第 24 卷第 3 期
2007 年 6 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 24 No. 3
Jun. 2007

文章编号：1000−8152(2007)03−0343−06

一种求解全局优化问题的新混合遗传算法

李　宏[1,2], 焦永昌[1], 张　莉[2], 王宇平[3]

(1. 西安电子科技大学 天线与微波国家重点实验室, 陕西 西安 710071;

2. 西安电子科技大学 理学院, 陕西 西安 710071; 3. 西安电子科技大学 计算机学院, 陕西 西安 710071)

摘要: 把简化的二次插值法融入实数编码遗传算法, 构成适于求解全局优化问题的混合遗传算法, 该混合算法可以较好解决遗传算法的早熟收敛问题, 提高了收敛速度, 改善了解的质量, 并减少了计算量. 由于该混合遗传算法对目标函数的性质没有要求, 适合求解大规模问题和工程实际问题. 通过对23个标准测试函数的仿真实验, 并和已有算法的比较, 结果表明本文提出的混合遗传算法是非常有效的.

关键词: 二次插值法; 遗传算法; 全局优化; 混合遗传算法

中图分类号: TP18　　**文献标识码**: A

Novel hybrid genetic algorithm for global optimization problems

LI Hong[1,2], JIAO Yong-chang[1], ZHANG Li[2], WANG Yu-ping[3]

(1. National Laboratory of Antennas and Microwave Technology, Xidian University, Xi'an Shaanxi 710071, China;
2. School of Science, Xidian University, Xi'an Shaanxi 710071, China;
3. School of Computer Science and Engineering, Xidian University, Xi'an Shaanxi 710071, China)

Abstract: A novel hybrid genetic algorithm for global optimization problems is proposed in this paper. A real-coded genetic algorithm is addressed. A simplified quadratic interpolation method is then integrated into the genetic algorithm. The hybrid genetic algorithm is capable of avoiding the premature convergence, improving the global search ability of the algorithm and the accuracy of the minimum function value, as well as reducing the computational burden. Simulation results on 23 benchmark problems show that the proposed hybrid genetic algorithm is efficient and effective in comparison with other existing algorithms.

Key words: quadratic interpolation method; genetic algorithm; global optimization; hybrid genetic algorithm

1　引言(Introduction)

考虑全局优化问题

$$\min_{\boldsymbol{x}\in\mathcal{B}} f(\boldsymbol{x}), \tag{1}$$

其中: $\boldsymbol{x} = (x_1, \cdots, x_n)^\top$, $\mathcal{B} = \{(l_1, u_1) \times \cdots \times (l_n, u_n) | l_i \leqslant x_i \leqslant u_i, i = 1, 2, \cdots, n\}$.

全局优化问题(1)具有广泛的工程应用背景, 其求解方法越来越受到人们的重视, 传统算法都是局部优化方法, 优化结果与初值有关, 并且对函数的性态有要求. 许多工程实际问题不但参数多, 而且目标函数性态差, 一般难以用基于梯度的传统算法来求解. 近年来, 进化算法(包括遗传算法、进化策略、进化规划)已成功地用于求解各种优化问题[1~6], 它不需要导数信息, 对函数的性态没有要求, 而且适用范围广、鲁棒性强, 适于并行运算, 与基于梯度的传统算法具有很好的互补性. 遗传算法作为进化算法的一种, 在各种问题的求解与应用中展现了其独特的魅力[1,3~6], 但在理论和应用上也暴露出诸多不足和缺陷, 如对某些复杂问题而言, 它易趋于早熟收敛而陷于局部最优解[1,4], 另外, 也存在收敛速度慢、计算量大等问题[1]. 为克服这些问题, 早在1989年Goldberg就提出混合方法的框架[1], 把GA与传统的、基于知识的启发式搜索技术相结合, 来改善基本遗传算法的局部搜索能力, 使遗传算法脱离早熟收敛状态而加速接近全局最优解. 混合遗传算法在一定程度上可以提高遗传算法的优化质量和收敛效率, 因此, 出现了各种各样的混合遗传算法来求解各类优化问题[1,3~6].

本文把简化的二次插值法作为一个局部搜索算子, 融入到设计的实数编码遗传算法中, 构成适于求

收稿日期: 2005−10−08; 收修改稿日期: 2006−04−26.
基金项目: 国家自然科学基金资助项目(60171045, 60374063).

解函数优化问题的混合遗传算法, 该混合方法可以较好解决遗传算法的早熟收敛问题, 并对目标函数的性质(可导性、连续性等)没有要求, 适合求解大规模问题和工程实际问题. 通过对23个标准测试函数的仿真实验, 并和已有算法做了比较, 结果表明本文提出的混合遗传算法是非常有效的.

2 实数编码的遗传算法(Real-coded genetic algorithm)

本文采用实数编码, 用向量$\boldsymbol{x} = (x_1, \cdots, x_n)^\top$作为一个个体(或染色体)来表示问题(1)的一个解.

1) 初始种群.

在n维界空间$\mathcal{B}$中随机产生pop个服从均匀分布的个体构成初始种群, pop表示种群规模.

2) 适应度函数.

由于所讨论的问题(1)是最小化问题, 为方便起见, 设定: 适应度越小, 结果越好. 适应度函数表示为:

$$\mathrm{fit}(\boldsymbol{x}) = f(\boldsymbol{x}). \tag{2}$$

判断个体$\boldsymbol{x} = (x_1, \cdots, x_n)^\top$是否满足界约束$\mathcal{B} = (l_1, u_1) \times \cdots \times (l_n, u_n)$. 对$i \in \{1, 2, \cdots, n\}$, 若分量$x_i \notin (l_i, u_i)$, 则令$\bar{x_i} = l_i + \lambda_i(u_i - l_i)$, 随机数$\lambda_i \in [0, 1]$; 若分量$x_i \in (l_i, u_i)$, 则令$\bar{x_i} = x_i$. 这样, $\bar{\boldsymbol{x}} = (\bar{x_1}, \cdots, \bar{x_n})^\top$ 就是满足界约束的新个体.

3) 交叉算子.

以杂交概率p_c随机选取两个父代个体, 记为$\boldsymbol{x}^1 = (x_1^1, x_2^1, \cdots, x_n^1)^\top$和$x^2 = (x_1^2, x_2^2, \cdots, x_n^2)^\top$. 则由下式产生后代个体$\boldsymbol{y} = (y_1, y_2, \cdots, y_n)^\top$:

$$\boldsymbol{y} = \begin{cases} \boldsymbol{x}^1 + \Gamma(\boldsymbol{x}^1 - \boldsymbol{x}^2), \mathrm{fit}(\boldsymbol{x}^1) \leqslant \mathrm{fit}(\boldsymbol{x}^2), \\ \boldsymbol{x}^2 + \Gamma(\boldsymbol{x}^2 - \boldsymbol{x}^1), \mathrm{fit}(\boldsymbol{x}^1) > \mathrm{fit}(\boldsymbol{x}^2), \end{cases} \tag{3}$$

其中$\Gamma = \mathrm{diag}(\gamma_1, \cdots, \gamma_n)$, $\gamma_i \in (-1, 1)$ 且$\gamma_i \neq 0$, $i = 1, \cdots, n$.

4) 变异算子.

设p_m是变异概率, 到目前为止最好的个体(保留的最好解)为$\boldsymbol{best} = (best_1, \cdots, best_n)^\top$.

对满足$\| \boldsymbol{best} - \boldsymbol{y}\| \geqslant \varepsilon_1$ ($\varepsilon_1 = 10^{-4}$)的每个$y = (y_1, \cdots, y_n)^\top$, 生成一个随机数$r \in (0, 1)$, 若$r < p_{\mathrm{m}}$, 则由下式产生变异后代$z = (z_1, \cdots, z_n)^\top$:

$$z = \boldsymbol{best} + V \cdot |c|, \tag{4}$$

其中V是如下的对角矩阵:

$$V = \mathrm{diag}(best_1 - y_1, \cdots, best_n - y_n).$$

$|\boldsymbol{c}| = (|c_1|, |c_2|, \cdots, |c_n|)^\top$是随机搜索步长, $\boldsymbol{c}$是服从n维标准正态分布的随机向量, 即: $\boldsymbol{c} \sim (N(0, 1), \cdots, N(0, 1))^\top$.

对不满足$\|\boldsymbol{best} - y\| \geqslant \varepsilon_1$ 的每个$y = (y_1, \cdots, y_n)^\top$, 生成一个随机数$r \in (0, 1)$, 若$r < p_{\mathrm{m}}$, 则由高斯变异, 产生后代$z = (z_1, \cdots, z_n)^\top$:

$$z = \boldsymbol{best} + \Delta\boldsymbol{v}, \tag{5}$$

其中$\Delta\boldsymbol{v} \sim N(0, \sigma^2) = (N(0, \sigma_1^2), \cdots, N(0, \sigma_n^2))^\top$, 即$\Delta\boldsymbol{v}$ 服从均值为$\boldsymbol{0} = (0, \cdots, 0)^\top$, 方差为$\sigma^2 = (\sigma_1^2, \cdots, \sigma_n^2)^\top$的$n$ 维正态分布.

5) 选择算子.

为了提高种群的多样性, 选择算子采取以下策略:

在当前种群(父代)及所有后代(杂交产生的后代以及变异产生的后代)个体中, 按适应度的升序排列, 选择前$pop - N_1$个个体, 再随机产生N_1 个个体, 把这两部分个体合起来作为下一代种群, 并且保留每一代产生的最好解.

6) 终止规则.

若算法执行了最大进化代数, 则停止, 所得最好解作为近似全局最优解.

3 二次插值法(Quadratic interpolation method)

三点二次插值法是一种简单有效的线搜索方法, 它不需目标函数的导数信息, 适用范围广, 而且计算量小, 适合作为启发式的搜索算子.

三点二次插值法在文[7]中用来进行全局搜索, 本文将其作为局部搜索算子, 插入到上述的实数编码的遗传算法中, 提高算法的搜索能力, 减少计算量, 从而使遗传算法跳出局部最优解, 快速向全局最优解靠近.

设$\boldsymbol{x}^a = (x_1^a, \cdots, x_n^a)^\top$, $\boldsymbol{x}^b = (x_1^b, \cdots, x_n^b)^\top$, $\boldsymbol{x}^c = (x_1^c, \cdots, x_n^c)^\top$, 计算这三点的适应度值$f_{\mathrm{a}} = \mathrm{fit}(\boldsymbol{x}^a)$, $f_{\mathrm{b}} = \mathrm{fit}(\boldsymbol{x}^b)$, $f_{\mathrm{c}} = \mathrm{fit}(\boldsymbol{x}^c)$.

假设$f_a > f_b$, $f_c > f_b$, 则由下式得到的近似极小值点$\bar{\boldsymbol{x}} = (\bar{x}_1, \cdots, \bar{x}_n)^\top$为:

$$\bar{x}_i = \frac{1}{2}\{\frac{\mathscr{A}}{\mathscr{B}}\}, i = 1, \cdots, n. \tag{6}$$

其中: $\mathscr{A} = [(x_i^b)^2 - (x_i^c)^2]f_{\mathrm{a}} + [(x_i^c)^2 - (x_i^a)^2]f_{\mathrm{b}} + [(x_i^a)^2 - (x_i^b)^2]f_{\mathrm{c}}$, $\mathscr{B} = (x_i^b - x_i^c)f_{\mathrm{a}} + (x_i^c - x_i^a)f_{\mathrm{b}} + (x_i^a - x_i^b)f_{\mathrm{c}}$. 每一代中三点$\boldsymbol{x}^a, \boldsymbol{x}^b, \boldsymbol{x}^c$的选择如下:

在当前种群(父代)及所有后代(杂交产生的后代以及变异产生的后代)个体中, 把适应度从小到大排列, 取$\boldsymbol{x}^b$, $\boldsymbol{x}^a$, $\boldsymbol{x}^c$ 依次为前3个最好个体, 即$f_b \leqslant f_a \leqslant f_c$. 若对某些$i \in \{1, 2, \cdots, n\}$, $(x_i^b - x_i^c)f_{\mathrm{a}} + (x_i^c - x_i^a)f_{\mathrm{b}} + (x_i^a - x_i^b)f_{\mathrm{c}} < \varepsilon_2$ ($\varepsilon_2 = 10^{-6}$) 则令$\bar{\boldsymbol{x}} = \boldsymbol{x}^b$, $\mathrm{fit}(\bar{\boldsymbol{x}}) = f_b$; 否则, 由式(6), 计算出$\bar{\boldsymbol{x}}$, 并计算其适应度$\mathrm{fit}(\bar{\boldsymbol{x}})$. 若$\mathrm{fit}(\bar{\boldsymbol{x}}) \leqslant f_b$, 则将

当前种群中的最差解用$\bar{\boldsymbol{x}}$替换.

4 混合遗传算法(Hybrid genetic algorithm)

遗传算法具有全局搜索的功能, 二次插值法具有局部搜索的特点, 将两者有机地结合, 来提高算法的收敛效率和全局寻优能力.

下面给出混合遗传算法的详细步骤:

Step 0 参数: 种群规模pop, 杂交概率$p_{\rm c}$, 变异概率$p_{\rm m}$, 最大进化代数N, 合适的整数N_1.

Step 1 初始化: 置$k = 0$, 产生初始种群$P(k) = \{\boldsymbol{x}^1(k), \cdots, \boldsymbol{x}^{\rm pop}(k)\}$, 并计算每个个体的适应度$\mathrm{fit}(\boldsymbol{x}^i(k)), i = 1, \cdots, pop$.

Step 2 杂交.

Step 2.1 在区间$(0, 1)$上产生随机数r_1, 同时在区间$[1, pop]$ 上随机产生两个整数r_2和r_3且$r_2 \neq r_3$. 若$r_1 < p_{\rm c}$, 则取父代的两个个体分别为$\boldsymbol{x}^{r_2}(k)$和$\boldsymbol{x}^{r_3}(k)$, 由交叉算子(3)产生它们的中间后代$\boldsymbol{y}(k)$.

Step 2.2 重复Step2.1 pop次, 设产生$pop_{\rm c}$ 个后代个体, 构成一个中间种群集, 记为$P_{\rm c}(k) = \{\boldsymbol{y}^1(k), \cdots, \boldsymbol{y}^{{\rm pop}_{\rm c}}(k)\}$, 并计算每个个体的适应度$\mathrm{fit}(\boldsymbol{y}^i(k)), i = 1, \cdots, pop_{\rm c}$.

Step 3 变异.

Step 3.1 对$i = 1, \cdots, pop_{\rm c}$, 若$\|\boldsymbol{best} - \boldsymbol{y}^i(k)\| \geqslant \varepsilon_1$ $(\varepsilon_1 = 10^{-4})$, 则产生一个随机数$r \in (0, 1)$; 若$r < p_{\rm m}$, 则由变异算子(4), 产生中间后代$\boldsymbol{z}(k)$. 否则, 产生一个随机数$r \in (0, 1)$, 若$r < p_{\rm m}$, 则由变异算子(5), 产生中间后代$\boldsymbol{z}(k)$.

Step 3.2 通过Step3.1, 设产生$pop_{\rm m}$ 个后代个体, 构成一个中间种群集, 记为: $P_{\rm m}(k) = \{\boldsymbol{z}^1(k), \cdots, \boldsymbol{z}^{{\rm pop}_{\rm m}}(k)\}$, 并计算每个个体的适应度$\mathrm{fit}(\boldsymbol{z}^i(k)), i = 1, \cdots, pop_{\rm m}$.

Step 4 局部搜索–简化的二次插值法.

Step 4.1 在当前种群(父代)及所有后代(杂交产生的后代以及变异产生的后代)个体中, 把适应度从小到大排列, 取$\boldsymbol{x}^b, \boldsymbol{x}^a, \boldsymbol{x}^c$依次为前3个最好个体, 即$f_b \leqslant f_a \leqslant f_c$.

Step 4.2 若对某些$i \in \{1, 2, \cdots, n\}$, $(x_i^b - x_i^c)f_{\rm a} + (x_i^c - x_i^a)f_{\rm b} + (x_i^a - x_i^b)f_{\rm c} < \varepsilon_2$ $(\varepsilon_2 = 10^{-6})$, 则令$\bar{\boldsymbol{x}} = \boldsymbol{x}^b$, $\mathrm{fit}(\bar{\boldsymbol{x}}) = f_b$; 否则, 由式(6), 计算出$\bar{\boldsymbol{x}}$, 并计算其适应度$\mathrm{fit}(\bar{\boldsymbol{x}})$.

Step 4.3 若$\mathrm{fit}(\bar{\boldsymbol{x}}) \leqslant f_b$, 则将当前种群中的最差解用$\bar{\boldsymbol{x}}$替换.

Step 5 选择: 由选择策略, 选出pop个个体构成下一代种群, 并保留每一代的最好解.

Step 6 判断: 若满足停止准则, 则停止, 输出所保留的最好解作为问题(1)的近似全局最优解. 否则, 令$k = k + 1$, 转Step2.

5 数值仿真结果及讨论(Numerical simulation results and discussion)

为了评估本文提出的算法的性能, 笔者测试了文献[2]中的23个标准测试函数, 详细的情况见文献[2]中表1或附录. $f_{01} \sim f_{13}$是中等维数(30维)的函数, 其中$f_{01} \sim f_{07}$是单峰函数, 而$f_{08} \sim f_{13}$ 是多峰函数, 且有许多局部极值点. $f_{14} \sim f_{23}$ 是一些低维的多峰函数, 含有少数的局部极值点.

本文算法的参数选取如下:

种群规模$pop = 100$, 杂交概率$p_{\rm c} = 0.8$, 变异概率$p_{\rm m} = 0.3$, 最大进化代数$N = 600$ (函数的计算次数最多85000), 合适的整数$N_1 = 10$.

对每个函数独立运行15 次, 记录15次所得最好值, 平均值, 中间值, 最差值, 标准差(简记作St. Dev.), 平均进化代数(简记作MNG), 及适应度的平均计算次数(简记作MNFE). 13个30维的测试函数$f_{01} \sim f_{13}$的统计结果见表1; 10个低维的测试函数$f_{14} \sim f_{23}$的统计结果见表2. 图1∼5给出了其中5个函数($f_{06}, f_{08}, f_{10}, f_{20}, f_{23}$) 的进化曲线图, 其中横轴表示进化代数, 纵轴表示每一代的最好个体的适应度值.

笔者把HGA所获得的平均结果、标准差、平均进化代数及适应度的平均计算次数, 与X.Yao等[2]提出的两种方法CEP和FEP分别做了比较. CEP和FEP所得结果也列在表1和表2中. 表中"NA"表示没有提供结果.

1) HGA分别与CEP和FEP所获得的平均结果和标准差的比较:

除了函数f_{07}, HGA所求出的23个测试函数的平均结果, 甚至最差的结果, 都明显优于FEP和CEP所找到的平均结果. 尤其是对函数f_{14}, f_{17}, $f_{19} \sim f_{23}$, HGA能找到比文[2]报道的已有的最优解更好的结果. 对每个测试函数, 独立运行15次, HGA所得结果与全局最优解的标准差比CEP和FEP 的标准差小的多, 表明HGA更具稳定性.

2) HGA与CEP和FEP的计算量的比较:

对23个测试函数, HGA求出最好解的平均进化代数为12∼600, 适应度的平均计算次数为1505∼84842; 而CEP和FEP求出最好解的进化代数为100∼20000, 适应度的计算次数为10000∼2000000. 表明HGA的计算量比CEP和FEP小的多.

表 1 对测试函数$f_{01} \sim f_{13}$, HGA 与CEP 和FEP 的性能比较

Table 1 Comparison of HGA with CEP and FEP for benchmark problem $f_{01} \sim f_{13}$

P	最优值	方法	最好值	平均值	中间值	最差值	St.Dev.	MNG	MNFE
		HGA	3.18×10^{-13}	2.36×10^{-12}	2.47×10^{-12}	3.78×10^{-12}	2.60×10^{-12}	600	84653
f_{01}	0	CEP	NA	2.2×10^{-4}	NA	NA	5.9×10^{-4}	1500	150000
		FEP	NA	5.7×10^{-4}	NA	NA	1.3×10^{-4}	1500	150000
		HGA	3.32×10^{-8}	1.15×10^{-7}	1.13×10^{-7}	2.53×10^{-7}	1.27×10^{-7}	600	84154
f_{02}	0	CEP	NA	2.6×10^{-3}	NA	NA	1.7×10^{-4}	2000	200000
		FEP	NA	8.1×10^{-3}	NA	NA	7.7×10^{-4}	2000	200000
		HGA	6.79×10^{-13}	2.99×10^{-12}	2.09×10^{-12}	8.79×10^{-12}	3.64×10^{-12}	600	84040
f_{03}	0	CEP	NA	5.0×10^{-2}	NA	NA	6.6×10^{-2}	5000	500000
		FEP	NA	1.6×10^{-2}	NA	NA	1.4×10^{-2}	5000	500000
		HGA	1.10×10^{-3}	4.07×10^{-3}	3.76×10^{-3}	9.66×10^{-3}	4.80×10^{-3}	600	82701
f_{04}	0	CEP	NA	2.0	NA	NA	1.2	5000	500000
		FEP	NA	0.3	NA	NA	0.5	5000	500000
		HGA	9.58×10^{-4}	0.8737	0.2532	3.7854	1.92	600	84709
f_{05}	0	CEP	NA	6.17	NA	NA	13.61	20000	2000000
		FEP	NA	5.06	NA	NA	5.87	20000	2000000
		HGA	0	0	0	0	0	260	35222
f_{06}	0	CEP	NA	577.76	NA	NA	1125.76	1500	150000
		FEP	NA	0	NA	NA	0	1500	150000
		HGA	1.21×10^{-2}	1.91×10^{-2}	1.63×10^{-2}	3.73×10^{-2}	2.02×10^{-2}	500	70660
f_{07}	0	CEP	NA	1.8×10^{-2}	NA	NA	6.4×10^{-3}	3000	300000
		FEP	NA	7.6×10^{-3}	NA	NA	2.6×10^{-3}	3000	300000
		HGA	-12569.5	-12569.5	-12569.5	-12569.4	3.18×10^{-2}	600	84730
f_{08}	-12569.5	CEP	NA	-7917.1	NA	NA	634.5	9000	900000
		FEP	NA	-12554.5	NA	NA	52.6	9000	900000
		HGA	3.75×10^{-12}	3.62×10^{-11}	1.45×10^{-11}	2.61×10^{-10}	7.20×10^{-11}	500	70396
f_{09}	0	CEP	NA	89.0	NA	NA	23.1	5000	500000
		FEP	NA	4.6×10^{-2}	NA	NA	1.2×10^{-2}	5000	500000
		HGA	1.21×10^{-7}	1.14×10^{-6}	1.28×10^{-6}	2.16×10^{-6}	1.31×10^{-6}	600	83887
f_{10}	0	CEP	NA	9.2	NA	NA	2.8	1500	150000
		FEP	NA	1.8×10^{-2}	NA	NA	2.1×10^{-3}	1500	150000
		HGA	8.97×10^{-12}	5.35×10^{-11}	4.64×10^{-11}	2.37×10^{-10}	7.64×10^{-11}	600	84842
f_{11}	0	CEP	NA	8.6×10^{-2}	NA	NA	0.12	2000	200000
		FEP	NA	1.6×10^{-2}	NA	NA	2.2×10^{-2}	2000	200000
		HGA	3.10×10^{-11}	9.05×10^{-10}	3.53×10^{-10}	4.99×10^{-9}	1.62×10^{-9}	600	84713
f_{12}	0	CEP	NA	1.76	NA	NA	2.4	1500	150000
		FEP	NA	9.2×10^{-6}	NA	NA	3.6×10^{-6}	1500	150000
		HGA	2.07×10^{-10}	8.61×10^{-8}	3.70×10^{-9}	1.13×10^{-6}	2.92×10^{-7}	600	84543
f_{13}	0	CEP	NA	1.4	NA	NA	3.7	1500	150000
		FEP	NA	1.6×10^{-4}	NA	NA	7.3×10^{-5}	1500	150000

表 2　对测试函数$f_{14} \sim f_{23}$, HGA 与CEP 和FEP 的性能比较

Table 2　Comparison of HGA with CEP and FEP for benchmark problem $f_{14} \sim f_{23}$

P	最优值	方法	最好值	平均值	中间值	最差值	St.Dev.	MNG	MNFE
		HGA	0.9980038	0.9980038	0.9980038	0.9980038	0	30	3866
f_{14}	1	CEP	NA	1.66	NA	NA	1.19	100	10000
		FEP	NA	1.22	NA	NA	0.56	100	10000
		HGA	0.00030749	0.00030749	0.00030749	0.00030749	0	200	27606
f_{15}	0.0003075	CEP	NA	4.7×10^{-4}	NA	NA	3.0×10^{-4}	4000	400000
		FEP	NA	5.0×10^{-4}	NA	NA	3.2×10^{-4}	4000	400000
		HGA	−1.0316285	−1.0316285	−1.0316285	−1.0316285	0	15	1724
f_{16}	−1.0316285	CEP	NA	−1.03	NA	NA	4.9×10^{-7}	100	10000
		FEP	NA	−1.03	NA	NA	4.9×10^{-7}	100	10000
		HGA	0.3978874	0.3978874	0.3978874	0.3978874	0	15	1727
f_{17}	0.398	CEP	NA	0.398	NA	NA	1.5×10^{-7}	100	10000
		FEP	NA	0.398	NA	NA	1.5×10^{-7}	100	10000
		HGA	3.0000	3.0000	3.0000	3.0000	0	12	1505
f_{18}	3	CEP	NA	3.0	NA	NA	0	100	10000
		FEP	NA	3.02	NA	NA	0.11	100	10000
		HGA	−3.9552	−3.9552	−3.9552	−3.9552	0	15	1890
f_{19}	−3.86	CEP	NA	−3.86	NA	NA	1.4×10^{-2}	100	10000
		FEP	NA	−3.86	NA	NA	1.4×10^{-5}	100	10000
		HGA	−3.3220	−3.3220	−3.3220	−3.3220	0	40	5288
f_{20}	−3.32	CEP	NA	−3.28	NA	NA	5.8×10^{-2}	200	20000
		FEP	NA	−3.27	NA	NA	5.9×10^{-2}	200	20000
		HGA	−10.1532	−10.1532	−10.1532	−10.1532	0	30	3886
f_{21}	−10	CEP	NA	−6.86	NA	NA	2.67	100	10000
		FEP	NA	−5.52	NA	NA	1.59	100	10000
		HGA	−10.4029	−10.4029	−10.4029	−10.4029	0	30	3971
f_{22}	−10	CEP	NA	−8.27	NA	NA	2.95	100	10000
		FEP	NA	−5.52	NA	NA	2.12	100	10000
		HGA	−10.5364	−10.5364	−10.5364	−10.5364	0	30	3829
f_{23}	−10	CEP	NA	−9.10	NA	NA	2.92	100	10000
		FEP	NA	−6.57	NA	NA	3.14	100	10000

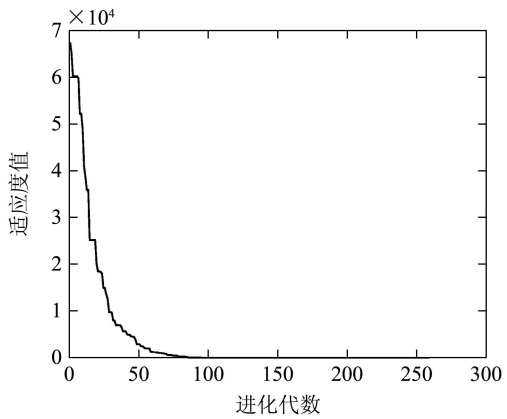

图 1　函数f_{06}的进化曲线图

Fig. 1　Convergence curve of f_{06}

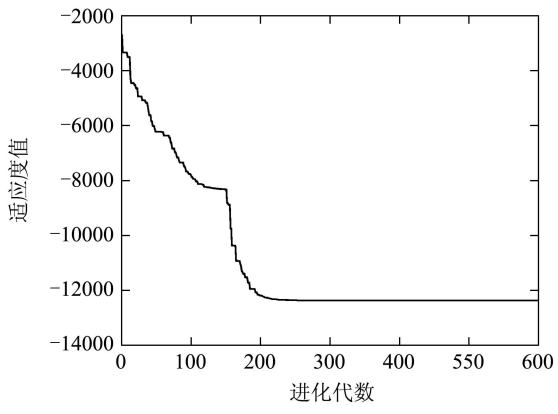

图 2　函数f_{08}的进化曲线图

Fig. 2　Convergence curve of f_{08}

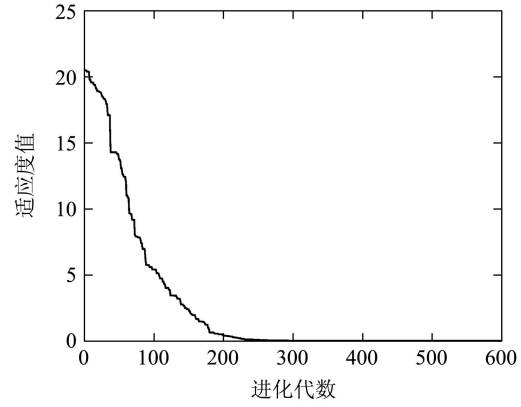

图 3 函数f_{10}的进化曲线图

Fig. 3 Convergence curve of f_{10}

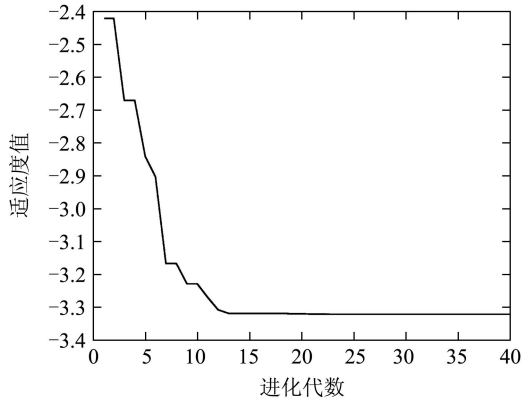

图 4 函数f_{20}的进化曲线图

Fig. 4 Convergence curve of f_{20}

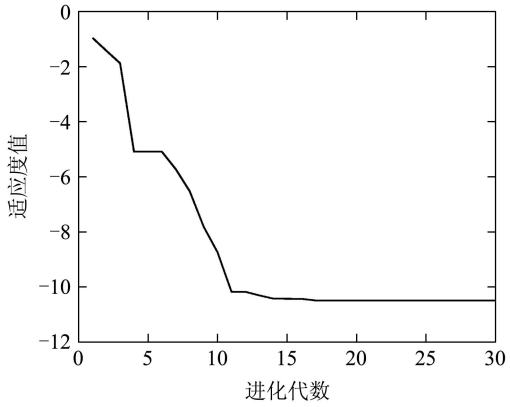

图 5 函数f_{23}的进化曲线图

Fig. 5 Convergence curve of f_{23}

6 结论(Conclusion)

二次插值法是一种简单有效的线搜索方法, 计算量小, 不需要导数信息, 把它应用到多维空间, 就是考虑到它简单方便, 避免了面搜索的计算量大的困难. 本文把简化的二次插值法作为一个局部搜索算子, 插入到设计的实数编码的遗传算法中, 构成一个混合遗传算法. 通过对23个标准测试函数的仿真实验, 充分表明该混合算法能克服早熟收敛, 能有效地改善算法的全局收敛能力, 并能提高解的质量和减少计算量.

笔者需要进一步研究的工作包括: 本文提出的混合遗传算法的全局收敛性的分析, 结合一些新的策略如邻域搜索来进一步减少算法的计算量, 以及将其应用于工程实际问题中来进一步检验混合算法的有效性.

参考文献(References):

[1] GOLDBERG D E. *Genetic Algorithms in Search, Optimization and Machine Learning*[M]. Reading, Ma: Addison Wesley, 1989.

[2] YAO X, LIU Y, LIN G M. Evolutionary programming made faster[J]. *IEEE Trans on Evol Comput*, 1999, 3(2): 82 – 102.

[3] LI H, JIAO Y C, WANG Y P. Integrating the simplified quadratic interpolation into the genetic algorithm for solving constrained optimization problems[C] *// Proc of Int Conf on Computational Intelligence and Security*. Germany: Springer, 2005, 1: 247 – 254.

[4] ONDREJ H, ANNA K. Improvements of real coded genetic algorithms based on differential operators preventing premature convergence[J]. *Advances in Engineering Software*, 2004, 35(1): 237 – 246.

[5] WANG H F, WU K Y. Hybrid genetic algorithm for optimization problems with permutation property[J]. *Computers & Operations Research*, 2004, 31(4): 2453 – 2471.

[6] 黄鹂, 陈森发, 周振国. 基于正交试验法的小生境混合遗传算法[J]. 控制理论与应用, 2004, 21(6): 1007 – 1010.
(HUANG Kun, CHEN Senfa, ZHOU Zhenguo. Orthogonal experiment method based on niche hybrid genetic algorithm[J]. *Control Theory & Applications*, 2004, 21(6): 1007 – 1010.)

[7] ALI M M, TÖORN A, VITANEN S. A numerical comparison of some modified controlled random search algorithms[J]. *J of Global Optimization*, 1997, 11(4): 377 – 385.

作者简介:

李 宏 (1972—), 男, 博士研究生, 西安电子科技大学讲师, 主要从事进化算法、优化算法与应用、智能信息处理方面的研究, E-mail: lihong@mail.xidian.edu.cn;

焦永昌 (1964—), 男, 博士, 西安电子科技大学教授, 博士生导师, 主要从事优化算法及应用、天线设计与测量等研究, E-mail: ychjiao@xidian.edu.cn;

张 莉 (1976—), 女, 博士研究生, 西安电子科技大学讲师, 主要从事神经网络、信号处理方面的研究;

王宇平 (1961—), 男, 博士, 西安电子科技大学教授, 博士生导师, 主要从事优化理论与应用、进化算法等研究.