

# 云遗传算法及其应用

戴朝华<sup>1</sup>, 朱云芳<sup>2</sup>, 陈维荣<sup>1</sup>, 林建辉<sup>3</sup>

(1. 西南交通大学电气工程学院, 四川成都 610031; 2. 西南交通大学峨眉校区计算机系, 四川峨眉 614202;  
3. 西南交通大学牵引动力国家重点实验室, 四川成都 610031)

**摘要:** 传统遗传算法搜索速度慢, 容易陷入局部最优解. 借鉴遗传算法(GA)的思想, 利用正态云模型云滴的随机性和稳定倾向性特点, 提出一种新的遗传算法—云遗传算法(CGA). CGA由正态云模型的Y条件云发生器实现交叉操作, 基本云发生器实现变异操作. 最后, 进行了函数优化实验和IIR数字滤波器优化设计, 并与标准GA、NQGA、CAGA和LARES等算法进行比较, 证明了该算法的有效性, 具有一定的参考和应用价值.

**关键词:** 遗传算法; 云模型; 云遗传算法; 函数优化; IIR滤波器设计

**中图分类号:** TP18      **文献标识码:** A      **文章编号:** 0372-2112(2007)07-1419-06

## Cloud Model Based Genetic Algorithm and Its Applications

DAI Chao hua<sup>1</sup>, ZHU Yun fang<sup>2</sup>, CHEN Wei rong<sup>1</sup>, LIN Jian hui<sup>3</sup>

(1. School of Electrical Engineering, Southwest Jiaotong University, Chengdu, Sichuan 610031, China;  
2. Department of Computer and Communication Engineering, E' mei Campus, Southwest Jiaotong University, E' mei, Sichuan 614202, China;  
3. State Key Laboratory of Traction Power, Southwest Jiaotong University, Chengdu, Sichuan 614202, China)

**Abstract:** Traditional genetic algorithm (GA) easily gets stuck at a local optimum, and often has slow convergent speed. As a novel genetic algorithm, cloud model based genetic algorithm (CGA) was originally proposed. CGA is based on both the idea of GA and the properties of randomness and stable tendency of a normal cloud model. In this algorithm, a Y conditional normal cloud generator is used as the cross operator of GA, and a basic normal cloud generator is used as the mutation operator. Finally, the experiments of function optimization and IIR digital filter design were conducted to compare CGA with standard GA, NQGA, CAGA and LARES. From the simulation results, it is believed that CGA is effective and will become a promising candidate of evolutionary algorithms.

**Key words:** genetic algorithm; cloud model; cloud genetic algorithm; function optimization; IIR digital filter design

## 1 引言

遗传算法(GA)是一类借鉴生物进化思想的随机优化算法. 最早由Holland教授在研究自然和人工自适应系统时提出<sup>[1]</sup>. 20世纪70年代, De Jong把基于GA的思想在计算机上进行了大量函数优化实验<sup>[2]</sup>. 20世纪80年代, Goldberg进行归纳总结, 形成了遗传算法的基本框架<sup>[3]</sup>. 由于遗传算法不限制模型结构形式、最优化准则形式、被估参数数目、有无约束, 具有自组织、自适应、自学习性和本质并行性等特点, 广泛应用于参数估计、模式识别、机器学习、神经网络、工业控制等众多领域. 但遗传算法在优化问题中存在的弱点限制了它更广泛的应用, 主要表现在搜索到最优解或满意解的速度太慢, 同时经常伴有早熟收敛的现象<sup>[4]</sup>.

遗传算法的操作规则是概率性而非确定性的. 传统GA采用二进制编码方式, 当交叉变异操作以一定概率执行后, 交叉点和变异位是随机产生的. 所以生成的下一代个体也是随机的, 即其进化的方向是随机的、不可控的, 也就是说GA是无记忆的, 以前的知识随着种群的变化而被破坏. 实数编码虽然很大程度上解决了GA算法精度和存储量的影响, 但其作为主要交叉算子的算术交叉都为凸运算<sup>[5]</sup>, 使整个群体寻找最优受到了限制, 并有未成熟收敛于局部最优的可能<sup>[6]</sup>, 而且同样没有解决进化方向的无记忆和随机性问题.

我国李德毅教授提出了“隶属云与语言原子模型”思想, 并将它逐步完善形成了云理论<sup>[7]</sup>. 该理论是对模糊理论隶属函数概念的创新与发展, 已成功应用于智能控制<sup>[8]</sup>、数据挖掘<sup>[9, 10]</sup>、入侵检测<sup>[11]</sup>、大系统评估<sup>[12]</sup>等

领域. 近年来, 进化算法领域也开始关注云模型. 其中文献[13]提出云自适应遗传算法(CAGA). CAGA采用传统的交叉、变异算子, 但交叉概率、变异概率由云模型产生, 取得了比传统自适应遗传算法更优的结果. 与文献[13]不同, 本文根据正态云模型云滴的随机性和稳定倾向性特点<sup>[7, 14]</sup>, 借鉴遗传算法思想, 利用云发生器代替传统的交叉、变异算子, 提出全新的云遗传算法(CGA).

## 2 云理论

云模型是用自然语言值表示的定性概念与其定量数据表示之间的不确定性转换模型, 主要反映客观世界中事物或人类知识中概念的模糊性和随机性, 并把二者完全集成在一起, 为定性定量相结合的信息处理提供了有力手段<sup>[7]</sup>.

### 2.1 基本概念

定义<sup>[7, 10]</sup> 设  $T$  为论域  $u$  上的语言值, 映射  $C_T(x): u \rightarrow [0, 1], \forall x \in u, x \rightarrow C_T(x)$ , 则  $C_T(x)$  在  $u$  上的分布称为  $T$  的隶属云, 简称云. 当  $C_T(x)$  服从正态分布时, 称为正态云模型.

正态云模型是一个遵循正态分布规律、具有稳定倾向的随机数集<sup>[14]</sup>, 用期望值  $Ex$ 、熵  $En$ 、超熵  $He$  三个数值来表征(如图1所示)<sup>[7]</sup>. 期望值  $Ex$ : 在数域空间最能代表这个定性概念

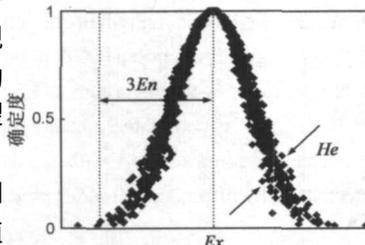


图1 正态云模型的三个数字特征示意图

的点, 反映了云的重心位置. 熵  $En$ : 一方面反映了在数域空间可被语言值接受的范围; 另一方面还反映了在数域空间的点能够代表这个语言值的概率, 表示定性概念的云滴出现的随机性. 它揭示了模糊性和随机性的关联性. 超熵  $He$ : 是熵的不确定度量, 即熵的熵, 反映了在数域空间代表该语言值的所有点的不确定度的凝聚性, 即云滴的凝聚度.

生成云滴的算法或硬件称为云发生器<sup>[7, 10]</sup>. 下面对与本文相关的三个主要云发生器的算法进行介绍.

### 2.2 基本云发生器

#### 算法 2-1 基本正态云云发生器<sup>[10]</sup>

INPUT:  $\{Ex, En, He\}, n$  // 数字特征和云滴数

OUTPUT:  $\{(x_1, \mu_1), \dots, (x_n, \mu_n)\}$  //  $n$  个云滴

FOR  $i = 1$  to  $n$

// 生成期望值为  $En$ 、方差为  $He$  的正态随机数

$En' = \text{RANDN}(En, He)$

$x_i = \text{RANDN}(Ex, En')$

$$\mu_i = e^{-\frac{(x_i - Ex)^2}{2(En')^2}}$$

drop( $x_i, \mu_i$ ) // 生成第  $i$  个云滴

### 2.3 X条件云发生器

给定云的三个数字特征( $Ex, En, He$ )和论域  $u$  上特定的值  $x_0$ , 产生云滴 drop( $x_0, \mu_i$ ), 这种云发生器称为 X 条件云发生器.

#### 算法 2-2 X条件云发生器<sup>[10]</sup>

INPUT:  $\{Ex, En, He\}, n, x_0$

OUTPUT:  $\{(x_0, \mu_1), \dots, (x_0, \mu_n)\}$

FOR  $i = 1$  to  $n$

$En' = \text{RANDN}(En, He)$

$$\mu_i = e^{-\frac{(x_0 - Ex)^2}{2(En')^2}}$$

drop( $x_0, \mu_i$ )

### 2.4 Y条件云发生器

给定云的三个数字特征( $Ex, En, He$ )和特定的确定度  $\mu_0$ , 产生云滴 drop( $x_i, \mu_0$ ), 这种云发生器称为 Y 条件云发生器.

#### 算法 2-3 Y条件云发生器<sup>[10]</sup>

INPUT:  $\{Ex, En, He\}, n, \mu_0$

OUTPUT:  $\{(x_1, \mu_0), \dots, (x_n, \mu_0)\}$

FOR  $i = 1$  to  $n$

$En' = \text{RANDN}(En, He)$

$$x_i = Ex \pm En' \sqrt{-2 \ln(\mu_0)}$$

drop( $x_i, \mu_0$ )

## 3 云遗传算法

在连续变量空间中, 全局最优解周围存在一个邻域. 在该邻域内, 以最优解为中心, 目标函数值由远及近逼近该值. 当前解的适应度较大时, 应该在较小的邻域内进行搜索, 相反则在较大的邻域内搜索. 从而可以逐步对最优解所在的区域进行定位, 并最终逼近最优解. 以寻找飞机“黑匣子”为例, 视飞机残骸散布密度为适应度. 残骸散布密度越大(适应度越大), 较小范围内存在黑匣子的可能性越大, 应该在较小的范围搜索; 相反, 残骸散布密度越小(适应度越小), 较小范围内黑匣子存在的可能性也越小, 应该在较大的范围搜索.

云遗传算法结合遗传算法思想, 沿用 GA 的交叉、变异操作概念, 由正态云模型的 Y 条件云生成算法实现交叉操作, 基本云生成算法实现变异操作. 由于正态云模型具有随机性和稳定倾向性的特点<sup>[7, 14]</sup>, 随机性可以保持个体多样性从而避免搜索陷入局部极值, 而稳定倾向性又可以很好地保护较优个体从而对全局最值进行自适应定位. CGA 采用实数编码, 由云模型进行个体更新.

#### 算法 3-1 云遗传算法

1) 初始化种群.

2) 计算适应度

3) 选择操作

4) 交叉

4. 1) 随机生成或人为指定确定度  $\mu$ 

$$4. 2) Ex = \frac{F_f}{F_f + F_m} x_f + \frac{F_m}{F_f + F_m} x_m \quad (1)$$

4. 3)  $En =$  变量搜索范围 /  $c_1$ 4. 4)  $He = En / c_2$ 

4. 5) 由算法 2- 3 产生一对儿女

5) 变异

5. 1)  $Ex$  取原个体5. 2)  $En =$  变量搜索范围 /  $c_3$ 5. 3)  $He = En / c_4$  (注:  $c_{1-4}$  为控制系数)5. 4) 执行算法 2- 1, 并生成随机数  $Temp$ , 当  $\mu > Temp$  时, 更新个体

6) 转 2), 直到满足停止条件

式(1)中,  $x_f$  和  $x_m$  分别为交叉操作的父个体和母个体;  $F_f$  和  $F_m$  则分别对应它们的适应度. 这意味着交叉操作中的  $Ex$  由父母双方按适应度大小加权确定, 并向适应度大的一方靠拢.

显然, 交叉操作实现了染色体(个体)的整体进化, 而变异操作则是反映染色体中某个基因在一定范围内的突变. CGA 不再引入交叉变异概率.

## 4 参数影响及性能分析

由正态云发生器可知<sup>[14]</sup>: 正态云是一种泛正态分布, 呈现“中间多, 两头少”的特点. 由  $En' \sim N(En, He^2)$  知, 期望  $EX = Ex + \sqrt{-2\ln\mu}En$ , 标准差  $D = \sqrt{-2\ln\mu}He$ . 所以, 参数  $Ex$ 、 $En$  的变化分别影响云模型的水平位置和陡峭程度, 而  $He$  和云滴的离散程度呈正比,  $\mu$  与之呈反比, 即  $He$  越大, 离散程度越大,  $\mu$  越小(对云的位置来说是越靠近山脚), 云滴越分散. 所有的云滴都在期望曲线附近随机波动, 而波动的大小是由  $He$  来控制的.

### 4.1 确定度

首先讨论算法 3- 1 中 4. 1) 中的确定度. 由图 1 可知,  $\mu$  越大, 云滴越靠近顶部, 变量的搜索范围越窄. 根据引言部分叙述的寻找黑匣子现象, 我们把  $\mu$  值自适应于适应度, 逐渐对最优值进行定位. 在此, 介绍两种方法, 一种是采用线性函数的确定性方法, 另一种是采用 X 条件云发生器的随机性方法.

#### 算法 4- 1 确定性线性函数法

$$\mu = \mu_{\max} - \frac{F_{\max} - F'}{F_{\max} - F_{\min}} (\mu_{\max} - \mu_{\min}) \quad (2)$$

式中,  $F_{\max}$  和  $F_{\min}$  分别代表当代种群的全局“最大”和“最小”适应度值,  $F'$  为交叉两父个体适应度的较大者,  $\mu_{\max}$  和  $\mu_{\min}$  为人为指定的确定度的最大值和最小值, 如  $\mu_{\max} = 0.95$ ,  $\mu_{\min} = 0.2$ .

#### 算法 4- 2 随机性 X 条件云发生器法

$$Ex = \frac{F_{\max}}{F_{\max} + F_{\min}} x_f + \frac{F_{\min}}{F_{\max} + F_{\min}} x_m$$

$$En = (F_{\max} - F_{\min}) / c_5$$

$$He = En / c_6$$

$$En' = \text{RANDN}(En, He)$$

$$\mu = e^{-\frac{(F' - E_n)^2}{2(En')^2}}$$

由算法 4- 1 和 4- 2 可知, 适应度越大的个体, 搜索范围越窄, 从而有利于保护较优个体的模式; 同时, 适应度最大的个体, 其确定度为 1, 交叉操作后生成的两个子个体都为式(1)中的  $Ex$ , 从而不利于种群多样性. 为此, 我们增加了一个限制条件: 当  $\mu \geq 0.95$  时,  $\mu = 0.95$ . 为了保护最优个体, 我们在选择操作中引入了最优保留策略. 根据图 1 和“ $3\sigma$ ”规则,  $c_5$  取略小于 3 的值, 我们取  $c_5 = 2.8$ . 同时建议  $5 \leq c_6 \leq 15$ , 我们取  $c_6 = 10$ .

### 4.2 $En$ 和 $He$

$En$  越大, 则云覆盖的水平宽度大, 从而使个体在交叉、变异操作时搜索范围大. 根据“ $3\sigma$ ”规则, 结合进化算法的速度和精度, 建议算法 3- 1 中  $6 \leq c_{13} \leq 3 \times p$  ( $p$  为种群大小). 随着进化代数的增加, 可取更大的值.  $He$  过大, 会在一定程度上丧失“稳定倾向性”;  $He$  过小, 又会在一定程度上丧失“随机性”. 建议  $c_{2,4}$  在 5~ 15 内取值.

同时, 为了进化初期能扩大搜索范围, 进化后期提高搜索精度, 可以根据代数、适应度等自适应动态调整参数. 例如  $c_{1-4}$  由进化代数的 sigmoid 函数或线性函数(单调递增)给定, 同时克服了人为指定带来的不便.

尽管  $En$  和  $He$  是云模型的重要参数, 但在 CGA 中, 两者的变化可以通过  $Ex$  和确定度的改变产生相同的进化结果; 所以, 经过若干代进化后,  $Ex$  和确定度的随机性部分掩盖了它们取值不同所带来的进化结果的差异. 因此,  $c_{1-4}$  在一定范围内的取值不同不会对最终的进化性能产生重大影响, 但进化差异总是可能存在, 所以参数的选取需要以后进一步研究. 本文实验取  $c_{1,3} = 3 * p$ ,  $c_{2,4} = 10$ .

## 5 典型函数优化

采用 16 个典型函数(见附录)进行函数优化实验. F1- F5 在文献[15]中用来验证作者提出的新量子遗传算法(NQGA), F2、F3、F6- F16 在文献[16]中用来验证作者提出的 LARES 算法. 同时与文献[13]提出的云自适应遗传算法(CAGA)进行比较.

种群大小均为 100, 最大代数分别与文献[15]和文献[16]相同. F1- F5 与文献[15]中的 NQGA 比较如表 1 (100 次独立实验), F2、F3、F7- F14 与文献[16]中的 LARES 算法比较(30 次独立实验), 结果见表 2. 表 1 中 NQGA 算法的优化结果引自文献[15], 表 2 中 GA 和 LARES 算法的优化结果引自文献[16]. 平均代数指当前

最佳适应度与参考最优适应度之差小于  $10^{-3}$  时进化代数多次独立实验的平均值。

CGA 利用了基于种群的进化体制和正态云模型的稳定倾向性、随机性特点。稳定倾向性能较好地保护较佳个体从而实现最优值的自适应定位, 随机性能保持个体多样性从而提高算法防止陷入局部极值的能力, 使 CGA 能较好地保持“勘探”和“开采”间的平衡。由表 1 和表 2 可知, CGA 在进化速度、鲁棒性和避免陷入局部最优能力上都明显优于文献[15, 16]中提出的 NQGA 和 LARES 算法。由于 NQGA 和 LARES 都优于传统 GA, 改进遗传算法(IGA)和最优保留遗传算法(OMGA)(比较结果见文献[15]和文献[16]), CGA 显然更优于

GA, IGA 和 OMGA。与 CAGA 相比, CGA 除对 F1、F2、F9、F12 和 F13 平均求解精度稍差外, 对其余的函数都达到了更好的精度。除 F8、F12- F14 外, CGA 对其余函数的收敛速度明显快于 CAGA。所以, CGA 整体性能优于 CAGA。

### 6 IIR 数字滤波器优化设计

下面采用 CGA 直接在频率域进行 IIR 数字滤波器的优化设计, 设计方法同文献[15]。

IIR 数字滤波器采用二阶节的级联结构形式, 即

$$H(z) = A \prod_{k=1}^N \frac{1 + a_k z^{-1} + b_k z^{-2}}{1 + c_k z^{-1} + d_k z^{-2}} \quad (3)$$

表 1 CGA 与 NQGA、CAGA 的性能比较

函数	参考值	最佳值			平均值			平均代数		
		NQGA	CAGA	CGA	NQGA	CAGA	CGA	NQGA	CAGA	CGA
F1	0	8.1080e-005	4.3368e-017	1.3147e-009	0.000527	4.3368e-017	1.4949e-007	17.80	26.13	3.60
F2	0	0.00003541	6.3056e-012	9.6253e-009	0.00043370	3.4925e-011	2.5749e-008	67.89	161.70	39.97
F3	3	3.000121	3.000000	3.000000	3.00058551	3.000000	3.000000	190.13	75.5	40.43
F4	0	1.8627e-005	1.1369e-012	2.2204e-016	0.00024215	1.7896e-06	1.3174e-007	329.13	535.80	80.23
F5	-1.0316	-1.031581	-1.031628	-1.031628	-1.0313833	-1.031628	-1.031628	79.82	83.54	57.43

表 2 GA、LARES、CAGA 与 CGA 的性能比较

函数	参考值	最佳值				平均值		平均代数	
		GA	LARES	CAGA	CGA	CAGA	CGA	CAGA	CGA
F2	0	0.00103	0.00051	6.3056e-012	9.6253e-009	3.4925e-011	2.5749e-008	161.70	39.96
F3	3	3.03713	3.00000	3.00000	3.00000	3.00000	3.00000	75.5	40.43
F6	0.998	1.24888	1.13103	0.998004	0.998004	0.998004	0.998004	116.2	109.83
F7	0	9.423925	0.00032	5.9009e-004	8.9394e-006	7.4938e-004	1.4909e-005	41015.62	37249.8
F8	7	7.00144	7.00064	7.00000	7.00000	7.00000	7.00000	19.18	19.37
F9	-186.73	-185.588	-186.604	-186.7309	-186.7309	-186.6914	-186.6267	8663.4	8906.53
F10	1	1.00559	1.00786	1.00000	1	1.00150	1.000584	7374.9	2482.37
F11	0	0.01058	0.00247	2.4399e-009	1.4891e-010	3.6361e-007	5.8789e-009	987.2	526.1
F12	0	0.00079	0.00064	2.3842e-006	3.4605e-005	2.3842e-006	0.00014	340.9	10746.37
F13	0	0.00153	0.00153	1.1921e-005	4.1921e-005	1.1921e-005	0.00013	194.7	10642.73
F14	0	0	0	1.7217e-019	1.1809e-017	3.3200e-017	2.5825e-017	23.6	77.67
F15	0	0.19275	0.08927	0.15315	3.5051e-008	0.19605	0.01170	200000	170477.1
F16	0	14.53298	4.9654	3.28032	5.1994e-007	8.40576	1.8529e-006	200000	15027.3

举例<sup>[15]</sup>: 设计一个 6 阶的带通 IIR 数字滤波器, 其技术指标为:

$$|H_d(e^{j\omega})| = \begin{cases} 0 & 0 \leq \omega \leq 0.28\pi, 0.72\pi \leq \omega \leq \pi \\ 1 & 0.32\pi \leq \omega \leq 0.68\pi \end{cases} \quad (4)$$

CGA 设计的 IIR 数字滤波器如式(5), 其频率响应如图 2 和图 3 所示。

$$H(z) = 0.12116 \times \frac{1 + 1.4745z^{-1} + z^{-2}}{1 - 0.94906z^{-1} + 0.83404z^{-2}} \times \frac{1 - 1.4731z^{-1} + z^{-2}}{1 - 0.00073464z^{-1} + 0.45374z^{-2}}$$

$$\times \frac{1 - 0.00073156z^{-1} - z^{-2}}{1 + 0.94899z^{-1} + 0.83394z^{-2}} \quad (5)$$

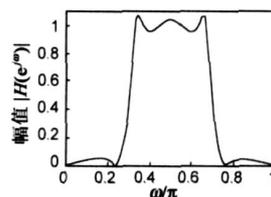


图 2 IIR 数字滤波器的幅频特性

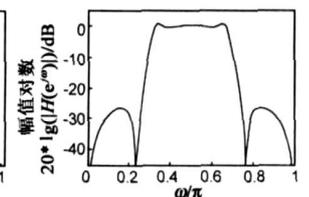


图 3 IIR 数字滤波器的对数幅频特性

表 3 对 CGA 与文献[15]中的四种算法的性能进行比较. 从表 3 可看出, 采用 CGA 设计的 IIR 数字滤波器的通带最大波动( $A_p$ ) 最小, 阻带最小衰减( $A_s$ ) 最大, 可见综合性能更佳.

表 3 各算法设计的滤波器性能

算法	OMGA	IGA	IQGA	NQGA	CGA
$A_p$ /dB	0.647	0.5764	0.732	0.3546	0.342
$A_s$ /dB	24.00	31.23	28.395	33.58	34.27

## 7 结论及下一步工作

本文利用正态云模型云滴的随机性和稳定倾向性特点, 结合遗传算法交叉、变异思想, 由云模型的 Y 条件云生成算法实现交叉操作, 基本云发生器算法实现变异操作, 巧妙地完成进化过程, 提出了全新的云遗传算法.

CGA 利用了正态云模型的稳定倾向性、随机性特点和基于种群的进化体制. 稳定倾向性可以较好地保护较佳个体从而实现对最优值的自适应定位, 随机性可以保持个体多样性从而提高算法防止陷入局部极值的能力. 基于种群的进化体制结合正态云模型的确定度特性, 使适应度较大的个体在较小的邻域内进行搜索, 适应度较小的个体在较大的邻域内进行搜索. 从而, CGA 能较好地保持“勘探”和“开采”间的平衡, 具备较好的搜索性能.

通过函数优化实验可知, CGA 算法不仅是可行的, 而且优于 GA、IGA、OMGA、NQGA、LARES 和 CAGA. 不仅进化代数少, 提高了进化速度, 而且获得最佳值的能力也较强, 多次实验的平均值更接近最优值, 从而具有较好的鲁棒性. IIR 数字滤波器优化设计的结果同时证明了算法具有较好的应用价值.

下一步的工作, 主要包括算法收敛性的理论证明、算法的参数分析与改进, 以及在工程优化中的其他应用.

### 附录: 16 个典型函数

#### F1: 简单平方和函数

$$F1 = x_1^2 + x_2^2; -5 \leq x_i \leq 5 \quad (6)$$

#### F2: DeJong's f2 函数

$$F2 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2; -2.048 \leq x_i \leq 2.048, i = 1, 2 \quad (7)$$

#### F3: Goldstein-Price 函数

$$F3 = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_2^2 + 27x_2^3)]; -2 \leq x_i \leq 2, i = 1, 2 \quad (8)$$

#### F4: DeJong's f6 函数

$$F4 = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2} - 100 \leq x_i \leq 100, i = 1, 2 \quad (9)$$

#### F5: Six-Peak 函数

$$F5 = (4 - 2.1x_1^2 + x_1^4/3)x_2^2 + x_1x_2 + (-4 + x_2^2)x_2^2; -3 \leq x_i \leq 3, i = 1, 2 \quad (10)$$

#### F6: DeJong's f5 函数

$$F6 = \left\{ 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{25} (x_i - a_j)^2} \right\}^{-1}; -65.536 \leq x_i \leq 65.536, i = 1, 2 \quad (11)$$

#### F7: Hyper-Ellipsoid 函数

$$F7 = \sum_{i=1}^{100} i^2 x_i^2; -1 \leq x_i \leq 1, 2 \quad (12)$$

#### F8: Goldstein 函数

$$F8 = x^6 - 15x^4 + 27x^2 + 250; -10 \leq x_i \leq 10 \quad (13)$$

#### F9: Shubert's 函数

$$F9 = \left\{ \sum_{i=1}^5 i \cdot \cos[(i+1)x_{i+1} + i] \right\} \left\{ \sum_{i=1}^5 i \cdot \cos[(i+1)x_{i+2} + i] \right\} + 0.5[(x_{i+1} + 1.42513)^2 + (x_{i+2} + 0.80032)^2]; -10 \leq x_i \leq 10 \quad (14)$$

#### F10: Yan and Ma 函数

$$F10 = \frac{x_1 + x_2}{2} - \cos(20\pi x_1) \cdot \cos(20\pi x_2) + 2; -10 \leq x_i \leq 10 \quad (15)$$

#### F11: Yan and Ma 函数

$$F11 = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\}; D = 5, -5 \leq x_i \leq 5 \quad (16)$$

#### F12: Schwefel's 2.21 函数

$$F12 = \max_i |x_i|, i = 1, \dots, D, -10 \leq x_i \leq 10 \quad (17)$$

#### F13: Schwefel's 2.22 函数

$$F13 = \sum_{i=1}^5 |x_i| + \prod_{i=1}^5 |x_i|; -10 \leq x_i \leq 10 \quad (18)$$

#### F14: Schwefel's 2.23 函数

$$F14 = \sum_{i=1}^5 x_i^{10}; -10 \leq x_i \leq 10 \quad (19)$$

#### F15: Griewangk's 函数

$$F15 = \sum_{i=1}^D \frac{x_i^2}{4000} + \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1; D = 10, -512 \leq x_i \leq 512 \quad (20)$$

#### F16: Rastrigin's 函数

$$F16 = 10D + \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i)); D = 15, -10 \leq x_i \leq 10 \quad (21)$$

### 参考文献:

[1] Holland J.H. Adaptation in Natural and Artificial System[M].

- Ann Arbor: The University of Michigan Press, 1975.
- [2] De Jong K A. An Analysis of the Behavior of a Class of Genetic Adaptive System[D]. PhD Dissertation. Ann Arbor: University of Michigan, USA, 1975.
- [3] D. E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning[M]. Reading, MA: Addison Wesley, 1989.
- [4] 王小平, 曹立明. 遗传算法 理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 2002: 20- 100.
- [5] 肖龙光. 部分智能优化算法的改进及其数学理论分析[D]. 上海: 东华大学, 2005. 12- 15.  
Xiao Longguang. Improvement of Partial Intelligent Optimization Algorithm and Analysis of Its Mathematical Theory[D]. Shanghai: Donghua University, 2005. 12- 15. (in Chinese)
- [6] 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, 31(1): 71- 74.  
Chen Xiaoping, Yu Shenglin. Improvement on crossover strategy of real-valued genetic algorithm[J]. Acta Electronica Sinica, 2003, 31(1): 71- 74. (in Chinese)
- [7] 李德毅, 孟海军, 史雪梅. 隶属云和隶属云发生器[J]. 计算机研究与发展, 1995, 32(6): 15- 20.  
Li Deyi, Meng Haijun, Shi Xuemei. Membership clouds and membership cloud generators[J]. Journal of Computer Research and Development, 1995, 32(6): 15- 20. (in Chinese)
- [8] 张飞舟, 范跃祖, 沈程智, 李德毅. 基于隶属云发生器的智能控制[J]. 航空学报, 1999, 20(1): 89- 92.  
Zhang Feizhou, Fan Yuezuo, Shen Chengzhi, etc. Intelligent control based membership cloud generators[J]. Acta Aeronautica et Astronautica Sinica. 1999, 20(1): 89- 92. (in Chinese)
- [9] Wang Shuliang, Li Deren, Shi wenzhong, et al. Cloud Model-Based Spatial Data Mining[J]. Geographical Information Science, 2003, 9(2): 67- 78.
- [10] 李兴生. 基于云模型和数据场的分类和聚类挖掘研究[D]. 南京: 中国人民解放军理工大学, 2003. 16- 19.  
Li Xingsheng. Study on Classification and Clustering Mining Based on Cloud Model and Data Field[D]. Beijing: PLA University of Science and Technology, 2003. 16- 19. (in Chinese)
- [11] 赵卫伟, 李德毅. 基于云模型的入侵检测方法[J]. 计算机工程与应用, 2003, 39(26): 158- 160.  
Zhao Weiwei, Li Deyi. Intrusion detection using cloud model[J]. Computer Engineering and Applications, 2003, 39(26): 158- 160. (in Chinese)
- [12] 宋远骏, 杨孝宗, 李德毅, 崔东华. 考虑环境因素的计算机可靠性评价[J]. 计算机研究与发展, 2001, 38(5): 631- 636.  
Song Yuanjun, Yang Xiaozong, Li Deyi. Reliability count evaluation of computers based on cloud models for environmental factors[J]. Journal of Computer Research and Development, 2001, 38(5): 631- 636. (in Chinese)
- [13] Zhu Yunfang, Dai Chaohua, Chen Weirong, Lin Jianhui. Adaptive probabilities of crossover and mutation in genetic algorithms based on cloud generators[J]. Journal of Computational Information Systems, 2005, 1(4): 671- 678.
- [14] 刘常昱, 李德毅, 杜 , 等. 正态云模型的统计分析[J]. 信息与控制, 2005, 34(2): 236- 239.  
Liu Changyu, Li Deyi, Du Yi etc. Some statistical analysis of the normal cloud model[J]. Information and Control, 2005, 34(2): 236- 239. (in Chinese)
- [15] 张葛祥, 李娜, 金炜东, 等. 一种新量子遗传算法及其应用[J]. 电子学报, 2004, 32(3): 476- 479.  
Zhang Gexiang, Li Na, Jin Weidong, etc. A novel quantum genetic algorithm and its application[J]. Acta Electronica Sinica, 2004, 32(3): 476- 479. (in Chinese)
- [16] Roberto Irizarry. LARES: an artificial chemical process approach for optimization[J]. Evolutionary Computation, 2004, 12(4): 435- 459.

#### 作者简介:



戴朝华 男, 1973年8月生于湖南新化, 西南交通大学电气工程学院博士研究生, 研究方向: 智能信息处理、模式识别。  
E-mail: ddzyf@126.com



朱云芳 女, 1973年7月生于山西晋城, 西南交通大学峨眉校区计算机系讲师, 工学硕士, 研究方向: 信号分析与处理、测控系统。

陈维荣 男, 1965年1月生于四川内江, 西南交通大学电气工程学院教授, 博士, 博士生导师, 研究方向: 电力系统及其自动化、工业监控技术。