

一种求解约束多目标优化问题的线性进化算法

汤可宗^{1,3} 杨静宇¹ 高尚^{2,3} 郑宇杰¹

(南京理工大学计算机科学与技术学院 南京 210094)¹ (江苏科技大学电子信息学院 镇江 212003)²
(苏州大学江苏省计算机信息处理技术重点实验室 苏州 215006)³

摘要 针对多目标优化问题,提出了一种新的基于实数编码的线性进化算法。新算法将约束优化问题的高维搜索空间通过线性变换映射到二维空间,在二维空间中探索原优化问题的解,并构造出一种线性适应度函数,重新设计了一种基于密度函数的交叉算子。对二组典型优化问题的测试表明,本算法是可行和有效的,解集分布的均匀性与多样性均较理想。

关键词 多目标优化,进化算法,Pareto 最优解,线性函数

中图分类号 TP301.6 **文献标识码** A

Linear Evolutionary Algorithm for Constrained Multi-objective Optimization Problems

TANG Ke-zong^{1,3} YANG Jing-yu¹ GAO Shang^{2,3} ZHENG Yu-jie¹

(School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China)¹

(School of Electric and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China)²

(Jiangsu Provincial Key Laboratory of Computer Information Processing Technology, Soochow University, Suzhou 215006, China)³

Abstract A new Multi-objective Linear Evolutionary Algorithm (MOLEA) based on real-coded for constrained multi-objective optimization was proposed. Search space of constrained dominance problems with high dimensions was compressed into two dimension in the LEA, which contains two main points. Firstly, gave a linear fitness function in two dimension space. Secondly, gave a crossover operator based on density function. In our tests, a few benchmark multi-objective optimization problem which was divided into two groups were taken to test this algorithm. The numerical experiments show that proposed approach is feasible and effective, and provide good performance in terms of uniformity and diversity of solutions.

Keywords Multi-objective optimization, Evolutionary algorithm, Pareto optimal solutions, Linear function

1 引言

最优化计算在实际应用中有着广泛的应用。许多多目标优化问题 (Multi-Objective Optimization Problem, MOP) 受各种不同约束条件的制约,且各目标之间存在无法比较或相冲突等特点,以致于难以用一般的优化方法去求解。常用的约束最优化方法可分为间接法和直接法两大类^[1]。(1)间接法,先将约束优化问题转化为一系列的无约束优化设计问题,再调用无约束优化方法来求解,常用的方法有罚函数法和最小二乘法等。(2)直接法,在选取下降方向和下降点时直接判断是否在可行区域内,常用的方法有:约束随机方向法、复合型法等。目前,针对不同的多目标优化方法各有其不同的转化策略,常用的多目标优化转化方法有目标转划法、最小二乘法、线性加权组合法和功效系数法等。从国内外的研究趋势看,基于 Pareto 排序方法现已被许多研究者所采纳,并已提

出多种基于 Pareto 的适应度赋值方案^[2],如(1)目标置换法;(2)变参数聚合法;(3)基于 Pareto 排序法。

近几年,基于种群随机搜索优化的方法,如进化算法 (EA, Evolutionary Algorithms)、进化策略 (ES, Evolutionary Strategies)、免疫算法 (IA, Immune Algorithms) 等已经成功地应用于求解多目标优化问题^[3]。虽然这些受生物激励机制而产生的启发式算法在求解多目标优化问题时已表现出极大的优越性,但由于 MOP 包括多个待优化目标,在具体应用这些算法时,会受到其自身的某些限制,文献^[2]中就已指出其用于多目标优化时须考虑的两个关键问题:(1)为了保证朝 Pareto 最优解集的方向搜索,如何实施适应度赋值和选择;(2)为了避免未成熟和获得均匀分布且范围最广的非劣解,如何保持群体的多样性。此外,多目标进化算法 (Multi-Objective Evolutionary Algorithm, MOEA) 也会受自身一些其它的限制,如较早收敛、探索新解能力随迭代增加而减缓等,要将

到稿日期:2008-05-29 本文受国家自然科学基金资助项目(60472060),江苏省计算机信息处理技术重点实验室开放课题基金资助项目(KJS0601),江苏省“青蓝工程”资助。

汤可宗(1978-),男,博士研究生,主要研究方向为智能信息处理、图像处理与模式识别, E-mail: tangkezong@126.com; 杨静宇(1941-),男,教授,博导,主要研究方向为模式识别理论与应用、计算机视觉、智能机器人; 高尚(1972-),男,副教授,主要研究方向为智能信息处理; 郑宇杰(1977-),男,博士,主要研究方向为模式识别、人工智能、人脸识别。

MOEA 更好地应用于 MOP 优化问题需对其做进一步的改进。

本文将各种约束条件与目标函数融合在一起,对进化算法中的个体适应度函数与交叉算子进行了重新设计,提出了一种新的快速而鲁棒的多目标线性进化算法(Multi-Objective Linear Evolutionary Algorithm, MOLEA)。对两组不同测试问题的数值实验结果证实了我们提出的算法是可行的、有效的。对含有不等式约束和等式约束的问题进行测试表明,新方法在解集分布的均匀性与多样性上均能达到较理想的结果。

2 MOP 的定义

不失一般性,我们考虑下列多目标优化问题:

$$\begin{aligned} \text{Min}_{x \in X} F(x), x \in R^d & \quad (1) \\ \text{st. } g_i(x) \leq 0, i=1, 2, \dots, k \end{aligned}$$

其中, $x = \{x_1, x_2, \dots, x_d\} \in s \subseteq \Omega$ 是一个 d 维决策变量, Ω 为目标函数 $F(x)$ 的 d 维搜索空间, s 为可行区域。 $l_i \leq x_i \leq u_i$, $F = \{f_1, f_2, \dots, f_m\}$ 是 m 个待优化的目标函数集合。

多目标优化问题的 Pareto 最优解时常采用一种“折衷”的办法取得^[4],即非劣解的每个目标函数值仅仅只有在降低不少于一个目标函数值的情况下才能得到改进。Pareto 偏序关系用于评价 MOP 中两个候选解之间的强度,自 Pareto 提出这种偏序关系以来,其偏序关系已经被广泛地应用于 MOP 的求解。其定义如下^[5]:

定义 1(解的优劣性) $f(x)$ 是向量目标函数,决策变量 x_a Pareto 优于 x_b ($x_a < x_b$) 或 x_b Pareto 劣于 x_a 满足以下关系:

$$x_a < x_b, \text{ if } f_{a,i} \leq f_{b,i}, \forall i = \{1, 2, \dots, d\}, \exists j = \{1, 2, \dots, d\}, f_{a,j} < f_{b,j}.$$

定义 2(Pareto 最优解) 决策变量 x_a 称为 MOP 的非劣解或 Pareto 最优解,当且仅当不存在向量 x_b 优于 x_a 。

3 线性适应度函数(LF)的构造

进化算法处理约束条件常见的方法为罚函数法,普通的适应度函数 $fitness(x) = f(x) + r \sum_{j=1}^m f_j(x)$ 可以看作是目标函数 $f(x)$ 和违反约束条件的程度函数 $\sum_{j=1}^m f_j(x)$ 的加权组合。文献[6]指出,实现罚函数法的困难在于其参数 r 难以选择和控制在现实世界中,求解 MOP,其决策变量的搜索空间往往处于高维空间中。就人的空间想象而言,空间维数小于 3 易于人们理解,我们能否将高维的搜索空间通过适当的方式转化为低维的空间,在低维空间求解 MOP。在此,我们给出如下的转换关系:

令 $y_1 = f(x), y_2 = \sum_{j=1}^m f_j(x)$, 如果适当选择 $y \rightarrow \bar{x}$ 的映射。则可以把上述适应度函数转化为 \bar{x} 的线性函数:

$$fitness(\bar{x}) = a^T \bar{x} = \sum_{i=1}^2 a_i \bar{x}_i \quad (2)$$

其中, $a = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 \\ r \end{bmatrix}, \bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = y$, 不失一般性,变量 \bar{x} 的表示法仍沿用 x 表示,我们将 a 称为权向量, x 为二维空间中的一点。 $a^T y = 0$ 在 x 空间确定了一个通过原点的超平面 H , 它将二维空间划分成两个子空间 Φ_1, Φ_2 , a 是 H 的

法向量,如图 1 所示。这种划分与原问题在高维搜索空间中对原决策变量 x 的划分是完全相同的,与 d 维搜索空间 Ω 相对应,此时,算法的搜索集中在 Φ_1 空间内,可行区域 s 与不规则形状区域 Φ_3 呈映射关系,并且点 x 满足以下关系:

$$\begin{aligned} fitness(x) > 0, \text{ if } x \in \Phi_1 \\ fitness(x) < 0, \text{ if } x \in \Phi_2 \\ fitness(x) = 0, \text{ if } x \in H \end{aligned} \quad (3)$$

进一步分析发现,线性函数 $fitness$ 可以看成是子空间 Φ_3 中某点 x 到超平面距离的一种代数度量:

$$r = \frac{a^T x}{\|a\|} \quad (4)$$

这样,我们就可以利用线性函数的简单性来解决复杂的 MOP。

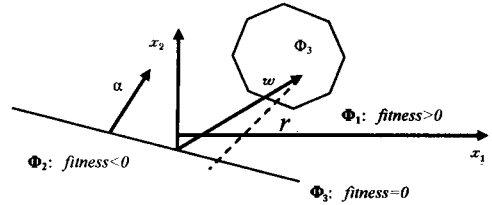


图 1 搜索空间的转化

通过以上分析,我们构造出一个新的用于评价个体的线性适应度函数:

$$fitness(x) = \frac{a^T x}{\|a\|} \quad (5)$$

在评价个体适应度时,可以直接使用式(5)去评价,而不需要关心个体是否可行。根据个体的适应度决定个体的优劣,计算时非常方便。

4 密度交叉算子与变异算子的设计

4.1 密度交叉算子(Density crossover operator, Dco)

基于密度的交叉算子采用两个父体重组产生后代,具体做法为:从种群中随机选出两个父体 $X = (x_1, x_2, \dots, x_3)$, $Y = (y_1, y_2, \dots, y_d)$, 并假定 $x_i < y_i$, 记:

$$\left[x_i - \frac{y_i - x_i}{2}, y_i + \frac{y_i - x_i}{2} \right] = [\bar{x}_i, \bar{y}_i] \quad (6)$$

我们构造如式(7)的密度函数。该函数取决于两个参数: α, β , 且 $\alpha, \beta \in [0, 1]$ 。由函数(7)可以生成一个累积分布函数(8), 且 $0 \leq G_{x_i, y_i} \leq 1$, 累积分布函数用于随机确定后代个体中的基因 η_i 。这一过程可通过函数(9)完成, G_{x_i, y_i}^{-1} 是 G_{x_i, y_i} 的反函数。

$$g_{x_i, y_i}(x, \alpha, \beta) : [\bar{x}_i, \bar{y}_i] \times \alpha \times \beta \rightarrow R \quad (7)$$

$$G_{x_i, y_i}(x, \alpha, \beta) : [\bar{x}_i, \bar{y}_i] \times \alpha \times \beta \rightarrow [0, 1] \quad (8)$$

$$\eta_i = G_{x_i, y_i}^{-1}(r, \alpha, \beta) \quad (9)$$

在我们的实验中,密度交叉算子能够较好地扩大解的探索区域,具有发现较好解的能力。采用密度交叉算子生成后代的具体步骤如下:

Step1 从种群中随机取出两个父体 $x = (x_1, x_2, \dots, x_d), y = (y_1, y_2, \dots, y_d)$ 。

Step2 对父代中每对基因 (x_i, y_i) 按照某种方式给定参数 α, β 。

Step3 生成一个介于 0 与 1 之间的随机数 r , 由函数(9)

确定后代基因 η_i 。

Step4 重复上述 Step1-Step3,由此确定后代的全部基因 $\eta_i, i=1,2,\dots,d$ 。个体 $\eta=\{\eta_1, \eta_2, \dots, \eta_d\}$ 生成结束。

Step5 记录本次迭代内产生的最佳个体,并记录其参数 α 与 β ,待下次迭代时将其作为参照值,按预先设定的方式生成对应的参数 α 与 β 。

4.2 邻域变异算子(Neighborhood mutate operator, Nmo)

变异算子采用文献[8]设计的邻域变异法。其具体做法是在个体的邻域空间内进行变异, $x'=x+\alpha \times r$ (x' 是新的个体, x 是原来的个体, $\alpha \in [-1, 1]$, r 是其邻域空间)。我们对邻域空间进行动态压缩,选择函数 $deratio=(1+0.1^a \times b)^{-t}$ ($deratio$ 为半径压缩率, a, b 为 0~9 之间的整数, t 为进化代数)。 $r=deratio \times r_0$ (r_0 为初始空间大小, r 为压缩后的邻域外空间大小),实验证明这种方法对空间内的搜索很有效。

5 数值实验

5.1 MOLEA 算法

本文提出的 LEA 算法的具体步骤如下:

Multi-Objective Linear Evolutionary Algorithm(MOLEA)

- 1) $t \leftarrow 0$;
 - 2) Initialize Population Pop_t ;
 - 3) Evaluate initial Population Pop_t ;
 - 4) While ($t < \text{MaxGenerations}$) do
 - 5) $t = t + 1$;
 - 6) Select two parents form Pop_t ;
 - 7) Obtain a new individual x_{new} by means of Dco;
 - 8) Mutate x_{new} by means of Nmo, generate x'_{new} ;
 - 9) Evaluate x'_{new} in terms of LF
 - 10) generate new Pop_{t+1} ;
 - 11) End
- End

5.2 仿真结果分析

数值实验环境: Intel Pentium 4, 2.26GHz, 512M 内存, Windows XP Professional, Matlab 7.0。

测试分两组 G1 和 G2 进行,选取的二组典型多目标优化函数分别来自文献[9,10]。第一组为仅含有不等式约束的测试问题,各目标函数最多只含有两个决策变量。第二组为既有不等式约束又有等式约束的单目标优化问题,目标函数中决策变量的数目不限。由于要考虑是否为可行解,所以计算中时间复杂度较高,利用我们定义的线性适应度函数,可以直接评价个体,而不需要关心个体是否为可行,计算时较为方便。初始参数的设置:种群个体 $Pop_t = 100$;运行的代数 200 代,对每个问题均在相同条件独立运行 20 次。

G1:第一组测试问题^[9]

(1)Test Problem1;BNH

$$\min F(x) = \{f_1(x), f_2(x)\};$$

$$f_1(x) = 4x_1^2 + 4x_2^2;$$

$$f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2.$$

s. t.

$$C_1(x) = (x_1 - 5)^2 + (x_2^2) \leq 25;$$

$$C_2(x) = (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7;$$

$$0 \leq x_1 \leq 5;$$

$$0 \leq x_2 \leq 3.$$

(2) Test Problem2;TNK

$$\min F(x) = \{f_1(x), f_2(x)\};$$

$$f_1(x) = x_1;$$

$$f_2(x) = x_2.$$

s. t.

$$C_1(x) = x_1^2 + x_2^2 - 0.1 \cos(16 \arctan \frac{x_1}{x_2}) - 1 \geq 0;$$

$$C_2(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5;$$

$$0 \leq x_1 < \pi;$$

$$0 \leq x_2 \leq \pi.$$

对于问题 1,约束条件为两个不等式约束,Pareto 最优解较好地分布于 Pareto 前沿,如图 2 所示。Pareto 前沿分别由解 $x_1^* = x_2^* \in [0, 3]$ 构成的区域与 $x_1^* \in [3, 5], x_2^* = 3$ 区域构成。对于问题 2,约束条件同样为两个不等式约束,Pareto 最优解全部均匀分布于 Pareto 前沿上。用户可以根据实际情况,从测试函数的决策变量的收敛值中选取最终的 Pareto 最优解如图 3 所示。

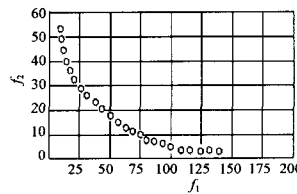


图 2 BNH 测试后得到的 Pareto 最优解

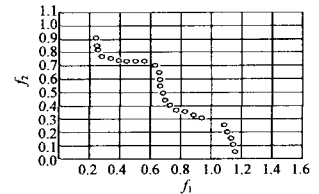


图 3 TNK 测试后得到的 Pareto 最优解

G2:第二组测试问题^[10]

(1)Test Problem1;

$$\min g(x) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$$

s. t.

$$x_4 - x_3 + 0.55 \geq 0, -x_4 + x_3 + 0.55 \geq 0;$$

$$1000 \sin(-x_3 - 0.25) + 1000 \sin(-x_4 - 0.25) + 894.8 - x_1 = 0;$$

$$1000 \sin(x_3 - 0.25) + 1000 \sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0;$$

$$1000 \sin(x_4 - 0.25) + 1000 \sin(x_4 - x_3 - 0.25) + 1294.8 = 0;$$

$$0 \leq x_i \leq 1200 (i=1, 2); -0.55 \leq x_i \leq 0.55 (i=3, 4).$$

(2)Test Problem2;

$$\min g(x) = e^{x_1 x_2 x_3 x_4 x_5}$$

s. t.

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0;$$

$$x_2 x_3 - 5x_4 x_5 = 0, x_1^3 + x_2^3 + 1 = 0;$$

$$-2.3 \leq x_i \leq 2.3 (i=1, 2); -3.2 \leq x_i \leq 3.2 (i=3, 4, 5).$$

与第一组测试问题不同,在第二组的两个测试问题中,两个问题均含有等式或不等式约束,且只对单目标优化问题进行测试。我们已提到真实世界中的许多 MOP 均可以转化为单目标优化问题求解。因此,对单目标优化问题的研究也是实际工程应用领域中非常值得重视的。

在第二组测试实验中,我们将新方法独立运行 30 次。对第一个测试问题,我们发现 22 次可以找到最优解 $x = (668.94675328910, 1013.10376657821, 0.10773654866, -0.39654576851)$,最优解距离为 $d(x) = 2.33232367833 \times 10^{-13}$,对应的最优值 $g(x) = 5198.5367$ 。第二个问题测试中,我们有 19 次找到最优解 $x = (-1.77365745561,$

1.45675898761, -1.5678456772, 0.66765756893, -0.75678765688), 与最优解对应的最优值 $g(x) = 0.055894567$, 对两个问题测试后所得到的解的均值及最差解如表 1 所示。

表 1 MOLEA 算法测试解

Value problems	Best	Mean	Worst
Test1	5198.5367	5199.3367	5199.9689
Test2	0.055894567	0.055999745	0.055999987

结束语 本文提出了处理约束优化问题的一种新方法——多目标线性进化算法(MOLEA),该方法将约束优化问题在高维搜索空间的求解转换到二维搜索空间中。利用给出的线性适应度函数,方便了对个体适应度值的评价。设计的基于密度的交叉算子可以较好地扩大解的搜索空间。数值实验显示出新算法是一种便于实现、通用性较强的方法。下一步我们的工作将集中在二维搜索空间中利用分类的方法改进 MOLEA,以提高算法的性能,并试图推广到其它的组合优化问题。

参考文献

[1] Deb K. Multi-objective genetic algorithms: problem difficulties and construction of test problem[J]. Evolutionary Computa-

tion,1999,7(3):205-230

[2] 崔逊学,林闯,方廷健.多目标进化算法的研究与进展[J].模式识别与人工智能,2003,16(3):306-313

[3] Deb K, Agrawal S, Pratap A, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197

[4] Beyer H-G, Deb K. On self-adaptive features in real-parameter evolutionary algorithms[J]. IEEE Transactions on Evolutionary Computation, 2001, 5(3): 250-270

[5] 边肇祺,张学工.模式识别.第二版[M].北京:清华大学出版社,2001:83-117

[6] Zitzler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithm: Empirical results[J]. Evolutionary Computation Journal, 2000, 8(2): 125-148

[7] Michalewicz Z. Genetic algorithms, Numerical optimization and constraints// Eshelman L J. ed. Proceedings of the 6th International Conference on Genetic Algorithms. San Mateo: Morgan Kaufmann Publishers, 1995: 151-158

[8] 邹秀芬,刘敏忠,吴志健,等.解约束多目标优化问题的一种鲁棒的进化算法[J].计算机研究与发展,2004,41(6):986-990

[9] Deb K. Multi-objective Optimization Using Evolutionary Algorithms. Chichester, UK: John Wiley & Sons, Ltd, 2001

[10] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization [J]. IEEE Transactions on Evolutionary computation, 2000, 4(3): 284-294

(上接第 226 页)

文计算方法如下:

$$\int_{-1}^1 \frac{1+x}{\sqrt[3]{3x+1}} dx = \int_{-1}^0 \frac{1+x}{\sqrt[3]{3x+1}} dx + \int_0^1 \frac{1+x}{\sqrt[3]{3x+1}} dx = \int_0^1 \frac{x}{\sqrt[3]{3x-2}} dx + \int_0^1 \frac{1+x}{\sqrt[3]{3x+1}} dx$$

本文算法中,使用图 2 所示的泛函网络结构,对所有的泛函神经元选取三角函数为基函数,参数的初始值随机选取,输入个数 $N=10$,泛函网络训练样本集为 $\{x_k = k\pi/100 | f(x_k), k=0,1,\dots,100\}$,计算结果为 $I=1.11190+0.71748i=1.82938$ 。表 5 列出了文献[12]中的结果和本文算法的计算结果。积分误差变化曲线如图 6 所示。

表 5 几种数值积分方法对应函数的积分值

f(x)	Gauss-Legendre 公式	Gauss-Per Kai 公式	本文算法	精确值
$(1+x)/\sqrt[3]{3x+1}$	1.436	1.7533	1.82938	1.8294

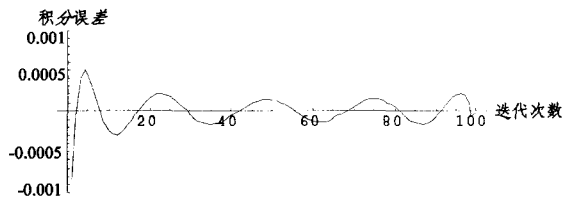


图 6 积分误差变化曲线

结束语 泛函网络是一个崭新的研究方向,其在某些方面的性能优于神经网络。本文给出了基于泛函网络求解数值积分的模型、算法,该算法适合于求任意函数的数值积分,因此,本文提出的数值积分计算方法适应性强。通过仿真实例可以看出,与传统计算方法相比,本文的计算精度更高,而且本文给出的积分算法不限制积分区间,是一种有效的数值积

分方法,在工程应用领域中具有较大的应用前景。

参考文献

[1] 林成森.数值计算方法[M].北京:科学出版社,1998:173-215

[2] 朱方生,刘大美,李素贞.计算方法[M].武汉:武汉大学出版社,2003

[3] Castillo E. Functional networks. Neural Processing Letters, 1998, 7: 151-159

[4] Castello E, Cobo A, Gutierrez J M. Working with Differential Functional and Difference Equations Using Functional Networks [J]. Appl. Math. Model, 1999, 23: 89-107

[5] Castello E, Gutierrez J M. Nonlinear Time Series Modeling and Prediction Using Functional Networks, Extracting Information Masked by Chaos[J]. Phys. Lett. A, 1998, 244: 71-84

[6] Yang Shioh - Shung, Tseng Ching - Shioh. An orthogonal neural network for function approximation. IEEE Transaction on Systems, Man, and Cybernetics, Part B; Cybernetics, 1996, 26(5): 779-785

[7] Burden R L, Faires J D. Numerical Analysis (Seventh Edition) [M]. Brooks/Cole, Thomson Learning, Inc., 2001: 190

[8] Burden R L, Faires J D. Numerical Analysis (Seventh Edition) [M]. Brooks/Cole, Thomson Learning, Inc., 2001: 212

[9] Burden R L, Faires J D. Numerical Analysis (Seventh Edition) [M]. Brooks/Cole, Thomson Learning, Inc., 2001: 206, 772

[10] Wang X H, He Y G, Zeng Z Z. Numerical integration study based on triangle basis neural network algorithm[J]. Journal of Electronics & Information Technology, 2004, 26(3): 394-399

[11] 周永权,张明,赵斌.基于进化策略方法求任意函数的数值积分[J].计算机学报,2008,31(2):196-206

[12] 朱磊.关于 Gauss_PerKai 型数值积分方法[J].合肥工业大学学报:自然科学版,2007,30(5):655-656