



## 目标空间映射策略的高维多目标粒子群优化算法

陈强, 王宇嘉, 梁海娜, 孙欣

引用本文:

陈强, 王宇嘉, 梁海娜, 等. 目标空间映射策略的高维多目标粒子群优化算法[J]. 智能系统学报, 2021, 16(2): 362–370.

CHEN Qiang, WANG Yujia, LIANG Haina, et al. Multi-objective particle swarm optimization algorithm based on an objective space papping strategy[J]. *CAAI Transactions on Intelligent Systems*, 2021, 16(2): 362–370.

在线阅读 View online: <https://dx.doi.org/10.11992/tis.202006042>

## 您可能感兴趣的其他文章

### 改进萤火虫优化算法的Renyi熵油污图像分割

Renyi entropy based on improved firefly optimization algorithm for image segmentation of waste oil  
智能系统学报. 2020, 15(2): 367–373 <https://dx.doi.org/10.11992/tis.201809002>

### 基于目标空间分解和连续变异的多目标粒子群算法

Decomposition and continuous mutation-based multi-objective particle swarm optimization  
智能系统学报. 2019, 14(3): 464–470 <https://dx.doi.org/10.11992/tis.201711015>

### 基于改进SPEA2算法的给水管网多目标优化设计

Multi-objective optimization design of water distribution systems based on improved SPEA2 algorithm  
智能系统学报. 2018, 13(1): 118–124 <https://dx.doi.org/10.11992/tis.201701012>

### 膜系统下的一种多目标优化算法

Multi-objective optimization algorithm based on membrane system  
智能系统学报. 2017, 12(5): 678–683 <https://dx.doi.org/10.11992/tis.201706013>

### 基于改进粒子群算法的移动机器人多目标点路径规划

Mobile robot multi-goal path planning using improved particle swarm optimization  
智能系统学报. 2017, 12(3): 301–309 <https://dx.doi.org/10.11992/tis.201606046>

### 一种用于大型舰船总体要素优化设计的约束多目标分解进化算法

A constrained multi-objective decomposition evolutionary algorithm for the overall element optimization design of large ship  
智能系统学报. 2016, 11(5): 635–640 <https://dx.doi.org/10.11992/tis.201605006>

微信公众平台



关注微信公众号, 获取更多资讯信息

DOI: 10.11992/tis.202006042

网络出版地址: <https://kns.cnki.net/kcms/detail/23.1538.TP.20201110.0926.002.html>

# 目标空间映射策略的高维多目标粒子群优化算法

陈强, 王宇嘉, 梁海娜, 孙欣

(上海工程技术大学 电子电气工程学院, 上海 201620)

**摘要:** 为了平衡优化算法在高维多目标优化问题中收敛性和多样性之间的关系, 增加算法的选择压力, 本文提出了一种基于目标空间映射策略的高维多目标粒子群优化算法 (many-objective particle swarm optimization algorithm based on objective space mapping strategy, MOPSO-OSM)。在求解高维多目标优化问题时, Pareto 准则难以从众多的非支配解中确定最优“折中”解, 因此将高维多目标空间映射为以收敛性和多样性评价指标的 2 维空间, 再将上述 2 维空间根据性能指标的优劣划分为 4 个不同区域。同时, 使用反向学习策略提高算法跳出局部最优的能力。实验表明, MOPSO-OSM 算法可以有效平衡收敛性和多样性之间的关系, 达到求解复杂多目标优化问题的目的。

**关键词:** 目标空间映射策略; 性能指标; 反向学习; 粒子群; 高维多目标优化; Pareto 准则; 收敛性; 分布性  
**中图分类号:** TP18 **文献标志码:** A **文章编号:** 1673-4785(2021)02-0362-09

中文引用格式: 陈强, 王宇嘉, 梁海娜, 等. 目标空间映射策略的高维多目标粒子群优化算法 [J]. 智能系统学报, 2021, 16(2): 362-370.

英文引用格式: CHEN Qiang, WANG Yujia, LIANG Haina, et al. Multi-objective particle swarm optimization algorithm based on an objective space mapping strategy [J]. CAAI transactions on intelligent systems, 2021, 16(2): 362-370.

## Multi-objective particle swarm optimization algorithm based on an objective space mapping strategy

CHEN Qiang, WANG Yujia, LIANG Haina, SUN Xin

(School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

**Abstract:** To balance the relationship between the convergence and diversity of the optimization algorithm in the multi-objective problem, the selection pressure of the algorithm is increased. A high-dimensional MOPSO-OSM (multi-objective particle swarm optimization algorithm based on objective space mapping strategy) is proposed in this paper. When solving high-dimensional multi-objective optimization problems, the Pareto based criterion cannot identify the best compromise solutions from many nondominated solutions. Therefore, the high-dimensional multi-objective optimization space is mapped into two-dimensional space based on indexes of convergence and diversity. Then, the two-dimensional space is divided into four regions according to the performance index. Simultaneously, the ability of the jumping local optimal solution is improved using the opposition learning strategy. The experimental results show that MOPSO-OSM can balance the relationship between convergence and diversity and solve complex problems.

**Keywords:** objective space mapping strategy; performance index; opposition learning; particle swarm optimization; high-dimensional multi-objective optimization; Pareto based criterion; convergence; diversity

在现实生活中存在大量多目标优化问题, 例如生产制造业、金融投资、航空气调度等。多目标

优化问题是由多个待优化目标函数组成, 当目标个数多于 3 个时, 该问题又被称为高维多目标优化问题 (many-objective optimization problem)<sup>[1]</sup>。由于各目标之间具有冲突性, 传统的优化算法很难得到一组最优解, 因此, 研究者大多采用启发式

收稿日期: 2020-06-24. 网络出版日期: 2020-11-10.

基金项目: 国家自然科学基金项目 (61403249).

通信作者: 王宇嘉. E-mail: [yjwangamber@sues.edu.cn](mailto:yjwangamber@sues.edu.cn).

方法来求解该问题。目前,用于求解高维多目标优化问题的启发式方法主要分为以下三类<sup>[2]</sup>:基于支配关系的进化算法、基于分解的进化算法和基于指标的进化算法。

基于支配关系的进化算法是通过 Pareto 支配策略来选择非支配解。NSGA-II<sup>[3]</sup> 算法通过快速非支配解排序策略来获得非支配解,但是该方法无法保证种群的多样性。NSGA-III<sup>[4]</sup> 作为 NSGA-II 的改进版本,为了增强算法处理高维多目标优化问题的能力,在筛选同一等级的个体时,采用了基于参考点的方法来代替拥挤距离,然而该算法仅在某些具有特定形状的 Pareto 前沿问题上表现较优。CNSGA-III<sup>[5]</sup> 算法在 NSGA-III 的基础上,通过非支配解层中添加一个聚类操作来增强算法种群的多样性,实验结果表明,该策略效果较差。Zou 等<sup>[6]</sup> 在 NSGA-II 框架基础上,提出了一种在关键层中选取非支配解的旋转网格策略,在一定程度上增强了算法的选择压力。Lin 等<sup>[7]</sup> 提出了一种基于平衡适应度估计策略的高维粒子群算法来解决高维多目标优化问题,该策略将目标空间划分为不同的区间并给每个区间中的目标赋予不同的权重,然而过多的权重设置限制了该方法的实际应用。

基于分解的进化算法是通过将多目标问题转化为多个单目标问题来处理。MOEA/D<sup>[8]</sup> 和 MOEA/D-M2M<sup>[9]</sup> 是两种常用的基于分解的多目标进化算法。MOEA/DD<sup>[10]</sup> 是通过目标空间分解与自适应权重分配相结合来解决高维多目标优化问题。但上述算法过于依赖权重的选取。此外为了减少 Pareto 前沿形状对分解算法性能的影响,Liu 等<sup>[11]</sup> 提出了一种基于模糊分解的多目标进化算法,结果表明该算法对于该问题具有较好的处理效果。

基于指标的进化算法是将解的评价标准作为选择支配解的一类算法。IBEA<sup>[12]</sup> 采用 Hypervolume 指标选取非支配解,该方法在处理高维多目标问题时无法保证分布性。Bader 等<sup>[13]</sup> 提出了另一种基于 Hypervolume 指标的进化算法,在一定程度上平衡了高维多目标优化问题的收敛性和分布性,但是 Hypervolume 指标的计算复杂度随着目标个数的增加呈指数增加<sup>[2]</sup>,进一步限制了该算法应用。此外,基于 GD<sup>[14]</sup>、IGD<sup>[15]</sup> 和 R2<sup>[16]</sup> 指标的进化算法在求解高维多目标优化问题时都取得了不错的效果。

基于 Pareto 支配策略的进化算法相对于另外两类算法,可以从搜索的深度上逼近 Pareto 前

沿,因此,对解决高维多目标优化问题方面仍具有巨大潜力。收敛性和分布性作为多目标优化问题中的两个重要指标,在种群演化过程中是相互冲突的<sup>[17]</sup>,因此设计一种能有效平衡收敛性和多样性的新型支配策略,对于解决高维多目标优化问题具有重要意义。本文提出了一种目标空间映射策略的高维多目标粒子群优化算法,该策略作为一种新的多样性保持机制,可从众多的候选解中筛选出收敛性和分布性都较优的个体,同时利用一种增强型反向学习策略帮助种群跳出局部最优。

## 1 粒子群优化算法

### 1.1 基本粒子群优化算法

粒子群算法 (particle swarm optimization, PSO)<sup>[18]</sup> 是通过研究鸟群觅食规律而发展出的一种群智能优化算法。粒子通过个体最优位置和群体最优位置动态更新自身的位置和速度,速度和位置的更新公式分别为

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

式中:  $\omega$  表示权重系数;  $v_{id}$  和  $x_{id}$  分别表示第  $i$  个粒子在第  $d$  维度上的速度和位置大小;  $c_1$  和  $c_2$  表示学习因子;  $r_1$  和  $r_2$  表示 (0, 1) 之间的随机数;  $p_{id}$  和  $p_{gd}$  分别表示个体最优位置和群体最优位置。

### 1.2 改进粒子群优化算法

粒子群算法在其发展过程中,为了提高算法的寻优性能,主要有以下几种改进策略。

1) 调整算法的参数。为了平衡种群勘探与开采的能力,Shi 等<sup>[19]</sup> 将  $\omega$  引入粒子群算法中,  $\omega$  值越大,勘探未知区域的能力越强,  $\omega$  值越小,小范围内的开采能力越强, Clerc 等<sup>[20]</sup> 建议  $\omega$  的取值为 0.729, Venter 等<sup>[21]</sup> 则采用非线性递减的策略来更新  $\omega$ 。此外还有部分研究者通过调整  $c_1$ 、 $c_2$  的取值来增强算法的搜索能力<sup>[22-23]</sup>。

2) 设计不同类型的拓扑结构。Kennedy<sup>[24]</sup> 通过分析种群拓扑结构与交互概率的关系,提出了一种 bare bones particle swarm (BBPS) 的模型。Yue 等<sup>[25]</sup> 提出了一种基于环形拓扑结构的粒子群算法并将其用于求解多模态的多目标问题。

3) 与其他策略相结合,形成混合粒子群算法。侯翔等<sup>[26]</sup> 为了提高算法求解问题的能力,对所有粒子的最优位置进行降维处理,形成一个参考全局最优解,同时使用该解来更新群体当前的

最优位置。Lin 等<sup>[27]</sup>将 MOPSO 同分解算法相结合,采用两种不同的搜索策略来更新粒子的速度,结果显示,算法对复杂问题的解决性能得到了加强。Zain 等<sup>[28]</sup>为了降低算法在约束问题上求解的难度,在标准 MOPSO<sup>[29]</sup>算法的基础上提出了一种基于动态搜索边界策略的 MOPSO。

## 2 目标空间映射策略的高维多目标粒子群算法

### 2.1 高维多目标优化问题

最小化高维多目标优化问题可以表示为

$$\begin{aligned} \min Y(\mathbf{x}) &= \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})\} \\ \text{subject to } \mathbf{x} &\in D \end{aligned} \quad (3)$$

式中:  $D$  表示决策空间,  $\mathbf{x}=(x_1, x_2, \dots, x_n)$ ;  $Y: D \rightarrow \mathbf{R}^m$  由  $m$  个目标函数  $f_1, f_2, \dots, f_m$  组成,  $\mathbf{R}^m$  表示目标空间;  $m \geq 4$  表示为目标函数的个数。通过式 (3) 得到一组最优解, 即 Pareto 解集, 表示为 PS, 如果  $PF = \{Y(\mathbf{x}) \in \mathbf{R}^m | \mathbf{x} \in PS\}$ , 称 PF 为 Pareto 前沿。

### 2.2 目标空间映射策略

由于非支配解个数增加导致算法无法收敛到完整的 Pareto 前沿, 本文采用目标空间映射策略来增强算法对非支配解的选择压力。

首先, 对每个个体在目标空间中的收敛性进行评价, 采用式 (4) 计算个体的收敛性:

$$F_i = \frac{d_{i,o}}{\sqrt{m}}, \quad i = 1, 2, \dots, n \quad (4)$$

式中:  $m$  表示目标函数的个数;  $d_{i,o}$  表示个体  $i$  到参考点  $o$  之间的欧式距离, 这里将参考点设置为原点。  $F_i$  的值越小, 个体的收敛性越好。

然后, 对每个个体在目标空间上的分布性进行评价, 采用式 (5) 计算个体的分布性:

$$\text{Dis}_i = \frac{\left( \sum_{s=1}^m \frac{f_s^{i+1}(\mathbf{x}) - f_s^{i-1}(\mathbf{x})}{\max(f_s(\mathbf{x})) - \min(f_s(\mathbf{x}))} \right)}{m}, \quad i = 1, 2, \dots, n \quad (5)$$

式中:  $f_s^{i+1}$  表示第  $i+1$  个个体在第  $s$  个目标函数上的目标值;  $\text{Dis}_i$  越大, 则第  $i$  个个体的分布越好。

此时, 每个非劣解都可以映射为以收敛性和分布性表征的 2 维空间, 即  $PS \rightarrow (F_i, \text{Dis}_i)$ 。

最后, 计算收敛性评价的平均值  $F_{\text{ave}}$  和分布性评价的平均值  $\text{Dis}_{\text{ave}}$ ,  $F_{\text{ave}}$  和  $\text{Dis}_{\text{ave}}$  如式 (6) 和 (7) 所示:

$$F_{\text{ave}} = \frac{\sum_{i=1}^n F_i}{n} \quad (6)$$

$$\text{Dis}_{\text{ave}} = \frac{\sum_{i=1}^n \left( \frac{1}{\text{Dis}_i} \right)}{n} \quad (7)$$

其中  $n$  表示个体的总数。

根据  $F_i$ 、 $\text{Dis}_i$ 、 $F_{\text{ave}}$ 、 $\text{Dis}_{\text{ave}}$  这 4 个参数将映射后的 2 维空间划分为 4 个不同的区域, 如图 1 所示。

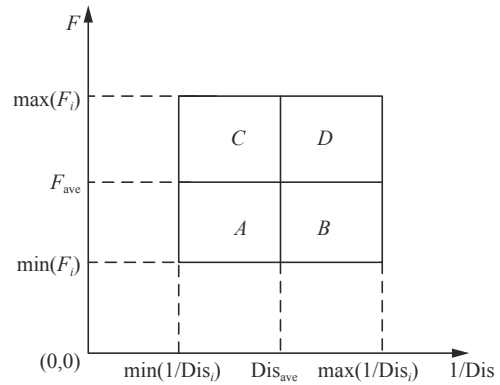


图1 映射后的 2 维空间划分结果

Fig. 1 2-dimensional space division result after mapping

设个体  $i$  对应的目标函数值为  $Y_i$ , 此时其映射到 2 维空间后存在以下 4 种情况:

$A: \{Y_i | F_i \leq F_{\text{ave}} \cap \frac{1}{\text{Dis}_i} \leq \text{Dis}_{\text{ave}}, Y_i \in A\}$ , 处于  $A$  区域的个体, 其收敛性和分布性均是最优。

$B: \{Y_i | F_i \leq F_{\text{ave}} \cap \frac{1}{\text{Dis}_i} > \text{Dis}_{\text{ave}}, Y_i \in A\}$ , 处于  $B$  区域的个体, 其收敛性较好, 分布性较差。

$C: \{Y_i | F_i > F_{\text{ave}} \cap \frac{1}{\text{Dis}_i} \leq \text{Dis}_{\text{ave}}, Y_i \in A\}$ , 处于  $C$  区域的个体, 其收敛性较差, 分布性较好。

$D: \{Y_i | F_i > F_{\text{ave}} \cap \frac{1}{\text{Dis}_i} > \text{Dis}_{\text{ave}}, Y_i \in A\}$ , 处于  $D$  区域的个体, 其收敛性和分布性都较差。

在不同区域内的非支配解质量从好到坏的顺序为:  $A > B = C > D$ 。

为进一步评价同区域内或并列区域内个体的优劣, 采用式 (8) 计算个体的序值。

$$\text{Value}_i = \alpha \left( \frac{1}{\text{Dis}_i} \right) + \beta F_i \quad (8)$$

式中:  $\alpha$  和  $\beta$  分别为分布性权重和收敛性权重。  $\text{Value}_i$  越小, 个体排序越好。对于处在区域  $A$  或  $D$  中的个体, 设置  $\alpha=\beta=1$ , 保证  $A$  和  $D$  区域中的个体信息不变。对于处在  $B$  和  $C$  中的个体, 其收敛性和分布性互不支配, 假设  $B$  区域中个体数量为  $k$ , 当个体处在  $B$  区域时, 参数设置如下:  $\beta=1, \alpha = \left( \min \left( \frac{1}{\text{Dis}_i} \right) + \text{rand} \left( \text{Dis}_{\text{ave}} - \min \left( \frac{1}{\text{Dis}_i} \right) \right) \right) \text{Dis}_i, i = 1, 2, \dots, k$ , 通过该设置,  $B$  区域中个体收敛信息保持不变, 分



布信息被缩放到  $(\min(1/\text{Dis}_i), \text{Dis}_{\text{ave}}), i = 1, 2, \dots, k$  区间内。当个体处于  $C$  区域时,假设  $C$  区域中个体数量为  $q$ ,参数设置如下:  $\beta = (\min(F_i) + \text{rand}(F_{\text{ave}} - \min(F_i))) \frac{1}{F_i}, i = 1, 2, \dots, q, \alpha = 1$ ,该设置将区域  $C$  中个体的收敛信息缩放到  $(\min(F_i), F_{\text{ave}}), i = 1, 2, \dots, q$  区间内,而分布信息保持不变。通过上述操作,处于区域  $B$  和  $C$  中的个体,其分布信息和收敛信息都被缩放到相同的区间之中,因此两区域中的个体可以被放在一起统一评价。

在选取非支配解时,首先选取区域  $A$  中的个体,当区域  $A$  中的个体个数大于外部文档大小时,选取  $\text{Value}_i$  值小的个体进入档案文件;当区域  $A$  中的个体个数小于外部文档大小时,再从  $B$  和  $C$  中选取剩余个体,当  $B$  和  $C$  中的个体个数大于剩余外部文档大小时,仍然选取  $\text{Value}_i$  值小的个体进入外部文档。当  $B$  和  $C$  中的个体个数不能满足条件时,最后选取  $D$  中的个体。

图2给出了目标空间映射策略的流程图。

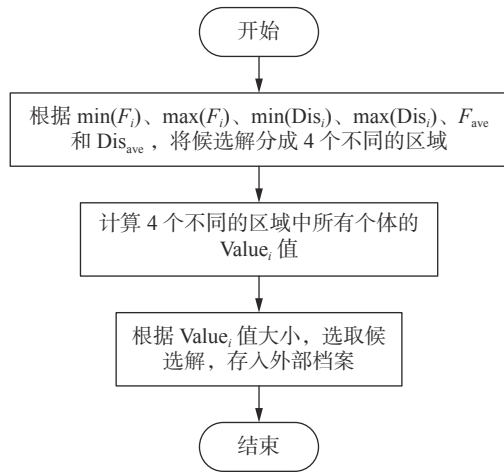


图2 目标映射策略的流程图

Fig. 2 Flowchart of the objective space mapping strategy

### 2.3 反向学习策略

优化过程中如果算法陷入局部最优,则利用反向学习策略作为跳出机制,其中文献[30]的反向学习策略如式(9)所示:

$$\begin{cases} x_{id}^* = k(\min(x_d) + \max(x_d)) - x_{id} \\ x_{id}^* = k(a_d + b_d) - x_{id} \\ x_{id}^* = \text{rand}(a_d, b_d), \text{ if } x_{id}^* < x_{d\min} \parallel x_{id}^* > x_{d\max} \end{cases} \quad (9)$$

式中:  $x_{id}^*$  表示第  $i$  个个体在第  $d$  维决策向量上得到的新位置;  $x_d$  表示所有个体在第  $d$  维上的位置;  $a_d$  和  $b_d$  分别表示种群个体在第  $d$  维目标向量上的最小和最大边界值;  $k$  表示  $(0, 1)$  间的随机数;  $[x_{d\min}, x_{d\max}]$  表示在第  $d$  维上的边界约束。

由式(9)可得,  $x_{id}^*$  的取值范围为  $[k(a_d + b_d) -$

$b_d, k(a_d + b_d) - a_d]$ , 当  $k=1$  时,  $x_{id}^*$  取得最大值为  $b_d$ 。因此,当最优解的决策向量位于  $b_d$  的右侧时,上述方法不能跳出局部最优。

本文对上述方法进行了改进,为提高算法跳出局部最优的能力,同时不忽略当前收敛信息,当  $x_{id}^* = x_{d\min}$  时,  $x_{id}$  执行式(10)给出的反向学习策略。

$$x_{id}^* = x_{d\min} + x_{d\max} - x_{id} \quad (10)$$

从式(10)可以看出,  $x_{id}^*$  的取值范围扩大到了  $[x_{d\min}, x_{d\max}]$ , 该粒子跳出局部最优区域。

为了判断算法是否陷入局部最优,本文采用文献[31]给出的判断准则作为反向学习的激发条件,如式(11)所示:

$$\begin{cases} \frac{|\min(f_i^t) - \min(f_i^{t-10})|}{|\min(f_i^t)|} < 0.005, t > 10 \\ \frac{|\max(f_i^t) - \max(f_i^{t-10})|}{|\max(f_i^t)|} < 0.005, t > 10 \end{cases} \quad (11)$$

式中:  $\min(f_i^t)$  和  $\max(f_i^t)$  分别表示在第  $t$  代时,第  $i$  个目标维度上的最小和最大值。 $\min(f_i^{t-10})$  和  $\max(f_i^{t-10})$  分别表示在第  $t-10$  代时,第  $i$  个目标维度上的最小和最大值。对于一个  $m$  维测试函数,通过式(11)对其所有的目标维度的变化率进行计算,当所有的变化率都小于 0.005 时,算法陷入局部最优。

### 2.4 MOPSO-OSM 流程

MOPSO-OSM 算法具体流程如下:

- 1) 算法初始化;
- 2) 判断是否满足停止条件,若条件满足,算法停止迭代,否则转到 3);
- 3) 判断种群是否陷入局部最优,执行反向学习策略;否则,直接转到 4);
- 4) 利用式(1)和(2)更新个体的速度和位置;
- 5) 计算个体的适应度值;
- 6) 对个体当前适应度值和前代适应度值进行比较来更新个体最优;
- 7) 选择非支配解;
- 8) 利用目标空间映射策略更新外部档案文件;
- 9) 从外部档案中随机选择一个个体来更新种群最优,并转到 2);

## 3 实验结果与分析

### 3.1 测试函数

为了评价算法性能的优劣,文中采取了 6 组 WFG 测试函数。参数设置如表 1 所示。

表 1 测试函数参数设置  
Table 1 Test functions parameter setting

测试函数	目标个数	决策变量个数	特征
WFG1	5	14	混合
	10	19	
WFG2	5	14	凸面, 不连续
	10	19	
WFG3	5	14	线性, 退化
	10	19	
WFG4	5	14	凹面, 多模
	10	19	
WFG5	5	14	凹面, 欺骗性
	10	19	
WFG6	5	14	凹面, 不可分
	10	19	

### 3.2 性能指标

文中利用世代距离 (GD)、间距 (SP) 和逆世代距离 (IGD) 3 个指标来评估算法的性能。GD 被用来评价种群的收敛性, GD 值越小, 收敛性越好, 其计算公式为

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (12)$$

式中:  $n$  表示非支配解的个数;  $d_i$  表示非支配解与 Pareto 最优解之间的欧式距离。

SP 指标通常被用来评价种群的分布性, 其计算如式 (13) 所示, 分布性好坏与其计算值成反比。

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (13)$$

式中  $\bar{d}$  为  $d_i$  的平均值。

IGD 指标被用来同时评价种群分布性和收敛性, IGD 越小, 算法展现出的性能越好, 其计算公式为

$$IGD = \frac{\sum_{x^* \in P^*} d(x^*, P)}{|P^*|} \quad (14)$$

式中:  $P$  和  $P^*$  分别表示 Pareto 最优解集和非支配解;  $|P^*|$  表示非支配个数。

### 3.3 对比算法及其设置

本文所得结果与目前较为流行的进化算法进行对比, 比较算法包括: NSGA-III<sup>[3]</sup>、RVEA<sup>[32]</sup>、MOEA/DD<sup>[10]</sup>、PESA-II<sup>[33]</sup> 和 NMPSO<sup>[7]</sup>。所有使用的算法其种群大小都设置为 100, 外部文档的大小为 100, 算法进化的次数为 700 次, 算法具体参

数设置如表 2 所示, 所有算法均运行 30 次, 计算收敛性和多样性指标, 取其平均值。

表 2 对比算法参数设置  
Table 2 Comparison algorithms parameter setting

对比算法	参数设置
MOEA/DD	$T = 20, p_m = 1/n, \eta_c = \eta_m = 20, \delta = 0.9, n_r = 2$
NSGA-III	$p_c = 1, p_m = 1/n, \eta_c = \eta_m = 20$
PESA-II	$p_c = 1, p_m = 1/n, \eta_c = \eta_m = 20, div = 10$
RVEA	$p_c = 1, p_m = 1/n, \eta_c = \eta_m = 20, \alpha = 2, f_r = 0.1$
NMPSO	$p_m = 1/n, \eta_m = 20, \omega \in [0.1, 0.5], c_1, c_2, c_3 \in [1.5, 2.5]$
MOPSO-OSM	$\omega \in [0.4, 0.9], c_1 = c_2 = 2$

对于 MOPSO-OSM 算法, 其权重  $\omega$  是随着迭代次数线性减少的。

### 3.4 结果与分析

表 3 和表 4 分别为所有算法在 5 目标和 10 目标测试函数时得到的 GD、SP 和 IGD 的平均值。

#### 3.4.1 收敛性分析

从表 3 中可以看出, 对于 5 目标测试函数, NMPSO 算法在 WFG1 和 WFG2 测试函数中得到了最好的 GD 值。NSGA-III 算法在 WFG4 和 WFG6 问题中取得了最佳的 GD 值。MOPSO-OSM 算法在 WFG3 和 WFG5 取得了最优的 GD 值。对于 10 目标的测试函数, 从表 4 中可以看出, 在 WFG1 和 WFG2 测试函数上, MOEA/DD 取得了最佳 GD 值。NSGA-III 在 WFG6 问题上取得最优 GD 值, 在 WFG4 和 WFG5 测试问题上, RVEA 取得的 GD 值排名第一。对于 WFG3 测试函数, MOPSO-OSM 取得了最优的 GD 值。从以上可以看出, 本文所提出的算法在大部分测试函数中并没有取得最优值。原因在于, 大多数算法在求解时, 仅仅追求收敛性, 忽视了分布性, 而本文在目标空间分配策略中, 同时考虑目标向量的收敛性和分布性, 所以在测试函数中, 并不能完全保证 GD 的最优性, 这也验证了“没有免费午餐”的原理。

#### 3.4.2 分布性分析

对于 5 目标测试函数, 从表 3 中可以看出, MOPSO-OSM 算法在所有测试函数中都取得了最好的 SP 值。对于 10 目标的测试函数, 从表 4 可以看出, MOPSO-OSM 算法同样在所有的测试函数中, 得到了最好的 SP 值。以上可以看出, MOPSO-OSM 算法在保持种群分布性上面具有很大的优势。

表 3 5 目标测试函数结果  
Table 3 Results of five objectives

指标	测试函数	MOEA/DD	NSGA-III	PESA-II	NMPSO	RVEA	MOPSO-OSM
GD	WFG1	0.049 84	0.192 06	0.188 98	<b>0.019 92</b>	0.186 08	0.202 70
	WFG2	0.013 58	0.068 36	0.075 10	<b>0.013 20</b>	0.064 48	0.050 50
	WFG3	0.321 44	0.344 88	0.201 76	0.204 10	0.377 42	<b>0.117 32</b>
	WFG4	0.027 68	<b>0.026 72</b>	0.050 06	0.028 46	0.027 94	0.030 19
	WFG5	0.029 99	0.026 89	0.049 64	0.034 32	0.027 84	<b>0.025 93</b>
	WFG6	0.091 36	<b>0.028 88</b>	0.059 10	0.046 78	0.028 94	0.031 61
SP	WFG1	1.004 48	0.954 02	0.192 30	0.993 90	1.017 32	<b>0.080 98</b>
	WFG2	0.238 28	0.736 02	0.407 62	0.336 22	0.447 74	<b>0.066 88</b>
	WFG3	1.231 42	0.651 98	0.260 22	0.334 96	0.718 32	<b>0.034 14</b>
	WFG4	1.295 76	0.778 64	0.710 48	0.598 66	0.771 82	<b>0.050 02</b>
	WFG5	1.306 08	0.782 14	0.778 62	0.620 52	0.781 74	<b>0.038 86</b>
	WFG6	1.316 84	0.775 52	0.717 20	0.622 30	0.774 96	<b>0.034 86</b>
IGD	WFG1	0.759 32	<b>0.367 25</b>	0.620 10	1.076 38	0.505 14	1.037 02
	WFG2	<b>0.581 56</b>	0.824 72	1.314 74	1.123 98	1.690 76	0.720 06
	WFG3	0.931 26	0.589 38	2.465 10	0.273 82	0.731 02	<b>0.230 22</b>
	WFG4	1.407 54	1.225 68	1.842 64	1.351 02	1.226 96	<b>1.215 34</b>
	WFG5	1.362 82	<b>1.215 06</b>	1.689 82	1.271 44	1.216 24	1.483 24
	WFG6	1.386 04	<b>1.216 18</b>	1.823 76	1.370 84	1.229 62	1.522 18

表 4 10 目标测试函数结果  
Table 4 Results of ten objectives

指标	测试函数	MOEA/DD	NSGA-III	PESA-II	NMPSO	RVEA	MOPSO-OSM
GD	WFG1	<b>0.064 20</b>	0.151 32	0.315 60	0.086 42	0.077 08	0.306 76
	WFG2	<b>0.041 10</b>	0.438 92	0.446 86	0.071 56	0.361 80	0.283 20
	WFG3	1.072 10	1.065 24	0.393 08	0.556 84	1.037 02	<b>0.230 66</b>
	WFG4	0.165 58	0.113 92	0.260 58	0.181 56	<b>0.082 68</b>	0.205 88
	WFG5	0.167 02	0.156 18	0.284 98	0.191 70	<b>0.059 46</b>	0.150 36
	WFG6	0.174 70	<b>0.127 48</b>	0.332 94	0.149 76	0.153 80	0.165 14
SP	WFG1	2.952 64	1.304 66	1.623 18	2.029 32	3.612 14	<b>0.181 42</b>
	WFG2	0.309 70	1.408 74	0.626 74	0.813 50	0.611 72	<b>0.068 72</b>
	WFG3	3.561 80	1.881 06	0.535 30	1.266 38	1.581 30	<b>0.076 24</b>
	WFG4	1.844 52	3.676 30	2.382 28	2.954 60	2.548 44	<b>0.147 96</b>
	WFG5	2.822 60	0.156 18	2.768 73	2.778 74	2.712 42	<b>0.111 14</b>
	WFG6	2.346 98	2.394 74	1.712 34	2.428 56	2.023 36	<b>0.135 82</b>
IGD	WFG1	<b>1.531 40</b>	2.074 74	3.353 78	4.071 31	1.690 12	3.210 88
	WFG2	<b>1.580 62</b>	7.084 92	4.728 54	2.117 6	8.839 20	3.640 01
	WFG3	3.422 18	3.738 76	6.422 06	0.986 80	4.811 42	<b>0.700 94</b>
	WFG4	7.863 98	6.129 48	6.250 24	<b>5.085 56</b>	5.890 66	5.424 32
	WFG5	7.598 92	5.897 46	7.131 26	<b>5.106 64</b>	5.813 56	5.646 24
	WFG6	7.681 06	7.190 22	9.466 46	5.454 48	6.204 72	<b>5.299 34</b>

### 3.4.3 整体性分析

从表 3 中可以看出, 对于 5 目标测试函数,

NSGA-III 在 WFG1、WFG5 和 WFG6 中取得了最优的 IGD 值。MOEA/DD 在 WFG2 中取得最好的

IGD 值。MOPSO-OSM 算法在 WFG3 和 WFG4 中取得的 IGD 值排名第一。对于 10 目标测试函数,从表 4 可以看出,MOEA/DD 在 WFG1 和 WFG2 中取得了 IGD 的最优值。NMPISO 在 WFG4 和 WFG5 中的 IGD 值最优。MOPSO-OSM 算法在 WFG3 和 WFG6 中的 IGD 取得了最佳值。从以上可以看出,本文所提算法在处理 WFG1 和 WFG2 问题时表现出较低的能力,主要原因在于目标映射策略要综合考虑个体的分布性和收敛性,导致了算法在该问题上的收敛性不足,无法得到一组完整的 Pareto 前沿,所以该算法在处理 WFG1 和 WFG2 问题时还有待提高。对于其余测试函数,虽然在部分问题上算法无法得到最优值,但是所得结果仍处于较好的排名。对于不同目标个数的

同一个测试函数,本文算法与其余算法相比较,随着目标个数的增加,对比算法的性能都在急剧下降,而本文所提算法展现出了更好的适应性。通过以上可以看出,本文采用的目标空间映射策略,将收敛性和多样性结合,用这两者对目标向量进行分类选择,最后得到的非支配解具有较好的性能。

图 3 分别给出了在 5 目标 WFG3 测试函数下得到的 Pareto 前沿。对于 WFG3 测试函数,所有算法都得不到一个完整的 Pareto 前沿。但是 MOPSO-OSM 算法的 Pareto 前沿相比于另外 5 种对比算法,其收敛性和分布性都是明显优于其他算法,表现出较好的性能。图 4 给出了 MOPSO-OSM 算法在 5 目标 WFG3 和 WFG4 测试问题中得到的 GD 值曲线。

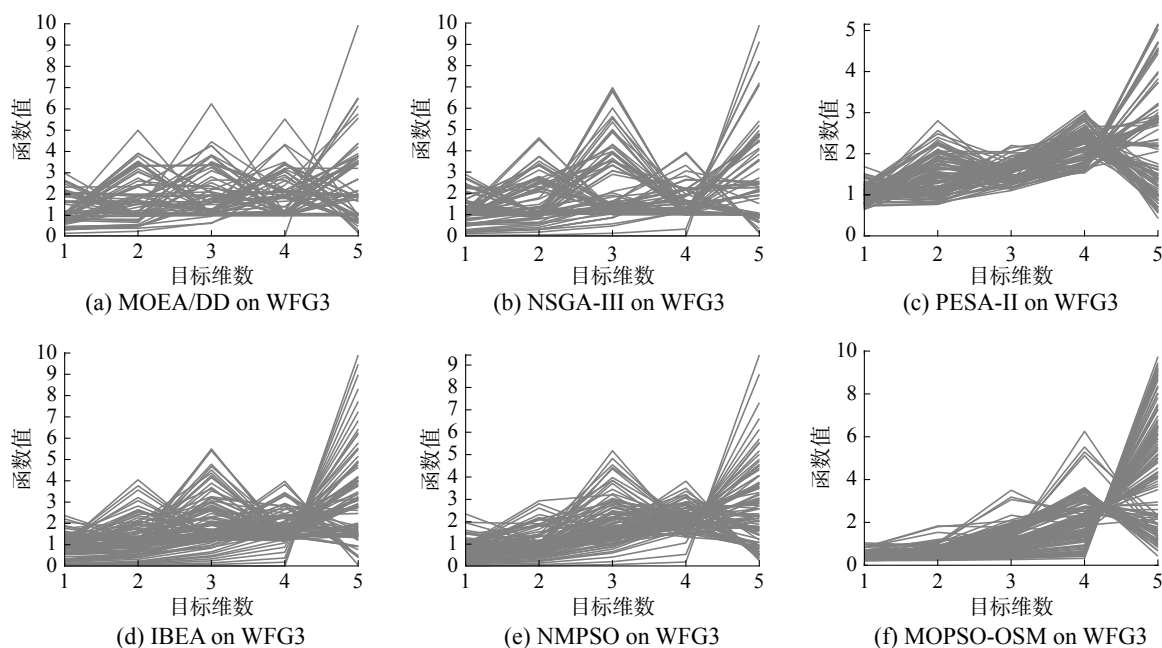


图 3 6 种算法在 5 目标 WFG3 测试函数所得 Pareto 前沿

Fig. 3 Six algorithms obtain Pareto Front in 5 objective WFG3 test function

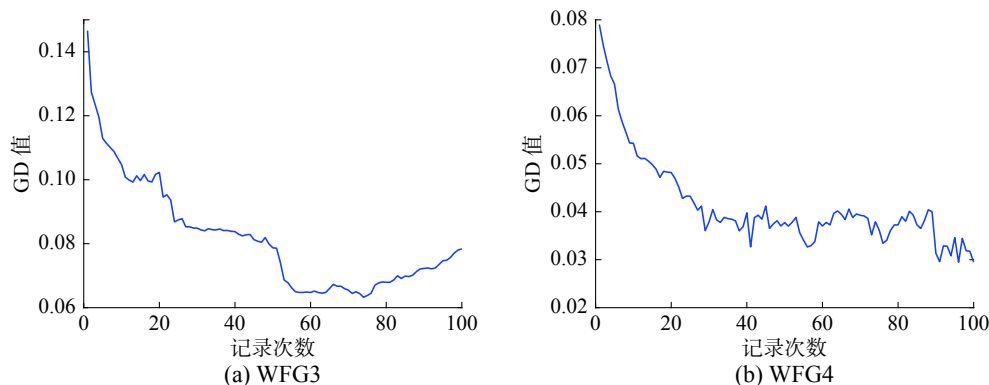


图 4 MOPSO-OSM 在 5 目标 WFG3 和 WFG4 上的 GD 曲线

Fig. 4 GD curves of MOPSO-OSM for WFG3 and WFG4 with five objectives



图4中横坐标为记录的次数,每7次迭代记录一次。从中可以看出,在迭代过程中,GD值都是逐渐减小,算法逐渐收敛。目标映射策略作为一种有效的筛选候选解的方法,对于算法在问题中的收敛性能的表现起到决定性的作用,因此,在高维多目标优化问题中,目标映射策略是可行且有效的。

## 4 结束语

本文提出了一种基于目标空间映射策略的高维多目标粒子群算法来求解高维多目标问题。该方法利用性能指标对目标空间进行划分,从而达到增强算法选择压力的目的。通过6组标准测试函数的仿真验证,实验结果表明,在处理高维多目标问题时,目标空间映射策略能够有效地提高种群的收敛性和分布性。将该算法应用在工程实例问题将是下一步的研究重点。

## 参考文献:

- [1] ISHIBUCHI H, TSUKAMOTO N, NOJIMA Y. Evolutionary many-objective optimization: a short review[C]//2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). Hong Kong, China, 2008: 266–271.
- [2] 刘建昌,李飞,王洪海,等.进化高维多目标优化算法研究综述[J].控制与决策,2018,33(5): 879–887.  
LIU Jianchang, LI Fei, WANG Honghai, et al. Survey on evolutionary many-objective optimization algorithms[J]. Control and decision, 2018, 33(5): 879–887.
- [3] 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.
- [4] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints[J]. IEEE transactions on evolutionary computation, 2014, 18(4): 577–601.
- [5] 汤恺祥,许峰.基于大数据聚类的改进NSGA-III算法[J].信息记录材料,2020,21(5): 109–112.  
TANG Kaixiang, XU Feng. Improved NSGA-III algorithm based on big data clustering[J]. Information recording materials, 2020, 21(5): 109–112.
- [6] ZOU Juan, FU Liuwei, ZHENG Jinhua, et al. A many-objective evolutionary algorithm based on rotated grid[J]. Applied soft computing, 2018, 67: 596–609.
- [7] LIN Qiuzhen, LIU Songbai, TANG Chaoyu, et al. Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems[J]. IEEE transactions on evolutionary computation, 2018, 22(1): 32–46.
- [8] ZHANG Qingfu, LI Hui. MOEA/D: a multiobjective evolutionary algorithm based on decomposition[J]. IEEE transactions on evolutionary computation, 2008, 11(6): 712–731.
- [9] LIU Hailin, GU Fangqing, ZHANG Qingfu. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems[J]. IEEE transactions on evolutionary computation, 2014, 18(3): 450–455.
- [10] LI Ke, DEB K, ZHANG Qingfu, et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition[J]. IEEE transactions on evolutionary computation, 2015, 19(5): 694–716.
- [11] LIU Songbai, LIN Qiuzhen, TAN K C, et al. A fuzzy decomposition-based Multi/many-objective evolutionary algorithm[J]. IEEE transactions on cybernetics, 2020: 1–15.
- [12] ZITZLER E, KÜNZLI S. Indicator-based selection in multiobjective search[C]//International Conference on Parallel Problem Solving from Nature. Berlin, Heidelberg: Springer, 2004: 832–842.
- [13] BADER J, ZITZLER E. HypE: an algorithm for fast hypervolume-based many-objective optimization[J]. Evolutionary computation, 2011, 19(1): 45–76.
- [14] MENCHACA-MENDEZ A, COELLO C A C. GDEMOEA: a new MOEA based on the generational distance indicator and  $\epsilon$ -dominance[C]//2015 IEEE Congress on Evolutionary Computation (CEC). Sendai, Japan: IEEE, 2015: 947–955.
- [15] TIAN Ye, CHENG Ran, ZHANG Xingyi, et al. An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility[J]. IEEE transactions on evolutionary computation, 2018, 22(4): 609–622.
- [16] LI Fei, CHENG Ran, LIU Jianchang, et al. A two-stage R2 indicator based evolutionary algorithm for many-objective optimization[J]. Applied soft computing, 2018, 67: 245–260.
- [17] LI Miqing, YANG Shengxiang, LIU Xiaohui. Bi-goal evolution for many-objective optimization problems[J]. Artificial intelligence, 2015, 228: 45–65.
- [18] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95 - International Conference on Neural Network. Perth, Australia, 1995: 1942–1948.
- [19] SHI Y, EBERHART R C. A modified particle swarm optimizer[C]//1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Con-

- gress on Computational Intelligence (Cat. No. 98TH8360). Anchorage, USA, 1998: 73–79.
- [20] CLERC M, KENNEDY J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space[J]. *IEEE transactions on evolutionary computation*, 2002, 6(1): 58–73.
- [21] VENTER G, SOBIESZCZANSKI-SOBIESKI J. Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization[J]. *Structural and multidisciplinary optimization*, 2004, 26: 121–131.
- [22] MA Borong, HUA Jun, MA Zhixin, et al. IMOPSO: an improved multi-objective particle swarm optimization algorithm[C]//2016 5th International Conference on Computer Science and Network Technology (ICCSNT). Changchun, China, 2016: 376–380.
- [23] QI Changxing, BI Yiming, HAN Huihua, et al. A hybrid particle swarm optimization algorithm[C]//2017 3rd IEEE International Conference on Computer and Communications (ICCC). Chengdu, China, 2017: 2187–2190.
- [24] KENNEDY J. Bare bones particle swarms[C]//Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706). Indianapolis, USA, 2003: 80–87.
- [25] YUE Caitong, QU Boyang, LIANG Jing. A multiobjective particle swarm optimizer using ring topology for Solving multimodal multiobjective problems[J]. *IEEE transactions on evolutionary computation*, 2018, 22(5): 805–817.
- [26] 侯翔, 蒲国林. 协同粒子群优化算法的改进与仿真[J]. *计算机工程与设计*, 2015, 36(6): 1530–1534.
- HOU Xiang, PU Guolin. Improvement of its cooperative particle swarm optimization algorithm and simulation[J]. *Computer engineering and design*, 2015, 36(6): 1530–1534.
- [27] LIN Qiuzhen, LI Jianqiang, DU Zhihua, et al. A novel multi-objective particle swarm optimization with multiple search strategies[J]. *European Journal of operational research*, 2015, 247(3): 732–744.
- [28] ZAIN M Z B M, KANESAN J, CHUAH J H, et al. A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization[J]. *Applied soft computing*, 2018, 70: 680–700.
- [29] COELLO C A C, LECHUGA M S. MOPSO: a proposal for multiple objective particle swarm optimization[C]//Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600). Honolulu, USA, 2002, 1051–1056.
- [30] WANG Hui, WU Zhijian, RAHNAMAYAN S, et al. Enhancing particle swarm optimization using generalized opposition-based learning[J]. *Information sciences*, 2011, 181(20): 4699–4714.
- [31] 马灿, 刘坚, 余方平. 混合模拟退火的布谷鸟算法研究[J]. *小型微型计算机系统*, 2016, 37(9): 2029–2034.
- MA Can, LIU Jian, YU Fangping. Research on cuckoo algorithm with simulated annealing[J]. *Journal of Chinese computer systems*, 2016, 37(9): 2029–2034.
- [32] CHENG Ran, JIN Yaochu, OLHOFFER M, et al. A reference vector guided evolutionary algorithm for many-objective optimization[J]. *IEEE transactions on evolutionary computation*, 2016, 20(5): 773–791.
- [33] CORNE D W, JERRAM N R, KNOWLES J D, et al. PESA-II: region-based selection in evolutionary multiobjective optimization[C]//Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. Morgan Kaufmann Publishers Inc, 2001: 283–290.

#### 作者简介:



陈强, 硕士研究生, 主要研究方向为进化计算和多目标优化。



王宇嘉, 副教授, 博士, 主要研究方向为进化计算、群智能和目标优化。发表学术论文 16 篇。



梁海娜, 硕士研究生, 主要研究方向为进化计算和群智能。