一种改进的基于差分进化的多目标进化算法

李 珂,郑金华
LI Ke,ZHENG Jin-hua

湘潭大学 信息工程学院,湖南 湘潭 411105
Institute of Information Engineering,Xiangtan University,Xiangtan,Hunan 411105,China
E-mail:jhzheng@xtu.edu.cn

LI Ke,ZHENG Jin-hua.Improved multi-objective evolutionary algorithm based on differential evolution.Computer Engineering and Applications,2008,44(29):51-56.

Abstract: Recently,the use of evolutionary algorithms(EAs) to solve the Multi-objective Optimization Problems(MOPs) has attracted much attention.EA is a population based optimized approach which can find a group of Pareto-optimal solutions in a single run.Differential Evolution(DE) is a branch of EA that is developed to handle problems over continuous domains.An improved Multi-objective Evolutionary Algorithm is proposed based on Differential Evolution(CDE) to solve MOPs.The proposed algorithm is compared to the other two classical Multi-objective Evolutionary algorithms(MOEAs) NSGA-II and SPEA2 with the experiment results.
Key words: multi-objective optimization;differential evolution;Multi-objective Optimization(CDE)

摘 要:近年来运用进化算法(EAs)解决多目标优化问题(Multi-objective Optimization Problems MOPs)引起了各国学者们的关注。作为一种基于种群的优化方法,EAs 提供了一种在一次运行后得到一组优化的解的方法。差分进化(DE)算法是 EA 的一个分支,最开始是用来解决连续函数空间的问题。提出了一种改进的基于差分进化的多目标进化算法(CDE),并且将它与另外两个经典的多目标进化算法(MOEAs)NSGA-II 和 SPEA2 进行了对比实验。
关键词:多目标优化;差分进化;多目标进化算法(CDE)
DOI:10.3778/j.issn.1002-8331.2008.29.014 **文章编号**:1002-8331(2008)29-0051-06 **文献标识码**:A **中图分类号**:TP18

1 引言

在实际应用中,人们经常遇到需要使用多个目标在给定可行区域上做尽可能最优的决策的问题。比方说在设计一种新型产品时,既要考虑产品的使用性能,又要考虑产品制造和生产的成本,另外包括产品的制造可行性、产品的使用可靠性、产品的可维护性等。诸如这些产品设计目标的改善可能相互冲突,譬如使用性能的提升将有可能导致产品的制造可行性的降低,因而必须在这些设计目标之间取得折衷和平衡。这种多个数值目标在给定区域上的最优化问题便是多目标优化问题(Multi-objective Optimization Problems,MOPs)。

与单目标优化问题不同,在多目标优化问题中,得到的并不是一个单一的最优解向量。因为对于 MOPs 来说各个优化指标之间是相互冲突的,一个解不可能对于所有指标同时达到最优。因此 MOPs 寻求的就是一个最优解的集合。而对于实际应用问题的最后方案决策,必须根据对问题的了解程度和决策人员的偏好,从多目标优化问题的最优解集合中挑选出一个或部分解作为所求的多目标问题的最优解。那么我们自然可以知道求解多目标优化问题的目标是:(1)尽可能地逼近最优解集;(2)使得最优解集中的解尽可能多。

最近大量的事例和迹象表明进化算法(EAs)的机理很适合求解 MOPs。而近年来,对于多目标进化算法(MOEAs)的研究引起了各国学者的兴趣,而期间提出了不少高效的 MOEAs 的改进算法。其中比较有名的包括:Zitziler 和 Thiele 提出的 SPEA2[2],Deb 等人提出的 NSGA-II[1],Knowles 和 Corne 提出的 PAES[3],Corne 等人提出的 PESA[4]等。而在国内,近年来也有不少有关多目标进化算法的专著出版[5-6],这为多目标进化算法的应用与研究提供了很大的帮助。

尽管 EAs 取得了很大的成功,但是在某种程度上由于受各自算法背景的影响,可以说还没有一种算法在解决所有的 MOPs 时都能得到最优的解集。这在某种程度上激励着广大的研究者们不断地对 MOEAs 进行改进,研究出更好的解决 MOPs 的方法来。

文章提出了一种新的改进的基于差分进化(Differential Evolution)的多目标进化算法,通过一系列的测试函数对其解决 MOPs 的能力进行了测试,并且将其与两个经典算法SPEA2,NSGA-II 进行了对比实验。

2 多目标进化算法中的相关定义

这一章将给出一些与本文工作相关的基本定义。

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60773047);湖南省教育厅重点科研项目(No.06A074)。
作者简介:李珂(1985-),男,硕士研究生,研究方向为进化计算;郑金华(1963-),教授,博士生导师,主要研究方向为进化计算,并行处理等。
收稿日期:2008-04-11 **修回日期**:2008-07-18

定义 1（多目标优化问题(MOPs)）一般 MOPs 由 n 个决策变量参数、k 个目标函数和 m 个约束条件组成，目标函数、约束条件与决策变量之间是函数关系，其形式如下所示：

$$\begin{aligned}&\max\ mization\quad \boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x})=(f_1(x),f_2(x),\cdots,f_k(x))\\&\text{subject to}\quad \boldsymbol{g}(\boldsymbol{x})=(g_1(x),g_2(x),\cdots,g_m(x))\leqslant 0\\&\text{其中}\quad \boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in\boldsymbol{X}\\&\qquad\ \boldsymbol{y}=(y_1,y_2,\cdots,y_m)\in\boldsymbol{Y}\end{aligned}\tag{1}$$

这里 $\boldsymbol{x}$ 表示决策向量，$\boldsymbol{y}$ 表示目标向量，$\boldsymbol{X}$ 表示决策向量 $\boldsymbol{x}$ 形成的决策空间，$\boldsymbol{Y}$ 表示目标向量 $\boldsymbol{y}$ 组成的目标空间，约束条件 $g(\boldsymbol{x})\leqslant 0$ 确定决策向量的可行取值范围。

定义 2（Pareto 支配）向量 $\boldsymbol{u}=(u_1,\cdots,u_k)$ 支配 $\boldsymbol{v}=(v_1,\cdots,v_k)$ 当且仅当满足关系：

$$\forall i\in(1,\cdots,k),u_i\leqslant v_i\wedge\exists i\in(1,\cdots,k)\text{有}\ u_i<v_i,\text{记作}\ \boldsymbol{u}\prec\boldsymbol{v}\tag{2}$$

定义 3（Pareto 最优解）向量 $\boldsymbol{x}_u\in F$（F 表示可行解域）是 Pareto 最优解当且仅当满足条件：

$$\nexists\,\boldsymbol{x}_v\in F\text{ 且 }\boldsymbol{x}_v\prec\boldsymbol{x}_u\tag{3}$$

定义 4（Pareto 前端）对于一个给定的多目标优化问题(MOPs)$\boldsymbol{f}(x)$和 *Pareto* 最优解集 $\boldsymbol{P}^*$，Pareto 前端(FP^*)定义如下：

$$FP^*\{\boldsymbol{f}=[f_1(x),\cdots,f_k(x)]|x\in P^*\}\tag{4}$$

定义 5（局部 Pareto 最优集）决策向量集 $\boldsymbol{A}\in\boldsymbol{F}$（$\boldsymbol{F}$ 表示可行解域）是局部 Pareto 最优集当且仅当满足：

$$\forall\boldsymbol{x}_u\in\boldsymbol{A}:\nexists\,\boldsymbol{x}_v\in\boldsymbol{F}:\boldsymbol{x}_v\prec\boldsymbol{x}_u\wedge\|\boldsymbol{x}_v-\boldsymbol{x}_u\|<\varepsilon\wedge\|f(\boldsymbol{x}_v)-f(\boldsymbol{x}_u)\|<\delta\tag{5}$$

其中 $\|\cdot\|$ 是相关距离的摸，并且 $\varepsilon>0,\delta>0$。

定义 6（全局 Pareto 最优集）决策向量集 $\boldsymbol{A}\in F$（F 表示可行解域）是全局 *Pareto* 最优集当且仅当满足：

$$\forall\boldsymbol{x}_u\in A:\nexists\,\boldsymbol{x}_v\in F:\boldsymbol{x}_v\prec\boldsymbol{x}_u\tag{6}$$

3 差分进化(Differential Evolution)算法

差分进化(DE)算法是进化算法的一个分支，它最早是由 R.Storn 和 K.Price 于 1995 年在解决连续区域的函数优化问题中提出的。在 DE 中，每一个解向量的值在染色体中被表示成一个实数向量。与传统的进化算法一样，DE 算法在初始化时先随机产生一个初始种群，并通过修补机制以保证每一个个体落在它们的约束范围内。再从种群中随机选出一个用来做比较替换的标志向量，然后另外又选择三个不同的个体向量作为母体向量，其中在这三个向量中有一个被选作基向量。在一定的概率下，基向量的值会通过加上另外两个向量差值的一定倍率而发生改变，从而产生一个被称为试验向量的新个体。实际上这个产生试验向量的过程可以看成是基向量在受到另外两个向量的扰动过程。如果产生的试验向量要优于选出的用来做比较替换的标志向量，那么便用该试验向量替换该标志向量，否则就保留该标志向量在种群中不做改变。如此循环，直到满足终止条件。

由上述可知，DE 与传统的进化算法有着以下的三点不同：

(1)DE 算法中都是使用实数编码，而传统的进化算法多以二进制编码为主，但是近期的进化算法也慢慢改用实数编码作为个体编码方式；

(2)在进化算法中，通常是选择两个个体进行交叉操作重组产生子代，而在 DE 算法中，子代的产生是通过三个母体的扰动作用而产生；

(3)在 DE 算法中新产生的个体保留与否是通过与被选出来进行比较替换的标志向量进行优劣对比而决定的。

DE 算法进化操作的过程可以用下面的图 1 来表示。

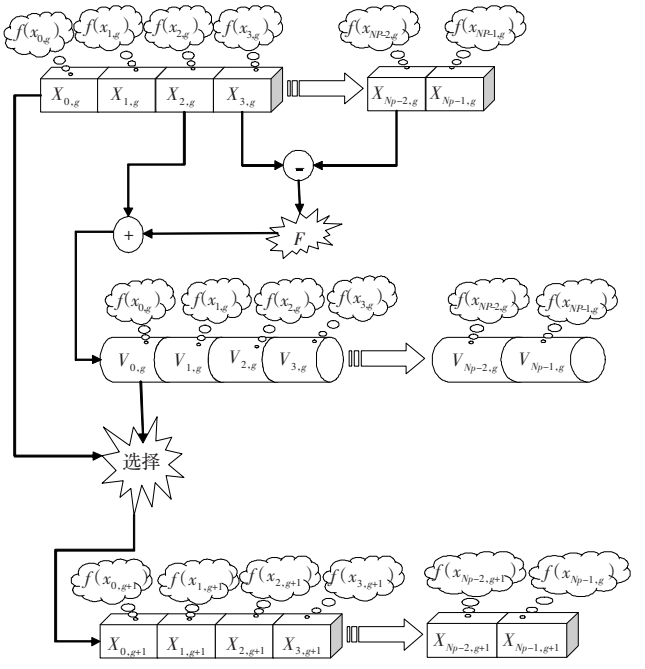


图 1 差分进化的进化操作示意图

在 DE 算法中，群体的初始化操作的是通过下面的表达式(7)进行的：

$$x_{i,G=0}^{j}=lower(x_i)+rand_i[0,1]\times(upper(x_i)-lower(x_i))\tag{7}$$

而 DE 算法的核心操作便是产生试验向量的操作，这个操作是依一定的概率根据表达式(8)进行的：

$$u_{i,G=g}^{j}=x_{i,G=g-1}^{r3}+F\times(x_{i,G=g-1}^{r1}-x_{i,G=g-1}^{r2})\tag{8}$$

随着 DE 算法的发展，逐渐吸引越来越多的学者对差分进化算法进行研究并不断改进，而且大量的将其应用于解决 MOPs 中去。其中比较成功的算法包括：H.Abbass 等人首先将所谓的 Pareto 支配的概念与 DE 算法结合提出了 PDE 算法[14]，S.Kukknonen 等人将约束处理机制引入 DE 算法中提出了 GDE 算法[10]，L.V.Santana-Quintero 等人将 ε 支配的概念引入 DE 算法中提出了 ε-MyDE 算法[16]，T.Robic 等人对前人提出的 DE 算法进行了扩展，将非支配排序和拥挤距离的概念与 DE 算法结合提出了 DEMO 算法[17]。

4 基于拥挤距离的差分进化算法(CDE)

提出的算法伪代码如下：

输入：$D,G_{max},NP\geqslant 4,CR\in[0,1],F\in(0,1+],q=0,MUT=1/NP$

初始变量边界：$lower(x_i),upper(x_i)$

初始化第 0 代种群 $P_{G=0}=\{\boldsymbol{x}_{G=0}^{1},\cdots,\boldsymbol{x}_{G=0}^{NP}\}$ 如下

for each 个体 $j\in P_{G=0}$

$x_{i,G=0}^{j}=lower(x_i)+rand_i[0,1]\times(upper(x_i)-lower(x_i))(i=1,\cdots,D)$

end for each

评价第 0 代种群 $P_{G=0}$，$g=1$

while $g<G_{max}$

forall $j\leqslant NP$

随机挑选 $r_1,r_2,r_3\in(1,\cdots,NP)$，

并且 $j\neq r_1\neq r_2\neq r_3$

生成一个随机数 $i_{rand}\in(1,\cdots,D)$

forall $i\leqslant D, u^{j}_{i,G=g}$

$$\begin{cases} x^{r3}_{i,G=g-1}+F\times(x^{r1}_{i,G=g-1}-x^{r2}_{i,G=g-1}) & \text{if } (rand[0,1)<CR\wedge i=i_{rand}) \\ x^{j}_{i,G=g-1} & \text{otherwise} \end{cases}$$

$$u^{j}_{i,G=g}=\begin{cases} lower(x_i)+rand[0,1)\times(upper(x_i)-lower(x_i)) & \text{if} \\ \quad (rand[0,1)<MUT) & \\ u^{j}_{i,G=g} & \text{otherwise} \end{cases}$$

如果产生的 $u^{j}_{i,G=g}$ 超出了变量范围则利用修补规则进行修补

end forall

找出与 $u^{j}_{G=g}$ 最邻近的个体 $\boldsymbol{x}^{*}_{G=g}$

$$\boldsymbol{x}^{j}_{G=g+1}=\begin{cases} \boldsymbol{u}^{j}_{G=g} & \text{if } \boldsymbol{u}^{j}_{G=g}<\boldsymbol{x}^{*}_{G=g} \\ \boldsymbol{x}^{j}_{G=g} & \text{otherwise} \end{cases}$$

$$\text{if}(\boldsymbol{x}^{j}_{G=g+1}=\boldsymbol{x}^{j}_{G=g}\wedge\boldsymbol{x}^{j}_{G=g}\neq\boldsymbol{u}^{j}_{G=g})\quad\begin{cases} q=q+1 \\ \boldsymbol{x}'_{p,G+1}=\boldsymbol{u}_{j,G} \end{cases}$$

end forall

将种群 $\boldsymbol{x}$ 与 $\boldsymbol{x}'$ 种群合并成新种群对,新种群进行非支配排序并计算其中个体的拥

挤距离,根据支配关系和拥挤距离对新种群进行修剪直到与初始种群大小相同。

$g=g+1$

end while

其中 CR 和 F 是在 DE 算法中用户自己定义的控制参数。CR 控制着交叉的概率,F 是对差分结果的加权。在算法的运行过程中参数 CR 控制着进化操作发生的机率,而参数 F 控制着整个算法运行的速度和鲁棒性,通过实验表明一个较小的 F 值可以加快收敛速度,但是同样这样容易陷入局部最优解[11]。另外还有一个在 CDE 算法中加入的参数 MUT,MUT 控制着变异的概率,在这里将 MUT 设为种群大小的倒数。

在通常的 DE 算法中产生的试验向量是与标志向量比较优劣关系,而在本算法中试验向量是与它最邻近的解向量比较支配关系,如果该试验向量支配与它最邻近的个体,则用该试验向量替换与它最邻近的个体,反之则保留标志向量进入下一代进化种群。另外如果标志向量与该试验向量互不支配,则将该试验向量加入一个临时种群。在一代进化操作运行完成之后,将临时种群与新产生的种群混合组成一个混合种群。根据上述的个体保留机制可知混合种群的大小有可能会扩大,利用对非支配个体的排序和个体的拥挤距离对该混合种群进行修剪直至混合种群大小恢复到原始种群的大小。另外在传统的 DE 算法中并没有单独使用变异操作,变异操作被视为交叉过程中的扰动而整合到产生试验向量的步骤中。但是在进化算法中变异操作却是不可或缺的一个步骤,对此尝试性地设计了一个变异算子将其加入进化过程中,变异操作是依据表达式(9)进行的:

$$lower(x_i)+rand[0,1)\times(upper(x_i)-lower(x_i)) \tag{9}$$

产生试验向量的过程有可能使得其超过变量的范围,这时采用了修补算子以使该试验向量恢复到设定的变量范围之内。该修补操作是依据表达式(10)进行的:

$$x^{*}_{i,G=g}=\begin{cases} upper(x_i)+(x^{j}_{i,G}-lower(x_i))/2 & \text{if } x^{j}_{i,G+1}<lower(x_i) \\ lower(x_i)+(x^{j}_{i,G}-upper(x_i))/2 & \text{if } x^{j}_{i,G+1}>upper(x_i) \\ x^{j}_{i,G+1} & \text{otherwise} \end{cases} \tag{10}$$

5　测试函数与评价方法

为了检验 CDE 算法的效果,对上述 CDE 算法分别在目标为二维和三维的情况下进行测试,并将其实验结果与两个经典算法 NSGA-II 和 SPEA2 进行了比较。

5.1　参数设置

三个算法均采用实数编码,NSGA-II 和 SPEA2 使用的交叉概率是 Pc=0.8 变异概率为 1/n(n 变量个数),而使用的交叉算子是模拟二进制交叉(SBX),使用的变异算子是多项式变异,其中 η_c=15,η_m=20。而在 DE 算法中 CR=0.95,F=0.4,MUT=1/NP。

5.2　使用的测试函数

实验使用到了 8 个测试函数[23-25],其中 7 个是二维目标,一个是三维目标的。对它们使用的参数设置如表 1。

表 1　测试函数参数设置表

测试函数	SCH	FON	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ-2
变量个数	1	3	30	30	30	10	10	10
种群大小	200	200	200	200	200	200	200	300
进化代数	300	300	300	300	300	300	300	500

5.3　评价方法

(1)Generation Distance(GD)[21]:用来估计算法的最终解集与全局非劣最优区域的接近程度,其表达式如下:

$$GD=\frac{\sqrt{\sum_{i=1}^{n}d_i^2}}{n} \tag{11}$$

n 是解集中个体的数目,d_i 是每个个体到全局非劣最优解的最小欧几里德距离。GD 的值越小就说明解集越靠近全局非劣最优区域。

(2)Spacing(SP)[22]:该方法通过计算解集中每个个体与邻居个体的距离变化来评价解集在目标空间的分布情况,其表达式如下:

$$SP=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d}-d_i)^2} \tag{12}$$

其中 $d_i=\min_j(|f^{j}_{1}(x)-f^{i}_{1}(x)|+|f^{j}_{2}(x)-f^{i}_{2}(x)|)$,$i,j=1,\cdots,n$。$n$ 是解集中个体的数目,$\bar{d}$ 是所有 d_i 的平均值。如果 SP=0 说明解集中所有个体之间的距离都相等且分布均匀,SP 的值越小说明解集分布越均匀。

6　实验结果

6.1　实验结果数据统计

算法共对 7 个二维测试函数和 1 个三维测试函数进行了测试,表 2 分别统计了三个 MOEAs 所得的解集的分布性和收敛性的评价值以及方差。其中在各项指标中表现最优的数据用加粗的字体标示。

6.2　实验结果

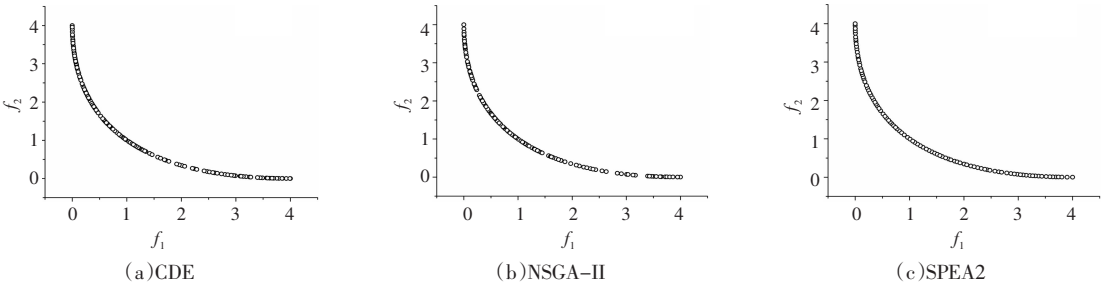
三个 MOEAs 对 8 个测试函数分别进行测试所得的最优边界如图 2 到图 9 所示。

表 2 三个 MOEAs 在 8 个测试函数上的评价值的比较表

测试函数	MOEAs	*GD*		*SP*	
		平均值	方差	平均值	方差
SCH	CDC	**0.000 271 33**	1.382E−005	**0.012 852 7**	0.0 006 038
	NSGA−II	0.000 425 70	1.654E−005	0.033 132 4	0.0 068 416
	SPEA2	0.000 391 07	1.934E−005	0.013 970 4	0.0 010 885
FON	CDC	0.000 281 17	1.653E−005	**0.003 124 4**	0.0 002 149
	NSGA−II	**0.000 161 12**	1.109E−005	0.006 098 8	0.0 002 661
	SPEA2	0.000 220 54	6.731E−006	0.003 245 8	0.0 002 112
ZDT1	CDC	0.000 271 25	1.587E−005	**0.002 578 9**	0.0 001 386
	NSGA−II	**0.000 188 56**	5.222E−005	0.007 804 4	0.0 007 836
	SPEA2	0.000 263 98	2.597W−005	0.003 028 2	0.0 001 807
ZDT2	CDC	0.000 110 40	3.856E−006	**0.002 999 4**	0.0 001 594
	NSGA−II	**0.000 107 14**	3.891E−005	0.007 705 6	0.0 015 039
	SPEA2	0.000 100 92	3.715E−006	0.003 155 8	0.0 002 961
ZDT3	CDC	**0.000 459 41**	2.612E−005	**0.003 139 1**	0.0 017 738
	NSGA−II	0.000 580 64	2.659E−005	0.008 397 6	0.0 004 040
	SPEA2	0.000 816 88	0.000 265 71	0.005 112 8	0.0 013 548
ZDT4	CDC	**0.000 273 54**	0.021 766 54	0.006 300 7	0.0 002 647
	NSGA−II	0.044 080 61	0.028 760 59	0.011 337 0	0.0 003 647
	SPEA2	0.075 771 20	0.059 578 41	**0.003 711 6**	0.0 001 929
ZDT6	CDC	**0.000 421 44**	1.687E−005	0.005 746 2	0.0 021 386
	NSGA−II	0.053 068 81	0.034 951 59	0.006 343 4	0.0 021 021
	SPEA2	0.000 562 05	7.473E−006	**0.001 827 4**	0.0 001 183
DTLZ−2	CDC	0.000 277 93	2.138E−005	0.028 518 7	0.0 025 607
	NSGA−II	0.000 107 99	3.456E−005	0.057 380 2	0.0 046 389
	SPEA2	**5.064E−005**	4.774E−005	**0.023 271 6**	0.0 015 607

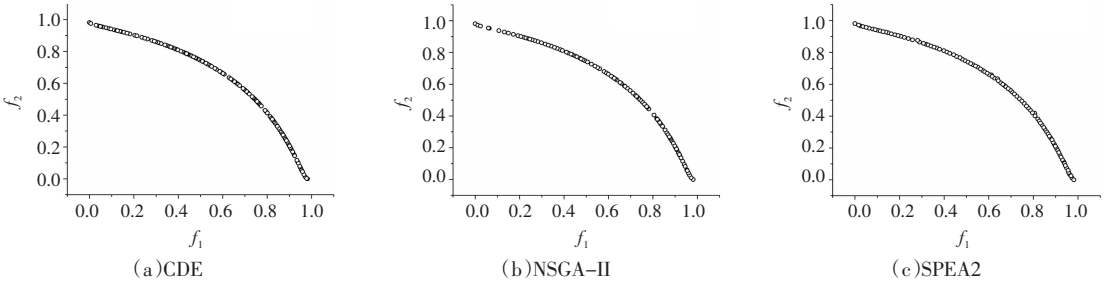
6.3 实验结果分析

SCH 测试问题的 PF_{true} 是分布于每维[0,4]之间的一段圆弧。如图 2 所示,3 个 MOEAs 所得的解集都覆盖了整个最优边界,CDE 的分布性比 NSGA−II 要理想,但是还是没有 SPEA2的解集分布的均匀。而表 2 的统计数据显示 CDE 在这个测试问题上的 *GD* 和 *SP* 值都是排第一的。FON 的 Pareto 面是一个凹平面,从图 3 可以看出 CDE 所得的解集的分布性更为均匀和平滑,而 NSGA−II 在整个最优边界上的解丢失的情况比较严重,而从表 2 的统计数据看出 CDE 在该测试问题上的 *GD* 值排第二,*SP* 值排第一。ZDT1,ZDT2,ZDT3 都是 30 个变量的测试问题,其中 ZDT3 是非连续问题,3 个 MOEAs 在分布性的方面表现相差不大,但是从图 4 到图 6 可以看出 CDE 所得的解集分布更为平滑,而比较表 2 的统计数据可以看到 CDE 在 *SP* 值上都是排第一,并且在 ZDT3 测试问题上 *GD* 值排第一。ZDT4 是一个比较难的测试问题,它有 21^9 个局部最优点,故通常 MOEAs 在该测试问题上容易陷入局部最优而难以收敛,由图 7 看出 CDE 不仅在该问题上很好地收敛,而且最优解集的分布也很均匀,而 SPEA2 却未能收敛到最优边界,表 2 的统计数据中同样 CDE 在该测试问题上的 *GD* 值是最好的。ZDT6 的 Pareto 面是非均匀分布的,因而通常 MOEAs 在该测试问题上也比较难收敛,从图 8 可以看出虽然三个 MOEAs 均收敛到了最优边界,但是在解集的分布上 CDE 要比 NSGA−II 和 SPEA2 都要均匀和平滑,而表 2 的统计数据表名 CDE 的收敛性是最好的。DTLZ2



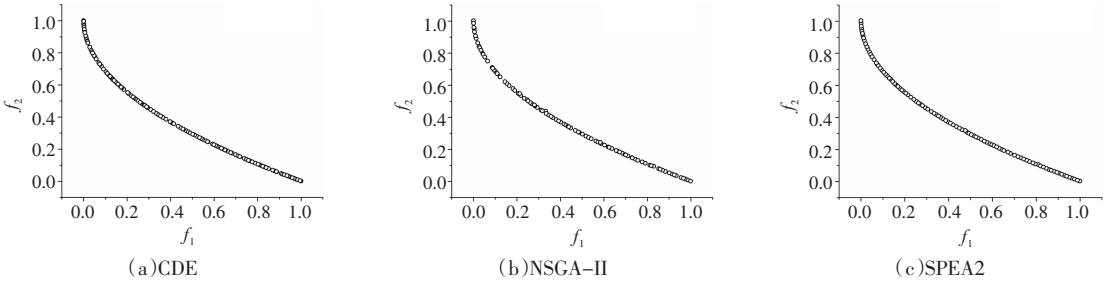
(a)CDE (b)NSGA−II (c)SPEA2

图 2 SCH



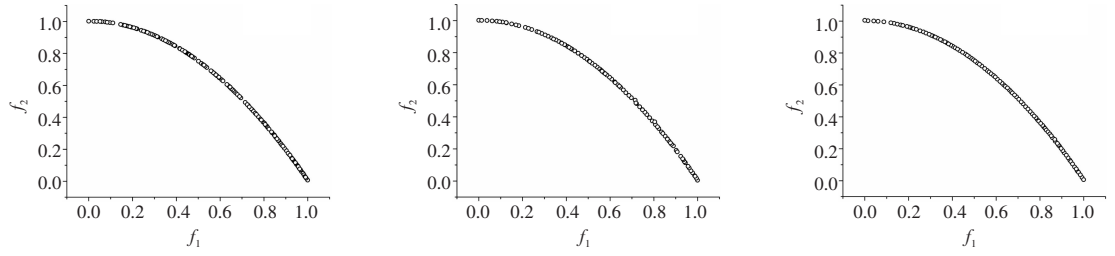
(a)CDE (b)NSGA−II (c)SPEA2

图 3 FON



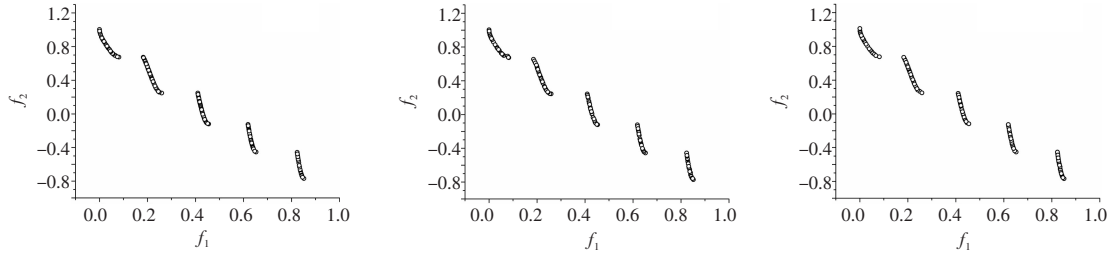
(a)CDE (b)NSGA−II (c)SPEA2

图 4 ZDT1



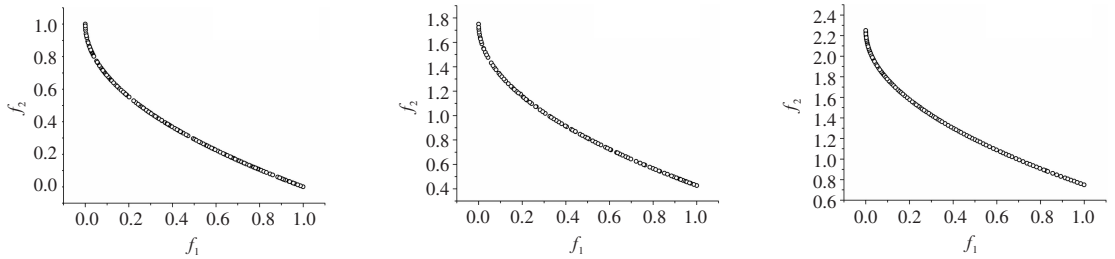
(a)CDE　(b)NSGA-II　(c)SPEA2

图 5　ZDT2



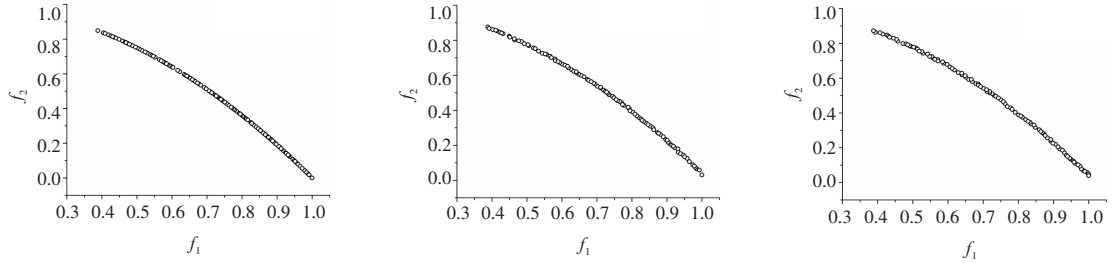
(a)CDE　(b)NSGA-II　(c)SPEA2

图 6　ZDT3



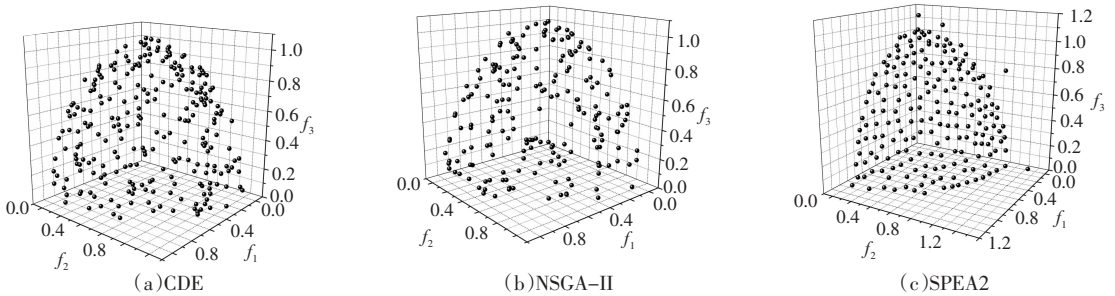
(a)CDE　(b)NSGA-II　(c)SPEA2

图 7　ZDT4



(a)CDE　(b)NSGA-II　(c)SPEA2

图 8　ZDT6



(a)CDE　(b)NSGA-II　(c)SPEA2

图 9　DLTZ-2

的 PF_{true} 为满足 $\sum_{i=1}^{M}(f_i)^2=1$ 的超球面。NSGA-II 在这一问题上所得的解集分布极不均匀,而且个体重叠现象较为严重,所以它的分布性很不理想。CDE 所得的解集的分布也比较林乱,但是所得的解的数目较 NSGA-II 要多,SPEA2 的得益于它的修剪算子,在该测试问题上所得的解集的分布比另外两个算法都要均匀,表 2 的统计数据同样表明 CDE 在 GD 和 SP 值上都不占优势。

7 总结

本文提出了一种改进的基于差分进化的多目标进化算法(CDE),它将非支配排序和拥挤距离的思想与差分进化算法结合,并且将变异算子引入差分算法中,以期能提高算法的收敛性。通过对 8 个测试函数进行对比实验可以看出 CDE 算法在二维测试问题上的表现比较出色,但是在三维测试问题上的表现却不如人意。

今后的研究工作将针对 CDE 在三维测试问题上的不足表现进行改进,并且在实验过程中发现 CDE 的表现与该算法中的几个控制参数 *CR* 和 *F* 的设置有一定关系,能够自适应地调整参数以得到好的实验结果也是今后研究工作的一个重点。

参考文献:

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

[2] Zitzler E,Laumanns M,Thiele L.SPEA2:improving the strength pareto evolutionary algorithm.Technical Report 103 [R].Computer Engineering and Networks Laboratory(TIK),ETH Zurich,Zurich,Switzerland,2001.

[3] Knowles J D,Corne D W.The Pareto archived evolution strategy:A new baseline algorithm for Pareto multiobjective optimization[C]//CEC'99:Proceedings of the 1999 Congress on Evolutionary Computation.Piscataway,New Jersey:IEEE Service Center,1999:98-105.

[4] Corne D W.The pareto envelope based selection algorithm for multiobjective optimization[C]//Schoenauer M.LNCS:Proc Parallel Problem Solving from Nature,PPSN VI,2000,1917:839-848.

[5] 郑金华.多目标进化算法及其应用[M].北京:科学出版社,2007.

[6] 崔逊学.多目标进化算法及其应用[M].北京:国防工业出版社,2006.

[7] Edgeworth F Y.Mathematical Psychics [M].London,England:P Keagan,1881.

[8] Pareto V.Cours D'Economie Politique:volume I and II.F.Rouge,Lausanne,1896.

[9] Kukkonen S,Lampinen J.Comparison of generalized differential evolution algorithm to other multiobjective evolutionary algorithms[C]//Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering(ECCOMAS2004),Jyv¨askyl¨a,Finland,July 2004:445.

[10] Kukkonen S,Lampinen J.An extension of generalized differential evolution for multi-objective optimization with constraints[C]//Proceedings of the 8th International Conference on Parallel Problem Solving from Nature(PPSN VIII),Birmingham,Finland,Sept 2004:752-761.

[11] Kukkonen S,Lampinen J.GDE3:The third evolution step of Generalized Differential Evolution[C]//Proceedings of the 2005 Congress on Evolutionary Computation(CEC2005),Edinburgh,Scotland,Sept 2005:443-450.

[12] Price K V.New ideas in optimization[M]//An Introduction to Differential Evolution.London:McGraw-Hill,1999:79-108.

[13] Price K V,Storn R,Lampinen J.Differential evolution:a practical approach to global op-timization[M].Berlin:Springer-Verlag,2005.

[14] Abbass H A,Sarker R,Newton C.PDE:a Pareto-frontier Differential Evolution approach for multi-objective optimization problems[C]//Proceedings of the 2001 Congress on Evolutionary Computation(CEC 2001),Seoul,South Korea,2001:971-978.

[15] Abbass H A.The self-adaptive Pareto differential evolution algorithm[C]//Proceedings of the 2002 Congress on Evolutionary Computation(CEC 2002),Honolulu,Hawaii,May 2002:831-836.

[16] Santana-Quintero L V,Coello Coello C A.An algorithm based on differential evolution for multi-objective problems[J].International Journal of Computational Intelligence Research,2005,1(2):151-169.

[17] Robi˘c T,Filipi˘c B.DEMO:differential evolution for multiobjective optimization[C]//Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization(EMO 2005),Guanajuato,Mexico,March 2005:520-533.

[18] Babu B V,Jehan M M L.Differential evolution for multi-objective optimization[C]//Proceedings of the 2003 Congress on Evolutionary Computation(CEC'2003),Volume 4,Canberra,Australia,December 2003.IEEE Press,2003,4:2696-2703.

[19] Zitzler E,Thiele L.Multiobjective optimization using evolutionary algorithms-a comparative Study[C]//Eiben A E.Parallel Problem Solving from Nature V.Amsterdam:Springer-Verlag,1998:292-310.

[20] Zitzler E,Laumanns M,Thiele L.SPEA2:Improving the strength Pareto evolutionary algorithm,TIK-Report 103[R].2001.

[21] Van Veldhuizen D A,Lamont G B.Multiobjective evolutionary algorithm research:a history and analysis,TR-98-03[R].Dept Elec Comput Eng,Graduate School of Eng,Air Force Inst Technol,Wright-Patterson AFB,OH,1998.

[22] Schott J R.Fault tolerant design using single and multi-criteria genetic algorithm optimization[D].Massachusetts Institute of Technology,Cambridge,1995-05.

[23] Kalyanmoy D.Multi-objective Genetic Algorithms:problem difficulties and construction of test problems[J].Evolutionary Computation,1999,7(3):205-230.

[24] Zitzler E,Kalyanmoy D,Thiele L.Comparison of mutiobjective evolutionay algorithms:empirical results[J].Evolutionary Computation,2000,8(2):173-195.

[25] Deb K,Thiele L,Laumanns M,et al.Scalable multiobjective optimization test problem[C]//Congress on Evolutionary Computation(CEC'2002),2002,1:825-830.