

# 一种基于混合概率选择算子的改进遗传算法

赵鑫宁, 喻 歆, 吴 锡

(成都信息工程大学计算机学院, 四川 成都 610225)

**摘要:**遗传算法(genetic algorithm, GA)已被成功应用于求解实值优化问题, 但在求解多峰实值优化问题时面临收敛较慢和早熟收敛的问题。为解决该问题, 提出一种基于高斯分布和柯西分布的概率选择算子。算子在执行选择操作时, 分别根据当前种群生成高斯和柯西分布, 通过对分布采样获得参加遗传操作的个体, 从而在保证选择压力的同时保持一定程度的种群多样性, 避免早熟收敛。此外, 基于种群算法投资组合(population-based algorithm portfolios, PAP)策略, 同时利用2种概率选择算子的优势, 采用2个子种群并行搜索, 提高算法在多种不同问题上的求解性能。在一系列多峰实值优化问题上的测试结果表明, 文中提出的改进方法能够显著提高遗传算法在收敛速度和求解精度两方面的性能。

**关键词:**计算机应用; 人工智能; 遗传算法; 多峰函数; 高斯分布; 柯西分布; 种群算法投资组合

**中图分类号:** TP301.6

**文献标志码:** A

## 0 引言

遗传算法<sup>[1-3]</sup>起源于20世纪60年代, 作为一种模拟自然界生物“优胜劣汰”的进化过程的自适应全局优化概率搜索算法, 至今仍被广泛关注并已成功应用在路径规划、结构设计和工业制造等多个领域。其基本原理<sup>[1]</sup>是模拟生物进化过程, 将实际问题的候选解种群编码并初始化以后, 对该种群反复执行选择、交叉和变异操作, 直到达到最大迭代次数或者找到满足要求的解。标准遗传算法(standard genetic algorithm, SGA)的算法流程图如图1所示。其中, 对选择、交叉、和变异3个算子进行研究和改进一直是大多数研究工作的重点<sup>[4]</sup>。

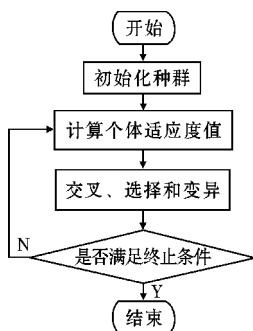


图1 标准遗传算法的流程图

虽然遗传算法在解决大多数优化问题时具备其他算法所没有的优势, 但在求解多峰实值优化问题时, 面临收敛速度较慢甚至早熟收敛的问题<sup>[5-6]</sup>。早在20世纪末, Xin Yao等<sup>[7]</sup>在采用演化规划算法求解实值

优化问题时提出基于柯西分布的变异算子, 在求解多数多峰函数时有显著效果, 但是柯西分布变异范围较大, 算法后期收敛较慢。陈有青等<sup>[8]</sup>提出一种对种群分组进行轮盘赌选择时采用高斯分布生成待遗传操作的种群的方法, 加快种群的收敛速度, 一定程度提高找到最优解的成功率, 但遇到较为复杂的多峰问题时仍然很难避免早熟收敛。谢柏桥等<sup>[9]</sup>提出算法前期使用一种变异算子, 后期使用另一种变异算子的方法, 只对简单函数较为有效, 由于无法精确确定何种问题上、处于何种阶段应该使用哪一种变异算子, 所以该方法在复杂函数上性能不理想。针对不同类型的优化问题所对应的最佳求解算法也不同的现象, Fei Peng等<sup>[10]</sup>提出一种基于种群算法投资组合的策略, 有效提高算法在不同类型优化问题上的求解性能。Jarno Martikainen等<sup>[11-14]</sup>也提出通过多种群混合算法来解决单种群不能有效求出解的问题。

遗传算法在求解多峰实值优化问题的过程中, 一方面需要维持一定的种群多样性以避免早熟收敛, 另一方面, 在搜索后期, 较大的种群多样性又会减缓收敛速度, 因此单纯依靠变异操作无法满足对种群多样性的适应性控制要求。此时如果在执行选择操作时, 产生符合高适应度个体特征的新个体参加后续遗传操作, 则能够在搜索前期维持一定的种群多样性, 而在搜索后期, 由于种群每个个体相似度较大, 因而该操作生成的新个体也比较接近, 从而有利于提高整个种群的收敛速度。基于该思想, 提出一种基于高斯分布和柯西分布的选择算子, 并根据PAP策略融合两种算子的优势, 以期改进遗传算法在多个不同实值优化问题上的总体性能。

## 1 改进的遗传算法及原理

### 1.1 分布函数

高斯分布是一种在理论研究和工程应用都非常重要的概率分布,能根据已经确定的均值和方差,在一定幅度范围内产生符合条件的随机变量。高斯密度函数为

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

其中,  $-\infty < x < \infty$ ,  $\mu$  为均值,  $\sigma$  为标准差,当  $\mu=0, \sigma=1$  时为标准正态分布时对应的密度函数。

柯西分布是一种类似于高斯分布的连续性分布函数,和高斯分布不同的是其期望和方差均不存在。

柯西密度函数为

$$f(x; x_0, \gamma) = \frac{1}{\pi} \frac{\gamma}{\gamma^2 + (x - x_0)^2} \quad (2)$$

其中,  $-\infty < x < \infty$ ,  $x_0$  是波峰所对应的  $x$  值,  $\gamma > 0$  为波形宽度比例参数。与密度函数对应的柯西分布的分布函数为

$$F(x; x_0, \gamma) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x - x_0}{\gamma}\right) \quad (3)$$

其中,  $-\infty < x < \infty$ 。

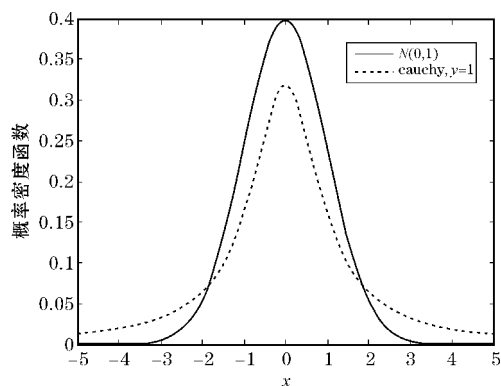


图2 高斯和柯西概率密度曲线

图2是高斯在  $\mu=0, \sigma=1$  和柯西在  $\gamma=1$  时的概率密度曲线对比图。从图中可以看出高斯分布和柯西分布的概率密度函数十分相似,但是高斯分布随机变量集中在均值附近,而柯西分布概率密度函数曲线虽然也呈钟型,但随机变量分布较为分散。因此,在产生随机变量时,和高斯分布相比,柯西分布生成的随机变量有较大机会分布在两侧距离中心较远的区域。

### 1.2 改进的遗传算法

#### 1.2.1 概述

针对SGA求解多峰实值优化问题时面临的收敛速度较慢和早熟收敛的问题,重点对选择算子进行改进。文献[8]提出的基于高斯分布选择的遗传算法(gaussian selection-based genetic algorithm, GSGA)一定程度上提高其在多峰实值优化问题上的性能,但在某些问题上收敛速度较慢,并且求解精度也不理想。文献[9]先使用基于柯西分布的变异算子,然后再使用基于高斯分布的变异算子,仅在部分多峰函数上取得较好的效果,并且在实际处理中也很难确定两种变异算子的切换时机。根据前文分析,在不同的搜索阶段,针对不同的优化问题,最佳的种群多样性控制机制也不相同。考虑到柯西分布和高斯分布的相似性以及控制在种群多样性方面的差异性,提出一种基于柯西分布选择的遗传算法(cauchy selection-based genetic algorithm, CSGA),并基于PAP策略,融合GSGA和CSGA的优势,以期提高算法在多种优化问题上的总体性能。具体来说,建立两个并行执行的子种群,每个子种群采用相同的设置参数,独立地进行初始化、选择、交叉和变异等操作。不同的是其中一个子种群采用基于高斯分布的选择算子,另一个子种群采用基于柯西分布的选择算子。每隔一定代数,两个子种群间进行优秀基因交流。容易发现,文中提出的方法可以同时利用两种选择算子的优点,而且避免了如何确定优先使用哪种选择算子和如何确定两种算子的切换时机等麻烦问题。算法的伪代码如下。

begin

- (1) 初始化种群大小均为  $N$  的子种群1和子种群2。
- (2) while 没有满足终止条件 do
  - (a) 对子种群1和子种群2所有个体进行适应度值评估。
  - (b) 子种群1采用基于高斯分布选择的改进遗传算法;子种群2采用基于柯西分布选择的改进遗传算法。
  - (c) if 遗传代数达到设定的交换代数
    - (i) 把子种群1的  $N$  个个体和子种群2的前  $M$  个优秀个体合并成一个新的种群。
    - (ii) 对新的种群按适应度由高到低排序。
    - (iii) 取新种群前  $N$  个个体作为子种群1的下一代。
    - (iv) 把子种群2的  $N$  个个体和子种群1的前  $M$  个优秀个体合并成一个新的种群。
    - (v) 对新的种群按适应度由高到低排序。
    - (vi) 取新种群前  $N$  个个体作为子种群2的下一代。

end if

end while

(3) 子种群1和子种群2中的适应度较高的最优解作为整个算法的最优解。

end

首先根据实际问题初始化两个大小均为  $N$  的子种群,其中一个子种群在选择阶段采用基于高斯分布的选择算子,另一个子种群在选择阶段采用基于柯西分布的选择算子,两个子种群独立进行评估。达到交换最优解的代数时,将其中一个子种群所有  $N$  个个体和另一个子种群按适应度值从大到小排序的前  $M$  个个体合并成一个大小为  $(N+M)$  的新种群,对新种群进行按适应度值从大到小进行排序,取前  $N$  个个体作为这个子种群的下一代,然后对另一个子种群采取同样操作。最终满足遗传操作终止条件后,两个子种群中适应度较高的最优解作为整个算法的最优解。

### 1.2.2 编码

遗传算法的编码方式有很多,较为常用的包括二进制编码和实值编码<sup>[15-16]</sup>。由于实值编码直接表示基因值,无需解码编码操作,因此文中采用实值编码方法。

### 1.2.3 适应度评估

种群个体的适应度评价是判断个体优劣的关键。适应度大较为优秀的个体被遗传到下一代的概率较大;反之,适应度较小的个体被遗传到下一代的概率较小。文中所用的测试函数为求最小值,所以对原有测试函数取负后求最大值,在得到结果后再取负。由于算法中采用轮盘赌选择算子的思想,每个个体的适应度值需为非负数,并且如果单纯对适应度值取绝对值,当适应度值同时存在正值和负值,会出现收敛异常的现象。因此,在进行适应度评估以后执行选择操作之前,对种群中所有个体按照适应度从大到小排序,然后每个个体的适应度值减去最小个体的适应度值,得到的新适应度值作为选择操作的依据。这样既能保证所有个体均大于等于零,又能在种群个体间适应度相差较小时,提高种群中较大个体在适应度和中所占的百分比。

### 1.2.4 选择

选择操作的改进是工作的重点。在遗传算法中,选择操作选出当前种群中适应度高的个体;交叉操作对选出的个体进行基因重组,发现潜在的较优解;变异通过突变产生新个体,避免搜索陷入局部最优。选择和交叉操作使种群能够收敛,但同时也会导致早熟,传统的遗传算法,主要靠变异操作避免早熟收敛的发生,

所以如果在选择阶段产生符合种群特征的更加优秀的基因型的同时引入随机性,将能在加快收敛速度的同时维持一定程度的种群多样性。因此,采用有助于局部搜索的高斯分布和有助于全局搜索的柯西分布独立地在选择阶段生成参与遗传操作的新个体,期望既能加快收敛又能避免早熟。

此外,当种群个体差异较大,适应度较大的个体被选中的概率大,也就意味着适应度较大的个体有更多的机会在选择阶段发生随机扰动,可加快种群的收敛速度;而当随着代数的增加,种群个体均逐渐趋于接近最优值时,每个个体在种群中占的比例接近,此时适应度较大的个体被轮盘赌选到的概率下降,也就意味着选择阶段发生的随机扰动几率逐渐减少,有利于种群的稳定性。为保证收敛,文中采用最优保存策略<sup>[3]</sup>,即保存历史最优个体,在每代结束后,用最优个体替换掉经过交叉、变异等遗传操作后所产生的适应度值最小的个体。

选择算子的具体操作方法为:

(1) 每个子种群在完成所有个体的适应度值计算后,对个体适应度值按从大到小排序,然后分组,根据分组个数  $N$  产生一个大小为  $N$  的候选池。

(2) 对适应度值较大的前  $M$  个分组,不同的子种群分别产生符合高斯分布和柯西分布的新个体。产生符合高斯分布新个体方法为:首先计算出每个分组每个基因位的均值和标准差,其中均值和标准差分别对应公式(1)中的  $\mu$  和  $\sigma$ 。然后使用 Box-Muller 法<sup>[17]</sup>产生符合该组个体特征的新个体。产生符合柯西分布新个体方法为:由于公式(3)分布函数是公式(2)概率密度函数在负无穷到正无穷的积分,所以利用公式(3)的反函数公式(4)即可得到相应所有的柯西随机变量,其中  $F(x; x_0, \gamma)$  取  $(0, 1)$ ,  $\gamma$  取 1,  $x_0$  对应所在分组每个基因位的均值。最后根据所求的柯西随机变量得到分组的新个体。

$$x = \tan[\pi(F(x; x_0, \gamma) - 0.5)]\gamma + x_0 \quad (4)$$

(3) 在求得所有前  $M$  个分组的新个体后,如果新个体的适应度大于所在分组最优个体适应度,则把分组  $i$  的新个体放入候选池的第  $i$  位;否则,将分组  $i$  的最优个体放入候选池的第  $i$  位。对于适应度值较小的后  $(N-M)$  个分组,取分组  $i$  中随机一个个体放入候选池的第  $i$  位。最后,候选池的  $N$  个个体分别对应  $N$  个分组。

(4) 对种群的  $N$  个分组进行轮盘赌选择,如果选中第  $i$  个分组,则选取候选池中第  $i$  个个体进行后面的遗传操作。



1.2.5 交叉和变异

交叉操作不仅是种群中优秀个体基因型的表达,还是遗传算法产生新个体的主要方法之一。适合浮点数的交叉方式有:单点交叉、双点交叉、多点交叉、离散交叉和算术交叉等<sup>[3]</sup>。文中采用单点交叉来获得新的基因型。

变异操作采取改变种群中一部分个体的基因值来扩大遗传算法的搜索范围,也是跳出当前局部最优的有效方法之一。文中只对按概率选中个体的某一位基因值进行变异,即将这位基因值替换成可行解空间中的一个随机值。

2 测试实例及实验结果分析

2.1 参数设置

采用 12 个实值优化函数<sup>[7]</sup>来测试改进算法的性能,函数如公式(5)~(16)所示,函数类型、定义域、维度和理论极小值如表 1 所示。其中式(5)~(10)为单峰函数,式(11)~(16)为多峰函数。测试中,分别对 SGA、GSGA、CSGA 和基于高斯和柯西混合机制的遗传算法(gaussian and cauchy selection-based genetic algorithm,GCSGA)进行对比,所设置遗传参数和函数评估次数均相同。具体参数为:SGA、GSGA 和 CSGA 对应种群个体数量为 200 个,交叉概率为 0.7,变异概率为 0.07,评估次数为 2000000 次,GCSGA 对应两个子种群个体数量均为 100 个,在选择时,均对种群前 16 % 的个体<sup>[8]</sup>进行高斯或者柯西扰动,每隔 500 代<sup>[10]</sup>,2 个种群互相交流最大值。仿真测试时,在每个测试函数上,每一个算法都采用相同的随机种子集合独立运行 50 次。

$$f_1(x) = \sum_{i=1}^n x_i^2 \tag{5}$$

$$f_2(x) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i| \tag{6}$$

$$f_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2 \tag{7}$$

$$f_4(x) = \sum_{i=1}^{n-1} [100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \tag{8}$$

$$f_5(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2 \tag{9}$$

$$f_6(x) = \sum_{i=1}^n ix_i^4 + random [0,1) \tag{10}$$

$$f_7(x) = \sum_{i=1}^n -x_i sin(\sqrt{|x_i|}) \tag{11}$$

$$f_8(x) = \sum_{i=1}^n [x_i^2 - 10cos(2\pi x_i) + 10] \tag{12}$$

$$f_9(x) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^n cos2\pi x_i) + 20 + e \tag{13}$$

$$f_{10}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n cos(\frac{x_i}{\sqrt{i}}) + 1 \tag{14}$$

$$f_{11}(x) = \frac{\pi}{n} \{ 10sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2 \} + \sum_{i=1}^n \mu(x_i, 10, 100, 4) \tag{15}$$

$$y_i = 1 + \frac{1}{4}(x_i + 1)$$
$$\mu(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < -a, \end{cases}$$

$$f_{12}(x) = 0.1 \{ sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + sin^2(2\pi x_n)] \} + \sum_{i=1}^n \mu(x_i, 5, 100, 4) \tag{16}$$

表 1 测试函数类型

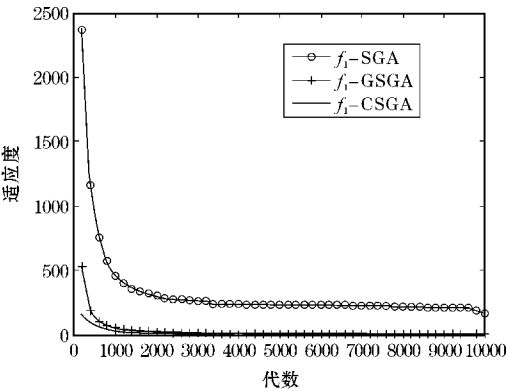
函数	函数类型	定义域 (n=30)	最小值 (f <sub>min</sub> )
f <sub>1</sub>	Sphere Model	[ - 100,100 ] <sup>n</sup>	0
f <sub>2</sub>	Schwefel's Problem 2.22	[ - 10,10 ] <sup>n</sup>	0
f <sub>3</sub>	Schwefel's Problem 1.2	[ - 100,100 ] <sup>n</sup>	0
f <sub>4</sub>	Generalized Rosenbrock's Function	[ - 30,30 ] <sup>n</sup>	0
f <sub>5</sub>	Step Function	[ - 100,100 ] <sup>n</sup>	0
f <sub>6</sub>	Quartic Function i. e. Noise	[ - 1.28,1.28 ] <sup>n</sup>	0
f <sub>7</sub>	Generalized Schwefel's Problem 2.26	[ - 500,500 ] <sup>n</sup>	-12569.5
f <sub>8</sub>	Generalized Rastrigin's Function	[ - 5.12,5.12 ] <sup>n</sup>	0
f <sub>9</sub>	Ackley's Function	[ - 32,32 ] <sup>n</sup>	0
f <sub>10</sub>	Generalized Griewank Function	[ - 600,600 ] <sup>n</sup>	0
f <sub>11</sub>	Generalized Penalized Functions	[ - 50,50 ] <sup>n</sup>	0
f <sub>12</sub>	Generalized Penalized Functions	[ - 50,50 ] <sup>n</sup>	0

2.2 实验结果分析

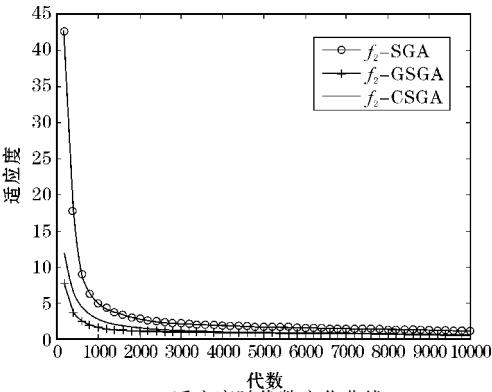
表 2 是对 12 个单峰和多峰 30 维函数分别用 SGA、GSGA、CSGA 和 GCSGA 所求解 50 次的均值和标准差对比结果。采用置信水平为 0.05 的 Wilcoxon 秩和检验确定比较结果的统计显著性。加粗的结果表明新提出的算法 GCSGA 在所有情况下都具有最优的求解精度(平均值最小)和求解稳定性(标准差最小)。

表 2 不同算法测试结果对比

函数	SGA		GSGA		CSGA		GCSGA	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
$f_1$	162.92549	298.84232	0.08815	0.13406	4.07077	4.45308	0.04682	0.04936
$f_2$	1.16002	1.28994	0.64019	0.61396	0.61970	0.44508	0.37131	0.35841
$f_3$	116737.33104	201398.43343	9.03218	19.36642	482.16624	509.51407	7.63085	11.28067
$f_4$	5248.62935	13880.22732	91.84581	275.54557	145.90468	94.64084	38.31525	33.70466
$f_5$	139.42000	199.67935	43.66000	78.42260	3.40000	4.48998	1.26000	2.84119
$f_6$	0.17082	0.34834	0.02791	0.02024	0.10590	0.16090	0.01795	0.01434
$f_7$	-12478.00622	91.65929	-12517.15331	89.89190	-12535.37676	53.06182	-12564.56238	23.19129
$f_8$	3.89380	3.91445	1.10084	1.44481	4.39832	4.13293	0.77023	0.95800
$f_9$	2.89543	1.92542	1.29503	1.08524	1.03552	0.80358	0.70798	0.66228
$f_{10}$	2.20201	1.87872	0.46698	0.63876	0.09975	0.03979	0.03469	0.02012
$f_{11}$	41273.94651	144407.05772	0.01134	0.01461	0.00074	0.00032	0.000008	0.000019
$f_{12}$	20723.81112	45419.07202	2.77755	5.17095	0.01595	0.00952	0.00041	0.00215

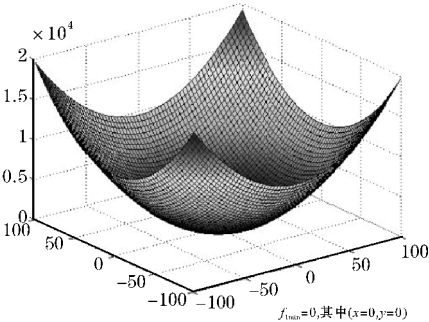


(a)  $f_1(x)$  适应度随代数变化曲线



(b)  $f_2$  适应度随代数变化曲线

图 3 3 种不同方法下  $f_1$  和  $f_2$  求解随代数变化的对比



(a)  $f_1$

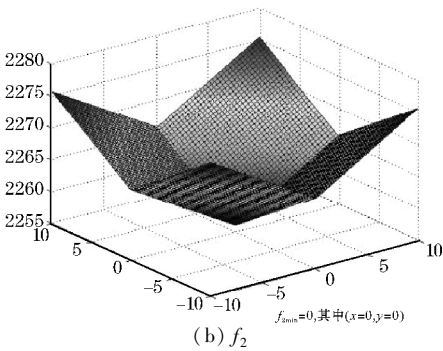


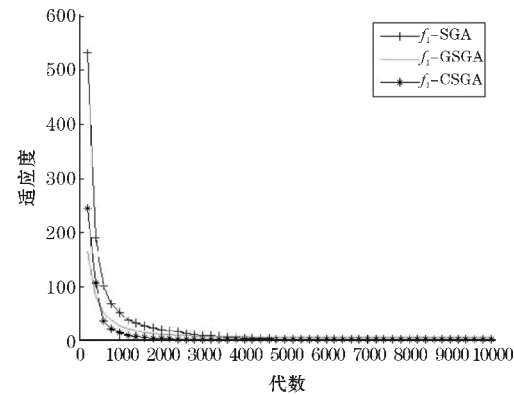
图 4  $f_1$  和  $f_2$  的二维空间结构图

从图 3 可以看出,单峰函数 $f_1(x)$  和 $f_2(x)$  上,GS- GA 和 CSGA 收敛速度均比 SGA 快。函数 $f_1(x)$  ,当  $n =2$  时,其空间结构如图 4(a) 所示,可以看出其只有一个波谷,使用 CSGA 有机会搜索到更大的解空间,因此 CSGA 比 GSGA 收敛较快。函数 $f_2(x)$  二维空间结构如图 4(b) 所示,虽然只有一个波谷,但由于定义域范围较小,生成符合柯西分布的新个体容易超出定义域范围,而最终只取分组的均值,没有生成更加优秀的个体。而在表 2 可以看出前期收敛速度较慢的函数 $f_1(x)$  使用 GSGA 的效果却较好,而前期收敛速度较慢的函数 $f_2(x)$  使用 CSGA 的效果却较好,说明不同的概率分布对函数收敛速度确实有影响,但函数本身的性质是求解快慢的主要因素。GSGA、CSGA 和 GCSGA 算法对函数 $f_1(x)$  和 $f_2(x)$  测试结果对比如图 5(a) 所示。通过对比,文中提出的改进算法采用 2 个分布独立地作用于不同种群,只在特定条件下进行优秀基因交流,的确能加快种群的收敛速度。

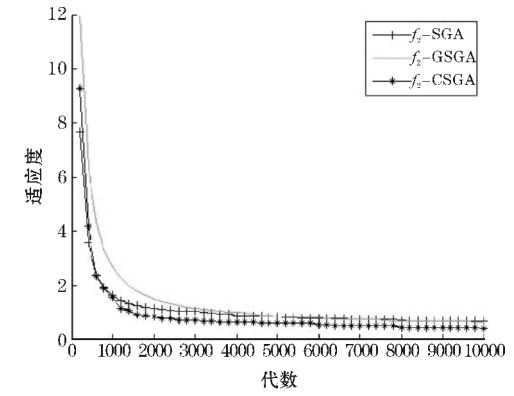
对于函数 $f_3(x)$  ,根据其函数表达式可以判断出是一个阶跃函数,定义域在  $[-100,100]^2$  的二维空间结构如图 6(a) 所示,函数只有一个波谷。其在较小定义域  $[-1,1]^2$  的二维结构如图 6(b) 所示,可以看

出在定义域区间内,不同变量对应着同一个解,此时高斯分布没有柯西分布对解空间搜索范围大,即在一定程度上不利于收敛。在表2中可以看出,CSGA比GS-GA效果好很多,从图7(a)看出用改进的算法在两个种群进行交换优秀基因后,收敛要快于其他方法,并且从表2得出其标准差和其他方法较小。

多峰函数 $f_7(x)$ 、 $f_8(x)$ 和 $f_9(x)$ 的二维结构图如图8所示。这3个函数存在较多的波谷,对照表1和表2发现他们均容易早熟,很难收敛到最小值附近,但是改进的遗传算法最终结果比其他改进方法提高一个数量级。他们的适应度随代数变化的曲线分别如图7(b)、图9(a)和图9(b)所示,可以较明显地看出改进的遗传算法无论收敛速度还是精确度均比其他方法较好。

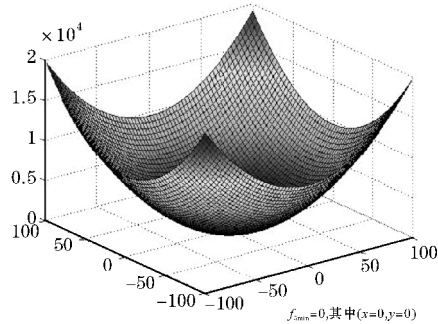


(a)  $f_1(x)$  适应度随代数变化曲线

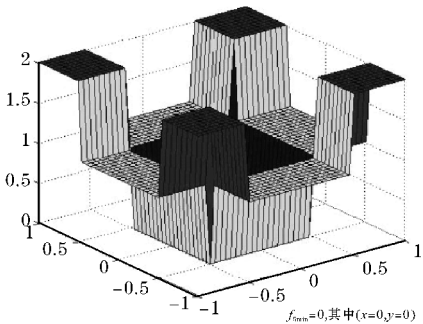


(b)  $f_2(x)$  适应度随代数变化曲线

图5 3种不同方法下 $f_1$ 和 $f_2$ 求解随代数变化的对比

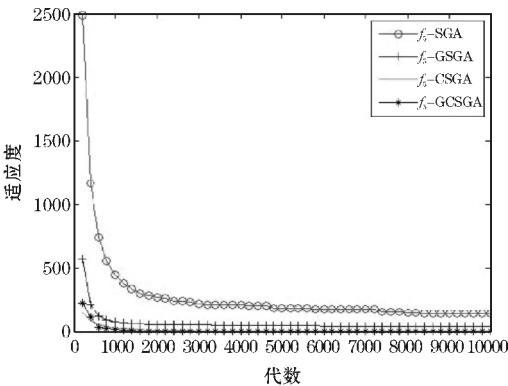


(a) 较大范围定义域

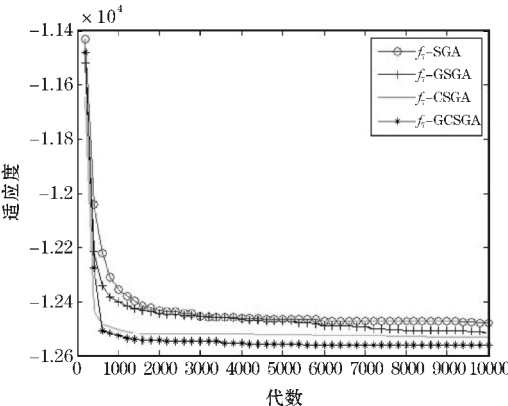


(b) 较小范围定义域

图6  $f_5$  的二维空间结构图

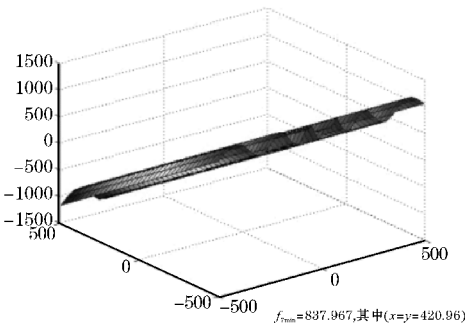


(a)  $f_5(x)$  适应度随代数变化曲线



(b)  $f_7(x)$  适应度随代数变化曲线

图7 4种不同方法下 $f_5$ 和 $f_7$ 求解随代数变化



(a)  $f_7$

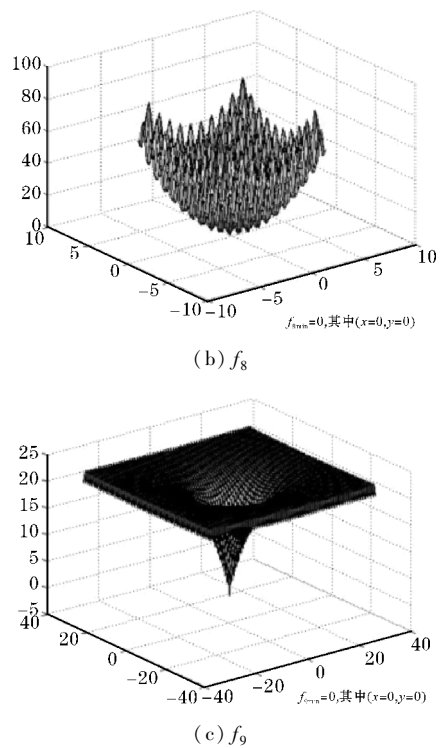
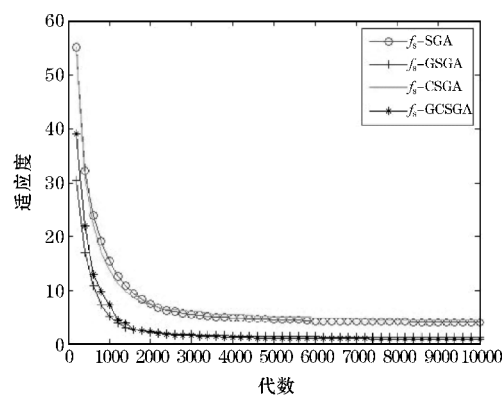
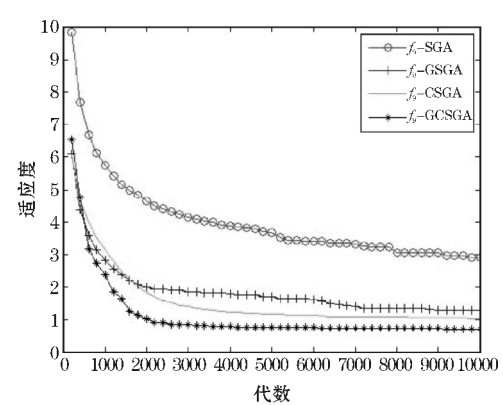


图8 函数 $f_7$ 、 $f_8$ 和 $f_9$ 的二维空间结构图



(a)  $f_8(x)$  适应度随代数变化曲线



(b)  $f_9(x)$  适应度随代数变化曲线

图9 4种不同方法下 $f_8$ 和 $f_9$ 求解随代数变化的对比

### 3 结束语

遗传算法在求解多峰实值优化问题时面临收敛较

慢和早熟收敛的难题。为解决这个问题,提出一种基于高斯和柯西分布的混合概率选择算子,并基于 PAP 策略同时利用两种选择算子的优点,同时避免了如何确定优先使用哪种算子和如何确定两种算子切换时机的难题。在多个测试问题上的仿真结果表明,新提出的方法显著提高了遗传算法在收敛速度和求解精度两方面的性能。未来的工作主要包括研究如何提高现有方法在高维问题上的求解性能,在并行计算机上优化现有算法性能,以及自适应地调整现有方法的参数如子种群交互周期等。

### 参考文献:

[1] Holland J H. Adaptation in Natural and Artificial Systems [M]. Ann Arbor: The University of Michigan Press, 1975.

[2] Goldberg D E. Genetic algorithm in search, optimization and machine learning [M]. Massachusetts: Addison-Wesley, 1989.

[3] 周明, 孙树栋. 遗传算法原理及应用 [M]. 北京: 国防工业出版社, 1999.

[4] Vasconcelos J A, Ramirez J A, Takahashi R H C. Improvements in Genetic Algorithms [J]. IEEE Transactions on Magnetics, 2002: 3414-3417.

[5] Janikow C Z, Michalewicz Z A specialized genetic algorithm for numerical optimization problems [C]. Proc of Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence, Herndon, VA: IEEE, 1990: 798-804.

[6] Jianwu Li, Yao Lu. An Efficient Real-Coded Genetic Algorithm for Numerical Optimization Problems [C]. Proc of third International Conference on Natural Computation, Haikou: IEEE, 2007: 760-764.

[7] Xin Yao, Yong Liu, Guangming Lin. Evolutionary Programming Made Faster [J]. IEEE Transactions on Evolutionary Computation, 1999, 3 (2): 82-102.

[8] 陈有青, 徐蔡星, 钟文亮. 一种改进选择算子的遗传算法 [J]. 计算机工程与应用, 2008, 44 (2): 44-49.

[9] 谢柏桥, 戴光明, 石红玉. 一种改进的求解约束函数优化问题的演化算法 [J]. 计算机应用与软件, 2008, 25 (7): 48-50.

[10] Fei Peng, Ke Tang, Guoliang Chen. Population-

- Based Algorithm Portfolios for Numerical Optimization [J]. IEEE Transactions on Evolutionary Computation, 2010, 14(5): 782–800.
- [11] Jarno Martikainen, Seppo J. Ovaska. Hierarchical two-population genetic algorithm [C]. Proc of the 2005 IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications, Espoo: IEEE, 2005: 91–98.
- [12] Taejin Park, Kwang Ryel Ryu. A Dual-Population Genetic Algorithm for Adaptive Diversity Control [J]. Evolutionary Computation, IEEE, 2010, 14(6): 865–884.
- [13] Doraghinejad M, Nezamabadi-pour H, Hashempour Sadeghian A. A hybrid algorithm based on gravitational search algorithm for unimodal optimization [C]. Proc of 2012 2nd International eConference on Computer and Knowledge Engineering. Mashhad: IEEE, 2012: 129–132.
- [14] 于蕾蕾. 双种群遗传算法的改进及其应用研究 [D]. 合肥: 合肥工业大学, 2009.
- [15] 张群超, 郑建国, 钱洁. 遗传算法编码方案比较 [J]. 计算机应用研究, 2011, 28(3): 819–822.
- [16] Herrera F, Lozano M, Verdegay J. L. Tackling real-coded genetic algorithms: operators and tools for behavioural analysis [J]. Kluwer Academic Publishers, 1998, 12(4): 265–319.
- [17] 任浩, 毛玉泉, 李思佳. Gauss 随机变量产生方法的比较研究 [J]. 船舶电子工程, 2010, 30(12): 87–90.

## Improving Genetic Algorithm based on A Hybrid Probabilistic Selector

ZHAO Xin-ning, YU Xin, WU Xi

(College of Computer Sciences, Chengdu University of Information Technology, Chengdu 610225, China)

**Abstract:** Genetic Algorithms (GA) have been applied to numerical optimization problems successfully. However, they are confronted with slow convergence and prematurity. Therefore, this paper proposes a hybrid probabilistic selector for a real-encoded GA based on Gaussian and Cauchy distributions. During the selection operation, a Gaussian and a Cauchy distribution are generated based on the selected groups of current population respectively. Then candidate individuals for genetic operations are generated by sampling the two distributions respectively. With the proposed selector, the selection pressure is guaranteed while a certain diversity is maintained which is beneficial to avoidance of prematurity. Furthermore, in order to improve the overall performance of GAs on a wide range of problems, two sub-populations, equipped with Gaussian and Cauchy based selectors respectively, are run parallel by adopting the strategy of Population-Based Algorithm Portfolios (PAP). Experimental results on a set of numerical optimization problems show that the proposed approach can significantly increase the performance of GAs with respect to the convergence rate and the solution quality.

**Key words:** computer application; artificial intelligence; genetic Algorithm; multimodal function; gaussian distribution; cauchy distribution; population-Based Algorithm Portfolios