第 34 卷 第 7 期　　自 动 化 学 报　　Vol. 34, No. 7
2008 年 7 月　　ACTA AUTOMATICA SINICA　　July, 2008

基于 Q 学习的适应性进化规划算法

张化祥 [1]　　陆 晶 [2]

摘 要 进化规划中, 个体选择变异策略特别重要. 适应性变异策略因在进化过程中动态选择个体变异策略, 能够取得较好的性能. 传统适应性变异策略都依据个体一步进化效果考察个体适应性, 没有从多步进化效果上对变异策略进行评价. 本文提出一种新的基于 Q 学习的适应性进化规划算法 QEP (Q learning based evolutionary programming), 该算法将变异策略看成行动, 考察个体多步进化效果, 并通过计算 Q 函数值, 学习个体最优变异策略. 实验表明, QEP 能够获得好的性能.

关键词 进化规划, 变异策略, Q 学习, 收益
中图分类号 TP18

An Adaptive Evolutionary Programming Algorithm Based on Q Learning

ZHANG Hua-Xiang[1]　　LU Jing[2]

Abstract Selection of mutation strategies plays an important role in evolutionary programming, and adaptively selecting a mutation strategy in each evolutionary step can achieve good performance. A mutation strategy is evaluated and selected only based on the one-step performance of mutation operators in classical adaptive evolutionary programming, and the performance of mutation operators in the delayed mutation steps is ignored. This paper proposes a novel adaptive mutation strategy based on Q learning — QEP (Q learning based evolutionary programming). In this algorithm, several candidate mutation operators are used and each is considered as an action. The evolutionary performance of delayed mutation steps is considered in calculating the Q values for each mutation operator and the mutation operator that maximizes the learned Q values is the optimal one. Experimental results show that the proposed mutation strategy achieves better performance than the existing algorithms.

Key words Evolutionary programming, mutation strategy, Q learning, reward

进化算法有遗传算法、进化规划和进化策略, 主要用于求解各种优化问题. 不同于遗传算法与进化策略, 在进化规划中, 个体只通过变异策略产生新的后代. 进化过程中通过变异实现个体的多样性, 从而可以在更大的区间内搜索问题的解, 对于学习问题的最优解具有重要意义. 进化过程中, 一方面要保持个体的多样性, 同时又要执行一定的策略使得群体中适应度好的个体以高概率在后代中保留下来, 这是进化规划中存在的搜索与执行的两难问题. 如果过分强调个体多样性, 个体很难收敛到问题的最优解, 而如果过分强调执行最优的个体, 往往会使得个体收敛到问题的局部最优解. 目前进化规划已广泛用于解决函数优化问题, 并已取得较好效果. 由于进化规划算法只通过变异单一手段实现个体多样性, 因此变异策略的选择直接影响到个体的进化速度和问题最终的求解.

个体通过变异, 实现在问题求解空间中的搜索. 目前进化规划中主要的变异策略有 Gaussian 变异 (Classical evolutionary programming, CEP)[1]、Cauchy 变异 (Fast evolutionary programming, FEP)[2]、Lévy 变异 (Lévy evolutionary programming, LEP)[3]、单点变异 (Single point mutation evolutionary programming, SPMEP)[4] 及混合策略变异[5]. 在 Gaussian 变异策略中, 个体通过增加一个正态分布随机数产生后代. 研究表明, 个体经过一定次数的变异后, 产生的后代偏离原个体的距离与进化步数的平方根成正比[3]. Cauchy 变异采用类似于 Gaussian 变异的策略, 只是将原来的正态分布随机数换成 Cauchy 随机数. Cauchy 分布的特点决定了 Cauchy 变异策略生成的个体与祖先的偏移量要大于 Gaussian 变异策略, 个体的进化更快一些. Lévy 策略产生的个体与祖先的偏移量更大, 从而导致个体进化更快.

上述个体变异在进化过程中执行单一的变异策略, 个体进化缺乏灵活性, 且该变异策略并非在每个进化步中最有效. 为此, 需要提出更加有效的变异策略. 研究表明, 传统的具有适应性的 Gaussian 变异策略具有更高的进化效率[6]; 适应性的 Lévy 变异[3, 7] 通过调整相应的参数使得进化更具有适应性, 从而加快个体进化; 单点变异通过随机选择变异的向量维, 强化后代的适应性, 加快个体进化. 因此, 变异策略是否具有适应性直接影响到个体的进化效果. 通过计算父代与子代适应度的差异, 调整个体进化步伐[8], 同样可以达到加速变异的目的.

为提高个体变异适应性, 文献 [5] 提出了一种混合进化算子. 该算子将上述四种变异策略同时应用到个体进化中. 每次变异时, 个体依据一定标准, 从四种策略中选择一种, 据此产生新的个体. 研究表明, 基于该混合策略的进化具有好的性能. 文献 [9] 给出了一种进行局部快速微调的进化算法, 通过在变异中对适应度高或低的个体分别进行诱导和随机动态区域变异, 增强个体的变异适应性, 加快个体进化. 结果表明该方法对全局最优解邻域进行搜索, 能较快找到高精度的数值解.

上述适应性变异策略的核心是在个体进化的每一步, 依据情况调整个体变异策略, 而不是在个体的整个进化过程中采用单一的变异策略. 同时, 上述适应性变异策略依据个体单步进化性能调整变异策略. 如果我们依据个体多步进化性能来调整个体的变异策略, 即通过向前考察几步个体进化效果, 并将该信息向回传递, 影响个体选择变异策略, 对个体进化将更加有利. 基于这样的想法, 本文提出一种新的适应性快速进化规划算法 QEP (Q learning based evolutionary programming), 将 QEP 应用于一些基准测试函数, 结果表明, QEP 优于上述算法或具有和其中最优算法相同的性能.

本文首先给出要求解的问题, 描述基本进化策略, 然后给出新算法, 并通过实验结果说明算法的有效性. 最后为本文结论.

1 基本进化策略

本文研究如何通过进化规划求解实值连续函数的全局最

收稿日期 2007-02-14　收修改稿日期 2007-08-11
Received February 14, 2007; in revised form August 11, 2007
国家自然科学基金 (90612003), 山东省中青年科学家科研奖励基金 (2006BS01020), 山东省自然科学基金 (Y2007G16) 资助
Supported by National Natural Science Foundation of China (90612003), Young Scholar Research Fund of Shandong Province (2006BS01020), and Natural Science Foundation of Shandong Province (Y2007G16)
1. 山东师范大学计算机系 济南 250014　2. 山东财政学院计算机系 济南 250014
1. Department of Computer Science, Shandong Normal University, Jinan 250014　2. Department of Computer Science, Shandong University of Finance, Jinan 250014
DOI: 10.3724/SP.J.1004.2008.00819

优解. 给定如下函数

$$\min f(\boldsymbol{x}), \quad \text{s.t.} \quad x_i \in [L_i, U_i]$$

其中, $\boldsymbol{x}$ 是一个 μ 维实值向量, $f(\boldsymbol{x})$ 为实值连续函数.

进化规划通过群体的进化可求解函数极值. 给定第 j 代的一个个体的任一个分量 $x_i(j)$ 及其对应的参数 $\sigma_i(j)$, Gaussian 变异策略如下

$$\sigma_i(j+1) = \sigma_i(j)\exp(\tau' N(0,1) + \tau N_i(0,1)) \tag{1}$$

$$x_i(j+1) = x_i(j) + \sigma_i(j+1)N_i(0,1) \tag{2}$$

其中 $N_i(0,1)$ 为针对不同的分量产生的正态随机数, $N(0,1)$ 为个体所有分量都相同的正态随机数. τ' 和 τ 分别为 $1/\sqrt{2\mu}$ 和 $1/\sqrt{2\sqrt{\mu}}$.

类同于 Gaussian 变异策略, Cauchy 变异策略与 Lévy 变异策略保持式 (1) 不变, 将式 (2) 分别变为式 (3) 和式 (4)

$$x_i(j+1) = x_i(j) + \sigma_i(j+1)\delta_i \tag{3}$$

$$x_i(j+1) = x_i(j) + \sigma_i(j+1)L_i(\beta) \tag{4}$$

其中式 (3) 中的 δ_i 为标准 Cauchy 分布随机数, 式 (4) 中的 $L_i(\beta)$ 为参数为 β 的 Lévy 随机数.

上述三种变异策略中, 个体每次变异在其每个分量上同时进行. 单点变异策略[4] 有所不同, 个体每次变异只在其中的一个分量上进行. 随机选择分量 x_i, 进化策略如下

$$\begin{aligned} \sigma_i(j+1) &= \sigma_i(j)\exp(-\alpha) \\ x_i(j+1) &= x_i(j) + \sigma_i(j+1)N_i(0,1) \end{aligned} \tag{5}$$

上述变异策略表明, Gaussian 与 Cauchy 变异策略对个体的所有分量同时采用相同的变异策略; 适应性 Lévy 变异策略通过调节参数 β, 在下一代中选择进化更好的个体, 不同的 β 值对应不同的后代, 通过增加产生后代的个数并从中选优, 提高进化过程的适应性; 单点变异通过随机选择个体的变异分量, 且在变异开始时, 变异的步子比较大, 随着变异步数的增加, 变异的步子逐步变小. 单点变异策略实际上是进化开始阶段在更大的假设空间随机搜索的成分比较大, 随着进化步数的增加, 进化的步子逐步减小, 个体在更小的空间内搜索, 此时执行的成分逐步增加. 通过这种适应性进化策略, 求解函数的最优解; 混合变异策略在进化开始时给定个体可选的几种变异策略, 并设定每种策略被选中的概率. 每个个体都以概率的方式选择变异策略, 并据此产生新的个体, 根据后代与父代个体的适应度调整个体各变异策略对应的概率. 该过程通过不断地调整变异策略对应概率实现在不同的进化步中改变进化策略, 达到更好进化的目的.

研究表明, 带参数的 Lévy 变异、单点变异及混合变异策略都比传统变异策略具有好的性能. 而上述具有适应性的变异策略都只是在进化的一步上考虑适应性. 我们知道, 作决策前, 如果能够多向前考虑几步, 做出正确决定的可能性就会更大. 为此, 在选择变异策略时, 我们试图通过多步进化效果, 评价变异策略的好坏, 增加个体进化适应性.

2 适应性进化策略

Q 学习[10] 通过选择最大化代理带折扣的累积收益的行动, 可以学习到代理的最优行动序列. 进化规划中, 若把个体变异策略看成行动, 这样个体选择最优变异策略就转化为代理选择最优行动. 基于带折扣方式计算代理累积收益, 并选择最大化累积收益对应行动的思想, 就是在选择行动前考虑行动的立即效果及多步滞后效果. 进化过程中, 个体在选择变异策略时, 以上各种适应性策略都只是考虑个体单步进化的效果. 如果我们通过考虑个体多步进化效果来选择个体的变异策略, 进化效果应该更好.

2.1 算法分析

对于群体中的任一个体, 假设我们向前看 $m+1$ 步, 个体开始选择的变异策略为 a. 在忽略行动变量的情况下, 可以计算个体采用行动 a 时的收益为

$$Q(a) = r(a) + \gamma Q(a^{(1)}) + \cdots + \gamma^m Q(a^{(m)}) \tag{6}$$

其中 $r(a)$ 为个体采用变异策略 a 的立即收益, 此时个体进化了一次. 新生成的个体采用 $a^{(1)}$ 生成新个体, 此时收益记为 $Q(a^{(1)})$, 依此类推, i 次进化后, 新生成的个体采用 $a^{(i)}$ 生成新个体, 此时的收益记为 $Q(a^{(i)})$. 式 (6) 为个体采用 $a, a^{(1)}, \cdots, a^{(m)}$ 变异策略序列的累积收益, $\gamma\,(0 \le \gamma < 1)$ 为折扣因子, γ 决定滞后收益对最优行动选择的影响. 当 $\gamma = 0$ 时, Q 值的计算降为只向前看一步; γ 接近 1 时, 滞后收益对最优行动选择的影响变得越来越大.

个体选择最优变异策略 $a^* = \arg\max_{a\in A}(Q(a))$, 其中 A 为个体可选变异策略的集合.

我们定义个体立即收益 $r(a) = f_p(a) - f_o(a)$, 其中 $f_p(a)$ 为父代个体对应的函数值, $f_o(a)$ 为采用变异策略 a 后生成的子代个体对应的函数值. 个体适应度函数定义为个体对应函数值的相反数. 此时, 给定个体可选变异策略个数 n, 一次进化后, 每个变异策略可产生一个后代, 此时共产生 n 个后代. 每个后代采用同样的方式变异, 二次进化后, 产生 n^2 个体. 同理可知 m 次进化后, 产生最新后代个数为 n^m. 要计算个体最优变异策略, 我们需进行指数级的计算, 该方法计算量太大, 不可行. 为此将问题简化如下: 个体第一次进化后产生 n 个个体, 该 n 个个体再次进化, 每个个体产生 n 个后代, 通过下面的方式从这 n 个后代中选择一个保留下来. 通过计算每个变异策略的立即收益, 并采用 Boltzmann 分布计算每个个体被保留下来的概率

$$p(a_i) = \frac{\mathrm{e}^{\frac{r(a_i)}{T}}}{\sum\limits_{i=1}^{n} \mathrm{e}^{\frac{r(a_i)}{T}}}$$

据此概率, 我们选择其中的一个变异策略并保留其后代. 被保留的后代再采用同样的方式产生 n 个后代, 并保留其中的一个后代. 通过这种简化后, 一个个体, 一次进化后产生 n 个后代, 每个后代再次进化, 只保留一个新的后代, 依此类推如图 1 (见下页) 所示. 设 m 为个体进化次数, 则 m 次进化后, 一个个体共产生的新的个体数为: 第一次进化产生 n 个后代; 第二次进化产生 n^2 后代, 保留其中的 n 个; 第三次进化产生 n^2 后代, 保留其中的 n 个; 第 m 次进化产生 n^2 后代, 保留其中的 n 个, 则 m 次进化后, 累计产生的个体数为 $n + (m-1)n^2$, 用于计算 Q 值需要保留的个体数为 mn, 此时问题时间复杂度降为多项式级.

将立即收益的定义代入式 (6) 并合并同类项后得

$$\begin{aligned} Q(a) = f_p(a) - (1-\gamma)f_o(a) - \gamma(1-\gamma)f_o(a^{(1)}) - \cdots - \\ \gamma^m f_o(a^{(m)}) \end{aligned} \tag{7}$$

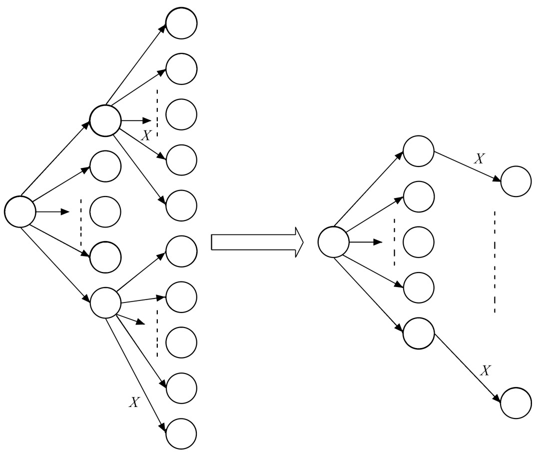

图 1　个体进化选择图

Fig. 1　The illustration of individual evolution and selection

$\forall m$, 有 $\lim_{\gamma\to 1}\gamma^m(1-\gamma)=0$ 和 $\lim_{\gamma\to 1}\gamma^m=1$. 根据式 (7), 可得 $\lim_{\gamma\to 1}Q(a)=f_p(a)-f_o(a^{(m)})$. 此时, 最优的变异策略为 $a^*=\arg\max_{a\in A}\lim_{\gamma\to 1}Q(a)=\arg\max_{a\in A}(f_p(a)-f_o(a^{(m)}))$. 可以看出, 当 γ 接近 1 时, 个体最优的变异策略为能够使得个体函数值与第 m 次变异后个体函数值的差最大化的变异策略.

基于前面的叙述, 从第二代起, 后代个体以概率的方式被选择. 因此, 可以计算 Q 值的期望值为

$$\begin{aligned}\mathrm{E}Q(a)=&f_p(a)-f_o(a)+\gamma\sum_{a^{(1)}\in A}P(a^{(1)})(f_o(a)-\\&f_o(a^{(1)}))+\gamma^2\sum_{a^{(2)}\in A}P(a^{(2)})(f_o(a^1)-f_o(a^{(2)}))+\\&\cdots+\gamma^m\sum_{a^{(m)}\in A}P(a^{(m)})(f_o(a^{(m-1)})-f_o(a^{(m)}))\end{aligned}$$

已知 $\sum_{a^{(k)}\in A}P(a^{(k)})=1$, 其中 $k\in\{1,\cdots,m\}$, 则上式可简化为

$$\begin{aligned}\mathrm{E}Q(a)=&f_p(a)-f_o(a)+\gamma(f_o(a)-\sum_{a^{(1)}\in A}P(a^{(1)})f_o(a^{(1)}))+\\&\gamma^2(f_o(a^{(1)})-\sum_{a^{(2)}\in A}P(a^{(2)})f_o(a^{(2)}))+\cdots+\\&\gamma^m(f_o(a^{(m-1)})-\sum_{a^{(m)}\in A}P(a^{(m)})f_o(a^{(m)}))\end{aligned}$$

Boltzmann 分布中, 当温度 $T\to 0$ 时, 最优变异策略被选中的概率趋于 1, 而非最优变异策略被选中的概率接近于 0. 此时, 如果有

$$\begin{aligned}&\lim_{k\to\infty}(f_o(a^{(k-1)})-\sum_{a^{(k)}\in A}P(a^{(k)})f_o(a^{(k)}))=\\&\lim_{k\to\infty}f_o(a^{(k-1)})-\lim_{k\to\infty}\sum_{a^{(k)}\in A}P(a^{(k)})f_o(a^{(k)})=\\&f_o(a^{(k-1)})-\lim_{k\to\infty}\sum_{a^{(k)}\in A}P(a^{(k)})f_o(a^{(k)})\end{aligned}$$

此时最优变异策略 a^* 对应概率 $P(a^*)\to 1$, 而其他变异策略对应的概率趋于 0. 故有 $\lim_{k\to\infty}\sum_{a^{(k)}\in A}P(a^{(k)})\cdot f_o(a^{(k)})=f_o(a^*)$. 此时, 我们得到下式

$$\lim_{k\to\infty}(f_o(a^{(k-1)})-\sum_{a^{(k)}\in A}P(a^{(k)})f_o(a^{(k)}))=f_o(a^{(k-1)})-f_o(a^*)\tag{8}$$

其中 $a^*=\arg\min_{a\in A}f_o(a)$.

上面的分析说明, 在进化的初始阶段, 个体在变异策略空间进行探索, 随着进化步数的增加, 个体探索的成分逐步减少, 越来越趋向于选择适应度函数大 (最优化函数值小) 的变异策略, 此时个体执行最优变异策略的成分逐步增加.

2.2　算法设计

为学习个体进化的最优变异策略, 需考虑变异策略单步收益及滞后收益, Q 学习可满足这方面的需要. 个体选择变异策略时, 借助搜索与执行的思想, 在进化的初始阶段, 设置比较高的温度参数, 通过 Boltzmann 分布选择后代个体. 将被保留下来的个体函数与其直接父代函数之间的差值作为变异策略的立即收益, 并利用学习搜索最优变异策略. QEP 算法设计如下:

QEP 进化算法 (QEP evolutionary algorithm)

1) 生成由 u 个个体组成的进化群体, 每个个体都表示为 (x,σ) 的形式, 并设定计算 Q 需要进化的步数 m;

2) 按照下面的方法学习每个个体最优变异策略

a) 对每个个体, 利用给定的 n 个变异策略产生新的后代, 并设定 $t=1$;

b) Do while $t<m$

每个后代利用式 (1) 的方式产生 n 个后代, 并依据 Boltzmann 分布选择其中一个; $t++$;

c) 计算每个变异策略对应的 Q, 并选择使 Q 最大化的变异策略作为个体的变异策略, 选择其对应的后代作为个体的后代, 同时忽略掉其他 $n-1$ 个后代.

3) 按照与 CEP 相同的方式选择个体组成新的群体;

4) 回到 2) 继续执行, 直到满足设定的停止条件.

3　实验结果分析

我们采用标准的测试函数[4] 验证算法的有效性, 这些函数可分为高维函数与低维函数、单模函数与多模函数以及连续函数与非连续函数. 实验中, 我们设定群体大小为 100, 折扣因子 $\gamma=0.5$ 及 $m=3$, 每种变异策略中参数的设置同原设置. 我们对每种算法进行 50 次计算, 并求得平均结果. 表 1 中, “Mean best” 为 50 个最终进化代中最优的个体对应的最小的函数值的平均, “Std dev” 为它们的方差. 从表 1 (见下页) 的实验结果看, 基于立即收益与滞后收益的适应性变异策略性能好于四种变异策略中任意一个. 经验表明做决定前向前多看几步, 决定的正确性就会增加. 而基于 Q 学习的适应性进化规划正是在算法中实现了这一思想才获得了好的性能.

γ 取值影响 Q 计算. 为此, 我们考察 γ 取不同值时变异策略的选择和算法的性能. 极端情况下, γ 取值过小, 滞后收益的作用太小, 当 $\gamma=0$ 时, 进化退化为单步适应性进化; γ 取值过大时, 产生的后代太多, 计算量过大. 表 2 给出了 γ 三种取值的实验结果. 数据表明, $\gamma=0.1$ 时算法性能最差, 因为此时滞后收益几乎不被考虑在内. 而 $\gamma=0.5$ 与 $\gamma=0.8$ 时, 算法的性能难于比较, 在某些函数上 $\gamma=0.5$ 时算法性能较好, 而对于另外一些函数, $\gamma=0.8$ 时算法性能较好. 直观理解, 如果设定 $\gamma=1$, 我们从个体进化的第一代一直可以看到最后一代, 理论上可以正确选择每一代的最优变异策略, 而算法只是向前考察了 3 步, 这可能是导致 $\gamma=0.8$ 时算法性能并不总是最好的原因.

表 1 QEP 与其他算法性能比较
Table 1 Performance comparison among QEP and other algorithms

F	N	QEP Mean best, Std Dev	SPMEP Mean best, Std Dev	LEP Mean best, Std Dev
f_1	1 500	16.7, 20.2	70.4, 29.8	180.9,132.7
f_2	1 500	$1.0\,e^{-4}$, $1.1\,e^{-5}$	$3.3\,e^{-4}$, $2.5\,e^{-4}$	$5.4\,e^{-4}$, $2.5\,e^{-4}$
f_3	2 000	$2.5\,e^{-4}$, $1.9\,e^{-5}$	$4.3\,e^{-4}$, $2.4\,e^{-4}$	$2.5\,e^{-3}$, $1.5\,e^{-3}$
f_4	5 000	$4.3\,e^{-4}$, $5.2\,e^{-4}$	$6.5\,e^{-3}$, $2.1\,e^{-3}$	0.82, 0.53
f_5	1 500	0, 0	0, 0	0, 0
f_6	3 000	$1.0\,e^{-4}$, $8.5\,e^{-3}$	$3.3\,e^{-4}$, $1.8\,e^{-3}$	$5.4\,e^{-4}$, $5.3\,e^{-3}$
f_7	9 000	−12 569.5, 0	−12 569.5, 0	−11 987.3, 58.5
f_8	5 000	$1.0\,e^{-5}$, $2.3\,e^{-6}$	$4.8\,e^{-5}$, $3.2\,e^{-5}$	$6.7\,e^{-3}$, $3.2\,e^{-3}$
f_9	1 500	$1.6\,e^{-3}$, $4.4\,e^{-4}$	$1.9\,e^{-3}$, $5.4\,e^{-4}$	$5.5\,e^{-2}$, $2.4\,e^{-2}$
f_{10}	2 000	$5.6\,e^{-4}$, $1.2\,e^{-3}$	$5.6\,e^{-3}$, $1.7\,e^{-3}$	$9.4\,e^{-3}$, $5.3\,e^{-3}$
f_{11}	1 500	$6.9\,e^{-7}$, $1.5\,e^{-7}$	$8.5\,e^{-7}$, $1.5\,e^{-7}$	$6.1\,e^{-6}$, $1.5\,e^{-6}$
f_{12}	1 500	$1.1\,e^{-5}$, $1.3\,e^{-6}$	$1.4\,e^{-5}$, $9.1\,e^{-6}$	$9.8\,e^{-5}$, $1.2\,e^{-5}$
f_{13}	100	1, 0	1, $1.2\,e^{-5}$	0.998, $1.3\,e^{-6}$
f_{14}	4 000	$3.08\,e^{-4}$, $2.2\,e^{-5}$	$4.46\,e^{-4}$, $1.5\,e^{-4}$	$4.2\,e^{-4}$, $2.7\,e^{-4}$
f_{15}	100	−1.03, 0	−1.03, 0	−1.03, $1.6\,e^{-4}$
f_{16}	100	0.398, 0	0.398, 0	0.398, 0
f_{17}	100	3, 0	3, 0	3, $5.4\,e^{-4}$
f_{18}	100	−3.86, 0	−3.86, 0	−3.9, $1.2\,e^{-3}$
f_{19}	200	−3.32, $1.6\,e^{-4}$	−3.25, $5.5\,e^{-2}$	−2.99, $4.8\,e^{-2}$
f_{20}	100	−10.3, $2.4\,e^{-2}$	−6.68, 0.74	−5.67, 1.35
f_{21}	100	−10.4, 0.24	−9.3, 0.89	−5.54, 1.69
f_{22}	100	−10.54, $1.1\,e^{-2}$	−6.89, 2	−8.43, 1.1

表 2 $\gamma = 0.1,\ 0.5,\ 0.8$ 时 QEP 的性能比较
Table 2 Performance comparison among QEPs when $\gamma = 0.1, 0.5$, and 0.8

F	N	$\gamma = 0.5$ Mean best, Std Dev	$\gamma = 0.1$ Mean best, Std Dev	$\gamma = 0.8$ Mean best, Std Dev
f_1	1 500	16.7, 20.2	28.3, 24.5	15.9, 16.8
f_2	1 500	$1.0\,e^{-4}$, $1.1\,e^{-5}$	$1.0\,e^{-4}$, $2.1\,e^{-4}$	$1.0\,e^{-4}$, $1.1\,e^{-4}$
f_3	2 000	$2.5\,e^{-4}$, $1.9\,e^{-5}$	$3.9\,e^{-4}$, $1.1\,e^{-4}$	$2.9\,e^{-4}$, $1.1\,e^{-5}$
f_4	5 000	$4.3\,e^{-4}$, $5.2\,e^{-4}$	$3.5\,e^{-3}$, $8.9\,e^{-4}$	$3.9\,e^{-4}$, $4.4\,e^{-4}$
f_5	1 500	0, 0	0, 0	0, 0
f_6	3 000	$1.0\,e^{-4}$, $8.5\,e^{-3}$	$5.6\,e^{-3}$, $1.1\,e^{-2}$	$2.1\,e^{-4}$, $8.4\,e^{-3}$
f_7	9 000	−12 569.5, 0	−12 569.5, $8.5\,e^{-3}$	−12 569.5, 0
f_8	5 000	$1.0\,e^{-5}$, $2.3\,e^{-6}$	$2.1\,e^{-5}$, $6.4\,e^{-5}$	$1.4\,e^{-5}$, $1.1\,e^{-5}$
f_9	1 500	$1.6\,e^{-3}$, $4.4\,e^{-4}$	$1.7\,e^{-3}$, $2.4\,e^{-3}$	$1.1\,e^{-4}$, $4.1\,e^{-5}$
f_{10}	2 000	$5.6\,e^{-4}$, $1.2\,e^{-3}$	$6.4\,e^{-4}$, $3.2\,e^{-3}$	$4.3\,e^{-4}$, $1.1\,e^{-3}$
f_{11}	1 500	$6.9\,e^{-7}$, $1.5\,e^{-7}$	$7.4\,e^{-7}$, $3.4\,e^{-7}$	$4.5\,e^{-7}$, $1.0\,e^{-7}$
f_{12}	1 500	$1.1\,e^{-5}$, $1.3\,e^{-6}$	$1.3\,e^{-5}$, $4.6\,e^{-6}$	$1.2\,e^{-5}$, $2.4\,e^{-6}$
f_{13}	100	1, 0	1, $3.2\,e^{-7}$	1, 0
f_{14}	4 000	$3.08\,e^{-4}$, $2.2\,e^{-5}$	$4.15\,e^{-4}$, $3.4\,e^{-5}$	$2.99\,e^{-4}$, $1.5\,e^{-5}$
f_{15}	100	−1.03, 0	−1.03, 0	−1.03, 0
f_{16}	100	0.398, 0	0.398, 0	0.398, 0
f_{17}	100	3, 0	3, 0	3, 0
f_{18}	100	−3.86, 0	−3.86, 0	−3.86, 0
f_{19}	200	−3.32, $1.6\,e^{-4}$	−3.32, $8.9\,e^{-4}$	−3.32, $1.1\,e^{-4}$
f_{20}	100	−10.3, $2.4\,e^{-2}$	−10.15, $6.9\,e^{-2}$	−10.28, $1.5\,e^{-2}$
f_{21}	100	−10.4, 0.24	−10.3, 0.54	−10.4, 0.08
f_{22}	100	−10.54, $1.1\,e^{-2}$	−6.78, $5.2\,e^{-2}$	−10.55, $1.1\,e^{-3}$

4 小结

个体在进化过程中如采用单一变异策略, 其变异不具有适应性. 当进化策略中有可调参数时, 如在 LEP 进化策略中有可调参数, 通过选择较好参数产生个体, 可加快个体进化. 当有多个变异策略同时存在时, 进化个体应在进化过程中动态选择变异策略, 以便更好进化. 进化过程中, 如果能够基于个体多步进化效果选择个体变异策略, 应该会得到更好的效果. 正是基于这样的思想, 我们提出了基于 Q 学习的适应性进化规划算法 QEP. QEP 直接关注后代个体函数与父代个体函数的差别, 而不是通过随机漫步决定二者的区别, 对于求解函数的最优值更直接. 实验结果表明, 该方法好于已提出的各种变异策略算法. 由于计算 Q 时需要产生多个个体, 因此, 算法增加了计算时间复杂度, 这是该算法的不足.

References

1 Fogel L J, Owens A J, Walsh M J. *Artificial Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming*. New York: Wiley-Interscience, 1999

2 Yao X, Liu Y, Lin G M. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 1999, **3**(2): 82−102

3 Lee C Y, Yao X. Evolutionary programming using mutations based on the Lévy probability distribution. *IEEE Transactions on Evolutionary Computation*, 2004, **8**(1): 1−13

4 Ji M J, Tang H W, Guo J. A single-point mutation evolutionary programming. *Information Processing Letters*, 2004, **90**(6): 293−299

5 Dong H, He J, Huang H, Hou W. Evolutionary programming using a mixed mutation strategy [Online], available: http://www.cs.bham.ac.uk/ jxh/hejunpl.html, December 20, 2006

6 Fogel D B. Evolving Artificial Intelligence [Ph. D. dissertation], California, USA: University of California, 1992

7 Iwamatsu M. Generalized evolutionary programming with Lévy-type mutation. *Computer Physics Communications*, 2002, **147**(1): 729−732

8 Lee S H, Jun H B, Sim K B. Performance improvement of evolution strategies using reinforcement learning. In: Proceedings of IEEE International Fuzzy Systems Conference. Seoul, Korea: IEEE, 1999. 639−644

9 Liu Xi-Chun, Yu Shou-Yi. A genetic algorithm with fast local adjustment. *Chinese Journal of Computers*, 2006, **29**(1): 100−105
(刘习春, 喻寿益. 局部快速微调遗传算法. 计算机学报, 2006, **29**(1): 100−105)

10 Sutton R S, Barto A G. *Reinforcement Learning: An Introduction*. Cambridge: MIT Press, 1998

张化祥 山东师范大学信息科学与工程学院计算机系教授, 博士. 主要研究方向为机器学习, 进化计算, Web 信息处理. 本文通信作者. E-mail: huaxzhang@163.com
(**ZHANG Hua-Xiang** Professor, Ph. D. in the Department of Computer Science, Shandong Normal University. His research interest covers machine learning, evolutionary computation, and Web information processing. Corresponding author of this paper.)

陆 晶 山东财政学院计算机系副教授. 主要研究方向为进化计算, 电子商务. E-mail: zhaogang@sdcz.gov.cn
(**LU Jing** Associate professor in the Department of Computer Science, Shandong University of Finance. Her research interest covers evolutionary computation and electronic commerce.)