

# 基于高斯混沌变异和精英学习的自适应多目标粒子群算法

韩敏, 何泳

(大连理工大学 电子信息与电气工程学部, 辽宁 大连 116023)

**摘要:** 为平衡多目标粒子群的全局和局部搜索能力, 提出一种基于高斯混沌变异和精英学习的自适应多目标粒子群算法. 首先, 提出一种新的种群收敛状态检测方法, 自适应调整惯性权重和学习因子的值, 以达到探索和开发的最佳平衡. 然后, 当检测到种群收敛停滞时, 采用一种带有高斯函数和混沌特性的变异算子协助种群跳出局部最优, 以增强全局搜索能力. 最后, 外部档案中的精英解相互学习, 增强算法的局部搜索能力. 在多目标标准测试问题上的仿真结果表明了所提出算法的有效性.

**关键词:** 变异; 精英学习; 自适应; 多目标优化; 粒子群

**中图分类号:** TP18

**文献标志码:** A

## Adaptive multi-objective particle swarm optimization with Gaussian chaotic mutation and elite learning

HAN Min, HE Yong

(Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116023, China.

Correspondent: HAN Min, E-mail: minhan@dlut.edu.cn)

**Abstract:** In order to keep the balance between the global and local searching abilities, an adaptive multi-objective particle swarm optimization algorithm based on Gaussian chaotic mutation and elite learning is proposed. Firstly, a method to detect the convergence state is proposed, then the inertia weight and acceleration coefficients are adjusted adaptively. Then, when the swarm is judged to be in the stagnation state, a mutation operator with the features of Gaussian function and chaotic sequence is proposed to help the swarm jump out from local optimum and enhance the global search. Moreover, the elite solutions in the external archive learn with each other to enhance the local search ability. The simulations on several standard test multi-objective problems show the effectiveness of the proposed method.

**Keywords:** mutation; elite learning; adaptive adjustment; multi-objective optimization; particle swarm optimization

## 0 引言

多目标优化问题是各类实际工程问题与决策的主要形式之一, 其特点是包含多个互相冲突的目标, 最优解为一组相互之间无法比较的折衷解, 称为 Pareto 最优解集. 由于基于群体搜索的进化计算方法可以并行处理所有目标函数, 适合求解多目标优化问题, 引起了学术界的广泛关注<sup>[1]</sup>. 其中粒子群算法 (PSO)<sup>[2]</sup> 由于具有操作简单、收敛速度快等优点, 已广泛应用于多目标优化问题的求解<sup>[3]</sup>.

在进化算法中, 如何平衡全局和局部搜索能力是一个关键问题, 即探索与开发的平衡问题. 粒子群在该方面存在一些不足: 在探索方面, 其收敛速度快的特性容易导致早熟收敛和陷入局部最优; 在开发

方面, 粒子群单一的搜索模式存在收敛精度不足的问题, 而且在多目标粒子群中, 全局最优解的频繁更换, 导致探索和开发的平衡问题更加突出. 对此, 一些学者通过调整粒子群参数的方式平衡探索和开发能力<sup>[4-5]</sup>. 为了达到探索和开发之间的最佳平衡, 要求能够动态监测种群的收敛状态, 并以此作为反馈信息进行自适应调整参数. 例如, Hu 等<sup>[6]</sup> 提出了基于平行格坐标系统的自适应多目标粒子群算法, 根据 Pareto 熵监测种群的进化状态, 自适应调整惯性权重和学习因子. 文献 [7] 引入单目标优化中的平均粒距概念检测种群收敛状态并自适应调整惯性权重和学习因子. 文献 [8] 提出了一种多目标优化环境下的种群收敛停滞判别准则. 由于在多目标问题中, Pareto 解集的特征使

**收稿日期:** 2015-05-21; **修回日期:** 2015-12-03.

**基金项目:** 国家重点基础研究发展规划项目(2013CB430403); 国家自然科学基金项目(61374154).

**作者简介:** 韩敏(1959—), 女, 教授, 博士生导师, 从事复杂工业系统建模与控制、智能技术及优化算法等研究; 何泳(1991—), 男, 硕士生, 从事复杂系统建模和智能优化算法的研究.

得直接检测种群收敛状态相比单目标问题更加复杂,与以上方法不同,本文提出以外部档案的收敛性提升程度间接反映种群的收敛状态,并以此作为自适应调整惯性权重和学习因子的依据,以达到平衡探索和开发的目的。

除了调节参数外,变异<sup>[9]</sup>和局部搜索策略也加入到粒子群中以提高探索和开发能力. 本文通过判断种群的收敛停滞状态自适应地进行变异和局部搜索,以最大化提升该两种策略的效率. 在变异方面,采用一种高斯混沌变异策略,利用高斯函数对离全局向导较近的粒子进行较大幅度的变异,较远的粒子进行较小幅度的变异,并引入混沌序列使之具有全局遍历性;在局部搜索方面,提出一种多目标优化中的精英学习策略,充分利用外部档案中非支配解的信息进行局部搜索,以提高收敛精度. 仿真实验部分采用多目标测试问题进行测试,仿真结果验证了所提出方法的有效性.

## 1 多目标问题和粒子群算法

为方便描述所提出的算法,首先对多目标问题和粒子群算法的一些相关概念进行简要说明.

### 1.1 多目标问题

具有  $n$  个决策变量、 $m$  个优化目标的多目标问题<sup>[9]</sup>可以表示为

$$\begin{aligned} \min F(x) &= (f_1(x), f_2(x), \dots, f_m(x))^T. \\ \text{s.t. } g_i(x) &\geq 0, \quad i = 1, 2, \dots, p; \\ h_j(x) &= 0, \quad j = 1, 2, \dots, q. \end{aligned} \quad (1)$$

其中:  $x = (x_1, x_2, \dots, x_n) \in \Omega \subset R^n$  为  $n$  维决策向量,  $\Omega$  为决策空间,  $F: R^n \rightarrow R^m$  包含了  $m$  个目标函数,  $g_i(x) \geq 0$  为不等式约束条件,  $h_j(x) = 0$  为等式约束条件.

**定义 1 (Pareto 支配)**<sup>[9]</sup> 称决策向量  $x_v = (v_1, v_2, \dots, v_n)$  对  $x_u = (u_1, u_2, \dots, u_n)$  是 Pareto 支配的, 记为  $x_v \prec x_u$ , 当且仅当  $\forall i = 1, 2, \dots, m, f_i(x_v) \leq f_i(x_u)$  且  $\exists j = 1, 2, \dots, m, f_j(x_v) < f_j(x_u)$ .

**定义 2 (Pareto 最优解集)**<sup>[9]</sup> 如果  $P^* = \{x \in R^n | \neg \exists x' \in R^n, x' \prec x\}$ , 则称  $P^*$  为 Pareto 最优解集.

**定义 3 (Pareto 前沿 (PF\*))**<sup>[9]</sup> Pareto 解集在目标函数上的映射为  $\text{PF}^* = \{F(x) | x \in P^*\}$ , 映射之后的集合称为 Pareto 前沿.

### 1.2 粒子群算法

粒子群算法是由 Kennedy 等<sup>[2]</sup>提出的一种基于种群搜索的进化计算技术. 在第  $k$  次迭代时, 种群中粒子  $i$  的速度和位置分别更新为

$$v_{i,d}^{(k+1)} =$$

$$\omega v_{i,d}^{(k)} + c_1 r_1 (p_{i,d} - x_{i,d}^{(k)}) + c_2 r_2 (p_{g,d} - x_{i,d}^{(k)}), \quad (2)$$

$$x_{i,d}^{(k+1)} = x_{i,d}^{(k)} + v_{i,d}^{(k+1)}. \quad (3)$$

其中:  $\omega$  为惯性权重,  $v_{i,d}^{(k)}$  和  $x_{i,d}^{(k)}$  分别为第  $k$  次迭代时粒子  $i$  的第  $d$  维变量的速度和位置,  $p_i$  和  $p_g$  分别为粒子  $i$  的历史最优和全局向导的位置,  $c_1$  和  $c_2$  为学习因子,  $r_1$  和  $r_2$  为  $[0,1]$  区间上均匀分布的随机值.

## 2 基于高斯混沌变异和精英学习的自适应多目标粒子群算法

本文算法(记为 AGE-MOPSO)首先提出粒子收敛性贡献的概念, 并作为自适应调整惯性权重和学习因子的依据; 然后给出收敛停滞判断的方法, 作为高斯混沌变异和精英学习的时机.

### 2.1 自适应调整惯性权重和学习因子

粒子群中的惯性权重  $\omega$ 、学习因子  $c_1$  和  $c_2$  可以通过调节来控制全局和局部搜索能力, 由于缺乏待优化问题的先验知识, 有必要根据种群收敛状态的反馈信息动态调整参数, 如何检测种群的收敛状态成为关键问题. 对此, 本文以粒子对档案收敛性提升的贡献为依据反映种群收敛状态. 当种群中的粒子加入到外部档案时, 计算其对外部档案中的解的支配程度, 以此得到该粒子对档案收敛性的贡献.

**定义 4** 假设一个由式(1)描述的多目标问题, 当粒子  $i$  (对应决策向量  $x_i$ ) 加入到档案  $A$  中时, 定义其对  $A$  的收敛性贡献为

$$C(x_i, A) = \max_{x \in A} \rho(x_i, x), \quad (4)$$

其中  $\rho(x_i, x)$  为  $x_i$  对  $x$  的支配程度, 计算如下:

$$\rho(x_i, x) = \begin{cases} \frac{1}{m} \sum_{j=1}^m \frac{|f_j(x_i) - f_j(x)|}{f_{j,\max} - f_{j,\min}}, & x_i \prec x; \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

$m$  为目标个数,  $f_{j,\max}$  和  $f_{j,\min}$  分别为  $A$  中解集在第  $j$  个目标上的最大值和最小值.

由定义 4 可见: 支配程度  $\rho(x_i, x)$  定量地给出了  $x_i$  比  $x$  更好的程度, 取值在  $[0,1]$  范围内;  $\rho(x_i, x)$  计算了在每个目标上  $x_i$  相对于  $x$  所提升的程度, 并进行了规范化处理, 使其值不受待优化问题的目标函数形式、Pareto 前沿形状的影响. 在支配程度定义的基础上, 当粒子  $i$  加入档案时, 以  $i$  对档案解的支配程度最大者作为其收敛性贡献. 显然, 如果  $i$  被档案中的解支配或者互不支配, 则其对档案的收敛性贡献为 0.

以粒子对档案的收敛性贡献反映当前收敛状态, 当所有粒子对档案的收敛性贡献趋近 0 时, 表明种群达到收敛稳定状态. 令  $f$  为粒子对外部档案的收敛性贡献最大值, 记为收敛因子, 即

$$f = \max_{i=1,2,\dots,N} C(x_i, A). \quad (6)$$

其中:  $N$  为种群大小,  $C(x_i, A)$  为粒子  $i$  对档案  $A$  的收敛性贡献,  $f \in [0, 1]$ . 图 1 为本文算法在单模态 ZDT1 和多模态 ZDT4 问题上的收敛因子随迭代次数变化的曲线. 图 1 中: ZDT1 的收敛因子曲线变化幅度较小且较快地趋近于 0, 表明算法很快达到收敛稳定状态; ZDT4 的收敛因子曲线变化幅度较大且变化周期较长, 表明算法不断产生新解替换档案内的旧解, 频繁跳出局部最优.

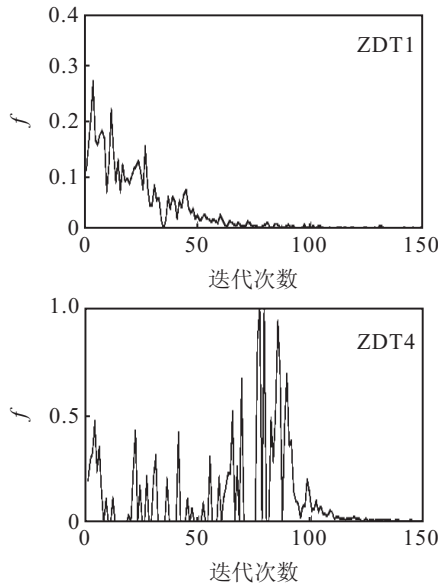


图 1 优化 ZDT1 和 ZDT4 时的收敛因子曲线

当  $f$  值较大时, 种群处于探索阶段, 应保持其全局搜索能力. 随着  $f$  值逐渐减小, 种群逐渐达到收敛稳定状态, 应加强局部搜索. 借鉴文献 [10] 的参数设置方法, 利用 Sigmoid 函数映射方式使得  $\omega$ 、 $c_1$  和  $c_2$  随着  $f$  的变化而变化. 首先假设  $\omega$  随  $f$  按如下公式变化:

$$\omega(f) = \frac{1}{1 + ae^{-bf}}. \quad (7)$$

设置当  $f$  从 1 减小到 0 时, 惯性权重从 0.9 递减到 0.4, 即当  $f = 0$  时,  $\omega = 0.4$ , 当  $f = 1$  时,  $\omega = 0.9$ . 代入式 (7) 求解得  $a = 1.5$ ,  $b = 2.6$ . 所以得到

$$\omega(f) = \frac{1}{1 + 1.5e^{-2.6f}} \in [0.4, 0.9], \quad \forall f \in [0, 1]. \quad (8)$$

同理, 设置当  $f$  从 1 减小到 0 时, 学习因子  $c_1$  从 2.5 递减到 0.5, 由于 Sigmoid 函数值范围为 (0,1), 假设

$$c_1(f) = \frac{5}{1 + ae^{-bf}}. \quad (9)$$

将  $f = 0$ ,  $c_1 = 0.5$  和  $f = 1$ ,  $c_1 = 2.5$  代入式 (9) 求解参数, 得到

$$c_1(f) = \frac{1}{0.2 + 1.8e^{-2.19f}} \in [0.5, 2.5], \quad \forall f \in [0, 1]. \quad (10)$$

设置当  $f$  从 1 减小到 0 时, 学习因子  $c_2$  从 0.5 递增到 2.5, 同理可得学习因子  $c_2$  变化为

$$c_2(f) = \frac{1}{0.2 + 0.2e^{2.2f}} \in [0.5, 2.5], \quad \forall f \in [0, 1]. \quad (11)$$

## 2.2 高斯混沌变异

本节给出收敛停滞状态的定义, 并在种群处于收敛停滞状态时, 采取一种带有高斯函数和混沌特性的变异方式协助种群跳出局部最优.

**定义 5** 若连续  $\tau$  次迭代中, 收敛因子  $f$  均小于预先指定的阈值  $\delta$ , 则定义种群处于收敛停滞状态.

文献 [11] 提出一种带有高斯混沌变异的单目标粒子群算法, 在此将该变异方法扩展到多目标粒子群. 被选中粒子的变异方法为

$$x_{i,d}^{(k)} = \begin{cases} x_{i,d}^{(k)} + \alpha h_{i,d}^{(k)} |p_{g,d} - x_{i,d}^{(k)}| (x_{\max,d} - x_{i,d}^{\prime(k)}), & \alpha \geq 0; \\ x_{i,d}^{(k)} + \alpha h_{i,d}^{(k)} |p_{g,d} - x_{i,d}^{(k)}| (x_{i,d}^{\prime(k)} - x_{\min,d}), & \alpha < 0. \end{cases} \quad (12)$$

其中:  $\alpha$  为  $[-1, 1]$  区间的随机数, 表明粒子位置是从正方向或负方向变异;  $|p_{g,d} - x_{i,d}^{(k)}|$  为粒子  $i$  在第  $d$  维上离全局最优的距离;  $x_{i,d}^{\prime(k)}$  为引入的 Logistic 混沌映射值, 使得变异具有各向遍历性, 具体为

$$x_{i,d}^{\prime(k)} = x_{\min,d} + z_{i,d}^{(k)} (x_{\max,d} - x_{\min,d}), \quad (13)$$

$$z_{i,d}^{(k)} = 4z_{i,d}^{(k-1)}(1 - z_{i,d}^{(k-1)}), \quad (14)$$

初始值  $z_{i,d}^{(0)} \in \text{rand}(0, 1)$  且  $z_{i,d}^{(0)} \neq 0.25, 0.5, 0.75$ ;  $x_{\max,d}$  和  $x_{\min,d}$  分别为第  $d$  维搜索空间的上界和下界;  $h_{i,d}^{(k)}$  为高斯函数, 表达式为

$$h_{i,d}^{(k)} = \exp\left(-\frac{(p_{g,d} - x_{i,d}^{(k)})^2}{2\sigma^2(k)}\right), \quad (15)$$

$$\sigma^2(k) = \sigma_0^2 e^{-\frac{k}{G_{\max}}}, \quad (16)$$

$k$  为当前的迭代次数,  $G_{\max}$  为最大迭代次数,  $\sigma_0$  为初始方差.

在所提变异算子中, 设置的高斯函数  $h_{i,d}^{(k)}$  具有两个特性使得变异更加有效: 首先, 粒子变异的幅度与粒子的位置有关. 如图 2 所示高斯曲线 1 中, 中心点为全局最优粒子的位置  $p_{g,d}$ , 离  $p_{g,d}$  较近的粒子  $x_{1,d}$  (即陷入局部最优、基本没有搜索能力的粒子) 具有较大的高斯函数值  $y_1$ , 再由式 (12) 计算得到较大幅度的变异, 重新获得搜索能力. 远离  $p_{g,d}$  的粒子  $x_{2,d}$  (还保持着搜索能力的粒子) 具有较小的高斯函数值  $y_2$ , 得到较小幅度的扰动, 从而避免过多干扰其搜索行为, 保持其搜索的连贯性. 粒子变异幅度随迭代次数的增大而逐渐减小. 因为随着迭代次数  $k$  逐渐增大, 式 (16) 中的  $\sigma^2(k)$  值将逐渐减小,  $\sigma^2(k)$  的值决定

了高斯函数曲线的陡峭程度, 其值越小, 高斯曲线越陡. 如图2中, 高斯曲线2的比高斯曲线1具有更小的 $\sigma^2(k)$ 值, 所以其曲线更加窄和陡峭, 使得同一位置的粒子 $x_{1,d}$ 在高斯曲线2所得的函数值更小( $y_3 < y_1$ ), 由式(12)计算所得的变异幅度值也更小. 所以, 整体种群的变异幅度逐渐减小, 保证了算法在迭代后期的开发能力.

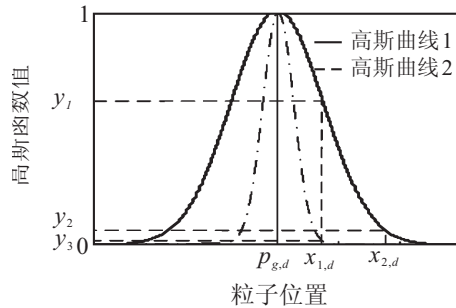


图2 带高斯函数的变异方法

### 2.3 精英学习

由于缺乏对待优化问题的先验知识, 当种群处于收敛停滞状态时, 即可能陷入局部最优, 也可能收敛到全局最优. 本文在高斯混沌变异的同时, 提出精英学习策略, 协助种群在收敛到全局最优附近时寻找精确解, 进一步提高收敛性.

选择外部档案中的某个非支配解 $i$ , 基于该解按下式进行搜索产生一个新解 $z$ :

$$x_{z,d} = \begin{cases} x_{i,d}, & d \neq d^*; \\ x_{i,d} + \phi \cdot (x_{j,d} - x_{i,d}), & d = d^*. \end{cases} \quad (17)$$

其中:  $d^*$  为随机选择的变量维,  $\phi$  为 $[-1, 1]$ 范围的随机数,  $j$  为档案中随机选择的解且 $j \neq i$ .

### 2.4 维护档案和选择全局最优向导

Deb等<sup>[12]</sup>提出的拥挤距离能快速判断非支配解之间的密集程度, 且能保持良好的解集分布性, 本文采用该策略来维护外部档案. 此外, 拥挤距离还用于为每个粒子确定全局最优向导: 随机选择档案中的两个个体, 将拥挤距离较大的个体作为该粒子的全局最优向导.

### 2.5 算法流程

AGE-MOPSO算法的总体流程如下所示.

**Step 1:** 初始化. 设定初始种群 $P^{(0)}$ , 种群规模 $N$ 和最大迭代次数 $G_{\max}$ ; 随机初始化所有粒子, 初始化粒子的个体最优 $p_i$ 为当前位置 $x_i$ ; 初始化档案 $A^{(0)} = \emptyset$ , 并给定最大容量 $\bar{A}$ ; 初始化进化因子 $f = 1$ ; 令迭代次数 $k = 0$ .

**Step 2:** 将 $P^{(0)}$ 中的非支配解加入 $A^{(0)}$ , 形成 $A^{(1)}$ .

**Step 3:** 令 $k := k + 1$ , 开始循环迭代.

**Step 4:** 由式(8)、(10)和(11)计算 $\omega$ 、 $c_1$ 和 $c_2$ , 种群中的粒子按式(2)和(3)更新位置, 形成 $P^{(k)}$ . 根据定义5判断种群是否陷入收敛停滞状态, 若是则进行高斯混沌变异和精英学习, 转至Step 5.

**Step 5:** 将种群中的个体、精英学习产生的个体与 $A^{(k)}$ 中的解进行Pareto支配比较, 如果不被 $A^{(k)}$ 中的解支配, 则加入到 $A^{(k)}$ 中, 并计算收敛因子 $f$ . 判断 $A^{(k)}$ 中解的数量是否大于 $\bar{A}$ , 如果是则删除拥挤距离最小的多余的解.

**Step 6:** 若不满足算法终止条件, 则返回Step 3继续迭代, 否则结束程序, 输出 $A^{(k)}$ 中的非支配解.

### 2.6 计算复杂度分析

假设优化问题含有 $m$ 个目标, 种群大小为 $N$ , 档案最大容量为 $\bar{A}$ . 种群函数评价复杂度为 $O(mN)$ , 拥挤距离的复杂度为 $O(m\bar{A} \log \bar{A})$  (假设使用快速排序算法), 种群和档案进行Pareto支配比较, 并检测种群收敛状态的计算复杂度为 $O(mN(N + \bar{A}))$ ; 高斯混沌变异的复杂度为 $O(n_g)$ ,  $n_g$ 为变异粒子的个数, 一般 $n_g < N$ ; 精英学习的复杂度为 $O(n_e)$ ,  $n_e$ 为学习次数, 一般 $n_e < \bar{A}$ . 如果档案最大容量等于种群数量, 则AGE-MOPSO的计算复杂度为 $O(mN^2)$ .

## 3 仿真分析

### 3.1 实验设置

为了验证AGE-MOPSO的有效性, 采用标准测试函数进行仿真实验, 分别为Kursawe<sup>[9]</sup>、ZDT<sup>[13]</sup>、DTLZ<sup>[14]</sup>和WFG系列<sup>[15]</sup>共15个测试问题. ZDT系列包含6个二目标测试问题, 即ZDT1~ZDT6. ZDT5为二进制测试问题, 需要二进制编码, 所以本文省去ZDT5的测试. DTLZ系列包含7个测试问题, 分别为DTLZ1~DTLZ7, 每个问题的变量维数和目标个数均可以变化, 本次仿真中的DTLZ问题均为3目标的测试问题.

为了比较AGE-MOPSO的性能, 选取5个典型的多目标粒子群算法进行对比, 分别为MOPSO<sup>[9]</sup>、MOPSO-CD<sup>[16]</sup>、OMOPSO<sup>[17]</sup>、TV-MOPSO<sup>[5]</sup>和NSGA-II<sup>[12]</sup>. 各算法的种群大小和档案最大容量均设置为100, 每个测试问题上最大评价次数为25 000次. 在AGE-MOPSO中, 设置收敛停滞判断所需的迭代次数 $\tau = 10$ , 阈值 $\delta = 0.05$ ; 每次迭代过程中高斯混沌变异和精英学习次数均为种群大小的1/3, 以平衡探索和开发. 其他算法的参数按照对应参考文献的推荐进行设置. 所有仿真实验在2.6 GHz双CPU和4 GB内存的电脑上运行.

为了评价算法所得解集的收敛性和多样性, 采用世代距离(GD)<sup>[9]</sup>和分布性指标( $\Delta$ )<sup>[18]</sup>作为性能评价

指标. GD度量近似Pareto前沿到真实的Pareto前沿之间的距离,其值越低,表明解集收敛性越好; $\Delta$ 指标度量所得解集的分布性,其值越低,表明解集分布性越好.

### 3.2 实验结果及比较分析

表1和表2分别列出了6个对比算法在15个测试问题上的GD、 $\Delta$ 的平均值和方差(括号内数据为方差),每个结果均为同一算法在同一测试问题上独立运行30次的统计结果,采用粗体字表示所有对比算法在对应行测试问题中的最小评价指标值.此外,为了测试所得结果是否具有统计学意义,采用Mann-Whitney秩和检验方法进行分析,表1和表2中的“+”、“-”和“=”分别表示在显著性水平为5%的Mann-Whitney统计检验下,AGE-MOPSO所得的指标值分别优于、劣于和等于对应列算法的指标值.为直

观展示各算法所得解集的收敛性和分布性,图3和图4给出了6种算法在部分测试问题上的Pareto前沿(所有图例均与第1个图相同).

由表1可见,AGE-MOPSO在15个测试问题中获得了9个最优GD值(其中6个显著性好于其他对比算法),1个不差于其他算法的GD值,4个次优值.MOPSO获得1个最优值,OMOPSO获得1个最优值,NSGA-II获得4个最优值.在一些较容易的优化问题如ZDT1(凸Pareto前沿)、ZDT2(非凸Pareto前沿)、ZDT3(非连续Pareto前沿)和WFG3(线性Pareto前沿)的优化中,所提出算法获得最好的收敛性(GD值最小).这是因为根据种群收敛状态自适应调整惯性权重和学习因子的策略,合理控制了探索和开发过程,使得算法较快速度过了探索阶段,并充分进行开发.且精英学习方法直接对档案解进行局部搜

表1 6种对比算法在标准测试问题上的GD指标对比结果

测试问题	AGE-MOPSO	MOPSO <sup>[9]</sup>	MOPSO-CD <sup>[16]</sup>	OMOPSO <sup>[17]</sup>	TV-MOPSO <sup>[5]</sup>	NSGA-II <sup>[12]</sup>
ZDT1	<b>1.23e-4(2.6e-5)</b>	3.57e-3(3.8e-4)	+ 3.51e-4(9.3e-5)	+ 1.44e-4(3.6e-5)	+ 2.14e-4(3.4e-5)	+ 3.03e-4(4.2e-5)
ZDT2	<b>5.85e-5(7.7e-6)</b>	2.78e-3(3.5e-4)	+ 3.02e-4(8.2e-5)	+ 8.54e-5(1.6e-5)	+ 1.80e-4(5.0e-5)	+ 2.96e-4(5.8e-5)
ZDT3	<b>1.19e-4(2.0e-5)</b>	2.72e-3(5.6e-4)	+ 1.27e-3(6.1e-4)	+ 1.70e-4(4.4e-5)	+ 1.58e-4(2.9e-5)	+ 1.62e-4(1.6e-5)
ZDT4	5.69e-1(3.2e-1)	8.59e+0(1.9e+0)	+ 6.63e+0(2.7e+0)	+ 9.69e-1(5.2e-1)	+ 7.37e-1(3.5e-1)	+ <b>1.38e-3(2.4e-3)</b>
ZDT6	7.80e-3(1.7e-2)	1.01e-1(3.4e-2)	+ 2.62e-1(2.9e-1)	+ 2.56e-2(3.1e-2)	+ 4.30e-2(2.3e-2)	+ <b>1.98e-3(2.7e-4)</b>
DTLZ1	4.98e+0(3.3e+0)	3.30e+1(3.0e+0)	+ 2.11e+1(3.7e+0)	+ 1.60e+1(2.0e+0)	+ 1.50e+1(1.5e+0)	+ <b>1.42e+0(1.0e+0)</b>
DTLZ2	3.88e-3(4.6e-4)	<b>1.45e-3(9.8e-4)</b>	- 6.71e-3(4.8e-4)	+ 3.09e-3(3.5e-4)	+ 4.10e-3(4.6e-4)	= 1.75e-3(1.4e-4)
DTLZ3	4.73e+0(2.3e+0)	4.35e+1(5.2e+0)	+ 3.65e+1(3.9e+0)	+ 2.84e+1(2.7e+0)	+ 2.97e+1(3.3e+0)	+ <b>1.43e+0(1.0e+0)</b>
DTLZ4	<b>5.58e-3(3.9e-4)</b>	8.01e-3(2.5e-3)	+ 1.30e-2(1.3e-3)	+ 5.87e-3(3.8e-4)	+ 6.26e-3(4.8e-4)	+ 5.63e-3(2.4e-4)
DTLZ5	<b>5.81e-4(3.8e-5)</b>	7.49e-4(6.3e-5)	+ 4.62e-3(5.0e-4)	+ 5.93e-4(4.7e-5)	= 6.21e-4(5.4e-5)	+ 6.26e-4(4.2e-5)
DTLZ6	5.60e-4(4.1e-5)	4.97e-1(4.1e-2)	+ 5.65e-4(5.9e-5)	= <b>5.59e-4(3.2e-5)</b>	= 5.64e-4(3.3e-5)	+ 1.39e-1(1.5e-2)
DTLZ7	<b>4.49e-3(6.1e-4)</b>	7.06e-3(1.6e-3)	+ 4.50e-3(7.9e-4)	= 5.27e-3(6.5e-4)	+ 5.52e-3(7.5e-4)	+ 4.97e-3(5.3e-4)
Kursawe	<b>1.54e-4(1.7e-5)</b>	1.20e-3(1.7e-4)	+ 1.93e-3(6.2e-4)	+ 5.53e-4(5.6e-5)	+ 7.24e-4(1.0e-4)	+ 2.07e-3(2.1e-5)
WFG3	<b>1.48e-4(8.8e-6)</b>	6.29e-4(5.4e-5)	+ 4.87e-4(2.7e-5)	+ 1.53e-4(9.3e-6)	= 1.82e-4(7.2e-6)	+ 1.89e-4(2.3e-5)
WFG4	<b>5.03e-4(5.9e-5)</b>	1.36e-3(8.3e-5)	+ 1.39e-3(9.8e-5)	+ 1.11e-3(1.0e-4)	+ 1.22E-4(6.4e-5)	+ 6.96e-4(8.0e-5)

表2 6种对比算法在标准测试问题上的 $\Delta$ 指标对比结果

测试问题	AGE-MOPSO	MOPSO <sup>[9]</sup>	MOPSO-CD <sup>[16]</sup>	OMOPSO <sup>[17]</sup>	TV-MOPSO <sup>[5]</sup>	NSGA-II <sup>[12]</sup>
ZDT1	<b>7.86e-2(1.2e-2)</b>	4.72e-1(4.5e-2)	+ 4.48e-1(8.0e-2)	+ 8.05e-2(7.9e-3)	+ 1.01e-1(1.6e-2)	+ 3.87e-1(4.7e-2)
ZDT2	<b>7.39e-2(1.2e-2)</b>	4.98e-1(4.2e-2)	+ 4.72e-1(9.6e-2)	+ 7.82e-2(1.2e-2)	+ 1.08e-1(1.4e-2)	+ 3.86e-1(5.0e-2)
ZDT3	<b>1.58e-1(2.8e-2)</b>	5.97e-1(4.3e-2)	+ 5.58e-1(7.4e-2)	+ 1.58e-1(2.3e-2)	= 1.78e-1(3.0e-2)	+ 3.72e-1(5.4e-2)
ZDT4	8.00e-1(7.9e-2)	9.52e-1(7.2e-2)	+ 8.85e-1(1.0e-1)	+ 8.92e-1(8.8e-2)	+ 8.00e-1(8.8e-2)	= <b>4.23e-1(1.1e-1)</b>
ZDT6	3.63e-1(5.1e-1)	1.42e+0(1.6e-1)	+ 1.51e+0(2.4e-1)	+ 7.04e-1(6.0e-1)	+ 1.06e+0(2.6e-1)	+ <b>3.29e-1(2.7e-2)</b>
DTLZ1	7.46e-1(1.7e-1)	<b>5.46e-1(5.8e-2)</b>	- 5.60e-1(5.0e-2)	+ 6.31e-1(9.8e-2)	+ 5.78e-1(5.4e-2)	+ 9.55e-1(3.2e-1)
DTLZ2	<b>3.73e-1(3.9e-2)</b>	7.94e-1(2.0e-2)	+ 3.87e-1(3.9e-2)	= 3.87e-1(3.5e-2)	= 3.93e-1(2.4e-2)	+ 5.07e-1(5.0e-2)
DTLZ3	7.11e-1(1.2e-1)	<b>6.96e-1(9.3e-2)</b>	= 7.77e-1(7.5e-2)	+ 8.16e-1(1.2e-1)	+ 7.62e-1(6.6e-2)	+ 1.02e+0(2.1e-1)
DTLZ4	<b>3.97e-1(4.0e-2)</b>	6.20e-1(9.9e-2)	+ 8.34e-1(8.6e-2)	+ 4.55e-1(1.0e-1)	+ 6.26e-1(6.8e-2)	+ 5.00e-1(5.0e-2)
DTLZ5	1.44e-1(1.7e-2)	5.61e-1(5.5e-2)	+ 4.41e-1(4.3e-2)	+ <b>1.26e-1(1.5e-2)</b>	- 4.61e-1(4.0e-2)	+ 4.44e-1(5.6e-2)
DTLZ6	1.03e-1(8.5e-3)	7.24e-1(7.6e-2)	+ 5.02e-1(1.2e-1)	+ <b>1.02e-1(1.6e-2)</b>	= 1.34e-1(1.3e-2)	+ 6.48e-1(4.9e-2)
DTLZ7	4.11e-1(3.8e-2)	6.75e-1(7.1e-2)	+ 4.52e-1(5.0e-2)	+ 4.26e-1(3.6e-2)	= <b>3.92e-1(3.6e-2)</b>	= 4.92e-1(4.2e-2)
Kursawe	<b>2.41e-1(6.5e-3)</b>	6.62e-1(6.3e-2)	+ 6.08e-1(7.6e-2)	+ 3.00e-1(3.0e-2)	+ 3.49e-1(5.8e-2)	+ 6.28e-1(6.5e-2)
WFG3	<b>3.75e-1(5.8e-3)</b>	6.71e-1(3.2e-2)	+ 5.35e-1(1.8e-2)	+ 3.74e-1(5.8e-3)	= 3.80e-1(6.3e-3)	+ 5.99e-1(3.3e-2)
WFG4	<b>1.62e-1(2.3e-2)</b>	5.50e-1(5.2e-2)	+ 4.50e-1(4.3e-2)	+ 3.74e-1(3.8e-2)	+ 4.11e-1(4.4e-2)	+ 3.62e-1(3.7e-2)

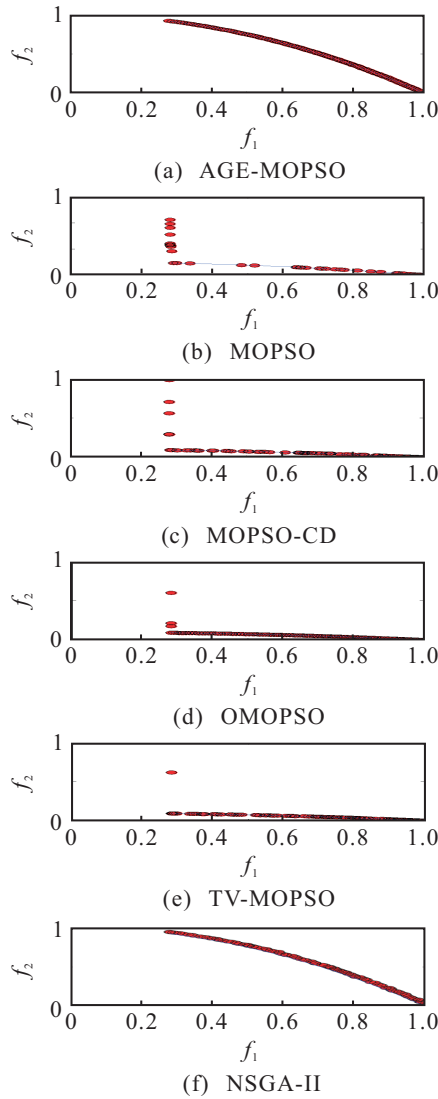


图3 6种对比算法在ZDT6上的Pareto前端比较

索,使档案解之间进行充分的信息交换,提高了收敛性. 所提出算法在较难的多模态问题如WFG4上得到最好收敛性,在ZDT4、DTLZ1和DTLZ3多模态问题上取得第2好的收敛性,仅次于NSGA-II算法,优于其他粒子群算法. 这表明高斯混沌变异能够缓解粒子群早熟收敛的现象,协助粒子群跳出局部最优. 图3为ZDT6(Pareto解集分布不均匀的测试问题)的优化结果,可以看出,只有AGE-MOPSO和NSGA-II所得解能都收敛到真实Pareto前沿附近,其他4种对比的粒子群算法虽然也寻找到全局最优区域,但有出现部分解远离真实Pareto前沿的情况.

由表2的 $\Delta$ 评价指标结果可见,所提出算法在15个测试问题所得 $\Delta$ 值中,获得8个最优值(其中5个显著性优于其他算法),6个次优值. MOPSO、OMOPSO和NSGA-II分别获得两个最优值,TV-MOPSO获得一个最优值. 这表明所提算法获得的解集分布均匀性也存在一定优势. 一般而言,在算法运行后期的开发程度高低能直接影响着分布性的好坏. 本文算法在检测到解集收敛性提升较小时,从

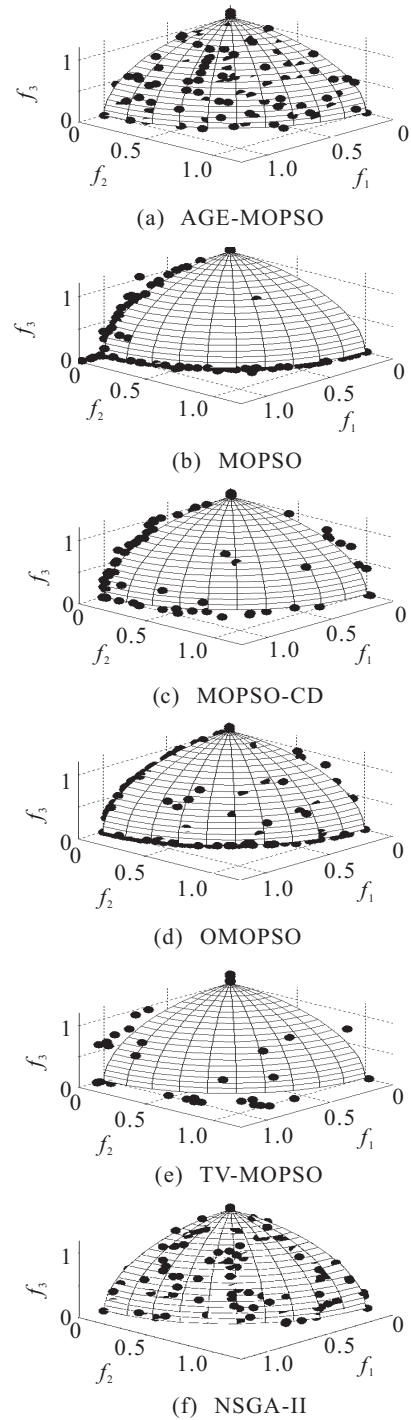


图4 6种对比算法在DTLZ4上的Pareto前端比较

探索模式切换到开发模式,以进行较充分地开发,增加获取分布性更好的解的机会. 而且,当判断收敛停滞时,精英学习策略更进一步协助种群获取分布性更好的解. 例如,DTLZ4的Pareto前沿为非分布均匀的解组成,可以检验算法在保持解分布性方面的能力,本文算法得到了最小的 $\Delta$ 值. 由图4可见,AGE-MOPSO和NSGA-II所得的近似Pareto前沿比较均匀地分布在真实Pareto前沿附近各处,其他算法获得的解大部分聚集在真实Pareto前沿的边界区域,中间区域的解较少,表明了自适应参数策略和精英学习策略的有效性.

## 4 结 论

本文提出了一种能有效平衡全局和局部搜索的多目标粒子群算法,根据档案解集的收敛性提升程度反映种群收敛状态,并自适应调整惯性权重和学习因子.判定种群收敛停滞,采用高斯混沌变异和精英学习策略进一步提升全局和局部搜索能力.与其他典型多目标粒子群在标准测试问题上进行仿真对比的结果表明,所提出算法具有较好的全局寻优和局部搜索能力,所得非支配解集的收敛性和分布性均较好.

### 参考文献(References)

- [1] 公茂果, 焦李成, 杨咚咚, 等. 进化多目标优化算法研究[J]. 软件学报, 2009, 20(2): 271-289.  
(Gong M G, Jiao L C, Yang D D, et al. Research on evolutionary multi-objective optimization algorithms[J]. J of Software, 2009, 20(2): 271-289.)
- [2] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Piscataway: IEEE, 1995: 1942-1948.
- [3] Reyes-Sierra M, Coello C C. Multi-objective particle swarm optimizers: A survey of the state-of-the-art[J]. Int J of Computational Intelligence Research, 2006, 2(3): 287-308.
- [4] 陈民铀, 张聪誉, 罗辞勇. 自适应进化多目标粒子群优化算法[J]. 控制与决策, 2009, 24(12): 1851-1855.  
(Chen M Y, Zhang C Y, Luo C Y. Adaptive evolutionary multi-objective particle swarm optimization algorithm[J]. Control and Decision, 2009, 24(12): 1851-1855.)
- [5] Tripathi P K, Bandyopadhyay S, Pal S K. Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients[J]. Information Sciences, 2007, 177(22): 5033-5049.
- [6] Hu W, Yen G. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system[J]. IEEE Trans on Evolutionary Computation, 2015(19): 1-18.
- [7] 夏立荣, 李润学, 刘启玉, 等. 基于动态层次分析的自适应多目标粒子群优化算法及其应用[J]. 控制与决策, 2015, 30(2): 215-221.  
(Xia L R, Li R X, Liu Q Y, et al. Adaptive multi-objective particle swarm optimization algorithm based on dynamic AHP and its application[J]. Control and Decision, 2015, 30(2): 215-221.)
- [8] 申晓宁, 郭毓, 陈庆伟, 等. 一种子群体个数动态变化的多目标优化协同进化算法[J]. 控制与决策, 2007, 22(9): 1011-1016.  
(Shen X N, Guo Y, Chen Q W, et al. A multi-objective optimization co-evolutionary algorithm with dynamically varying number of subpopulations[J]. Control and Decision, 2007, 22(9): 1011-1016.)
- [9] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 256-279.
- [10] Zhan Z H, Zhang J, Li Y, et al. Adaptive particle swarm optimization[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 2009, 39(6): 1362-1381.
- [11] Han M, Fan J C, Wang J. A dynamic feedforward neural network based on Gaussian particle swarm optimization and its application for predictive control[J]. IEEE Trans on Neural Networks, 2011, 22(9): 1457-1468.
- [12] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: Nsga-II[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.
- [13] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results[J]. Evolutionary Computation, 2000, 8(2): 173-195.
- [14] Deb K, Thiele L, Laumanns M, et al. Scalable test problems for evolutionary multiobjective optimization[C]. Proc of IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 2002: 26-33.
- [15] Huband S, Hingston P, Barone L. A review of multiobjective test problems and a scalable test problem toolkit[J]. IEEE Trans on Evolutionary Computation, 2006, 10(5): 477-506.
- [16] Raquel C R, Naval P C. An effective use of crowding distance in multiobjective particle swarm optimization[C]. Proc of Genetic and Evolutionary Computation Conf. New York: IEEE, 2005: 257-264.
- [17] Sierra M R, Coello C A C. Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance[C]. Proc of Evolutionary Multi-Criterion Optimization. Berlin: Springer-Verlag, 2005: 505-519.
- [18] Aimin Z, Yaochu J, Qingfu Z, et al. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion[C]. Proc of IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 2006: 892-899.

(责任编辑: 郑晓蕾)