文章编号:1001-3997(2010)03-0218-02

# 遗传算法中适应度函数的改进*

金 芬 孙春华 钟 鸣
(苏州市职业大学 机电工程系,苏州 215104)

# Improvement of fitness function in genetic algorithm

JIN Fen, SUN Chun-hua, ZHONG Ming
(Department of Mechanical and Electrical Engineering, Suzhou Vocational University, Suzhou 215104, China)

【摘 要】遗传算法引导搜索的主要依据就是个体的适应度值,因此适应度函数的设计显得尤为重要。本文兼顾保持种群的多样性和算法的收敛性,提出了一种基于指数变换的、指数系数可随进化代数动态调整的非线性适应度函数。以两个典型的测试函数为例,在相同的遗传操作和参数下,分别采用本文提出的适应度函数、线性拉伸变换及一般的指数变换适应度函数进行优化计算,计算结果表明采用提出的新适应度函数能极大地提高算法的优化精度、收敛速度和收敛概率。

**关键词:遗传算法;适应度函数;指数变换;优化计算**

**【Abstract】** *The primary basis of genetic algorithm guiding the search is the individual fitness value, so the design of fitness function is particularly important. To keep the diversity of population and the convergence of algorithm, it proposed a non-linear fitness function which based on index transformation. The index coefficient in it can adapt to evolutionary process of algorithm. Under the same genetic operators and the same parameters, calculating with the proposed fitness function, linear scaling transformation fitness function of Goldberg and general index transformation fitness function respectively, The simulation results of two typical testing functions show that the proposed fitness function can greatly improve the accuracy of optimization algorithms, the convergence speed and the probability of convergence.*

**Key words: Genetic algorithm; Fitness function; Index transformation; Optimization computation**

**中图分类号:**TH16,TP301.6 **文献标识码:**A

## 1 引言

遗传算法[1](GA Genetic Algorithm)是一种模拟生物群体遗传和进化机理的启发式优化算法,达尔文的"适者生存,优胜劣汰"是其基本的优化思想,其引导搜索的主要依据就是个体的适应度值。也就是说遗传算法依靠选择操作来引导算法的搜索方向,而选择操作是以个体的适应度作为确定性指标,从当前群体中选择适应值高的个体以生成交配池,再进行交叉和变异操作,寻找问题的最优解。如果适应度函数选得不合适可能会造成群体中基因信息的丢失,则群体中个体平均相似度增加,最终造成遗传算法早熟。因此适应度函数的选取至关重要,直接影响到遗传算法的收敛速度以及能否找到最优解。许多学者针对遗传算法中的适应度函数提出了许多有益的改进措施,Goldberg[4]在简单遗传算法的基础上提出线性拉伸方法以扩大个体之间的差异;文献[5]对线性拉伸方法中常数的选取进行了讨论,同时提出了一种非线性的动态适应度函数。本文分析了指数变换法的特点,提出了一种基于指数变换的非线性的动态适应度函数,以典型的遗传算法测试函数验证本文提出的适应度函数的有效性与可行性。

## 2 适应度函数的分析与改进

### 2.1 指数变换法的特点

指数变换法:$F^*(X)=e^{-aF(X)}$ (1)

式中:$F^*(X)$—经过指数变换后得到的新适应度;$F(X)$—原适应度,可直接取目标函数,或根据问题本身的要求及其目标函数值域的变化范围通过线性变换得到;为指数的系数,取正数。

### 2.2 改进的指数变换法

根据新思想提出的新的适应度函数如下:

$$\begin{cases} F^*(X)=e^{-aF(X)} \\ a=\sqrt[m]{t}\,/(F_{avg}+\varepsilon),\ m=1+\lg(T) \end{cases} \quad (2)$$

式中:$F^*(X)$—经过指数变换后得到的新适应度;$F(X)$—原适应度,若问题本身是求最小化,则直接取目标函数,若是最大化问题,则要先将问题作变换,转化为求最小值问题;系数 $a$ 不再是一个常数,而是一个随进化代数增加而逐渐增加的动态变化的正数;$t$—当前的进化代数,$T$—最大遗传代数;平均适应度 $F_{avg}$—当前种群的平均值,由于是正数,必须为非负数,因此当目标函数值可能为负数时,$F(X)$可取目标函数加上一个足够大的正数;$\varepsilon$——足够小的正数,防止 $F_{avg}$ 为 0 时,算式的分母为零。

## 3 算法测试与分析

### 3.1 测试函数

函数 1:$f_1(x,y)=-(x^2+2y^2-0.4\cos(3\pi x)-0.6\cos(4\pi y))$,

$-10\le x,y\le 10$ (3)

该函数是一个非线性、对称、不可分离的多峰函数,但只有 1 个全局最大点,在点(0,0)处,最大值为 1。其三维图形,如图 1 所示。

*来稿日期:2009-05-11 *基金项目:苏州市职业大学科研资助项目(SZD09L14)

函数 2:

$$f_2(x,y)=100(y-x^2)^2+(1-x)^2, -2.048\le x,y\le 2.048 \quad (4)$$

该函数是一个非线性、不对称、不可分离的病态单峰函数,很难找到最优解,有一个全局最小点(1,1),最小值为 0。

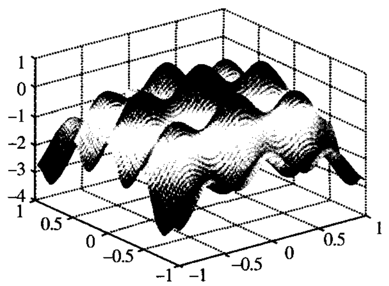


图 1 函数 $f_1(x,y)$ 的三维图形

## 3.2 测试用的三种适应度函数

本文采用以下三种适应度对上述 2 个函数进行求解计算,三种适应度函数分别为:

(1)算法 1(FGA):采用普通的指数变换法,指数系数 $a$ 为常数;

函数 1 为求最大值,则原适应度函数需要作变换:

$$F_1(x,y)=-f_1(x,y) \quad (5)$$

函数 2 为求最小值,则原适应度函数可直接取目标函数:

$$F_2(x,y)=f_2(x,y) \quad (6)$$

(2)算法 2(LGA):采用 Goldberg 的线性拉伸法,适应度函数为:

$$F'(X)=\frac{(c-1)F_{avg}}{F_{max}-F_{avg}}F(X)+\frac{F_{max}-cF_{avg}}{F_{max}-F_{avg}}F_{avg} \quad (7)$$

式中:$F'(X)$—经过线性拉伸得到的新适应度函数;$F(X)$—原适应度函数,可根据问题本身的要求及其目标函数值域的变化范围,直接取目标函数或将目标函数线性变换得到;$F(X)$必须为非负数;$F_{avg}$—当前群体原适应度的平均值;$F_{max}$—当前群体中原适应度最大值;$c$—最佳个体期望复制数量,取 1~2,当群体规模大小为(50~100)时,可取 1.2~2。显然,适应度函数的好坏与常数 $c$ 的取值相关[5],本实验取 $c$=2。

函数 1 为求最大值,但其目标函数值可能为负数,因此原适应度函数需要作变换:

$$F_1(x,y)=K+f_1(x,y) \quad (8)$$

式中:$K$—常数;保证 $F_1(x,y)$—非负数。

函数 2 为求最小值,而适应度线性拉伸适合求解问题的最大值,因此要作变换,原适应度函数:

$$F_2(x,y)=-f_2(x,y)+K \quad (9)$$

式中:$K$—常数;保证 $F_1(x,y)$—非负数。

(3)算法 3(MGA):采用本文提出的动态非线性指数变换法;

函数 1 为求最大值,而且其目标函数值可能为负数,为保证 $F_{avg}(x,y)$是非负数,原适应度函数要作变换:

$$F_1(x,y)=K-f_1(x,y) \quad (10)$$

式中:$K$—常数;保证 $F_1(x,y)$为非负数;

函数 2 求最小值,且其目标函数值为非负数,故原适应度函数可直接取目标函数:

$$F_2(x,y)=f_2(x,y) \quad (11)$$

## 3.3 实验结果与分析

为了便于算法搜索性能的比较,对于同一个函数,三种算法均采用文献[9]提出的改进实数遗传算法,并选用同样的测试参数。选取的种群规模均为 100;总进化代数为 100;允许最优性能无改善的代数为 50;交叉概率为 0.9;变异概率为 0.01;算法终止条件为达到总进化代数算法终止或连续运行 50 代最优解没有改进则算法终止或达到规定的寻优精度算法终止。

在 Genuine Intel(R)CPU T2300(内存 512M)微机上分别用三种算法实验 100 次,函数最优解保留小数后六位,2 个测试函数在独立运行 100 次中获得的最优值、最优值对应的变量值、首次搜索到最优解时的运行代数和运行时间,如表 1 所示。运行时间代表该方法的收敛速度。如表 2 所示,给出了 2 个测试函数的最优值分别收敛于不同精度时三种算法的收敛状况。表 2 中的结果为每个函数收敛于每一个精度,分别用三种算法独立运行 100 次得到。表 2 中的平均收敛代数用 $g$ 表示、收敛概率用 $p$ 表示、误差用 $\varepsilon$ 表示。$\varepsilon$ 为函数优化最优计算值与理论最优值之差。

**表 1 测试函数分别采用三种适应度函数的运行结果**

| 函数 | 方法 | 变量 x | 变量 y | 函数最优解 | 首次搜索到最优解的运行代数 | 运行时间(s) |
|---|---|---|---|---|---|---|
| $f_1$ | FGA | −5.50967e−005 | −0.000322168 | 0.999995 | 36 | 0.21875 |
| | LGA | 5.14891e−005 | −2.14548e−005 | 1.000000 | 38 | 0.265625 |
| | MGA | 2.82077e−006 | 9.7901e−006 | 1.000000 | 15 | 0.125 |
| $f_2$ | FGA | 1.00077 | 1.00289 | 1.82812e−004 | 61 | 0.421875 |
| | LGA | 1.00066 | 1.00135 | 5.59045e−007 | 67 | 0.46875. |
| | MGA | 0.999931 | 0.99986 | 5.37358e−009 | 20 | 0.15625 |

从表 1、2 的运行结果可以看出 MGA 方法在收敛精度、收敛速度和收敛的概率方面明显优于算法 FGA 和 LGA,LGA 算法次之,FGA 算法最差;平均运行代数 FGA 有时小于 LGA,但其值也与收敛概率有关。

**表 2 三种适应度函数在测试函数收敛于不同精度时的收敛情况**

| 收敛精度与收敛情况 / 测试函数与方法 | | $\varepsilon\le10^{-2}$ | | $\varepsilon\le10^{-3}$ | | $\varepsilon\le10^{-4}$ | | $\varepsilon\le10^{-5}$ | | $\varepsilon\le10^{-6}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | g | p(%) | g | p(%) | g | p(%) | g | p(%) | g | p(%) |
| f1 | FGA | 7 | 70 | 9 | 15 | 10 | 5 | 36 | 1 | / | 0 |
| | LGA | 18.79 | 75 | 27.76 | 54 | 33.65 | 26 | 38.81 | 16 | 45.55 | 11 |
| | MGA | 8.85 | 100 | 11.27 | 100 | 13.80 | 100 | 16.12 | 100 | 18.12 | 100 |
| f2 | FGA | 12.84 | 32 | 12.25 | 4 | / | 0 | / | 0 | / | 0 |
| | LGA | 24.53 | 41 | 35.56 | 16 | 53.8 | 5 | 52.67 | 4 | 66.75 | 3 |
| | MGA | 10.08 | 70 | 14.55 | 60 | 20.02 | 51 | 24.4 | 50 | 25.45 | 48 |

# 4 结束语

适应度函数表明个体或解的优劣性。一般而言,适应度函数可通过对目标函数的值域进行某种映射变换得到,适应度函数的设计要综合考虑问题本身的要求及其值域的变化范围等。本文提出的基于指数变换、指数系数随进化代数增加而逐渐增加的动态变化的适应度函数,既保证了算法初期种群的多样性,又能在算法后期提高较优个体的适应度,有助于算法的收敛。通过两个典型函数的优化计算,验证了该改进算法的有效性。

## 参考文献

1 陈国良,王煦法等. 遗传算法及其应用[M]. 北京:人民邮电出版社,1996(6):10~14

2 张思才,张方晓. 一种遗传算法适应度函数的改进方法[J]. 计算机应用与软件,2006,23(2):108~110

3 陈小平,于盛林. 实数遗传算法交叉策略的改进 [J]. 电子学报,2003,1(31):71~74