

基于进化策略方法求任意函数的数值积分

周永权¹⁾ 张 明²⁾ 赵 斌³⁾

¹⁾ (广西民族大学数学与计算机科学学院 南宁 530006)

²⁾ (大连水产学院理学院 辽宁 大连 116023)

³⁾ (中央民族大学理学院 北京 100081)

摘 要 提出了两种基于进化策略求任意函数数值积分的新方法,其中方法一是基于混合基函数进化策略的数值积分算法;方法二是基于不等距点分割的进化策略数值积分算法.两种算法都采用适用于高维优化问题的单基因突变进化策略,使得该算法不但能计算通常意义下任意函数的定积分,而且能计算奇异函数积分和振荡函数积分.最后给出几个数值积分算例,并与传统数值积分方法作了比较,仿真结果分析表明,两种算法十分有效,能够快速有效地获得任意函数的数值积分值.

关键词 混合基函数;不等距点分割;适应度;进化策略;单基因突变;数值积分

中图法分类号 TP18

Solving Numerical Integration Based on Evolution Strategy Method

ZHOU Yong-Quan¹⁾ ZHANG Ming²⁾ ZHAO Bin³⁾

¹⁾ (College of Mathematics and Computer Science, Guangxi University for Nationalities, Nanning 530006)

²⁾ (School of Science, Dalian Fisheries University, Dalian, Liaoning 116023)

³⁾ (School of Science, Central University for Nationalities, Beijing 100081)

Abstract In this paper, two new kinds of calculating numerical integration methods based on evolution strategy algorithm are proposed. One of which is based on mixing basis function evolution strategies for solving numerical integration, and the other is based on inequality point's segmentation for solving numerical integration. Both of these two algorithms adopt single-gene mutation evolution strategies that suit for high dimensional optimization, which can not only compute usual definite integral for any functions, but also compute singular integral and oscillatory integral. Finally, several experimental results show that the two proposed numerical integration methods are more efficient and feasible in computing the arbitrary functions numerical integration compared with traditional numerical integration methods.

Keywords hybrid basis functions; inequality point's segmentation; fitness; evolution strategies; single-gene mutation; numerical integration

1 引 言

在工程技术与自然科学领域,许多现象的定量

分析往往可以抽象地归结为求解特定的数学问题.一般来说,这些数学问题不易甚至无法求得它们的精确解,需要借助近似方法得到问题的近似数值解,

其中,数值积分 $\int_a^b f(x) dx$ 是一个非常基本的计算

问题. 但是, 众所周知, 除了有限的几种类型外, 对于一般的被积函数 $f(x)$ 往往无法求得其原函数. 因此, 用合适的有限和近似代替所求被积函数的定积分, 即

$$\int_a^b f(x) dx \approx \sum_{k=0}^m a_k f(x_k), \text{ 这里, } x_0, x_1, \dots, x_m \text{ 是区间}$$

$[a, b]$ 上的某些点, a_0, a_1, \dots, a_m 是与 $f(x)$ 无关的常数. 若按这个近似计算公式需要一些复杂的计算, 且常数 a_0, a_1, \dots, a_m 一般不易确定. 于是, 出现了计算定积分的 Newton-Leibniz 公式: $\int_a^b f(x) dx = F(b) - F(a)$, ($F'(x) = f(x)$). 到目前为止, 在计算定积分时, 人们通常采用的方法是先求出被积函数的原函数, 然后由 Newton-Leibniz 公式计算, 事实上很多定积分不易或者不能直接用上述方法来计算, 因为许多问题被积函数的原函数不能显式地表示出来; 或者原函数表达式过于复杂而不适于计算, 因此有必要提出一种数值积分新方法, 以解决定积分的近似计算方法所存在的不足. 此外, 数值积分虽然计算方法很多, 如梯形求积方法、抛物线求积方法、Newton-Cotes 方法、Romberg 方法、Gauss 方法等^[1-2], 其中梯形求积方法和抛物线求积方法适用于光滑性较差的被积函数, 但是对于其它被积函数精度较低; Newton-Cotes 方法是一种利用插值多项式来构造数值积分的常用方法, 但高阶的 Newton-Cotes 方法的收敛性没有保证, 实际计算中很少使用; Romberg 方法虽然收敛速度快, 计算精度较高, 但是计算量较大; Gauss 方法积分精度高、数值稳定、收敛速度较快, 但节点与系数的计算较困难等等. 针对这些问题, 文中首先提出了一种基于混合基函数进化策略算法的数值积分近似计算方法, 其基本思想是利用混合基函数进化策略算法来逼近被积函数以实现定积分的近似计算.

另一方面, 由于以上提到的数值积分公式, 它们大都是基于等距节点分割求和的求积公式, 或是整个计算过程中规定等距节点不变, 或是在选好的等距节点的基础上继续等距分割. 所有这些具有等距节点的求积公式并不能根据被积函数的形状来划分, 要想得到较高的精度必须需要更多的节点, 最好的分割并不是预先给定的等间距, 而是根据被积函数的形状, 即根据被积函数曲线的凹凸形状随机地选定分割点, 即根据函数值变化情况随机产生不等距的子区间. 这样分割求和, 所得到函数的积分值比传统的方法精度要高. 针对这个思想, 文中接着又提出了另一种基于不等距点分割的进化策略数值积分

方法, 其基本思想是: 首先, 在积分区间上随机地产生一些分割点, 然后对这些分割点进行优化, 最后依据这些优化后得到的割点, 作为分割点 (未必是等距离) 再作和的方法, 便可得到更准确的积分值. 该算法的好处在于不但能计算通常意义下函数的定积分, 而且还能计算奇异函数积分和振荡函数积分.

针对文中提出的两种进化策略算法用于求数值积分的新方法, 通过 5 个仿真实验与传统的数值积分方法作了比较. 分析表明, 该算法十分有效, 能够快速有效地获得任意函数的数值积分, 且不需要给定被积函数, 因此能有效地解决建模困难的系统或未知系统的求积分问题, 可看作是对传统的数值积分方法的一种改进, 在工程实际中有较大的应用价值.

2 进化策略简介

1963 年, 德国柏林工业大学的 Schwefel 等为了研究风洞中流体力学问题, 提出进化策略 (Evolution Strategies, ES)^[3-6]. 由于设计中描述物体形状的参数难以用传统方法进行优化, 因而利用生物突变的思想来随机改变参数值, 获得了较好的结果. 随后, 他们对这种方法进行了深入地研究和发展产生了此算法. ES 是专门为求解参数优化问题而设计的, 而且在 ES 算法中引进了自适应机制, 隐含并行性和群体全局搜索性是它的两个显著特征, 而且具有较强的鲁棒性, 对于一些复杂的非线性系统求解具有独特的优越性能. ES 从 $(1+1)$ -ES, $(\lambda+1)$ -ES 演变为现在最常用的 $(\mu+\lambda)$ -ES 和 (μ, λ) -ES, 统称为 CES (Classical Evolution Strategies), 提高 ES 的收敛速度和全局搜索能力, 一直是 ES 研究的主要内容之一. CES 使用 Gauss 突变算子, Yao^[7] 和 Kappler^[8] 引入了 Cauchy 突变算子, 利用其较强的分散性获得了较好的跳出局部极值点的能力, 但收敛速度有所降低; 林丹^[9], Gunter^[10] 通过理论分析表明 Gauss 突变算子具有较强的局部搜索能力, 而均匀突变算子和 Cauchy 突变算子具有较强的局部逃逸能力. Chang^[11] 对 ES 中使用组合算子的效果进行了探索, 一组测试函数的计算结果表明组合算子对单模态函数的优化性能有改进, 而对多模态函数的效果并不显著, 提高收敛速度和全局搜索能力, 仍然是 ES 和其它进化算法的研究热点, 对于高维优化问题 (维数大于 10) 尤其重要, 于是, 王湘中^[12] 等提出了适用于高维优化问题的改进进化策略, 建立了单基因突变与均匀突变相结合、使用精英繁殖、

递减型策略参数、小种群规模的 $(\mu + \lambda + k)$ -ES,该算法特别对于高维优化问题效果显著。

在文献[12]方法的基础上,文中建立了单基因 Gauss 突变,单基因 Cauchy 突变和均匀突变相结合、使用精英繁殖的 $(\mu + 2\lambda + k)$ -ES,采用 3 种突变方式结合的目的是充分发挥各自的优势,使算法收敛速度加快。在此基础上,文中设计出了两种基于进化策略的方法求任意被积函数的积分,其中,方法一是基于混合基函数进化策略算法求解数值积分的新方法;方法二是基于不等距点分割的进化策略求解数值积分的新方法。两种方法的共同之处不但能计算通常意义下函数的定积分,而且能计算奇异积分和振荡积分的近似值。最后,通过实验与传统的数值积分方法作了比较,模拟实验表明,本文所提出的两种算法,具有积分精度高、自适应性强、收敛速度快等特点,能够快速有效地获得任意函数的数值积分值。

3 两种用于求数值积分的进化策略模型

3.1 混合基函数进化策略模型

本节提出的基于进化策略的数值积分计算方法,其基本思想是利用进化策略实现用一组基函数的线性组合来逼近被积函数以实现定积分的数值计算。基函数如何选取是关键,对于一些实际问题,有时会有一些先验知识,它会启示我们用某一基函数(如三角函数、多项式、Fourier 函数、B 样条基函数等),或者是一些混合基函数的组合,如本节的方法我们可选用以下混合基函数。

3.1.1 混合基函数

不妨设混合基函数由三角函数、指数函数和幂函数组成,令

$$c_j(x) = \begin{cases} \cos(jx), j=0, 1, 2, \dots, N/4 \\ \sin[(j-N/4)x], \\ \quad j=N/4+1, N/4+2, \dots, 2N/4 \\ e^{x/(j-2N/4)}, \\ \quad j=2N/4+1, 2N/4+2, \dots, 3N/4 \\ x^{1/(j-3N/4)}, \\ \quad j=3N/4+1, 3N/4+2, \dots, N \end{cases} \quad (1)$$

其中, N 为 4 的倍数, 设 w_j 为进化策略优化权值, $c_j(x)$ 为三角基函数、指数基函数和幂基函数, 且 $x \in [0, 2\pi]$, 进化策略优化参数矩阵为 $\mathbf{W} = (w_0, w_1, \dots, w_N)^T$, 基函数矩阵为 $\mathbf{C}(x) = (c_0(x), c_1(x), \dots,$

$c_N(x))^T$, 则有

$$g(x) = \sum_{j=0}^N w_j c_j(x) = \mathbf{W}^T \mathbf{C}(x) \quad (2)$$

误差函数定义为

$$e(t) = f(x_t) - g(x_t), \quad t = 0, 1, \dots, n-1 \quad (3)$$

其中 n 为积分区间内的等距节点数, $f(x)$ 为被积函数, 即目标函数, $g(x)$ 为理想函数, 用来逼近于被积函数 $f(x)$ 。

3.1.2 积分定理

定理 1. 设 a, b 为积分上下限, 且 $0 \leq a, b \leq 2\pi$, w_j 为进化策略优化参数, 则有

$$\begin{aligned} I = \int_a^b f(x) dx &\approx (b-a)w_0 + \\ &\sum_{j=1}^{N/4} \frac{1}{j} w_j [\sin(jb) - \sin(ja)] + \\ &\sum_{j=1}^{N/4} \frac{1}{j} w_{N/4+j} [\cos(ja) - \cos(jb)] + \\ &\sum_{j=1}^{N/4} j w_{2N/4+j} [e^{b/j} - e^{a/j}] + \\ &\sum_{j=1}^{N/4} \frac{j}{j+1} w_{3N/4+j} [b^{(j+1)/j} - a^{(j+1)/j}]. \end{aligned}$$

证明。

$$\begin{aligned} I = \int_a^b f(x) dx &\approx \int_a^b g(x) dx \\ &= \int_a^b \left[\sum_{j=0}^{N/4} w_j \cos(jx) + \sum_{j=1}^{N/4} w_{N/4+j} \sin(jx) + \right. \\ &\quad \left. \sum_{j=1}^{N/4} w_{2N/4+j} e^{x/j} + \sum_{j=1}^{N/4} w_{3N/4+j} x^{1/j} \right] dx \\ &= \int_a^b \left[w_0 + \sum_{j=1}^{N/4} w_j \cos(jx) + \sum_{j=1}^{N/4} w_{N/4+j} \sin(jx) + \right. \\ &\quad \left. \sum_{j=1}^{N/4} w_{2N/4+j} e^{x/j} + \sum_{j=1}^{N/4} w_{3N/4+j} x^{1/j} \right] dx \\ &= (b-a)w_0 + \sum_{j=1}^{N/4} w_j \int_a^b \cos(jx) dx + \\ &\quad \sum_{j=1}^{N/4} w_{N/4+j} \int_a^b \sin(jx) dx + \\ &\quad \sum_{j=1}^{N/4} w_{2N/4+j} \int_a^b e^{x/j} dx + \sum_{j=1}^{N/4} w_{3N/4+j} \int_a^b x^{1/j} dx = \\ &= (b-a)w_0 + \sum_{j=1}^{N/4} \frac{1}{j} w_j [\sin(jb) - \sin(ja)] + \\ &\quad \sum_{j=1}^{N/4} \frac{1}{j} w_{N/4+j} [\cos(ja) - \cos(jb)] + \\ &\quad \sum_{j=1}^{N/4} j w_{2N/4+j} [e^{b/j} - e^{a/j}] + \\ &\quad \sum_{j=1}^{N/4} \frac{j}{j+1} w_{3N/4+j} [b^{(j+1)/j} - a^{(j+1)/j}]. \end{aligned}$$

注意以上我们给出的混合基函数中每类基函数阶次均为 $N/4$, 从定理的证明过程可看出, 实际上每类基函数可取不同的阶次. 一般来讲, 对混合基函数尽量选取一些初等函数作为基函数, 做到其积分值简单、易计算. 在实际应用中可考虑以下两种取法:

(1) 混合基函数选取正交基函数(如不同频的三角函数、正交多项式等)作为基函数, 其基函数的线性组合可实现对被积函数的逼近^[13].

(2) 混合基函数选取近似计算中最常用的 B 样条基函数(如单变量基函数), 其基函数的线性组合可实现对被积函数以任意精度逼近定义在致密区间上的连续实函数^[14-15].

3.2 标准进化策略

标准进化策略^[3-4,6]的实现过程可简述如算法 1.

算法 1.

1. 二元表达式. 这种表达式中个体由目标变量 X 和标准差 σ 两部分组成, 每部分又有 n 个分量, 即

$$(X, \sigma) = ((x_1, x_2, \dots, x_i, \dots, x_n), (\sigma_1, \sigma_2, \dots, \sigma_i, \dots, \sigma_n)),$$

其中, X 和 σ 之间的关系

$$\begin{cases} \sigma'_i = \sigma_i \cdot \exp(r' \cdot N(0, 1) + r \cdot N_i(0, 1)) \\ x'_i = x_i + \sigma_i \cdot N_i(0, 1) \end{cases} \quad (4)$$

式中, (x_i, σ_i) 为父代个体的第 i 个分量; (x'_i, σ'_i) 为子代新个体的第 i 个分量; $N(0, 1)$ 为服从标准正态分布的随机数; $N_i(0, 1)$ 为针对第 i 个分量重新产生一次符合标准正态分布的随机数; r' 为全局系数, 常取 1; r 为局部系数, 常取 1.

上式表明, 新个体是在旧个体基础上随机变化而来.

2. 随机生成初始群体, 并计算其适应度. 进化策略中初始群体由 μ 个个体组成, 每个个体 (X, σ) 内又可以包含 n 个 x_i, σ_i 分量. 产生初始个体的方法是随机生成. 为了便于和传统的方法相比较, 可从某个初始点 $(X(0), \sigma(0))$ 出发, 通过多次突变产生 μ 个初始个体, 该初始点可从可行域中用随机方法选取. 初始个体的标准差 $\sigma(0) = 3.0$.

3. 计算初始个体的适应度, 若满足条件, 终止; 否则, 往下进行.

4. 根据进化策略, 用下述操作产生新群体:

4.1. 重组. 将两个父代个体交换目标变量和标准差, 产生新个体. 一般目标变量采用离散重组, 标准差采用中值重组.

4.2. 突变. 对重组后的个体添加随机量, 按式(1)产生新个体.

4.3. 计算新个体适应度.

4.4. 选择. 按照 (μ, λ) 选择策略, 在新产生的 λ 个新个体择优选取 μ 个个体作为下一代群体, 这里要求 $\lambda > \mu$.

5. 反复执行步 4, 直到达到终止条件, 选择最优个体作为最终的结果.

注意: 在步 2, 初始个体的标准差 $\sigma(0)$, 可用公式 $\sigma(0) = \Delta X / \sqrt{n}$ 来计算, 其中 $\Delta X, n$ 分别表示初始点与最优点的距离和个体中所含分量的个数; 由于

ΔX 在初始时不便确定, 可取 $\sigma(0) = 3.0$. $\sigma(0)$ 不宜取得太大, 若 $\sigma(0)$ 太大而且 μ 也太大, 选择力度不够, 容易造成群体过于分散. 尽管 $\sigma(0)$ 较小, 但在进化的过程中通过个体的自适应调整仍可使搜索点很快分布在可行域.

其次, 为了控制群体的多样性和选择的力度, 比值 μ/λ 是一个重要的参数, 对算法的收敛速度有很大的影响. 若 μ 太小(比如 $\mu = 1$), 则群体太单调; 若 μ 太大, 则计算量过大. 通常, μ 取 15 或更多一些. λ 数值的大小, 类似于 μ 的作用, 也要适当. 研究表明, 比值 μ/λ 宜取 1/7 左右. 本文最后一部分 5 个仿真实验也验证了比值 μ/λ 取法的正确性.

4 适用于高维优化的进化策略

4.1 标准差

标准差指 σ , 它控制突变的幅度. 进化策略中每个个体基因都有各自的标准差, 传统进化策略选取的标准差随机性过大, 不能很好跟踪进化的进程^[16]. 事实上, 随着不断进化, 优秀个体逐步接近极值点, 突变的幅度应该随之减小(代价是牺牲了全局搜索能力), 而且每个基因的变化在统计意义上是同步的. 在 $(\mu + 2\lambda + k)$ -ES 中, 一个种群的所有个体、所有基因共用一个相同的标准差 σ , 使用确定性递减策略, 即如果若干代(如 30 代, 称为一个进化时段)没有进化, 则确定性地减小 σ , 如递减系数为 0.75. 这样, 标准差的计算开销远比传统方法小.

4.2 精英繁殖

进化策略采用共同繁殖方式产生子代, 进化的初始阶段, 共同繁殖方式使每个父个体都有相同的繁殖机会, 相当于多点同时搜索, 有利于扩大搜索范围, 但在进化后期, 父个体逐步向最优秀的个体附近聚集, 此时, 其他较差个体很难产生具有足够生存竞争力的优秀个体, 更优秀的后代主要由最优秀的精英个体产生, 进化实际是精英个体产生的后代完成, 其它个体产生的后代基本没有生存能力, 反而增加了计算开销. $(\mu + 2\lambda + k)$ -ES 使用精英繁殖方式, 即每个子个体都由父代中最优秀的精英个体产生, 以增加产生更优秀后代的机会, 提高收敛速度. 就计算复杂性而言, 精英繁殖方式的附加计算开销略少于共同繁殖方式.

4.3 突变算子

传统的进化策略采用全基因高斯突变算子, 一般认为, 这样有利于从各个方向搜索, 事实上, 当问

题规模比较大时,由于突变方向的随机性,往往使突变的效果互相抵消,很难得到进化的后代个体,导致进化策略在求解大规模问题时的收敛速度很低,甚至导致不收敛,而且这种突变的计算开销也大.在 $(\mu+2\lambda+k)$ -ES中,采用单基因突变方式,即在产生后代时,只随机选择其中的一个基因突变,其它基因直接从父个体继承.因为 Gauss 突变具有很强的局部搜索能力和较弱的全局搜索能力,而 Cauchy 突变和均匀突变全局搜索能力强局部搜索能力弱,所以把三者结合充分发挥各自的优势,使进化速度更快,准确度更高.

综上所述, $(\mu+2\lambda+k)$ -ES 繁殖与突变算子描述如下:

假定当前最优父代个体 $x_i = \max\{x_i, i=1, 2, \dots, \mu+2\lambda+k\}$, 产生 λ 个单基因 Gauss 突变子个体,即

$$\begin{cases} x'_k(j) = x_i(j) + \sigma_i N(0, 1), & j = \text{random}(1, n) \\ x'_k(i) = x_i(i), & k=1, 2, \dots, \lambda, i=1, 2, \dots, n, i \neq j \end{cases} \quad (5)$$

然后产生 λ 个单基因 Cauchy 突变子个体,即

$$\begin{cases} x'_k(j) = x_i(j) + \sigma_i (\eta + N(0, 1)), & j = \text{random}(1, n) \\ x'_k(i) = x_i(i), & k=1, 2, \dots, \lambda, i=1, 2, \dots, n, i \neq j \end{cases} \quad (6)$$

最后产生 k 个均匀突变子个体,即

$$\begin{cases} x'_k(j) = x_i(j) + \xi_j, \\ j = \text{random}(1, n), \xi_j = \text{random}(x_{j_{\min}}, x_{j_{\max}}), \\ x'_k(i) = x_i(i), & k=1, 2, \dots, \lambda, i=1, 2, \dots, n, i \neq j \end{cases} \quad (7)$$

其中, σ_i 是标准差,它控制突变的幅度;参数 η 是当 $t=1$ 时的一个柯西分布的随机变量比例参数. 计算 $(2\lambda+k)$ 个新个体的适应值,再从 $(\mu+2\lambda+k)$ -ES 个个体中选择 μ 个最优的个体作为下一代的父个体.

5 两种基于进化策略的数值积分算法

5.1 基于混合基函数进化策略的数值积分算法

根据第 3.1 节混合基函数及积分定理,基于混合基函数进化策略的数值积分算法可描述如算法 2.

算法 2.

1. 确定个体的二元表达式:表达式中个体由目标变量 W 和标准差 σ 两部分组成,每部分有 $(N+1)$ 个分量,即

$$(W, \sigma) = ((w_0, w_1, \dots, w_N), (\sigma_0, \sigma_1, \dots, \sigma_N)),$$

其中, (w_0, w_1, \dots, w_N) 表示进化策略优化权值; $(\sigma_0, \sigma_1, \dots, \sigma_N)$ 表示各个权值对应的标准差,用来调整各个权值的变化幅度.

2. 随机生成初始群体:进化策略中初始群体由 μ 个个体组成,每个个体 (W, σ) 内包含 $(N+1)$ 个 w_i, σ_i 分量,产生初始个体的方法是随机生成.为了便于和传统的方法比较,可以从某个初始点 $(W(0), \sigma(0))$ 出发,通过多次突变产生 μ 个初始个体,该初始点可从可行域中用随机方法选取.初始个体的标准差 $\sigma(0) = 2.0$.

3. 计算适应度:按式(3)中的误差函数规定适应度函数

$$G = \sum_{t=0}^{n-1} |e(t)| \quad (8)$$

其中, $t=0, 1, \dots, n-1$; n 为积分区间内的等距节点数.适应度越趋近 0,表明该个体越优良.终止条件规定最大进化代数 T ,当进化代数大于 T 时程序运行终止.

4. 如果满足条件,终止,选出最优解;否则,继续往下进行.

5. 根据进化策略,按照下述操作产生新群体:

5.1. 突变.在父代群体中挑出最优个体,然后对该个体进行单基因高斯突变、单基因柯西突变和均匀突变,产生 $(2\lambda+k)$ 个新个体.

5.2. 计算新个体适应度.

5.3. 选择.在 $(\mu+2\lambda+k)$ 个个体中挑选优良的 μ 个个体组成下一代群体.

6. 反复执行步 5,直到到达终止条件,选择最佳个体作为进化策略的结果.

7. 条件终止时,选出最优解 $W_{\text{best}}(w_0, w_1, \dots, w_N)$,再按定理 1 的公式计算出所要求被积函数的积分值.

5.2 基于不等距点分割的进化策略数值积分算法

根据第 4 节高维优化进化策略,基于不等距点分割的进化策略数值积分算法,可描述如下:

算法 3.

1. 确定个体的表达式.表达式中个体由目标变量 X 和标准差 σ 两部分组成,每部分有 $(n+1)$ 个分量,即

$$(X, \sigma) = ((x_1, x_2, \dots, x_n), (\sigma_1, \sigma_2, \dots, \sigma_n)),$$

其中, n 表示积分区间内的节点数; (x_1, x_2, \dots, x_n) 表示积分区间内的节点; $(\sigma_1, \sigma_2, \dots, \sigma_n)$ 表示各个节点对应的标准差,用来调整各个节点的变化幅度.

2. 随机生成初始群体.进化策略中初始群体由 μ 个个体组成,每个个体 (X, σ) 内包含 n 个 x_i, σ_i 分量,产生初始个体的方法是随机生成.为了便于和传统的方法比较,可以从某个初始点 $(X(0), \sigma(0))$ 出发,通过多次突变产生 μ 个初始个体,该初始点可从可行域中用随机方法选取.初始个体的标准差 $\sigma(0) = 3.0$.

3. 计算适应度.将随机产生的每一个个体分别置于积分区间的左端点和右端点之间,各自按照升序排好顺序.这样,对应每个个体积分区间总共有 $n+2$ 个节点和 $n+1$ 个小段,然后分别计算这 $n+2$ 个节点相邻节点之间的距离 $d_j, j=1, 2, \dots, n+1$,再计算出这 $n+2$ 个节点对应的函数值和每小段中间节点的函数值.找出每小段左端点、中间节点和右端点的函数值中的最小值 w_j 和最大值 $W_j, j=1, 2, \dots, n+1$,则该个体的适应度为

$$f(j) = \frac{1}{2} \sum_{j=1}^{n+1} d_j |w_j - w_j| \tag{9}$$

适应度越趋近于 0,表明该个体越优良. 终止条件选择一个很接近 0 的值 ϵ ,当最小适应度小于 ϵ 时,程序运行终止.

4. 如果满足条件,终止,选出最优解;否则,继续往下进行.
5. 根据进化策略,按照下述操作产生新群体:

5.1. 突变. 在父代群体中挑出最优个体,然后对该个体进行单基因高斯突变、单基因柯西突变和均匀突变,产生 $(2\lambda + k)$ 个新个体.

5.2. 计算新个体适应度.

5.3. 选择. 在 $(\mu + 2\lambda + k)$ 个个体中挑选优良的 μ 个个体组成下一代群体.

5.4. 如果一个进化时段(如 30 代)的最优个体没有进化,则 $\sigma_{t+1} = 0.75\sigma_t$, 否则 $\sigma_{t+1} = \sigma_t$.
6. 反复执行步 5,直到到达终止条件,选择最佳个体作为进化策略的结果.

7. 条件终止时,所求积分值就近似等于 $\sum_{j=1}^{n+1} m_j d_j$. 其中, $(m_1, m_2, \cdots, m_{n+1})$ 为由积分区间的左端点、最优个体和右端点划分成的 $n+1$ 个小段的中点对应的函数值; $(d_1, d_2, \cdots, d_{n+1})$ 为由积分区间的左端点、最优个体和右端点划分成的 $n+1$ 个小段对应的距离.

6 仿真实验

为了验证本文提出的两种进化策略数值积分算法的有效性和正确性,本文选取了参考文献[13-17]中给出的一些实例,并与传统的梯形法、Simpson 方法、Composite Simpson 方法和 Romberg 方法比较.

例 1. 文献[17]中用梯形法和 Simpson 方法在 $[0, 2]$ 积分区间分别计算被积函数 $x^2, x^4, \sqrt{1+x^2}, 1/(1+x), \sin x, e^x$ 等 6 个函数的积分,结果如文献[17]中的表 4. 7 所示.

若用本文基于混合基函数的 ES 的进化策略算法,取 $\mu=5, \lambda=7 \times \mu, k=8, n=20, N=20, T=1000$;用本文基于不等距点分割的进化策略数值积分方法,取 $\mu=15, \lambda=7 \times \mu, n=60$. 表 1 给出了文献[17]中的结果与本文算法计算的结果.

表 1 几种数值积分方法对应各函数的积分值						
$f(x)$	函数 1	函数 2	函数 3	函数 4	函数 5	函数 6
	x^2	x^4	$\sqrt{1+x^2}$	$1/(1+x)$	$\sin(x)$	e^x
梯形法	4.000	16.000	3.326	1.333	0.909	8.389
Simpson 方法	2.667	6.667	2.964	1.111	1.425	6.421
基于混合基函数的 ES 方法	2.659	6.338	2.956	1.090	1.419	6.390
基于不等距点分割的 ES 方法	2.666	6.398	2.9577	1.098	1.416	6.388
精确值	2.667	6.400	2.958	1.099	1.416	6.389

图 1~图 3 分别给出了基于混合基函数的进化策略(ES)在计算 6 个函数积分时积分误差的变化曲线.

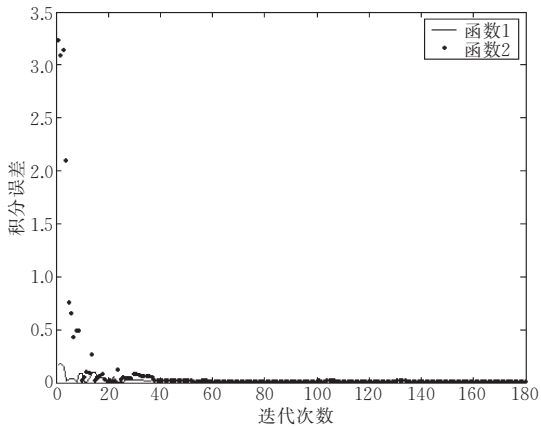


图 1 函数 1 和函数 2 的积分误差的变化曲线 (例 1,基于混合基函数)

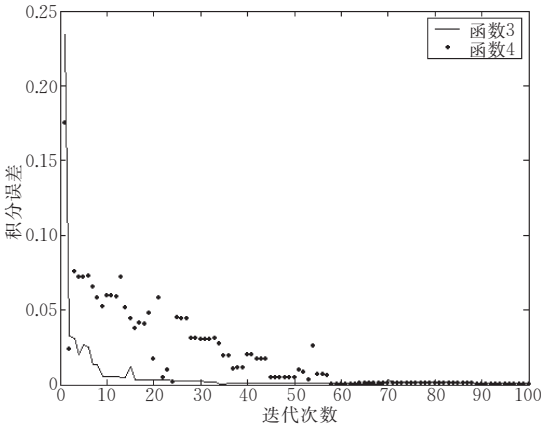


图 2 函数 3 和函数 4 的积分误差的变化曲线 (例 1,基于混合基函数)

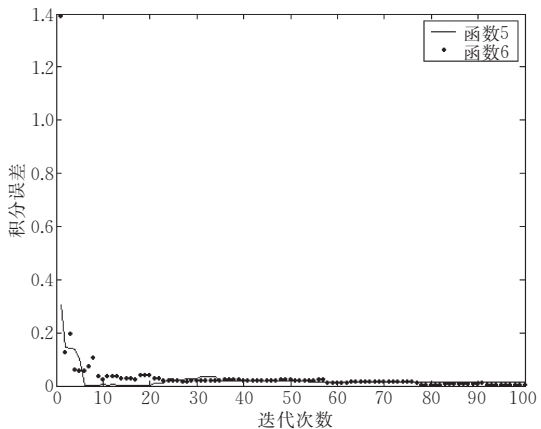


图 3 函数 5 和函数 6 的积分误差变化曲线 (例 1,基于混合基函数)

图 4~图 6 分别给出了基于不等距点分割的进化策略(ES)在计算 6 个函数积分时积分误差的变化曲线.

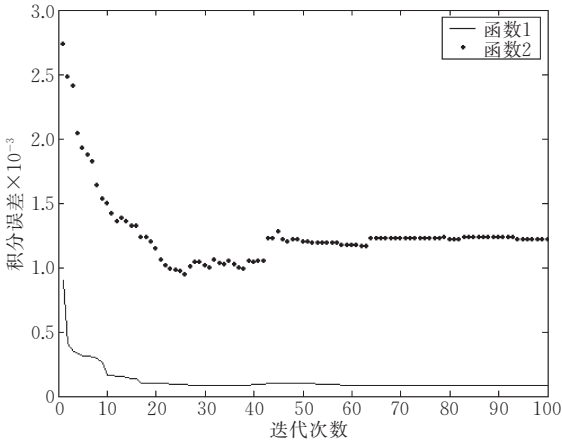


图 4 函数 1 和函数 2 的积分误差的变化曲线
(例 1, 基于不等距点分割)

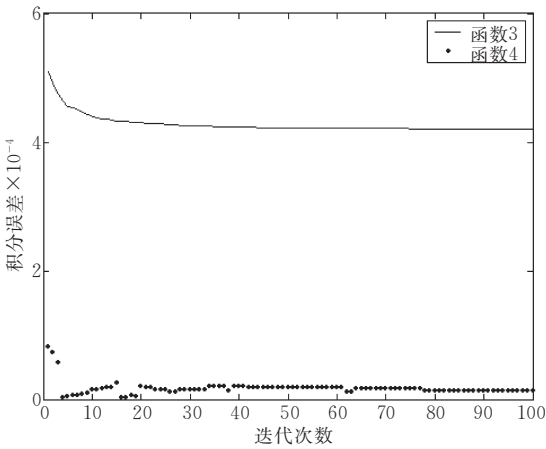


图 5 函数 3 和函数 4 的积分误差的变化曲线
(例 1, 基于不等距点分割)

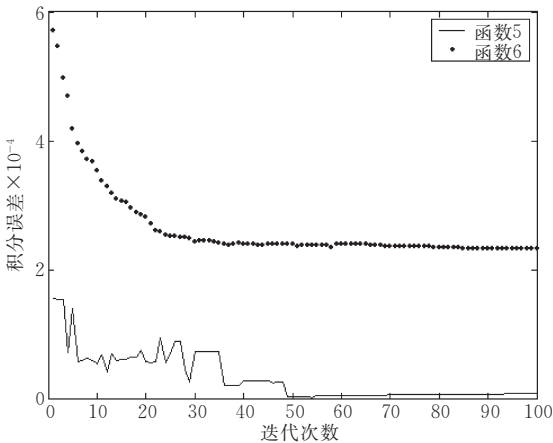


图 6 函数 5 和函数 6 的积分误差变化曲线
(例 1, 基于不等距点分割)

例 2. 计算积分 $\int_0^{48} \sqrt{1+(\cos x)^2} dx$.

用 Romberg 方法计算该积分时遇到了困难^[18], Composite Simpson' rule 计算时, 将积分区

间 $[0, 48]$ 等分成 100 个均匀子区间, 计算结果为 58.47082^[19], 文献[20]中采用神经网络算法得到的结果为 58.5205, 精确值是 58.4704691. 考虑到被积函数是以 π 为周期的函数, 且 $48=15\pi+0.8761$, 因此, 计算方法如下:

$$\int_0^{48} \sqrt{1+(\cos x)^2} dx = 15 \int_0^\pi \sqrt{1+(\cos x)^2} dx + \int_0^{0.8761} \sqrt{1+(\cos x)^2} dx.$$

若用本文基于混合基函数的 ES 方法, 取 $\mu=15$, $\lambda=7 \times \mu$, $k=8$, $n=60$, $N=20$, $T=500$, 计算结果为 $\int_0^{48} \sqrt{1+(\cos x)^2} dx \approx 58.48828$. 对应的积分误差变化曲线如图 7 所示.

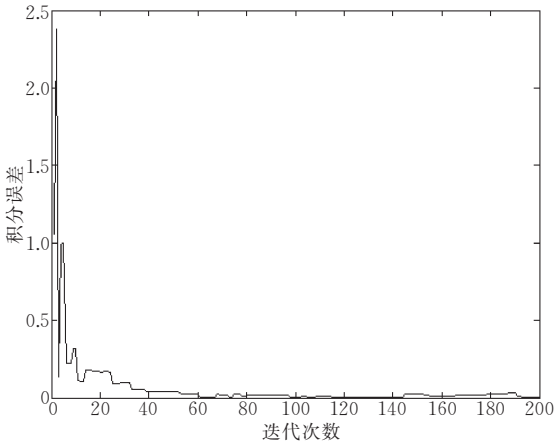


图 7 函数积分的误差变化曲线
(例 2, 基于混合基函数)

若用本文基于不等距点分割的进化策略数值积分方法, 取 $\mu=15$, $\lambda=7 \times \mu$, $n=100$, 计算结果为 $\int_0^{48} \sqrt{1+(\cos x)^2} dx \approx 58.47065$. 对应的积分误差变化曲线如图 8 所示.

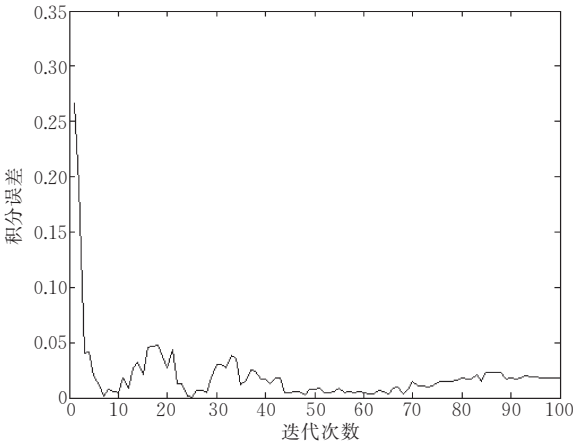


图 8 函数积分的误差变化曲线
(例 2, 基于不等距点分割)

例 3. 计算如下奇异积分：

$$f(x) = \begin{cases} e^{-x}, & 0 \leq x < 1 \\ e^{-x/2}, & 1 \leq x < 2. \\ e^{-x/3}, & 2 \leq x \leq 3 \end{cases}$$

该函数的精确积分值是 1.546036, 文献[20]中采用神经网络计算得到的结果为 1.5467.

若用本文基于混合基函数的 ES 的方法计算, 取 $\mu=5, \lambda=7 \times \mu, k=8, n=30, N=28, T=250$, 得到的结果为 1.5455. 对应的积分误差变化曲线如图 9 所示.

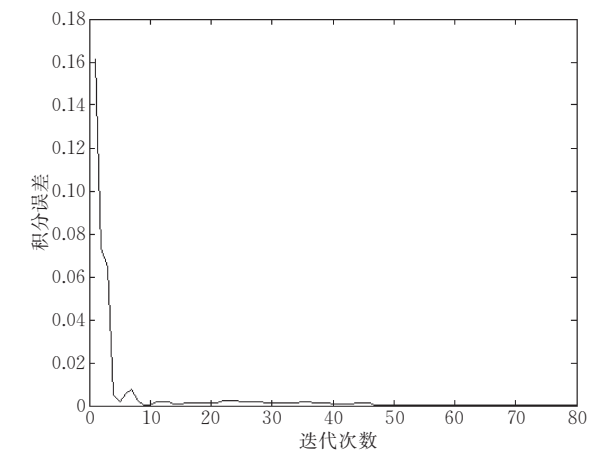


图 9 函数积分的误差变化曲线(例 3, 基于混合基函数)

若用本文基于不等距点分割的进化策略数值积分方法, 取 $\mu=25, \lambda=7 \times \mu, n=30$, 得到的结果为 1.5459805. 由此看出此算法具有较强的奇异积分处理能力. 对应的积分误差变化曲线如图 10 所示.

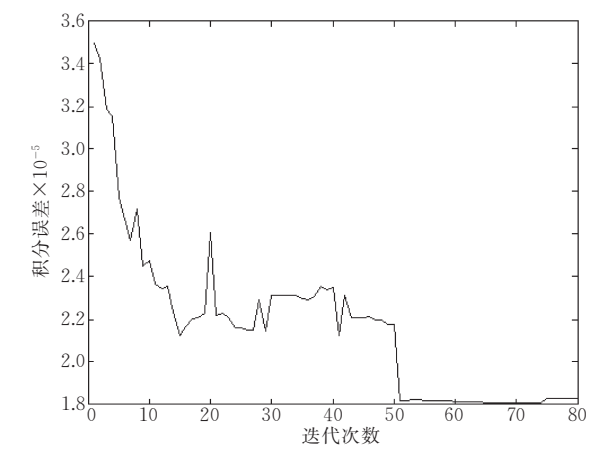


图 10 函数积分的误差变化曲线(例 3, 基于不等距点分割)

例 4. 计算振荡积分^[21] $I = \int_0^{2\pi} x \cos x \sin mx \, dx$.

用 G_{32} 代表 32 个点的高斯公式的计算值, 用 $2n \times L_5$ 表示以 $\sin mx$ 的零点划分的子区间(共 $2n$ 个)的五点高斯-罗巴托-勒让德公式计算值.

若用本文基于混合基函数进化策略的数值积分算法, 取 $\mu=5, \lambda=7 \times \mu, k=8, m=10, 20, 30; N=20$; 对应的积分误差变化曲线如图 11 所示.

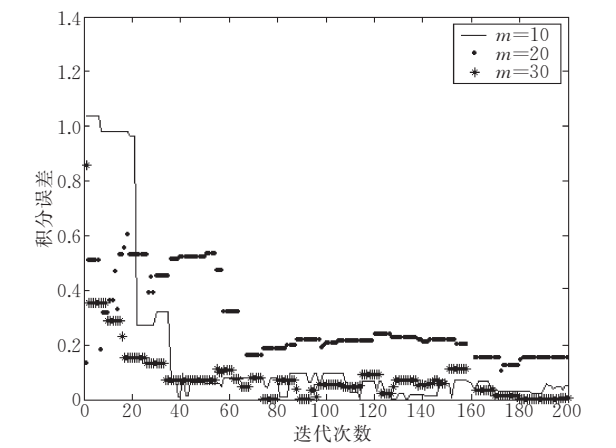


图 11 函数积分的误差变化曲线
(例 4, 基于混合基函数)

若用本文基于不等距分割的进化策略数值积分算法, 取 $\mu=25, \lambda=7 \times \mu, n=120$, 参数 $m=10, 20, 30$; 对应的积分误差变化曲线如图 12 所示.

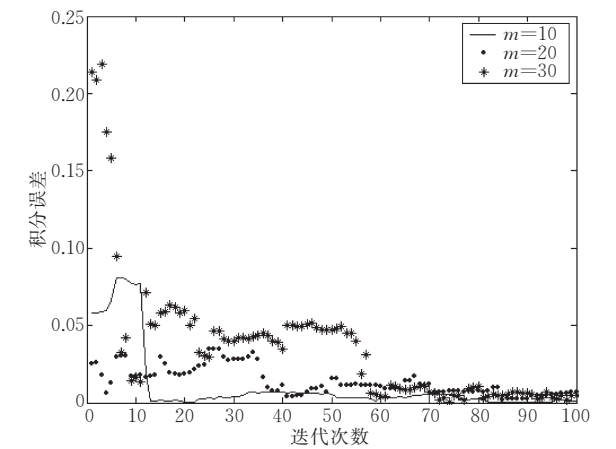


图 12 函数积分的误差变化曲线
(例 4, 基于不等距点分割)

两种算法的计算结果的对比如表 2 所示.

表 2 几种数值积分方法对应各函数的计算结果					
m	精确值	G_{32}	$2n \times L_5$	基于混合基函数的 ES 方法	基于不等距点分割的 ES 方法
10	-0.63466518	-0.6340207	-0.55875940	-0.68134052	-0.65034080
20	-0.31494663	-1.2092524	-0.27789620	-0.37280425	-0.30583435
30	-0.20967243	-1.5822272	-0.18508448	-0.17305621	-0.23556815

例 5. 计算积分 $\int_0^1 e^{-x^2} dx$.

此积分的被积函数的原函数不是初等函数,所以无法用牛顿-莱布尼茨公式来计算这个积分. 文献[20]中分别用矩形法、梯形法和 Simpson 方法计算该积分. 取 $\mu=20, \lambda=7 \times \mu, n=30, N=40, T=80$, 基于混合基函数的进化策略积分误差变化曲线如

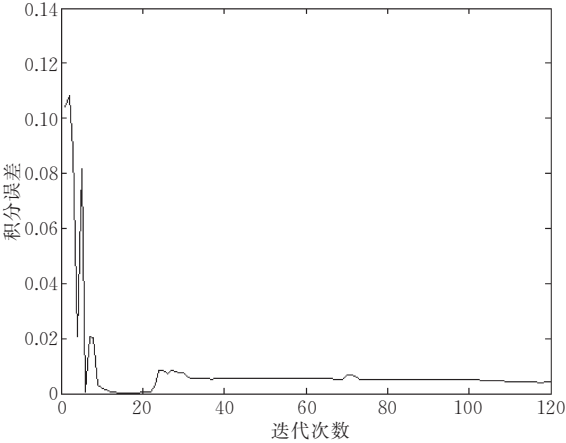


图 13 函数积分的误差变化曲线(例 5,基于混合基函数)

图 13 所示. 基于本文不等距分割的进化策略数值积分算法,取 $\mu=20, \lambda=7 \times \mu, n=30$, 则积分误差变化曲线如图 14 所示.

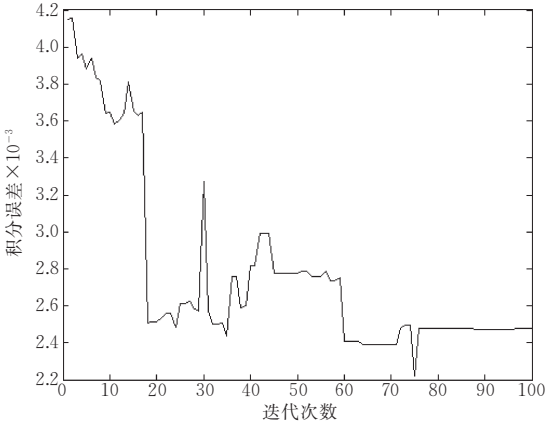


图 14 函数积分的误差变化曲线
(例 5,基于不等距点分割)

表 3 列出了文献[20]中的结果与本文算法的计算结果.

表 3 几种数值积分方法对应函数的积分值

$f(x)$	矩形法	梯形法	Simpson 方法	基于混合基函数的 ES 方法	基于不等距点分割的 ES 方法	精确值
e^{-x^2}	0.77782	0.74621	0.74683	0.74652	0.74683	0.746824

通过 5 个不同类型的数值积分仿真例子可看出,当参数 $\mu/\lambda=1/7$, 阶次 $N \geq 20$, 节点数 $n \geq 20$, $k=8$ 时,用本文提出的两种进化策略求数值积分算法,相比传统的一些方法(如矩形法、梯形法、Simpson 方法等)精度要高.

6 结 论

文中给出了两种基于进化策略方法求任意被积函数积分的方法,两种算法都采用适用于高维优化的单基因突变进化策略算法,充分发挥了进化策略算法的群体搜索和并行的特性. 通过 5 个数值积分算例表明,该算法十分有效,不但可以计算原函数不易求得的被积函数的积分,而且还可以计算奇异积分和振荡积分,可看作是对传统的数值积分方法的一种推广. 设计适用于更高维优化的进化策略算法和应用此算法计算多重积分问题是今后我们所要做的工作.

参 考 文 献

[1] Xue Mi. Numerical Mathematical and Computation. Shang-

hai; Fudan University Press, 1991(in Chinese)
(薛密. 数值数学和计算. 上海: 复旦大学出版社, 1991)
[2] Liu Yu-Lian, Fu Pei-Ren. Mathematical Analysis. Beijing: Higher Education Press, 1992(in Chinese)
(刘玉琏, 傅沛仁. 数学分析讲义. 北京: 高等教育出版社, 1992)
[3] Schwefel H P, Back T. Evolution strategies I: Variants and their computational implementation//Winter G ed. Proceedings of the Genetic Algorithms in Engineering and Computer Science. Wiley, 1995: 111-126
[4] Schwefel H P, Back T. Evolution strategies II: Theoretical aspects//Winter G. Proceedings of the Genetic Algorithms in Engineering and Computer Science. Wiley, 1995: 127-140
[5] Wang Xiao-Ping, Cao Li-Ming. Genetic Algorithms—Theory, Applies and Software Realization. Xi'an: Xi'an Jiaotong University Press, 2002(in Chinese)
(王小平, 曹立明. 遗传算法—理论、应用与软件实现. 西安: 西安交通大学出版社, 2002)
[6] Yun Qing-Xia. Evolutionary Algorithms. Beijing: Beijing Matallurgy Press, 1997(in Chinese)
(云庆夏. 进化算法. 北京: 北京冶金工业出版社, 1997)
[7] Yao Xin, Liu Yong. Fast evolution strategies//Angeline P J ed. Proceedings of the 6th Annual Conference on Evolutionary Programming. Berlin: Springer-Verlag, 1997: 151-161
[8] Kappler C. A evolutionary algorithms improved by large mutation//Voigt H M, Ebeling W, Rochenberg I et al. Pro-

- ceedings of the Parallel Problem Solving from Nature (PPSN) IV. Lecture Notes in Computer Science 1141. Berlin, Germany: Springer-Verlag, 1996: 346-355
- [9] Lin Dan, Li Min-Qiang, Kou Ji-Song. On research of mutation operator in evolutionary programming and evolutionary strategies. *Journal of Tianjing University (Science and Technology)*, 2000, 33(5): 627-630(in Chinese)
(林丹, 李敏强, 寇纪淦. 进化规划和进化策略中突变算子的若干研究. *天津大学学报(自然科学版)*, 2000, 33(5): 627-630)
- [10] Gunter R. Local convergence rates of simple evolutionary algorithms with Cauchy mutations. *IEEE Transactions on Evolutionary Computation*, 1997, 4(1): 249-258
- [11] Chang Ming, Kazuhiro O, Kanji U. Some experimental observations of (μ, λ) evolution strategies//*Proceedings of the 2001 Congress on Evolutionary Computation*. Seoul: Coex Center, 2001, 1: 663-670
- [12] Wang Xiang-Zhong, Yu Shou-Yi. Improved evolution strategies for high-dimensional optimization. *Control Theory and Applications*, 2006, 23(1): 147-151(in Chinese)
(王湘中, 喻寿益. 适用于高维优化问题的改进进化策略. *控制理论与应用*, 2006, 23(1): 147-151)
- [13] Yang Shiow-Shung, Tseng Ching-Shiow. An orthogonal neural network for function approximation. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 1996, 26(5): 779-785
- [14] Yu Dong-Jun, Wang Shi-Tong. Constructive theory for B-spline neural network. *Journal of Computer Research & Development*, 1999, 36(5): 534-540(in Chinese)
(於东军, 王士同. B样条神经网络的构造理论. *计算机研究与发展*, 1999, 36(5): 534-540)
- [15] Guarnieri S et al. Multilayer feedforword networks with adaptive spline activation functions. *IEEE Transactions on Neural Network*, 1999, 10: 672-683
- [16] Peng Hong, Yang Li-Hong, Zheng Xian-Yi et al. A new evolutionary strategy for solving engineering optimization problem. *Journal of South China University of Technology*, 1997, 25(12): 17-21(in Chinese)
(彭宏, 杨立洪, 郑咸义等. 计算工程优化问题的进化策略. *华南理工大学学报*, 1997, 25(12): 17-21)
- [17] Burden R L, Faires J D. *Numerical Analysis*. 7th Edition. Brooks/Cole, Thomson Learning, Inc, 2001: 190
- [18] Burden R L, Faires J D. *Numerical Analysis*. 7th Edition. Brooks/Cole, Thomson Learning, Inc, 2001: 212
- [19] Burden R L, Faires J D. *Numerical Analysis*. 7th Edition. Brooks/Cole, Thomson Learning, Inc, 2001: 206, 772
- [20] Wang Xiao-Hua, He Yi-Gang, Zeng Zhe-Zhao. Numerical integration study based on triangle basis neural network algorithm. *Journal of Electronics and Information Technology*, 2004, 26(3): 394-399(in Chinese)
(王小华, 何怡刚, 曾昭昭. 三角基函数神经网络算法在数值积分中的应用研究. *电子与信息学报*, 2004, 26(3): 394-399)
- [21] Tongji University Mathematical Staff room. *Modern Numerical Mathematical and Computation*. Shanghai: Tongji University Press, 2004(in Chinese)
(同济大学数学教研室. *现代数值数学和计算*. 上海: 同济大学出版社, 2004)



ZHOU Yong-Quan, born in 1962, Ph. D., professor. His current research interests include neural network, computation intelligent.

ZHANG Ming, born in 1978, master, assistant. His current research interests include evolution computation, neural network.

ZHAO Bin, born in 1964, Ph. D., associate professor. His current research interests include computation theory and applies.

Background

This work is supported by the National Natural Science Foundation of China under grant No. 60461001 and the Natural Science Foundation of Guangxi under grant No. 0542048.

In engineering technology, the numerical integration is a very basic computation problem. So far, the method which people usually adopt is to solve the primitive function of the integrated function at first, then compute the definite integral by the Newton-Leibniz formula. In fact, it is not easy or direct to compute the solutions of many definite integral by

using the above method, as some integrated functions' explicit primitive functions can not be shown or the primitive functions which are so complex are not suitable for computation. Therefore, it is necessary to propose a novel computation method to solve numerical integration, in order to solve the defect of the approximate definite integral computing method. According to these problems, an approximate numerical integral evolution strategy computing method was proposed based on hybrid basis functions, whose basic idea is

to use hybrid basis functions evolution strategies algorithm to approximate integrated function in order to accomplish the definite integral's approximate computing.

On the other hand, nowadays people usually use the numerical integral formula based on equally spaced nodes Summed-division or the regulated equally spaced nodes which are invariable in the whole computing process. All the integral formulas having equally spaced nodes can not be divided according to the form of the integrated functions, we need more nodes to obtain higher precision, the best division is not the equally spaced previously chosen, but to choose division points randomly in terms of the form of the integral function which means the concave and protruding form of the integral

function curve. Thus, divide first, sum next, the precision of the function's integral value is higher than the traditional method. In the light of the idea, another Evolution Strategies numerical integral method based on unequally spaced nodes division is proposed, not only this Algorithm can calculate the usual significant functions' definite integral but also calculate the singular integral and oscillating integral.

The two algorithms in this paper proposed can not only calculate the usual significant functions' definite integral but also calculate the singular integral and oscillating integral can be regarded as the improvement of the traditional numerical integral method, they have widespread application in the engineering technology and science computation.