

# 进化学习策略收敛性和逃逸能力的研究<sup>1)</sup>

阎岭 蒋静坪

(浙江大学电气工程学院 杭州 310027)  
(E-mail: xiaozhao9912@sina.com)

**摘要** 分析了基于拉马克主义的进化学习策略 (简称 LELS) 和基于达尔文主义的进化学习策略 (简称 DELS) 的异同: 前者的生物学依据是后天获得性遗传, 而后者的依据是自然选择; 前者的表现型和基因型在学习的过程中同时被优化; 而后者表现型的变化不会直接导致基因型的改变. 利用马尔可夫链理论证明了此类算法的收敛性, 并且在理论上分析了 DELS 具有更强的局部逃逸能力. 在仿真试验中应用 8 个标准测试函数进行测试, 结果表明此类算法具有较好的全局优化能力和较快的收敛速度, 其中 DELS 的逃逸能力更强.

**关键词** 进化学习策略, 拉马克主义, 达尔文主义, LELS DELS 后天获得性遗传  
**中图分类号** TP273; TP30

## Convergence and Escape Capacity Research of Evolution Learning Strategies

YAN Ling JIANG Jing-Ping

(College of Electrical Engineering, Zhejiang University, Hangzhou 310027)  
(E-mail: xiaozhao9912@sina.com)

**Abstract** The Lamarckian evolution learning strategy (LELS) and Darwinian evolution learning strategy (DELS) are discussed in terms of their similarities and differences in learning implementation. The former is based on inheritance of acquired character, *i.e.*, both phenotype and genotype can be optimized through learning while the later only optimizes phenotype based on Darwinian selection. The convergence of ELS is proved using Markov chain theory. And we also theoretically demonstrate that DELS has stronger escaping capacity. These algorithms are applied to 8 standard test functions. Simulation results show that LELS and DELS yield faster convergence and better global optimization ability than standard evolution strategies; moreover, DELS also leads to better escaping capacity. Finally, limitations of the work as well as the future study are discussed.

**Key words** ELS, Lamarckian, Darwinian, LELS, DELS, inheritance of acquired character

## 1 问题的提出

进化策略 (Evolution Strategy) 与传统优化算法相结合, 是近年来的研究热点之一. 相关的文献有: [1] 将遗传算法与单纯形法相结合, 即后代个体利用单纯形法进行局部搜索,

1) 国家自然科学基金 (60405012) 资助

Supported by National Natural Science Foundation of P. R. China (60405012)

收稿日期 2004-6-28 收修改稿日期 2004-12-28

Received June 28, 2004; in revised form December 28, 2004

以提高适应度; [2] 明确提出将单纯形法作为学习算子; [3] 提出是基于拉马克主义的, 即后天获得性遗传; [4] 给出了混和遗传算法的收敛性证明; 此外 [5~7] 是将遗传算法与梯度法或其他局部优化算法相结合, 进行相关的研究.

但是目前的工作, 多是将一些新的混和进化算法 (常属作者自己提出的新方法) 与经典进化算法就某些数字仿真例子进行比较, 以此验证新算法的性能或效率更优, 而理论分析则非常有限, 尤其是有关收敛性、逃逸能力以及收敛速度的研究大大滞后于其在工程中的应用<sup>[8~10]</sup>. 而且现有的学习方式大多基于拉马克主义 (虽然许多文献并未明确标出), 基于达尔文主义的尚不多见, 两者的区别以及理论分析也未做足够的研究. 针对进化策略的研究也较少见. 此类方法的名称也较混杂, 早期的工作大多称之为“混和进化算法 (Hybrid EA)”, 但是本文认为这不足以体现学习的概念, 因此建议改为“进化学习算法”; 将进化策略与学习算法相结合的, 称之为“进化学习策略”.

基于上述考虑, 本文分析了基于拉马克主义的进化学习策略 (Lamarckian evolution learning strategy, 简称 LEELS) 和基于达尔文主义的进化学习策略 (Darwinian evolution learning strategy, 简称 DELS) 的异同, 证明了此类算法的收敛性, 在理论上分析了 DELS 具有更强的局部极值逃逸能力; 最后给出了仿真试验的结果.

## 2 预备知识

### 2.1 学习、基因型、表现型与适应度之间的关系

上述 4 者的关系如图 1 所示. 一个生物个体, 要在生存竞争中获胜, 除了要具有良好的遗传基因以外, 还应该进行较好的学习, 以获得较好的表现型, 否则就不能充分发挥其优良基因的潜力. 个体的适应度, 是由其表现型决定的, 与基因型没有必然的关系. 选择只能根据适应度针对表现型来进行, 而无法发生在基因型这一层面上. 学习是沟通基因型与表现型之间的桥梁. 如果图 1 中虚线成立, 即表现型在后天学习中也会影响基因型, 就是拉马克主义; 否则就是达尔文主义, 这将在 2.2 节中详细论述.

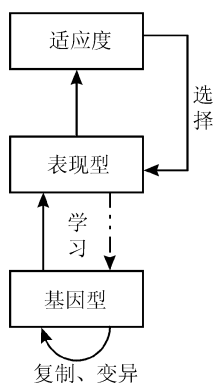


图 1 学习、基因型、表现型与适应度

Fig. 1 The relationship of learning, genotype, phenotype and fitness

早期的进化算法是生物进化过程的一个极其简单的模拟, 并没有考虑生物后天学习的情况. 因此, 在进化算法中以适当的方式加入学习的环节, 就有可能在本质上提高其性能和效率.

### 2.2 拉马克的用进废退和达尔文的自然选择

拉马克主义最重要的一个观点就是“后天获得性遗传”, 即所谓“用进废退”说, 最著

名的例子就是“长颈鹿”:长颈鹿的祖先经常伸长了脖子去吃树枝高处的叶子,脖子受到了锻炼,变长了,而这一点可以遗传,因此其后代就要比父母的脖子长一些,经过一代又一代的进化,脖子就会变得越来越长.概括起来讲,就是生物在后天的学习中获得了表现型的改变(长颈鹿的脖子在锻炼中拉长),可以直接影响基因型的改变,进而导致了进化的产生.

达尔文对此有着截然不同的解释,他认为自然选择在起作用.在长颈鹿繁殖的过程中,会有变异的情况产生,有的后代脖子会变长,有的会变短,当然还有的长度不变.这种变异是随机的,即“上帝是掷骰子的”.但是,只有脖子长的鹿,才有可能获得更多的食物,存活下来的可能性也就越大.这种变异和选择的过程不断发展,使得长脖子的性状在群体中扩散开来,就会进化出长颈鹿这样一个新的物种.脖子长度改变这样一种变异,对于每一只鹿来说只是一个偶然事件,但是在自然选择的作用下,在其种群的层面上来看,却呈现出向长颈鹿进化的方向.通过后天学习而获得的表现型的改变(长颈鹿的脖子在锻炼中拉长),虽然可以提高其存活的可能性,但是不会导致基因型的改变,更不会直接遗传给后代<sup>[11]</sup>.

自 20 世纪以来,基因论确立并与自然选择学说完美的结合在一起之后,拉马克主义也就被彻底的抛弃了.但是由于拉马克主义可以在一定程度上解释一定层次上的某些进化问题,所以也可以为我们改进进化策略提供一些有益的启发.本文分别以拉马克主义和达尔文主义为依据,并不意味着作者认为拉马克主义在生物进化界也是正确无误的,只是以该理论为启发来讨论算法问题.

### 3 进化学习策略描述

#### 3.1 基于达尔文主义的进化学习策略 (DELS) 算法流程

本算法中父代和子代个体的数目分别用  $\mu$  和  $\lambda$  表示,简记为  $(\mu, \lambda)$ -LELS,描述如下:

1) 初始化:随机生成  $\mu$  个个体,构成第 1 代群体的父代.其中每个个体的不同是由个体的基因型不同而决定的.

2) 变异:每个父代个体将自己的基因经过变异后产生  $\lambda/\mu$  个子代,总共生成  $\lambda$  个子代个体.此  $\lambda$  个子代个体与其各自的父代个体在基因上有所不同.

3) 学习: $\lambda$  个子代个体各自进行学习.这一步可以采用不同的学习算法,本文采用的是简单有效的单纯形法.学习后得到的个体,表现型被优化,适应度提高,但是基因型不会发生变化.

4) 选择:每个个体由于对环境的适应性不同而表现出不同的生命力或适应度,它由遗传基因和后天学习的情况共同决定. $\lambda$  个个体中只有生命力较强的  $\mu$  个个体成为下一代的父辈个体.如果终止条件满足,则算法结束.否则回到步骤 2).在进行新一轮的繁殖时,只有基因型经过变异遗传给下一代,表现型在此不起作用.

在 DELS 算法中,基因型与表现型没有必然的关系,即基因型好的个体,表现型不一定好;表现型好的个体,基因型不一定好.

#### 3.2 基于拉马克主义的进化学习策略 (LELS)

**假设 1.** 在 LELS 算法中,表现型等同于基因型.

一般来说,基因型与表现型之间的关系较为复杂,但是由于 LELS 算法采用拉马克主义,表现型的改善会直接导致基因型的改善,因此两者之间的函数关系是单调递增的.本文为简单起见,在 LELS 算法中,将表现型等同于基因型,以此来体现“后天获得性遗传”的观点.

LELS 在学习和变异操作上, 与 DELS 有所不同. 学习后得到的个体, 表现型被优化, 适应度提高. 由于采用拉马克主义, 以及假设 3.2, 所以基因型也同样被优化.  $\lambda$  个个体中表现型较好, 适应度较高的  $\mu$  个个体成为下一代的父代. 需要注意的是, 在父代个体将自己的基因遗传给下一代时, 已经包括了后天学习中获得的基因型 (表现型) 的改变.

## 4 收敛性证明和局部极值逃逸能力分析

### 4.1 进化学习策略收敛性证明

本节所指的进化学习策略 (ELS), 包括 LELS 和 DELS 算法. 假设 ELS 在第  $t (t \geq 0)$  代的种群为  $\mathbf{X}(t) = (\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \dots, \mathbf{x}_{t,N}), \mathbf{x}_{t,i} \in \mathbf{S}$ , 则  $\{\mathbf{X}(t)\}_{t \geq 0}$  构成一个离散时间的随机过程, 其状态空间为  $\mathbf{E} = \mathbf{S}^N, t \in \mathbb{Z}^+ = (0, 1, \dots, n, \dots)$ . 由于新种群  $\mathbf{X}(t+1)$  仅以一种概率方式依赖于当前种群  $\mathbf{X}(t)$ , 因此随即过程  $\{\mathbf{X}(t)\}_{t \geq 0}$  构成一个离散时间的马尔可夫链.

**定义 1.** 对于 ELS 求解数值优化问题, 设随机变量  $T = \min\{t \in \mathbb{Z}^+ : F_t = f^*\}$  表示首次击中全局极值点的时间, 如果  $P\{T < \infty\} = 1$  且与初始种群无关, 则称 ELS 以概率 1 在有限时间内访问到全局极值点. 函数值  $f^* = f(x^*) < +\infty$  称为一个全局极大值当且仅当  $\forall x \in \mathbf{S} \Rightarrow f(x) \leq f(x^*)$  成立时, 被称为一个全局极大值点<sup>[12]</sup>.

**假设 2.** 在每一进化代  $t$ , 对父代种群  $\mathbf{X}(t)$ 、变异算子作用一次后得到的后代种群  $\mathbf{X}(t)$  以及任意的  $y \in \mathbf{S}$ , 如果  $y \notin \mathbf{X}(t)$ , 则存在  $p(t) > 0$ , 使得  $y \in \mathbf{X}(t)$  的概率  $\geq p(t)$ .

这个假设保证了在进化的每一代中, 每一个个体可以以严格大于 0 的概率通过一次变异成为搜索空间中其他任意一个个体. 对于绝大多数实际情况, 该假设都是成立的.

**定义 2.** 考察连续参数优化问题 (Continuous parameter optimization problem), 即给定搜索空间为非空  $n$  维欧式空间  $R^n$  中的有界集合  $\mathbf{S} = \prod_{i=1}^n [a_i, b_i]$ , 其中  $a_i < b_i, i = 1, 2, \dots, n$ . 定义  $\mathbf{S}_1$  为全局极大点子空间,  $\mathbf{S}_2$  为非全局极大点子空间, 当且仅当  $\mathbf{S}_1 \cap \mathbf{S}_2 = \Phi, \mathbf{S}_1 \cup \mathbf{S}_2 = \mathbf{S}$ , 全局极大点  $\mathbf{x}^* \in \mathbf{S}_1$ , 且梯度  $\nabla f(\mathbf{x}) \neq 0, \forall \mathbf{x} \in \mathbf{S}_1, \mathbf{x} \neq \mathbf{x}^*$ . 显然  $\mathbf{S}_1$  为一凸集, 与此对应的函数  $f(\omega), \omega \in \mathbf{S}_1$  为凸函数.

**定理 1.** 如果 ELS 满足假设 2, 则其在有限次进化后以概率 1 访问到全局极大点, 亦即  $P\{T < \infty\} = 1$ , 且与初始分布无关.

**证明.** 由于可以证明单纯形算法对于凸函数全局极值点搜索的收敛性, 因此, 根据定义 2, 只要证明在任一进化代  $t (t < \infty)$ , 若父代个体  $\mathbf{X}(t) \notin \mathbf{S}_1$ , 而通过变异算子变异后得到的任一子代个体  $\hat{\mathbf{x}}_t \in \mathbf{S}_1$ , 则通过学习算子的作用, 必可以得到最优个体  $\hat{\mathbf{x}}_t = \mathbf{x}^*$ , 即以概率 1 访问到全局极大点. 其中  $\hat{\mathbf{x}}_t$  为学习后的个体.

由假设 4.1 知, 在任一进化代  $t$ , 通过变异算子的作用可以以一个下界为  $P(t)$  的概率将  $\mathbf{X}(t)$  中任一个体变异为搜索空间中任意一个其他个体. 因此, 在变异过程中对任一个体  $\mathbf{x}_t \notin \mathbf{S}_1$ , 通过变异算子作用后变为  $\mathbf{S}_1$  子空间中任意一点的概率都不小于  $p(t)$ , 则落入  $\mathbf{S}_1$  子空间的概率不小于  $\int_{\mathbf{S}_1} p(t)$ , 且与个体的具体实例无关. 因此, 在  $t$  代进化以后, 还未有任一个体落入  $\mathbf{S}_1$  子空间的概率  $P_{not}(t)$  满足:

$$P_{pot}(t) = \prod_{i=1}^t (1 - \int_{\mathbf{S}_1} P(t)) \quad (1)$$

则

$$P\{T < \infty\} = 1 - p_{not}(t) \quad (2)$$

上式两端令  $t \rightarrow \infty$ , 可得

$$P\{T < \infty\} = 1 - \prod_{i=1}^{\infty} (1 - \int_{S_1} p(t)) \quad (3)$$

$$\because 0 < \int_{S_1} p(t) < 1 \quad (4)$$

$$\therefore 0 < 1 - \int_{S_1} p(t) < 1 \quad (5)$$

$$\therefore \prod_{i=0}^{\infty} (1 - \int_{S_1} p(t)) = 0 \quad (6)$$

即可得到

$$P\{T < \infty\} = 1 \quad (7)$$

由证明过程可以看出, 结论的成立与初始群体是无关系的.

证毕.

#### 4.2 局部极值逃逸能力分析

逃逸能力 (Escaping capacity)<sup>[13]</sup> 是指跳出局部极值点的足够大邻域的能力. 文献<sup>[13]</sup> 对高斯变异算子采用精英和非精英选择两种情况下的逃逸能力进行比较, 受此启发, 本文用类似的方法分析 DELS 和 LELS 的逃逸能力.

由于 ELS 的复杂性, 本文采用两种简化的方式来产生新的搜索点, 分别代替 LELS 和 DELS 来进行讨论. 假设当前点  $x_0$  是局部最优点, 方法一产生新的搜索点为:  $x_0 + N_i(0, 1)$ , 其中  $i = 1, 2, \dots, k$ ,  $N_i(0, 1)$  重复独立产生. 由于 LELS 的表现型的改进直接引起基因型的改进, 所以陷入局部极值点的个体, 在变异后产生的下一代, 如果仍然在局部极值点邻域内, 必将经过学习操作又收敛于该极值点, 因此与方法一相类似. 方法二产生新的搜索点为:  $x_i + N_{i+1}(0, 1)$ , 其中  $i = 0, 1, \dots, k-1$ ,  $N_{i+1}(0, 1)$  也是重复独立产生的, 即对于每一个新产生的点, 都接受它并用它产生新的搜索点. 由于 DELS 的表现型的改进不能引起基因型的改变, 所以基因型不受学习的影响而只经过变异操作遗传给下一代, 因此与方法二相类似.

对于  $x_0$  的某个邻域  $S(R) = \{x : |x - x_0| < \varepsilon\}$ ,  $\varepsilon$  为事先给定的一个小数, 考虑在产生  $k$  个个体中, 至少有一个个体跳出该邻域的概率为  $P_{esp}$ .

对于方法一, 显然有

$$\begin{aligned} P_{esp1} &= 1 - \prod_{i=1}^k P(|x_i - x_0| < R) = 1 - [P(x_i - x_0 < R)]^k = 1 - [P(|N(0, 1)| < R)]^k = \\ &= 1 - \left[ \int_{-R}^R \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right) dt \right]^k = 1 - \left( \operatorname{erf}\left(\frac{\sqrt{2}}{2}R\right) \right)^k \end{aligned} \quad (8)$$

对于方法二, 有

$$P_{esp2} = 1 - \prod_{i=1}^k P(|x_i - x_0| < R) = 1 - \left[ \int_{-R}^R \frac{1}{\sqrt{2\pi k}} \exp\left(-\frac{1}{2k}t^2\right) dt \right]^k = 1 - \operatorname{erf}\left(\frac{R}{\sqrt{2k}}\right) \quad (9)$$

根据上式对  $P_{esp1}$  和  $P_{esp2}$  作图, 分别取  $k = 2, 5, 10, 100$ , 如图 2 所示.

分析图 2 可以看出, 局部极值点的邻域越大, 跳出局部极值点的可能性就越小; 进化代数越多, 跳出局部极值点的可能性就越大, 两种方法都符合该规律; 当  $R$  较小时, 两种

方法跳出局部极值点的可能性都较大,但是拉马克主义更大一些;而当  $R$  较大时,拉马克主义迅速衰减,而达尔文主义仍保持较高的可能性,尤其在  $k$  较大的时候。

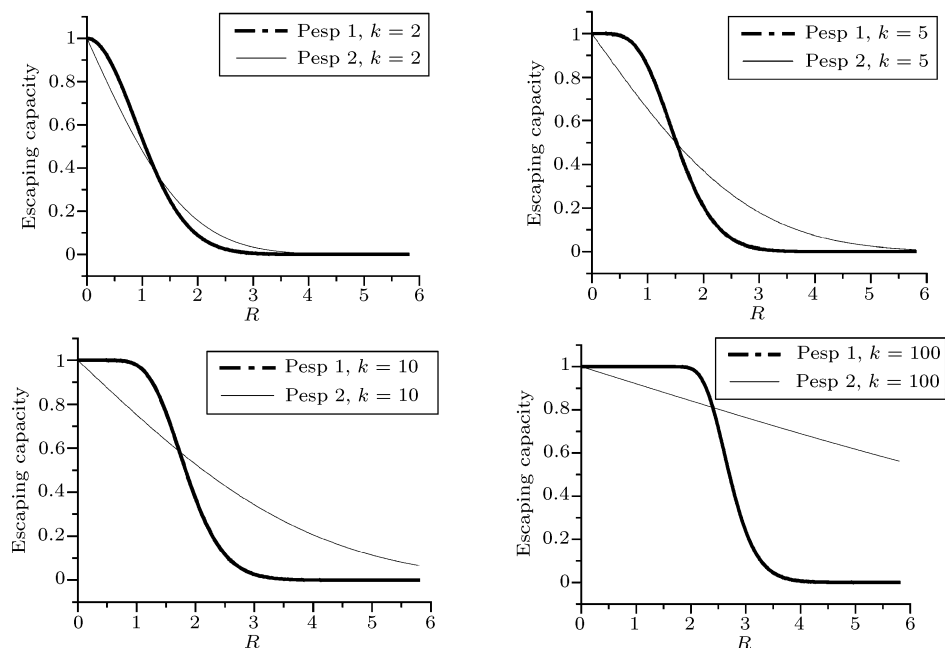


图 2 LELS 和 DELS 逃逸能力比较

Fig. 2 The comparison of escaping capacity of LELS/DELS

## 5 仿真实例

本节采用 8 个标准测试函数来测试 LELS/DELS 的性能,如表 1 所示.其中 LELS/DELS 算法的学习操作应用单纯形法实现,设置其最大迭代次数为 100.对每个函数独立运行 50 次,并记录 1) 平均进化代数, 2) 运行时间(秒), 3) 解的平均值, 4) 平均每代运行时间(秒).

表 1 本文采用的 8 个标准测试函数

Table 1 8 standard test functions

测试函数	最值
$F_1 = \sum_{i=1}^2 x_i^2 x_i^2, (-5.12 < x_i < 5.12)$	0
$F_2 = \sum_{i=1}^{M-1} [100(x_{i+1} - x_i)^2 + (1 - x_i)^2 + (1 - x_i)^2], (-5.12 < x < 5.12, i = 1, 2, \dots, M)$	0
$F_3 = \left\{ \sum_{i=1}^5 i \cos[(i+1)x + i] \right\} \left\{ \sum_{i=1}^5 i \cos[(i+1)y + i] \right\}, (-10 < x, y < 10)$	186.7309
$F_4 = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2, (100 < x, y < 100)$	1.0316
$F_5 = -\sum_{i=1}^M x_i \sin(\sqrt{ x_i }), (-500 < x_i < 500, i = 1, 2, \dots, M)$	0
$F_6 = (x_1^2 + x_2^2)^{0.25} [\sin^2(50 \times (x_1^2 + x_2^2)^{0.1}) + 1.0], -10 \leq x_1, x_2 \leq 10$	0
$F_7 = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}, -10 \leq x_1, x_2 \leq 10$	0.999
$F_8 = MA + \sum_{i=1}^M (x_i^2 - A \cos(2\pi x_i)), (-5.12 < x < 5.12, i = 1, 2, \dots, M)$	0

测试平台: AMD XP 2200+,DDR256M/ WinXP/Matlab6.5. 由于循环是 Matlab 运行速度的瓶颈<sup>[14]</sup>,而学习子程序(单纯形法)会用到大量的循环,所以将 LELS/DELS 算法与 ES

算法做速度方面的比较时, 会有失偏颇. 因此本文将学习子程序用 C 语言编制 MEX 文件实现, 提高了学习速度; 而 LEELS/DELS 的其他部分采用 Matlab 实现, 与 ES 算法完全一样. 又由于 Matlab 一般只是用于仿真, 系统实施时一般采用 C/C++ 等其他语言实现, 没有因循环而导致的速度瓶颈, 因此本文的做法是较公正的.

仿真结果如表 2 所示. 分析该表可以看出, LEELS/DELS 所需的进化代数较少, 约为 ES 的  $1/840 \sim 1/2$ ; 搜索精度也略好于 ES. 而为此付出的代价只是每代多做若干次单纯形迭代, 以及平均每代花费的时间是经典算法的约 2~3 倍. 但是由于进化代数远远小于经典算法, 所以总体运行时间仍然比经典算法少. 其中最突出的是 F7 函数, DELS 的进化代数仅是 ES 算法的约  $1/840$ , 总体运行时间约为  $1/270$ ; LEELS 算法相应的结果分别是  $1/574$  和  $1/197$ .

表 2 LEELS/DELS 算法和 ES 算法的性能比较  
Table 2 The comparison of performance of LEELS/DELS

测试函数	优化算法	进化代数	运行时间 (秒)	解的平均值	平均每代运行时间 (秒)
$F_1$	(2,14)-LEELS	1	0.0141	2E-05	0.0141
	(2,14)-DELS	1	0.0109	4E-05	0.0109
	(2,14)-ES	20.5	0.0164	0.00042	0.0008
$F_2$	(2,14)-LEELS	7.9	0.0172	0.00060	0.00218
	(2,14)-DELS	25	0.0408	0.00040	0.00163
	(2,14)-ES	776.2	0.1812	0.00046	0.00023
$F_3$	(2,14)-LEELS	43.2	0.0469	186.7306	0.00109
	(2,14)-DELS	31.5	0.0440	186.7304	0.00140
	(2,14)-ES	5523.4	2.3749	186.7301	0.00040
$F_4$	(2,14)-LEELS	20.3	0.0265	1.03127	0.00131
	(2,14)-DELS	18.8	0.0124	1.03122	0.00066
	(2,14)-ES	114	0.0314	1.03128	0.00028
$F_5$	(2,14)-LEELS	352.2	0.1610	837.966	0.00046
	(2,14)-DELS	136.6	0.0907	837.966	0.00066
	(2,14)-ES	473.4	0.1092	837.965	0.00023
$F_6$	(2,14)-LEELS	614.1	0.3766	0.00088	0.00061
	(2,14)-DELS	339.9	0.1655	0.00087	0.00049
	(2,14)-ES	795.8	0.1670	0.00090	0.00021
$F_7$	(2,14)-LEELS	19.3	0.0234	0.99963	0.00121
	(2,14)-DELS	13.2	0.0171	0.99963	0.00130
	(2,14)-ES	11089.3	4.6188	0.99944	0.00042
$F_8$	(2,14)-LEELS	36.1	0.0267	0.0004	0.00074
	(2,14)-DELS	29.5	0.0125	0.0005	0.00042
	(2,14)-ES	160.3	0.0390	0.0005	0.00024

大多数情况下, DELS 比 LEELS 所需要的进化代数少, 总体运行时间更短. 例如对于  $F_5$  函数, DELS 的进化代数是 LEELS 算法的约  $1/2.58$ , 总体运行时间约为  $1/1.78$ ; 相应的  $F_6$  结果是  $1/1.81$  和  $1/2.28$ . 只有对于  $F_1$  和  $F_2$  函数例外. 其中对于  $F_1$ , DELS/LEELS 都只需要一代就可以搜索到最优值, 并且 LEELS 搜索精度更高一些, 这是因为  $F_1$  函数只有一个极值点; 对于  $F_2$  函数 DELS 的进化代数和总体运行时间是 LEELS 的 3 倍左右, 这主要是因为  $F_2$  函数的局部极值较少, 更适合 LEELS 发挥其后天获得性遗传的优势. 这说明局部极值较少的时候, LEELS 性能更好一些; 而局部极值较多时 DELS 更好一些. 这也印证了本文提出的 DELS 算法局部极值逃逸能力强于 LEELS 的观点.

众所周知, 实时性是进化算法的一个瓶颈, DELS/LEELS 算法在此做了一个较好的突破.

## 6 结语

自然环境是根据适应度针对表现型进行选择, 表现型的好坏, 一部分取决于基因是否优良, 另一部分则取决于个体在后天学习中是否充分发挥了优良基因的潜力. 因此本文讨论基于拉马克主义的进化学习策略 (简称 LELS) 和基于达尔文主义的进化学习策略 (简称 DELS) 在学习和变异操作上的异同, 证明了此类算法的收敛性, 并且进行了局部极值逃逸能力的分析. 仿真试验表明, 此类算法具有较好的全局优化能力和较高的实时性. 在测试函数局部极值较少的情况下, LELS 算法的收敛速度较快, 而测试函数局部极值较多时, DELS 算法的收敛速度较快, 这也与本文提出的 DELS 具有更强的局部极值逃逸能力的观点相吻合.

但是本文只能在仿真中得到进化学习策略收敛速度较快的结果, 尚无法给出理论证明, 这也是今后的努力方向.

## References

- 1 John Yen, Bogju Lee. A simplex genetic algorithm hybrid. In: Proceedings of IEEE International Conference on Evolutionary Computation, Indianapolis, Indiana, USA, 1997. 13~16: 175~180
- 2 Seront G, Bersini H. Simplex GA and hybrid methods. In: Proceedings of IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 1996. 845~848
- 3 Mashhadi H R, Shanechi H M, Lucas C. A new genetic algorithm with Lamarckian individual learning for generation scheduling. *IEEE Transactions on Power Systems*, 2003, **18**(3): 1181~1186
- 4 Xiang Li, Gu pei-liang. Hybrid genetic algorithm with quick convergence, *Control and Decision*, 2002, **17**(1):
- 5 Jian Xun Peng, Thompson S, Kang Li. A gradient-guided niching method in genetic algorithm for solving continuous optimization problems. In: Proceedings of the 4th World Congress on Intelligent Control and Automation, 2002. **4**: 3333~3338
- 6 Kazarlis S A, Papadakis S E, Theocharis J B, Petridis V. Microgenetic algorithms as generalized hill-climbing operators for GA optimization. *IEEE Transactions on Evolutionary Computation*, 2001, **5**(3): 204~217
- 7 Magyar G, Johnsson M, Nevalainen O. An adaptive hybrid genetic algorithm for the three-matching problem. *IEEE Transactions on Evolutionary Computation*, 2000, **4**(2): 135~146
- 8 Liu min, Wu cheng. Hybrid genetic algorithm method for identical parallel machine earliness/tardiness scheduling problem. *Acta Automatica Sinica*, 2000, **26**(2):
- 9 Liu X, Lee B. Optimal design of fiber Raman amplifier based on hybrid genetic algorithm. *Photonics Technology Letters, IEEE*, 2004, **16**(2):
- 10 Balakrishnan P V, Gupta R, Jacob V S. Development of hybrid genetic algorithms for product line designs, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 2004, **34**(1):
- 11 Fang Zhou-Zi. New Commentary of Evolutionism. Hong Kong: New World Press, 1997
- 12 Torn A, Zilinskas A. Global Optimization. Berlin: Springer-Verlag, 1989
- 13 Beyer G H, Fogel D B. A note on the escape probabilities for two alternative methods of selection under Gaussian mutation. In: Proceedings of the Sixth Annual Conference on Evolutionary Programming, Angeliene, P. J. (ed.), Berlin: Springer-Verlag, 1997: 264~274
- 14 MATLAB Application Program Interface Guide. MathWorks Inc. 1998

阎 岭 浙江大学电气学院控制理论专业博士生. 目前感兴趣的领域为进化计算、模糊控制和神经网络.

(YAN Ling Ph.D. candidate at College of Electrical Engineering, Zhejiang University. His research interests include evolutionary computation, fuzzy control, and neural network.)

蒋静坪 浙江大学电气学院教授、博士生导师. 目前感兴趣的领域为进化计算、大系统理论和模糊控制等.

(JIANG Jing-Ping Professor in College of Electrical Engineering at Zhejiang University. His research interests include evolutionary computation, theory of large scale system, and fuzzy control.)