在多个子群间并行进行,子群的竞争决定了子群的变异能力,子群重组导致携带适应度信息的个体在子群间流动。给出了算法的流程,并使用典型的进化

算法测试函数对算法的性能进行了比较,结果表明,MCEP算法在全局收敛性、收敛快速性和参数鲁棒性上均具有良好的性能。

#### 参考文献:

- [ 1 ] Yao X, Liu Y. A new evolutionary system for evolving artificial neural networks[J]. IEEE Transactions on Neural Networks, 1997, 8(3): 694 - 713.
- [ 2 ] Sebald A V, Schlenzig J. Minimax design of neural net controllers for highly uncertain plants[J]. IEEE Transactions on Neural Networks, 1994, 5(1): 73- 82.
- [ 3 ] Bomholdt S, Graundenz D. General asymmetric neural networks and structure design by genetic algorithms[J]. IEEE Transactions on Neural Networks, 1992, 5(5): 327- 334.
- [ 4 ] Fogel D B. Applying evolutionary programming to selected traveling salesman problem[J]. Cybernetics and System, 1993, 24(1): 27- 36.
- [ 5 ] Yao X, Liu Y, Lin G. Evolutionary programming made faster[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82- 102.
- [ 6 ] 宋爱国, 陆佑人. 一种基于排序操作的进化算子自适应遗传算法[J]. 电子学报, 1999, 27(1): 85- 88.
- [ 7 ] 刘芳, 李人厚. 基于自适应变异规则的一种有效的进化规划[J]. 控制与决策, 2002, 17(2): 148- 150.
- [ 8 ] 郭崇慧, 唐焕文. 一种改进的进化规划算法及其收敛性[J]. 高等学校计算数学学报, 2002, 3(1): 51- 56.
- [ 9 ] Sachfler J D, Caruana R, EsheIman L J, et al. A study of control parameters affection on line performance of genetic algorithms for function optimization[A]. Proceedings of the 3<sup>th</sup> International Conference on Genetic Algorithm[C]. San Mateo, CA: Morean Keufmann, 1991. 51- 60.

- [ 10 ] 周晴, 李衍达. 考虑基因表达过程的进化算法[J]. 电子学报, 2002, 30(1): 114- 117.
- [ 11 ] Banzhaf W, Beckman F H. Evolution and Biocomputation[M]. Berlin, New York: Springer, 1995.
- [ 12 ] De Jong K. An analysis of the behavior of a class of genetic adaptive systems[D]. Michigan: University of Michigan, 1975.
- [ 13 ] Gordon V S, Whitely D. Serial and parallel genetic algorithms as function optimizers[A]. Proceedings of the 5<sup>th</sup> International Conference on Genetic Algorithm[C]. San Mateo, CA: Morean Keufmann, 1993. 177 - 183.

#### 作者简介:



**王向军** 男, 1973 年 11 月生于河北省滦南县, 分别于 1996, 1999, 2004 年在海军工程大学获得工学学士、硕士和博士学位, 现为海军工程大学电气工程系讲师, 研究兴趣为电力电子技术, 进化计算, 神经网络。



**嵇斗** 男, 1975 年 9 月生于江苏省灌南县, 1997 年在中国矿业大学获得工学学士学位, 2001 年在海军工程大学电子工程学院获得工学硕士学位, 现为海军工程大学电气工程系讲师, 研究兴趣为水声信号处理, 进化计算, 神经网络。