

# 自适应控制参数的通用差异演化算法研究\*

简平<sup>a</sup>, 邹鹏<sup>b</sup>, 熊伟<sup>b</sup>, 陈治科<sup>a</sup>

(解放军装备学院 a. 研究生管理大队; b. 重点实验室, 北京 101416)

**摘要:** 为提高求解多目标优化问题效率,对通用差异演化(GDE)算法及其自适应参数控制问题进行了研究。首先,分析了GDE3算法的编码、交叉、变异、选择等原理和算法流程;然后,利用个体的适应度作为参数调整的依据,并结合一定的调整概率提出一种新的对缩放因子和交叉概率参数自适应控制策略,提高算法的搜索能力;最后,通过典型的多目标函数对自适应控制参数的通用演化算法(selfGDE3)、GDE3和非劣分层遗传算法2(NSGA-II)的性能进行比较分析,结果表明,selfGDE3算法具有良好的搜索性能。

**关键词:** 多目标优化; 非劣分层遗传算法2; 通用差异演化算法; 自适应控制参数

**中图分类号:** TP18 **文献标志码:** A **文章编号:** 1001-3695(2012)08-2899-04

doi:10.3969/j.issn.1001-3695.2012.08.025

## Research on generalized differential evolution algorithm based on adaptive controlling parameter

JIAN Ping<sup>a</sup>, ZOU Peng<sup>b</sup>, XIONG Wei<sup>b</sup>, CHEN Zhi-ke<sup>a</sup>

(a. Postgraduate Management Team, b. Key Laboratory, the PLA Equipment Academy, Beijing 101416, China)

**Abstract:** In order to solve the multi-objective optimization problem efficiently, this paper researched on generalized differential evolution algorithm and the method of adaptive controlling parameter. Firstly, it analyzed the principle and process of generalized differential evolution algorithm 3, including coding, crossover, mutation. Secondly, the algorithm put forward adaptive controlling strategy to crossover and mutation parameter based on the fitness of individual and the adjusting probability, which improved the performance of algorithm. Finally, it compared the performance of selfGDE3, GDE3 and NSGA-II through testing some benchmark functions. The results show the feasibility of selfGDE3.

**Key words:** multi-objective optimization; NSGA-II; GDE (generalized differential evolution) algorithm; adaptive controlling parameter

一般来说,求解优化问题的理想情况是快速有效地得到全局最优解。在面对某些大型问题时,需要在复杂而庞大的搜索空间中寻找最优解或准最优解,传统优化方法在计算速度、收敛性、初值敏感性等方面都不能满足要求,甚至无法求解。近年来人们通过模拟自然界生态系统机制,提出了一系列求解复杂优化问题的仿生优化算法,如遗传算法、蚁群算法、粒子群算法等,丰富了现代优化技术,并广泛地应用到组合优化问题。演化算法、差异演化算法的提出,以及 Lampinen 等人<sup>[1-3]</sup>先后提出三代通用差异演化算法,进一步发展了多目标优化方法,提高了对问题的全局搜索能力。在设计演化算法时,一方面需要针对具体问题选择适当的编码方案及相应的遗传算子,另一方面需要选择算法的控制参数。参数的不同选取会对算法的性能产生较大的影响。对如何确定算法较好地控制参数,目前没有相关理论和方法指导,主要依靠经验设定<sup>[4]</sup>。本文对通用差异演化算法的原理进行研究,并对影响算法的参数进行自适应控制,以提高算法的搜索能力。

## 1 多目标优化问题及优化算法

### 1.1 多目标优化问题概述

多目标优化问题可以简要定义为:寻找一组既满足约束条

件又使总目标函数最优化的决策变量的取值。一般多目标优化问题由  $n$  个决策变量、 $k$  个目标函数组成。优化的目标为

$$\min y = f(x) = (f_1(x), f_2(x), \dots, f_k(x))$$
$$\text{约束条件 } e_i(x) \leq 0 \quad i=1, 2, \dots, m \quad (1)$$

$$x = (x_1, x_2, \dots, x_n) \in X, y = (y_1, y_2, \dots, y_k) \in Y$$

其中: $x$ 表示决策向量; $y$ 表示目标向量, $e_i(x)$ 表示总共  $m$  个约束条件的第  $i$  个; $X$ 表示  $x$ 形成的决策空间; $Y$ 表示  $y$ 形成的目标空间。多目标优化问题中各个子目标可能是相互冲突的,不能使其同时达到最优,只能进行折中处理,使各个子目标尽可能达到最优。在实际应用问题中,通常从 Pareto 最优解集中挑选一组解作为多目标优化问题的最优解。设  $x^* \in X$ ,若不存在  $x \in X$ ,使得  $f(x)$  优于  $f(x^*)$ ,则称  $x^*$  为多目标优化问题的 Pareto 最优解,所有 Pareto 最优解的集合称为 Pareto 最优集。Pareto 最优解对应的目标向量称为非支配目标向量,所有非支配目标向量的集合称为多目标优化问题的 Pareto 前沿。Pareto 前沿的散布性能将反映非支配目标向量在目标空间  $Y$  上分布的广泛性和均匀性,广泛性和均匀性越好, Pareto 前沿更优。

### 1.2 典型优化算法

传统的多目标优化方法是将多个目标转换为一个目标的

收稿日期: 2011-12-27; 修回日期: 2012-02-22 基金项目: 装备预先研究资助项目

作者简介: 简平(1985-),男,江西新余人,博士研究生,主要研究方向为信息系统设计与优化(jianping85730@sina.com); 邹鹏(1957-),男,教授,博导,主要研究方向为高性能计算机及应用; 熊伟(1971-),男,教授,博导,主要研究方向为信息系统设计与优化。

优化问题,用单目标优化方法分别求解,其解集区近似多目标优化问题的 Pareto 最优集。常见的方法有权求和法、 $\varepsilon$ -约束法、最小—最大法等<sup>[5]</sup>。加权求和法就是将多目标优化中的各个目标函数加权求和,转换为单目标优化问题进行求解,通过选取不同的权重组合可以获得不同的 Pareto 最优解。其缺点是权重的选取与各个目标的相对重要程度有很大关系,很难在 Pareto 最优前端的非凸部分上求得解。 $\varepsilon$ -约束法是将某个目标函数作为优化目标来约束其他目标函数的方法以求解多目标优化问题。最小—最大法是通过最小化各个目标函数值与预设目标值之间的最大偏移量来寻求问题的最优解。以遗传算法为代表的演化算法具有适应性、通用性、隐并行性、扩展性等特点,已经广泛应用于多目标优化问题的求解上。常用的几种多目标遗传算法是并列选择法、非劣分层遗传算法(NSGA)和非劣分层遗传算法 2(NSGA-II)、基于目标加权法的遗传算法、多目标粒子群算法(MOPSO)、微遗传算法(micro-genetic algorithm, micro-GA)等。演化算法<sup>[6]</sup>可以在单轮模拟过程中找到多个 Pareto 最优解,并通过逐代积累寻找具有某些特征的个体,使种群有效逼近多目标优化问题的整个 Pareto 前沿;此外,演化算法不局限于 Pareto 前沿的形状和连续性,易于处理不连续的、凹形的 Pareto 前沿。

## 2 通用差异演化算法

差异演化算法是一种基于种群演化的实数编码多目标全局优化算法,其演化过程与遗传算法类似,也包括种群初始化、变异、交叉和选择操作。根据父代种群个体间的差异产生子代,其子代的产生受变异概率和交叉概率的影响很大。基于种群操作的演化算法可以隐并行地搜索解空间的多个解,并能利用不同解之间的相似性来提高求解问题效率,是一种基于 Pareto 最优概念的多目标优化的演化算法。通用差异演化算法<sup>[1-3,7]</sup>共发展了三代,GDE1 修改了基本演化算法 DE 的选择算子,交叉算子和变异算子没有变化,从而可以支持有约束的多目标优化问题。其选择的基本思想是:如果实验个体约束优于上一代的个体,则在下一代用实验个体取代父代个体,但在优化过程中没有对非劣个体进行排序,不具备维持解的分布性和扩展性的机制,优化解对控制参数的依赖大;GDE2 引入拥挤距离对在目标函数空间互不占优的实验个体和父代个体进行决策,提高了了解的扩展性和分布性,但是降低了种群的聚集程度;针对这些问题,GDE3 对原有的 GDE 和 GDE2 进行了改进,将 DE 方法应用到具有  $M$  个目标函数和  $V$  个约束函数的问题。GDE3 算法的基本流程描述如下:

a) 输入初始化参数  $D, G_{\max}, NP \geq 4, F \in (0, 1 + ]$ , 初始基向量为  $\mathbf{x}^{(lo)}, \mathbf{x}^{(hi)}$ 。

b) 初始化父代种群。 $\forall i \leq NP \wedge \forall j \leq D: x_{j,i,G=0} = x_j^{(lo)} + \text{rand}_j[0,1] \cdot (x_j^{(hi)} - x_j^{(lo)}), i = \{1, 2, \dots, NP\}, j = \{1, 2, \dots, D\}, G=0, m=0, \text{rand}_k[0,1] \in [0,1]$ 。

c) 交叉变异产生父代个体的实验个体。任意选择  $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$  用来生成  $x_{j,i,G}$  的实验个体,其中  $r_1, r_2, r_3$  与  $i$  两两互异,并产生随机参数  $j_{\text{rand}} = \text{rand}(1, D)$ , 则实验个体  $u_{j,i,G}$  的每个决策变量有

for( $j=1; j \leq D; j=j+1$ )

```

{
  if( randj[0,1) < CR ∨ j = jrand )
    uj,i,G = xj,r3,G + F(xj,r2,G - xj,r1,G)
  else
    uj,i,G = xj,i,G
}

```

CR 控制交叉操作,代表实验个体选择三个父代个体线性组合的概率; $F$  控制问题解的搜索速度和鲁棒性。采用这种交叉策略可以确保下一代个体中至少一个染色体来源于实验个体  $u_{j,i,G}$ 。

d) 选择操作。对父代个体和其实验个体进行比较选择产生下一代个体,遵循的规则为:对于两个不可行解,选择实验个体和父代个体中占优的解,如果互不占优,则选择父代个体;对于可行解和不可行解,选择可行的个体;如果实验个体和父代个体均可行,选择占优的解,如果互不占优,同时选择两个解,且  $m = m + 1$ 。

e)  $i = i + 1$ , 如果  $i < NP$  转步骤 c)。

f) 非劣排序。选择合适个体,保持  $G + 1$  代种群规模为  $NP$ 。对  $G + 1$  代的种群  $P_{G+1} = \{x_{1,G+1}, x_{2,G+1}, \dots, x_{NP,G+1}, \dots, x_{NP+m,G+1}\}$  (规模为  $NP \sim 2NP$  间),通过非劣排序和计算拥挤距离的方法,使得种群规模为  $NP$ 。

g) 若  $G < G_{\max}$ ,  $G = G + 1$ , 返回步骤 c) 操作。

h) 输出当前种群即为 Pareto 最优解。

## 3 控制参数自适应调整策略

控制参数对差异演化算法的优化能力和收敛速度等性能会产生较大的影响。在通用差异演化算法中主要有种群规模  $NP$ 、缩放因子  $F$  和交叉概率  $CR$  三个控制参数。种群规模太小,会使种群失去多样性,陷入局部极小点;种群规模太大,会导致大量的个体适应度评估运算。 $F$  决定了差向量对基向量的影响程度, $F$  取值太大,使算法迭代增多,收敛慢; $F$  取值太小,会加快算法的收缩速度,从而使算法错过全局最优解,陷入局部极小点。 $CR$  用来控制交叉操作生成的向量中哪些分量由变异向量贡献,哪些分量由目标向量贡献, $CR$  的值越大,变异向量的贡献就越大。

在通用差异演化算法中,这三个参数是保持不变的。对一个具体问题,要获得最优的参数设置比较困难,因此,有学者提出了自适应调整控制参数的策略<sup>[8]</sup>。在演化过程中  $F$  和  $CR$  以概率的形式随种群发生变化,但未考虑个体的适应度和多目标的情况,具有一定的盲目性。相对于  $NP$ ,差异演化算法的性能对  $F$  和  $CR$  更敏感<sup>[9]</sup>,自适应技术主要应用于  $F$  和  $CR$  参数。当利用当前的  $F$  和  $CR$  产生的新个体的适应度优于父代个体时,说明这两个参数是合适的,则无须对这两个参数进行调整。基于这种思想并借鉴单目标优化自适应参数控制方法<sup>[10]</sup>,将个体的适应度作为参数调整的依据,并结合一定的调整概率提出一种新的对  $F$  和  $CR$  参数自适应控制策略,以提高算法的搜索能力。在调整过程中加入了个体适应度的比较,目标是让缩放因子向当前种群中较好的参数方向取值。具体方法是:对种群个体采用不同的  $F$  和  $CR$ ,在演化过程中则按式(2)和(3)自动调整(以最小化目标函数为例)。

$$F_{i,G+1} = \begin{cases} \alpha \times F_{\text{best},G} + (1 - \alpha) \times \text{rand}_1 & \text{if } u_{i,G} < x_{i,G} \text{ and } \text{rand}_2 < \tau_1 \\ F_{i,G} & \text{otherwise} \end{cases} \quad (2)$$

$$CR_{i,G+1} = \begin{cases} \text{rand}_3 & \text{if } u_{i,G} < x_{i,G} \text{ and } \text{rand}_4 < \tau_2 \\ CR_{i,G} & \text{otherwise} \end{cases} \quad (3)$$

其中:  $\text{rand}_k, k \in \{1, 2, 3, 4\}$  是  $[0, 1]$  间的随机数;  $F_{\text{best},G}$  表示当前种群中非劣排序后第一个体对应的缩放因子;  $u_{i,G} < x_{i,G}$  表示实验个体劣于(非支配)父代个体。从式(2)和(3)可以看出,如果个体  $i$  用其对应的  $F_i$  和  $CR_i$  值生成的新个体的适应度优于该个体当前适应度,则表明该个体的缩放因子和交叉概率取值是有效的,保留该缩放因子和交叉概率到下一代。只有在  $u_{i,G}$  的适应度劣于  $x_{i,G}$  的适应度值,并且随机数小于调整概率  $\tau_1, \tau_2$  时,才生成新的缩放因子和交叉概率。 $\alpha$  是向  $F_{\text{best},G}$  的趋向率,由式(4)确定:

$$\alpha = \frac{1}{1 + e^{\frac{1}{M} \sum_{j=1}^M \frac{f_j(x_{\text{best},G}) - f_j(x_{i,G})}{f_j(x_{\text{best},G})}}} \quad (4)$$

其中:  $f_j(x_{\text{best},G})$  表示当前种群中非劣排序后的第一个体对应的第  $j$  个目标适应值,  $M$  为优化目标数。随着个体  $x_{i,G}$  与种群最优个体几何平均距离的增加,其对应参数  $F_{i,G+1}$  向  $F_{\text{best},G}$  的趋向率  $\alpha$  也越大,相应的随机扰动越小;当个体  $x_{i,G}$  与最优个体接近时,其对应参数的随机扰动就会增加。参数的自适应控制使种群中适应度较差的个体向着种群中的最优个体趋近,提高了算法的性能。基于自适应控制参数的通用差异演化算法(selfGDE3)描述如下:

- 设定种群规模  $NP$ 、最大演化代数  $G_{\max}$  及  $\tau_1$  和  $\tau_2$  的值;
- 随机初始化种群及种群中每个个体对应的参数  $F_i$  和  $CR_i, G=0$ ;
- $i=0$ ;
- 利用个体  $x_i(t)$  对应的参数  $F_i$  和  $CR_i$ , 按照 GDE3 算法的工作策略执行变异、交叉和选择操作,生成新一代个体  $x_{i,G+1}$ ;
- 按照式(4)计算趋向率  $\alpha$ ;
- 按照式(2)和(3)更新个体  $x_{i,G+1}$  对应的  $F_{i,G+1}$  和  $CR_{i,G+1}$ ;
- $i=i+1$ ;
- 如果  $i < NP$ , 转步骤 d);
- 通过快速非劣排序保持种群规模为  $NP$ ;
- 如果  $G < G_{\max}$ ,  $G=G+1$ , 转步骤 c) 继续执行;
- 输出当前种群即为 Pareto 最优解。

## 4 算法性能分析

### 4.1 性能指标

本文对 NSGA-II、GDE3 和 selfGDE3 算法求得的非劣解的宽广性  $M$ 、分散性  $SP$  和运行时间指标进行比较分析,  $M^{[11]}$  用来计算解集  $P$  中的所有极值解构成的多面体的周长,  $SP^{[12]}$  度量算法所求非劣解的多样性, 可通过非劣解的分布情况衡量, 即通过非劣解的间隔距离  $SP$  度量,  $M$  和  $SP$  指标分别由式(5)和(6)表示。

$$M = \sqrt{\sum_{k=1}^m (\max_i^n f_k^i - \min_i^n f_k^i)^2} \quad (5)$$

其中:  $n$  是非劣解的数目,  $m$  表示目标空间的维数。  $M$  值越大, 说明解集  $P$  的范围越广。

$$SP = \sqrt{\frac{1}{n} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (6)$$

其中:  $d_i = \min(|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|), i, j = 1, 2, \dots, n, \bar{d}$  是所有  $d_i$  的均值;  $n$  是非劣解的数目。若  $SP=0$ , 则表示非劣解前端上所有解呈均匀分布, 因此  $SP$  越小, 非劣解分布越均匀。

运行时间体现了算法的复杂度, NSGA-II 算法的复杂度为  $O(GkN^2)$ , GDE3 算法的复杂度为  $O(GN^2 \log^{k-1} N)$ 。其中:  $G$  为进化总代数;  $k$  为目标函数的个数;  $N$  为种群规模。对于 selfGDE3, 由于调整概率取值不大, 同时  $F$  和  $CR$  的调整原则比较简单, 与 GDE3 算法相比, 没有增加算法的时间复杂度。

### 4.2 实验结果分析

选用三个多目标检测函数  $F_1, F_2, F_3$ ,  $F_1, F_2$  包含两个目标函数和三维变量,  $F_3$  包含两个目标函数和二维变量, 其表达式如式(7)~(9)所示。算法迭代次数为 200, 种群规模为 200, NSGA-II、GDE3 算法中  $F$  和  $CR$  均取 0.8, 每种算法分别运行 50 次, 以消除随机性。

#### 1) 函数 $F_1$

$$\begin{cases} \min f_1(x) = \sum_{i=1}^2 (-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2})) \\ \min f_2(x) = \sum_{i=1}^3 (|x_i|^{0.8} + 5 \sin(x_i)^3) \end{cases} \quad (7)$$

s. t.  $-5 \leq x_1, x_2, x_3 \leq 5$

#### 2) 函数 $F_2$

$$\begin{cases} \min f_1(x) = 1 - \exp(-\sum_{i=1}^3 (x_i - \frac{1}{\sqrt{3}})^2) \\ \min f_2(x) = 1 - \exp(-\sum_{i=1}^3 (x_i + \frac{1}{\sqrt{3}})^2) \end{cases} \quad (8)$$

s. t.  $-4 \leq x_1, x_2, x_3 \leq 4$

#### 3) 函数 $F_3$

$$\begin{cases} \min f_1(x) = \sin(x_1^2 + x_2^2 - 1) \\ \min f_2(x) = \sin(x_1^2 + x_2^2 + 1) \end{cases} \quad (9)$$

s. t.  $0 \leq x_1, x_2 \leq 4\pi/3$

图 1~3 分别是三种算法对函数  $F_1, F_2$  和  $F_3$  所求的非劣解的分布图, 可以看出三种算法都能得到效果较好的非劣解。表 1~3 是三种算法对函数  $F_1, F_2$  和  $F_3$  的非劣解在宽广性、分散性和时间指标的统计结果。可以看出, selfGDE3 在  $F_1, F_2$  和  $F_3$  函数测试中的  $SP$  平均值都低于 NSGA-II 和 GDE3,  $F_1$  中  $M$  平均值高于 NSGA-II, 略低于 GDE3,  $F_2$  和  $F_3$  中  $M$  平均值都高于 NSGA-II 和 GDE3, 运行时间  $t$  在两个函数中均处于中间位置。由此可见, 在非劣解的分散性、均匀性上 selfGDE3 表现最好, 解的宽广性上 selfGDE3 和 GDE3 都较好, 算法的运行时间 NSGA-II 最少, selfGDE3 次之。这是由于  $F$  和  $CR$  的调整原则较简单, 与 GDE3 相比没有增加算法的时间复杂度, 同时 selfGDE3 使个体向适应度较好的染色体变异, 解的均匀性好, 减少了非支配排序所耗时间。

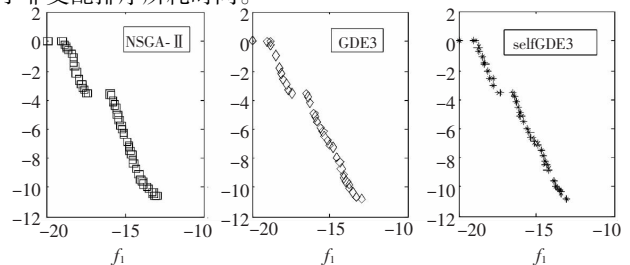


图 1 函数  $F_1$  非劣解分布

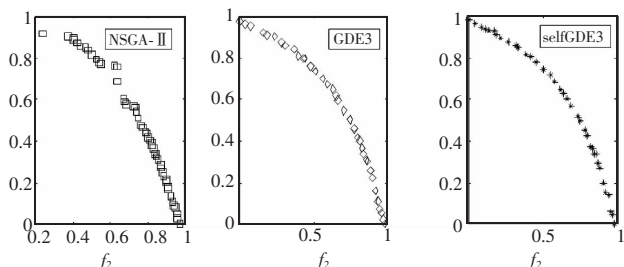


图 2 函数  $F_2$  非劣解分布

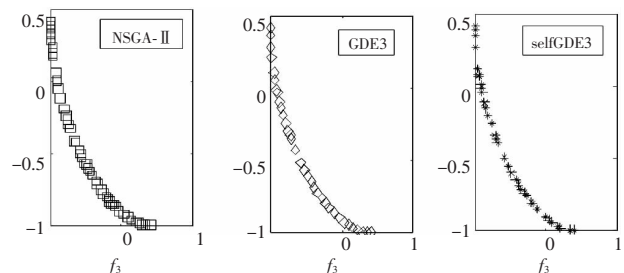


图 3 函数  $F_3$  非劣解分布

表 1 三种算法对函数  $F_1$  的性能

| $F_1$    | SP      |         |         | M        |          |          | t        |          |          |
|----------|---------|---------|---------|----------|----------|----------|----------|----------|----------|
|          | best    | mean    | worst   | best     | mean     | worst    | best     | mean     | worst    |
| NSGA-II  | 0.018 5 | 0.055 2 | 0.089 2 | 13.092 8 | 12.676 5 | 8.886 4  | 16.265 0 | 17.717 2 | 20.875 0 |
| GDE3     | 0.038 6 | 0.043 4 | 0.082 1 | 12.876 4 | 12.830 5 | 12.796 3 | 43.860 0 | 49.540 2 | 58.125 0 |
| selfGDE3 | 0.036 3 | 0.042 9 | 0.079 9 | 12.842 4 | 12.827 0 | 12.802 4 | 32.235 0 | 41.129 5 | 54.718 0 |

表 2 三种算法对函数  $F_2$  的性能

| $F_2$    | SP      |         |         | M       |         |         | t        |          |          |
|----------|---------|---------|---------|---------|---------|---------|----------|----------|----------|
|          | best    | mean    | worst   | best    | mean    | worst   | best     | mean     | worst    |
| NSGA-II  | 0.004 6 | 0.005 2 | 0.005 6 | 1.388 4 | 1.387 5 | 1.386 1 | 15.281 0 | 16.951 8 | 19.515 0 |
| GDE3     | 0.004 5 | 0.005 0 | 0.005 5 | 1.388 9 | 1.388 0 | 1.387 0 | 35.484 0 | 36.629 5 | 41.078 0 |
| selfGDE3 | 0.002 8 | 0.003 3 | 0.004 0 | 1.389 0 | 1.388 2 | 1.387 3 | 31.797 0 | 36.031 5 | 41.125 0 |

表 3 三种算法对函数  $F_3$  的性能

| $F_3$    | SP      |         |         | M       |         |         | t        |          |          |
|----------|---------|---------|---------|---------|---------|---------|----------|----------|----------|
|          | best    | mean    | worst   | best    | mean    | worst   | best     | mean     | worst    |
| NSGA-II  | 0.026 9 | 0.033 4 | 0.051 2 | 1.416 0 | 1.415 1 | 1.414 3 | 15.875 0 | 16.563 2 | 17.750 0 |
| GDE3     | 0.003 3 | 0.004 3 | 0.005 0 | 1.416 3 | 1.416 1 | 1.415 9 | 39.016 0 | 42.920 9 | 44.157 0 |
| selfGDE3 | 0.003 0 | 0.003 5 | 0.005 0 | 1.677 3 | 1.639 3 | 1.563 9 | 38.484 0 | 40.200 7 | 49.172 0 |

## 5 结束语

最优化问题是工程实践和科学研究中主要的问题形式之一,其中许多优化问题都属于多目标优化,多目标差异演化算法是目前优化问题研究领域的热点之一。本文基于多目标优化的差异演化算法理论和单目标优化参数自适应控制方法,系统研究了通用差异演化算法的原理和流程,针对算法中控制参数固定不变的缺陷,提出了面向个体适应度的参数自适应调整

方法,使缩放因子向当前种群中较好的参数方向取值,提高算法的搜索能力。通过对典型函数的测试表明,相对于 NSGA-II 和 GDE3 算法,本文的 selfGDE3 算法可以获得比较广泛和均匀的非劣解,同时算法运行时间也较少。进一步工作是将 selfGDE3 算法运用到实际复杂的多目标优化问题。

## 参考文献:

- [1] LAMPINEN J. DE's selection rule for multiobjective optimization [EB/OL]. (2001-07-22). <http://www.it.lut.fi/kurssit/03-04/010778000/MODE.pdf>.
- [2] KUKKONEN S, LAMPINEN J. An extension of generalized differential evolution for multi-objective optimization with constraints [C]//Proc of the 8th International Conference on Parallel Problem Solving from Nature. 2004;752-761.
- [3] KUKKONEN S, LAMPINEN J. GDE3: the third evolution step in generalized differential evolution [C]//Proc of Congress on Evolutionary Computation. 2005;443-450.
- [4] TEO J. Exploring dynamic self-adaptive populations evolution in differential [J]. Soft Computing: A Fusion of Foundations, Methodologies and Applications, 2006, 10(8): 673-686.
- [5] 马小姝, 李宇龙. 传统多目标优化方法和多目标遗传算法的比较综述 [J]. 电气传动自动化, 2010, 32(3): 48-53.
- [6] RAINER S, KENNEDY J. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous space [R]. Berkeley: University of California, 1995.
- [7] 程思微, 张辉, 沈林成, 等. 基于 GDE3 算法的侦察卫星星座优化设计 [J]. 系统仿真学报, 2009, 21(2): 586-589.
- [8] SWAGATAM D, AMIT K. Two improved differential evolution schemes for faster global search [C]// Proc of Conference on Genetic and Evolutionary Computation. 2005;991-998.
- [9] BREST J, BOSKOVIC B, GREINER S. Performance comparison of self-adaptive and adaptive differential evolution algorithms [J]. Soft Computing: A Fusion of Foundations, Methodologies and Applications, 2007, 11(7): 617-629.
- [10] 武志峰. 差异演化算法及其应用研究 [D]. 北京: 北京交通大学, 2009.
- [11] DEB K. Multi-objective optimization using evolutionary algorithms [M]. Hoboken: Wiley, 2001.
- [12] SCHOTT J. Fault tolerant design using single and multicriteria genetic algorithm optimization [D]. Cambridge: Massachusetts Institute of Technology, 1995.