

# 基于目标空间分区的稳态高维多目标进化算法

李 飞<sup>1</sup>, 刘建昌<sup>1</sup>, 朱佳妮<sup>1</sup>, 李晨曦<sup>2</sup>

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819; 2. 南京航空航天大学 自动化学院, 江苏 南京 211106)

**摘 要:** 针对高维多目标优化中 Pareto 非劣候选解所占比例很大,常用的先考虑收敛性再考虑分布性的多目标进化算法面临选择压力衰减的问题,提出一种先考虑分布性再考虑收敛性的高维多目标进化算法——基于目标空间分区的稳态高维多目标进化算法(SS-OSP).该算法先采用目标空间分区策略将种群按照权重向量分为多个子空间,在每个子空间中按照分解方法中的聚合函数选择个体;然后,考虑到常规的PBI聚合函数的罚参数在进化过程中一直保持不变的情况,提出一种自适应PBI聚合函数;最后,仿真实验结果表明所提出的算法与其他三种算法相比,具有更好的收敛性和分布性.

**关 键 词:** 高维多目标优化问题;稳态进化算法;目标空间分区;自适应PBI聚合函数;分解策略

**中图分类号:** T 18      **文献标志码:** A      **文章编号:** 1005-3026(2018)03-0305-06

## Steady-State Many-Objectives Evolutionary Algorithm Based on Objective Space Partition

LI Fei<sup>1</sup>, LIU Jian-chang<sup>1</sup>, ZHU Jia-ni<sup>1</sup>, LI Chen-xi<sup>2</sup>

(1. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 2. College of Automation Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China. Corresponding author: LI Fei, E-mail: lanceleeneu@126.com)

**Abstract:** Due to the sharp increasing of the proportion of Pareto non-dominated candidate solutions for many-objective optimization problems, the commonly used many-objective evolutionary algorithms encounter the selection pressure deterioration problem considering the convergence-first-and-diversity-second selection approach. This paper proposed a many-objective evolutionary algorithm with the diversity-first-and-convergence-second selection strategy-steady-state many-objective evolutionary algorithm based on objective space partition (SS-OSP). Firstly, it divided objective space into a large number of subspaces using a set of weight vectors. Then, one individual in each subspace was selected via adopting aggregation function. In addition, since the penalty parameter of PBI aggregation function remained constant in evolutionary process, an adaptive PBI aggregation function was proposed. Finally, the experimental results show that better convergence and diversity can be obtained using the proposed algorithm.

**Key words:** many-objective optimization problems; steady-state evolutionary algorithm; objective space partition; adaptive PBI aggregation function; decomposition strategy

进化算法(evolutionary algorithm, EA)是一种基于种群的随机全局搜索算法,算法在一次运行后得到一组解,因此被认为是适合解决多目标优化问题(multi-objective optimization problems, MOPs)的方法之一<sup>[1-2]</sup>.但是当优化问题的目标

大于或等于3时,此类问题称为高维多目标优化问题(many-objective optimization problems, MaOPs),随着维数的增加,种群中 Pareto 非劣个体所占比例很大,候选解的选择压力就会衰减,这种现象称为“维数灾难”,常规的多目标进化算法

收稿日期: 2016-10-12

基金项目: 国家自然科学基金资助项目(61773106, 61374137); 流程工业综合自动化国家重点实验室基础科研业务项目(2013ZCX02-03).

作者简介: 李 飞(1988-),男,安徽太和人,东北大学博士研究生; 刘建昌(1960-),男,辽宁黑山人,东北大学教授,博士生导师.

难以有效解决此类问题<sup>[3-4]</sup>.

目前研究者已经提出了许多求解 MaOPs 的算法,根据种群的选择机制,可将这些算法分为以下三类:基于 Pareto 支配关系的算法<sup>[5]</sup>、基于性能指标的算法<sup>[6]</sup>和基于分解的算法<sup>[7]</sup>. 基于 Pareto 支配关系的算法通过 Pareto 支配关系对种群中的个体进行排序,保留 Pareto 支配关系占优的个体,淘汰 Pareto 支配关系差的个体. 由于个体间存在非支配的关系,通常在这类算法中还需考虑个体的密度信息,通过 Pareto 支配关系促进种群收敛,利用个体密度信息维持种群的分布度. 在求解 MaOPs 时,种群中的非支配个体数目所占比例急剧增加,所以该方法不能直接应用到 MaOPs 中,通常都需要修改常规的 Pareto 支配关系. He 等<sup>[8]</sup>引入模糊逻辑理论<sup>[9]</sup>,提出一种基于模糊 Pareto 支配关系的高维多目标进化算法 FD - NSGA - II. 基于性能指标的算法<sup>[6]</sup>最早由 Zitzler 等在 2004 年提出,此类算法采用性能指标引导种群搜索,可将多目标优化问题转化为单目标优化问题求解. 常用的性能指标是 Hypervolume 指标<sup>[6]</sup>和 R2 指标<sup>[10]</sup>. 针对 Hypervolume 指标计算复杂度高的问题, Bader 等<sup>[11]</sup>提出一种基于 Hypervolume 指标的快速高维多目标进化算法 HypE, HypE 采用蒙特卡洛采样法估计解的 Hypervolume 指标值. Hernandez 等<sup>[10]</sup>提出一种基于 R2 指标的高维多目标进化算法 MOMBI - II,并将该算法用于解决高维多目标优化问题. 随着 MOEA/D<sup>[7]</sup>的提出,基于分解的多目标进化算法在求解 MaOPs 时得到了蓬勃的发展,具有代表性的算法有 DBEA<sup>[12]</sup>和  $\theta$  - DEA<sup>[13]</sup>等.

常规多目标进化算法大都采用先考虑收敛性再考虑分布性的思想,该思想适合求解 MOPs,但在求解 MaOPs 时难以选择合适的候选解逼近真实前沿. 针对上述问题,本文提出一种先考虑分布性再考虑收敛性的高维多目标进化算法——基于目标空间分区的稳态高维多目标进化算法 ( steady-state many-objective evolutionary algorithm based on objective space partition, SS - OSP ).

## 1 背景知识

### 1.1 多目标优化

不失一般性,以最小化问题为例,多目标优化问题可以表示为以下的形式<sup>[1]</sup>:

$$\begin{aligned} \min F(x) &= [f_1(x), f_2(x), \dots, f_m(x)] . \\ \text{s. t. } &g_j(x) \leq 0 (j = 1, 2, \dots, J) , \\ &h_k(x) = 0 (k = 1, 2, \dots, K) , \\ &x \in \Omega . \end{aligned} \tag{1}$$

其中:  $J$  和  $K$  表示不等式约束和等式约束个数;  $\Omega = \prod_{i=1}^n [a_i, b_i] \subseteq \mathbf{R}^n$  代表决策变量空间;  $x = \{x_1, x_2, \dots, x_n\}$  表示一个候选解;  $F: \Omega \rightarrow \mathbf{R}^m$  包含了  $m$  个相互冲突的目标,  $\mathbf{R}^m$  称为目标空间.

求解多目标优化算法的主要目的是得到一组具有代表性的解集,该解集尽可能逼近真实的 Pareto 前沿且分布均匀.

### 1.2 聚合函数

分解方法给每个个体赋予一个不同的权重向量,通过聚合方法将多目标优化问题转化成单目标优化问题. 通常有三种聚合方法可供选择<sup>[13]</sup>:

1) Weighted Sum 聚合方法. Weighted Sum 聚合方法的计算公式为

$$\min g^{\text{ws}}(x | \mathbf{w}) = \sum_{j=1,2,\dots,m} w_{i,j} f_j(x) . \tag{2}$$

其中:  $\mathbf{w}$  是权重向量且满足  $w_{i,j} \geq 0$ ;  $\sum_{j=1,2,\dots,m} w_{i,j} = 1$ .

2) Tchebycheff 聚合方法. Tchebycheff 聚合方法的计算公式为

$$\min g^{\text{tc}}(x | \mathbf{w}) = \max_{j=1,2,\dots,m} \{w_{i,j} (f_j(x) - z_j^*)\} . \tag{3}$$

其中:  $z_j^*$  是参考点,它的计算公式为

$$z_j^* = \min \{f_j(x) | x \in \Omega\} . \tag{4}$$

3) PBI 聚合方法. PBI 聚合方法的计算公式为

$$g^{\text{pbi}}(x | \mathbf{w}) = d_1 + \theta d_2 . \tag{5}$$

其中:  $d_1 = \frac{\| (f(x) - \mathbf{z}^*)^T \mathbf{w} \|}{\| \mathbf{w} \|}$ ;

$d_2 = \| (f(x) - \mathbf{z}^*) - d_1 \frac{\mathbf{w}}{\| \mathbf{w} \|} \|$ ;  $\theta$  为罚参数.

## 2 算法实现

### 2.1 SS - OSP 基本流程

本文提出的 SS - OSP 算法的伪码如图 1 所示. 在完成算法参数和权重向量初始化后,算法进入迭代过程. 算法采用模拟二进制交叉和多项式变异产生一个子代个体,然后将子代个体与父代个体合并,最后丢掉一个决策者所不偏好的候选解.

**Procedure Algorithm of SS – OSP:**

```

begin
    initialize algorithm parameters;
    initialize weight vector set (subproblems)  $W$ ;
    generate randomly and evaluate initial
    population  $P_0$ ;
    while ( $t < \text{MaxGen}$ )
        for  $i = 1 : |P_0|$ 
             $Q_{t,i} = \text{generation}(P_t)$ ;
             $P_t = \text{selection}(P_t \cup Q_{t,i})$ ;
        end for
    until a termination condition is met
    output the population  $P_t$ .
end
    
```

图 1 SS – OSP 伪码图

Fig. 1 Pseduo-code for SS-OSP

2.2 SS – OSP 的选择策略

常见的高维多目标进化算法的选择策略通常是先考虑收敛性再考虑分布性. 本文提出一种先考虑分布性再考虑收敛性的选择策略,如图 2 所示. 先将目标空间按权重向量分为多个子空间,然后在子空间中选择个体. 该策略分为三个部分:目标空间归一化、分区处理和个体选择. 下面将分别介绍这三个部分.

首先对目标空间进行归一化处理,如式(6)所示:

$$\tilde{f}_j(x) = \frac{f_j(x) - z_j^*}{z_j^{\text{nad}} - z_j^*}. \tag{6}$$

其中:理想点  $z_j^*$  可由种群中的第  $j$  个目标值的最小值得到; $z_j^{\text{nad}}$  采用每个目标函数的最大值.

然后对种群进行分区处理. 计算每个个体与所有权重向量的夹角,找到夹角最小的权重向量,将该个体放入该权重向量所对应的目标子空间中,如式(7)和式(8)所示:

$$\theta_i(x) = \arccos \frac{\tilde{f}WT(x)^T w_i}{\|\tilde{f}(x)\| \|w_i\|}, \tag{7}$$

$$\Phi_i(x) = \{x | \theta_k(x) \leq \theta_j(x)\}, i, k = 1, 2, \dots, N. \tag{8}$$

其中: $N$  表示权重向量的个数; $\theta_i$  表示  $\tilde{f}(x)$  与权重向量  $w_i$  的夹角; $\Phi_i(x)$  表示个体  $x$  属于权重向量  $w_i$  所在的区域.

**Procedure Algorithm of Selection Operator:**

```

begin
    normalize population  $P$  using (6)
    for  $i = 1 : |P|$  do
        for  $j = 1 : |W|$  do
            calculate the angle between  $P_i$ 
            and  $W_j$  using (7);
        end for
        calculate the subspace  $\Phi_i(x)$  using
        (8);
    end for
    for  $i = 1 : |W|$  do
        if the number of individuals in the  $i$ th
        subregion is greater than zero
            find the individual with the smallest
            aggregation function value in the  $i$ th
            subregion;
        else
            find the individual with the smallest
            aggregation function value in the
            other subregion which is not be
            chosen;
        end if
    end for
end
    
```

图 2 选择操作伪码图

Fig. 2 Pseduo-code for selection operator

最后从每个种群中采用分解方法选取个体. 如果某个权重向量所在的区域中只有 1 个个体,则直接保留这个个体;如果某个权重向量所在的区域中的个体数目大于 1,则选择该区域内聚合函数值最小的个体;如果某个权重向量所在区域中没有个体,则选择整个种群中聚合函数值最小的个体.

常规 PBI 聚合函数中, $\theta$  用来平衡收敛性和分布性, $\theta$  较小,PBI 强调种群的收敛性,会选择出收敛性较好的个体; $\theta$  较大,PBI 强调种群的分布性,会选择分布性较好的个体. 在 PBI 聚合函数中, $\theta$  取 5.0. 在种群的进化前期强调收敛性而后期强调分布性,在种群的进化过程中始终保持  $\theta$  值不变显然不合理, $\theta$  应该随着迭代次数的增加而增加. 因此, $\theta$  的计算公式如下:

$$\theta = \theta_0 \cdot \frac{\max(z^{\max})}{\min(z^{\max})} \cdot \frac{t}{\text{MaxGen}}. \tag{9}$$

其中: $\theta_0 = 5.0$ ;  $z^{\max}$  表示当前种群中每个目标的最大值组成的向量;  $\frac{\max(z^{\max})}{\min(z^{\max})}$  表示向量  $z^{\max}$  中最大值与最小值的比值;  $t$  表示当前的代数; MaxGen 表示最大的代数.

图 3 是上述选择策略的示意图. 图中有 4 个权重向量  $w_1 \sim w_4$ , 图 3 所示的候选解中存在 5 个个体, 因此需要对候选解进行修剪.  $\Phi_1$  区域中有 2 个个体  $x_1$  和  $x_2$ ,  $x_1$  在权重向量  $w_1$  上的聚合函数值更小, 选择  $x_1$  个体;  $\Phi_2$  区域中只有一个个体  $x_3$ , 因此直接选择  $x_3$  个体. 由于  $x_2$  相对于  $x_3$  在权重向量  $w_2$  上的 PBI 聚合函数值更小, 若没有分区处理则会选择  $x_2$  个体, 而一般认为  $x_3$  个体更有利于种群在  $\Phi_2$  区域搜索;  $\Phi_3$  区域中有 2 个个体  $x_4$  和  $x_5$ ,  $x_4$  在权重向量  $w_3$  上的聚合函数值更小, 选择  $x_4$  个体;  $\Phi_4$  区域中没有个体, 则不选择个体. 上述的候选解修剪策略有利于引导种群向权重向量靠近, 实现种群的均匀分布.

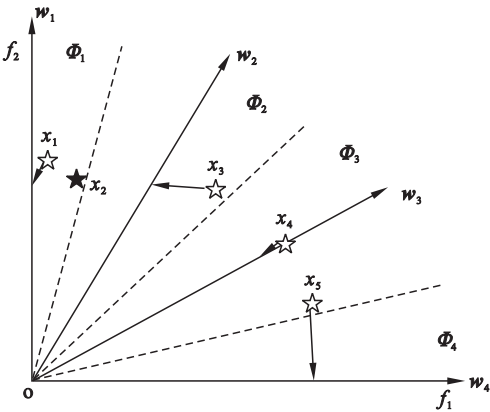


图 3 选择操作示意图

Fig. 3 Schematic diagram of selection operator

3 算法测试及结果分析

为了验证 SS - OSP 算法性能, 选取 WFG 系列测试函数作为研究对象, 将 SS - OSP 与 DBEA<sup>[12]</sup>, MOMBI - II<sup>[10]</sup> 和 MOEA/D - PBI<sup>[14]</sup> 进行比较.

本文采用超体积指标 (hypervolume indicator,  $I_H$ ) 比较算法的性能,  $I_H$  越大说明算法的收敛性和分布性越好.  $I_H$  的计算公式如下:

$$I_H(A, r) = \text{volume}(\cup_{f \in A} [f_1, r_1] \times \cdots \times [f_m, r_m])$$

(10)

其中:  $r$  是参考点, 取  $r = [3, 5, \cdots, 2m + 1]$ ;  $A$  是最终得到的解集. 根据文献[11]中提出的方法, 采

用蒙特卡洛采样方法估算得到  $I_H$  指标值, 采样个数取为  $10^6$ .

为了保证公平性, 每种算法都进行 30 次独立实验. 每个测试函数的迭代次数见表 1 中的第二列. SS - OPS 中的分布系数  $\eta_c = 30$ , 交叉概率  $p_c = 1.0$ , 种群的数目与权重向量数目一样, 分别是 5 维问题取 210 个、8 维问题取 156 个、10 维问题取 275 个和 15 维问题取 135 个, 其他算法的参数设置参考相应的文献<sup>[10, 12, 14]</sup>. 表 1 是 4 种算法在不同测试函数不同维数上的性能指标评价结果, 每个表格中的 2 个数分别代表性能指标的平均值和标准差. 表 1 中的“+”、“=”和“-”表示本文算法获得的  $I_H$  值在显著性水平为 5% 的 Wilcoxon 检验中分别优于、等于和劣于对应列的对等算法在对应行的测试问题上的显著性区分结果, 并在最后一行列出了本文算法在所有测试问题上显著优于、等于和劣于其他算法的个数之和.

图 4 和图 5 分别列出了各算法在 WFG6 和 WFG9 的 15 维问题上得到的 Pareto 前沿的平行坐标图. 由图可知, SS - OSP 在这两个问题上得到的 Pareto 前沿的收敛性和分布性都比较好; DBEA 在这两个问题上分布性都不好, 特别在 WFG9 的 15 维问题上出现种群聚集情况; MOMBI - II 在这两个问题分布性也都不好; 而 MOEA/D - PBI 在 WFG6 的 15 维问题上只能得到部分前沿, 而在 WFG9 的 15 维问题上收敛性可以接受, 但是分布性比 SS - OSP 差.

从表 1 可知, 本文提出的 SS - OSP 算法整体性能最好, DBEA 性能次之, 而 MOEA/D - PBI 算法性能最差. 其中 DBEA 在 36 个测试问题中, 有 7 个测试问题的性能好于 SS - OSP, 但是有 20 个测试问题的性能比 SS - OSP 差; 而 MOMBI - II 在 10 个问题上的性能好于 SS - OSP, 但是有 24 个测试问题的性能比 SS - OSP 差; MOAE/D - PBI 除了在 WFG2 的 15 维问题、WFG3 的 10 维问题以及 WFG9 的 15 维问题上与 SS - OSP 性能相似外, 其他问题上的性能都比它差. 此外, 从表 1 中还可以得到, MOMBI - II 在 WFG1 ~ WFG3 上的性能更好, 而 SS - OSP 和 DBEA 在 WFG4 ~ WFG9 上的性能更好. 综上所述, 通过与 DBEA, MOMBI - II 和 MOEA/D - PBI 比较可知, 本文提出的 SS - OSP 是一种求解高维多目标优化问题的有效算法.



表 1 各算法  $I_H$  性能指标的平均值和标准差  
Table 1 Means and standard deviations of the  $I_H$  performance indicator calculated by each algorithm

HV	Obj. (MaxGen)	SS – OSP	DBEA	MOMBI – II	MOEA/D – PBI
WFG1	5(750)	4.59e3(6.55e1)	4.28e3(4.00e1) +	<b>7.53e3(1.62e2) –</b>	1.02e3(9.19e1) +
	8(1 500)	1.49e7(5.52e5)	1.99e7(7.60e6) –	<b>3.26e7(1.20e6) –</b>	3.64e6(2.53e5) +
	10(2 000)	5.68e9(3.91e8)	9.67e9(3.86e9) –	<b>1.33e10(2.64e7) –</b>	1.60e9(8.75e7) +
	15 (3 000)	1.09e17(1.401e + 16)	6.22e16(6.24e16) =	<b>1.78e17(5.06e15) –</b>	2.26e16(9.05e14) +
WFG2	5(750)	9.85e3(6.73e2)	<b>9.71e3(7.70e2) =</b>	<b>1.02e4(4.18e1) –</b>	9.35e3(7.39e2) +
	8(1 500)	3.02e7(3.02e6)	2.42e7(3.18e6) +	3.33e7(3.15e5) =	2.82e7(2.67e6) +
	10(2 000)	<b>1.26e10(1.19e9)</b>	9.01e9(3.10e9) +	1.33e10(1.21e8) +	1.17e10(9.63e8 +
	15 (3 000)	<b>1.56e17(1.70e16)</b>	7.95e16(3.16e16) +	1.61e17(7.17e15) +	<b>1.44e17(2.46e16) =</b>
WFG3	5(750)	<b>6.90e3(1.11e2)</b>	6.56e3(1.30e2) +	6.66e3(9.66e1) +	5.97e3(2.75e2) +
	8(1 500)	1.67e7(9.80e5)	2.01e6(4.90e3) +	<b>2.28e7(1.80e5) –</b>	1.50e7(7.07e5) +
	10(2 000)	5.52e9(2.01e8)	6.52e8(8.53e5) +	<b>9.22e9(6.97e7) –</b>	5.51e9(3.51e8) =
	15 (3 000)	5.68e16(2.63e15)	6.12e15(2.11e13) +	<b>1.17e17(1.55e15) –</b>	4.97e16(7.90e15) +
WFG4	5(750)	<b>8.50e3(6.69e1)</b>	8.44e3(5.49e1) +	7.99e3(2.98e2) +	8.34e3(1.82e2) +
	8(1 500)	2.96e7(4.38e5)	<b>3.02e7(3.46e5) –</b>	1.93e7(1.00e6) +	1.76e7(5.35e6) +
	10(2 000)	1.23e10(1.61e8)	<b>1.30e10(5.23e7) –</b>	8.04e9(6.82e8) +	4.38e9(1.97e9) +
	15 (3 000)	<b>1.72e17(4.65e15)</b>	7.81e16(4.13e16) +	7.16e16(1.04e16) +	5.52e16(2.61e16) +
WFG5	5(750)	8.46e3(2.39e1)	<b>8.44e3(2.86e1) –</b>	7.36e3(1.68e2) +	8.00e3(4.45e2) +
	8(1 500)	<b>2.98e7(1.96e5)</b>	2.96e7(1.80e5) +	1.77e7(6.84e5) +	1.47e7(6.41e6) +
	10(2 000)	<b>1.22e10(7.28e7)</b>	1.23e10(5.24e7) +	6.75e9(2.32e8) +	3.46e9(1.81e9) +
	15 (3 000)	<b>1.75e17(8.05e14)</b>	3.98e16(5.31e16) +	6.13e16(5.32e15) +	3.37e16(2.90e16) +
WFG6	5(750)	<b>8.59e3(5.97e1)</b>	<b>8.57e3(8.10e1) =</b>	6.97e3(1.51e2) +	7.63e3(5.56e2) +
	8(1 500)	<b>3.05e7(4.20e5)</b>	<b>3.06e7(2.24e5) =</b>	1.70e7(6.75e5) +	1.32e7(5.73e6) +
	10(2 000)	<b>1.26e10(1.22e8)</b>	<b>1.26e10(9.32e7) =</b>	6.89e9(3.58e8) +	3.26e9(1.83e9) +
	15 (3 000)	<b>1.78e17(2.69e15)</b>	7.10e16(4.29e16) +	5.94e16(3.50e15) +	3.28e16(2.49e16) +
WFG7	5(750)	7.93e3(1.36e3)	<b>8.98e3(2.85e2) –</b>	7.65e3(6.73e2) =	1.93e3(1.56e3) +
	8(1 500)	<b>2.01e7(6.23e6)</b>	1.72e7(4.07e6) +	1.69e7(2.70e6) +	3.73e6(4.50e6) +
	10(2 000)	<b>8.65e9(2.54e9)</b>	7.167e9(2.010e9 +	8.04e9(5.33e8) +	1.38e9(5.99e8) +
	15 (3 000)	<b>1.78e17(2.69e15)</b>	7.10e16(4.29e16) +	5.94e16(3.50e15) +	3.28e16(2.49e16) +
WFG8	5(750)	<b>5.06e3(1.28e2)</b>	<b>4.92e3(1.54e2) =</b>	4.03e3(2.19e2) +	3.48e3(4.29e2) +
	8(1 500)	1.22e7(1.10e6)	<b>1.44e7(8.82e5) –</b>	<b>1.39e7(4.55e5) –</b>	5.75e6(1.05e6) +
	10(2 000)	4.52e9(4.77e8)	1.37e9(4.87e8) +	<b>5.82e9(1.63e8) –</b>	1.90e9(8.60e8) +
	15 (3 000)	<b>5.12e16(1.46e16)</b>	<b>5.21e16(1.26e16) =</b>	6.75e16(1.24e16) +	1.75e16(1.80e16) +
WFG9	5(750)	<b>7.81e3(2.07e1)</b>	7.80e3(1.60e1) +	6.44e3(1.05e2) +	7.43e3(3.47e2) +
	8(1 500)	<b>2.57e7(1.07e6)</b>	<b>2.62e7(2.02e5) =</b>	1.63e7(6.86e5) +	2.06e7(3.09e6) +
	10(2 000)	<b>1.03e10(4.67e8)</b>	<b>1.02e10(1.63e9) =</b>	6.55e9(2.52e8) +	9.07e9(1.25e9) +
	15 (3 000)	<b>1.04e17(2.02e16)</b>	4.89e15(5.86e13) +	5.72e16(5.84e15) +	<b>1.08e17(1.69e16) =</b>
– / + / =			7/20/9	10/24/2	0/33/3

注：黑体表示最优  $I_H$  指标值；“–”表示 SS – OSP 比对比算法差；“+”表示 SS – OSP 比对比算法好；“=”表示算法性能相似。

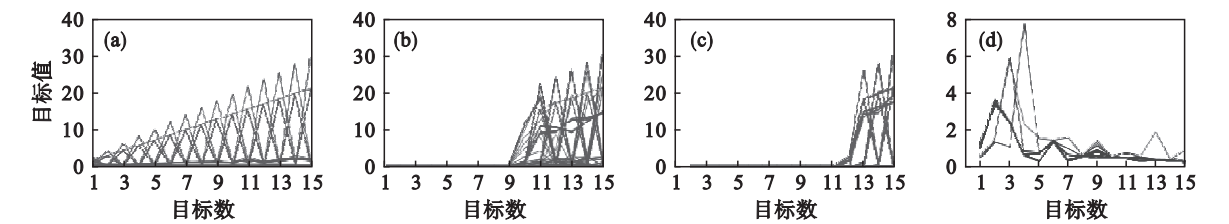


图 4 各算法在 WFG6 的 15 维问题上得到的 Pareto 前沿的平行坐标图  
Fig. 4 Parallel coordinates of Pareto front obtained by each algorithm on 15-objective WFG6  
(a)—SS – OSP；(b)—DBEA；(c)—MOMB I – II；(d)—MOEA/D – PBI.

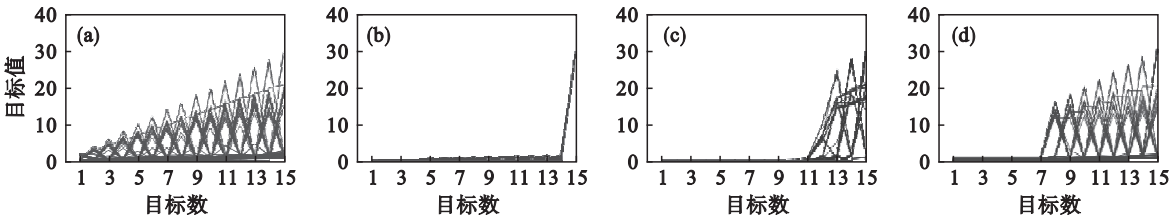


图 5 各算法在 WFG9 的 15 维问题上得到的 Pareto 前沿平行坐标图  
Fig. 5 Parallel coordinates of Pareto front obtained by each algorithm on 15-objective WFG9  
(a)—SS-OSP; (b)—DBEA; (c)—MOMB I - II; (d)—MOEA/D-PBI.

4 结 论

由于高维多目标优化问题中 Pareto 非劣个体比例过大,常规多目标进化算法无法选择出决策者偏好的个体,本文提出一种先考虑分布性再考虑收敛性的基于目标空间分区的稳态高维多目标进化算法.通过目标空间分区策略提高种群的分布性;然后在每个子空间中采用自适应的 PBI 聚合函数选择个体,提高种群的收敛性.通过与其他三个高维多目标进化算法在 WFG1 ~ WFG9 测试函数上的仿真结果表明,该算法具有较好的收敛性和分布性.

参考文献:

[ 1 ] Coello C C, Lamont G B, van Veldhuizen D A. Evolutionary algorithms for solving multi-objective problems [ M ]. New York: Springer Science & Business Media, 2007: 1 - 30.  
[ 2 ] 付亚平,王洪峰,黄敏.面向多目标优化问题的基于 Species 的遗传算法[ J ]. 东北大学学报(自然科学版), 2016, 37(3): 314 - 318.  
(Fu Ya-ping, Wang Hong-feng, Huang Min. Species-based genetic algorithm for multiobjective optimization problems [ J ]. Journal of Northeastern University (Natural Science), 2016, 37(3): 314 - 318.)  
[ 3 ] Li B, Li J, Tang K, et al. Many-objective evolutionary algorithms: a survey[ J ]. ACM Computing Surveys, 2015, 48 (1): 1 - 35.  
[ 4 ] Giagkiozis I, Fleming P J. Methods for multi-objective optimization: an analysis[ J ]. Information Sciences, 2014, 293: 338 - 350.  
[ 5 ] 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.  
[ 6 ] Zitzler E, Künzli S. Indicator-based selection in multiobjective search [ C ]// International Conference on Parallel Problem Solving from Nature. Berlin: Springer Heidelberg, 2004: 832 - 842.  
[ 7 ] Zhang Q, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition[ J ]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712 - 731.  
[ 8 ] He Z, Yen G G, Zhang J. Fuzzy-based pareto optimality for many-objective evolutionary algorithms [ J ]. IEEE Transactions on Evolutionary Computation, 2014, 18 (2): 269 - 285.  
[ 9 ] Zadeh L A. Fuzzy sets[ J ]. Information and Control, 1965, 8 (3): 338 - 353.  
[ 10 ] Hernandez G R, Coello C C A. Improved metaheuristic based on the R2 indicator for many-objective optimization [ C ]// Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. Madrid: ACM, 2015: 679 - 686.  
[ 11 ] Bader J, Zitzler E. HypE: an algorithm for fast hypervolume-based many-objective optimization [ J ]. Evolutionary Computation, 2011, 19(1): 45 - 76.  
[ 12 ] Asafuddoula M, Ray T, Sarker R. A decomposition based evolutionary algorithm for many objective optimization[ J ]. IEEE Transactions on Evolutionary Computation, 2014, 19 (3): 445 - 460.  
[ 13 ] Yuan Y, Xu H, Wang B, et al. A new dominance relation-based evolutionary algorithm for many-objective optimization [ J ]. IEEE Transactions on Evolutionary Computation, 2016, 20(1): 16 - 37.  
[ 14 ] Li K, Deb K, Zhang Q, et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition [ J ]. IEEE Transactions on Evolutionary Computation, 2015, 19(5): 694 - 716.