

# 控制与决策

Control and Decision

## 求解约束高维多目标问题的分解约束支配NSGA-II优化算法

顾清华, 莫明慧, 卢才武, 陈露

引用本文:

顾清华, 莫明慧, 卢才武, 等. 求解约束高维多目标问题的分解约束支配NSGA-II优化算法[J]. 控制与决策, 2020, 35(10): 2466–2474.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0116>

---

### 您可能感兴趣的其他文章

Articles you may be interested in

#### [一种基于参考点约束支配的NSGA-III算法](#)

A reference point constrained dominance-based NSGA-III algorithm

控制与决策. 2019, 34(2): 369–376 <https://doi.org/10.13195/j.kzyjc.2017.1067>

#### [利用种群扩张与稀疏化策略改进NSGA-II-DE算法](#)

Using population expansion and sparsity strategy to improve NSGA-II-DE algorithm

控制与决策. 2019, 34(1): 55–62 <https://doi.org/10.13195/j.kzyjc.2017.0973>

#### [基于分解和差分进化的多目标粒子群优化算法](#)

Multi-objective particle swarm optimization algorithm based on decomposition and differential evolution

控制与决策. 2017, 32(3): 403–410 <https://doi.org/10.13195/j.kzyjc.2016.0186>

#### [一种求解多目标优化问题的进化算法混合框架](#)

A hybrid framework of evolutionary algorithm for solving multi-objective optimization problems

控制与决策. 2017, 32(10): 1729–1738 <https://doi.org/10.13195/j.kzyjc.2016.0937>

#### [一种双链结构的多目标进化算法DCMOEA](#)

A multi-objective evolutionary algorithm based on double chains

控制与决策. 2015(4): 577–584 <https://doi.org/10.13195/j.kzyjc.2013.1737>

# 求解约束高维多目标问题的分解约束支配 NSGA-II 优化算法

顾清华<sup>1,2†</sup>, 莫明慧<sup>2</sup>, 卢才武<sup>1</sup>, 陈露<sup>2</sup>

(1. 西安建筑科技大学 资源工程学院, 西安 710055; 2. 西安建筑科技大学 管理学院, 西安 710055)

**摘要:** 针对多目标进化算法处理约束高维多目标优化问题时出现解的分布性和收敛性差、易陷入局部最优解问题, 采用 Pareto 支配、分解与约束支配融合的方法, 提出一种基于分解约束支配 NSGA-II 优化算法 (DBC DP-NSGA-II)。该算法在保留 NSGA-II 中快速非支配排序的基础上, 首先采用 Pareto 支配对种群进行支配排序; 然后根据解的性质采用分解约束支配 (DBC DP) 惩罚等价解, 保留稀疏区域的可行解和非可行解, 提高种群的分布性、多样性和收敛性; 最后采用个体到权重向量的垂直距离和拥挤度距离对临界值进行再排序, 直到选出  $N$  个最优个体进入下一次迭代。以约束 DTLZ 问题中 C-DTLZ1、C-DTLZ2、DTLZ8、DTLZ9 测试函数为例, 将所提出的算法与 C-NSGA-II、C-NSGA-III、C-MOEA/D 和 C-MOEA/DD 进行对比分析。仿真结果表明, DBC DP-NSGA-II 所得最优解分布更加均匀, 具有更好的全局收敛性。

**关键词:** 约束高维多目标; Deb 约束支配; MOEA/D; NSGA-II; 分布性; 收敛性

中图分类号: TP273 文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0116

**引用格式:** 顾清华, 莫明慧, 卢才武, 等. 求解约束高维多目标问题的分解约束支配 NSGA-II 优化算法 [J]. 控制与决策, 2020, 35 (10): 2466-2474.

## Decomposition-based constrained dominance principle NSGA-II for constrained many-objective optimization problems

GU Qing-hua<sup>1,2†</sup>, MO Ming-hui<sup>2</sup>, LU Cai-wu<sup>1</sup>, CHEN Lu<sup>2</sup>

(1. School of Resources Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China; 2. School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China)

**Abstract:** The distribution and convergence of solutions are poor when constrained many-objective optimization problems are solved with multi-objective evolutionary algorithm, which tend to fall into local optimal solutions. We propose a decomposition-based constrained dominance principle NSGA-II (DBC DP-NSGA-II) based on the fusion of Pareto dominance, decomposition and constraint dominance. In the study, based on retaining the fast non-dominant ranking in NSGA-II, Pareto dominance is employed firstly to dominate population. Then according to the nature of the solution, the DBC DP is adopted to punish the equivalent solutions. The feasible and infeasible solutions in sparse regions are preserved to improve the distribution, diversity and convergence of the population. Finally, the critical values are reordered by the vertical distance and the crowding distance from the individual to the weight vector until  $N$  optimal individuals are selected for the next iteration. Using constrained DTLZ as an example, the algorithm is compared with C-NSGA-II, C-MOEA/D, C-MOEA/DD and C-NSGA-III. The results show that it has more uniform distribution and better global convergence performance than the other four algorithms.

**Keywords:** constrained many-objective optimization; Deb's constraint handling strategy; MOEA/D; NSGA-II; distribution; convergence

## 0 引言

高维多目标优化问题 (many-objective optimization problems, MaOPs)<sup>[1]</sup> 中的无约束问题研

究已较为广泛。而约束优化问题研究较少, 主要原因是增加了约束条件后, 导致可行域不连通或者可行域变得较为复杂<sup>[2]</sup>。但在实际工程应用中, 如复杂物流

收稿日期: 2019-01-24; 修回日期: 2019-04-27.

基金项目: 国家自然科学基金项目 (51774228, 51404182); 陕西省自然科学基金项目 (2017JM5043); 陕西省教育厅专项科研计划项目 (17JK0425).

责任编辑: 巩敦卫.

†通讯作者. E-mail: qinghuagu@126.com.

车辆调度<sup>[3]</sup>、城市水资源管理<sup>[4]</sup>等都是典型的约束高维多目标问题,因此,研究约束高维多目标优化对实际工业问题的解决具有重要的理论意义和实际价值。

为了求解约束MaOPs,其关键性问题之一是保持目标与约束的平衡,大多数进化算法都是以此为目的进行约束MaOPs求解<sup>[5]</sup>。例如,惩罚函数法采用惩罚因子来保持目标与约束之间的平衡,将约束MaOPs转换为无约束MaOPs。文献[6]引入惩罚函数来惩罚非可行解,对多目标差分进化算法(MOEA/DE)<sup>[7]</sup>进行了改进以处理约束多目标问题。采用惩罚函数简单且易实现,但是针对不同的约束条件,惩罚函数中的参数往往难以确定。为了避免调整惩罚参数,学者们提出了一种分别比较目标和约束的约束处理方法,如Deb约束支配原则<sup>[8]</sup>、Epsilon约束处理(EC)<sup>[9]</sup>、随机排序方法(SR)<sup>[10]</sup>等。由于Deb约束支配原则简单且没有额外的参数,它是目前使用最多的约束处理方法。由于解决约束问题时MOEA-DE需要两个惩罚参数,为此,文献[11]对MOEA-DE进行了改进,引入Deb约束准则,提出了C-MOEA/D;同时,通过在NSGA-III的精英选择算子中引入Deb约束准则,扩展了NSGA-III在约束问题中的应用。文献[12]对MOEA/DD更新过程中融入了Deb约束准则,而其他部分则保持不变,成功地将MOEA/DD引入约束多目标领域,提出了C-MOEA/DD。这3种算法在支配关系选择中,当两个解都是非可行解时,选择保留稀疏区域的非可行解;当一个解为可行解,另一个为非可行解时,仍选择可行解。这导致算法过于保护可行解,摒弃在稀疏区域且靠近可行域的非可行解。但是,在求解约束高维多目标问题时,保护在稀疏区域且靠近可行域的非可行解对于提高种群的分布性至关重要。文献[13]将Deb约束准则与NSGA-III中参考点相融合,通过小生境技术和关联参考点判断支配关系,提出了一种基于参考点的约束支配关系(RPCDP),该算法通过提高稀疏区域非可行解的保护,从而提高最优解的分布性和收敛性。但RPCDP支配没有融入Pareto支配,导致该算法过于依赖参考点。在求解约束MaOPs时,以上算法在对解进行支配排序方面仍存在局限性,需要进一步深入研究。

为了提高NSGA-II求解约束高维多目标问题的收敛性、分布性和多样性,本文提出一种求解约束高维多目标优化问题的分解约束支配NSGA-II优化算法(decomposition-based constrained dominance principle NSGA-II),利用分解约束(decomposition-based constrained dominance principle, DBCDP)支配概念,在保持NSGA-II中快速非支配排序的基础上,通过基于分解中的权重向量,将目标空间均匀分割和

解到权重向量的距离与约束处理技术相融合,提高种群的分布性和收敛性,然后通过保护稀疏区域的可行解和非可行解提高种群的多样性。

## 1 基本概念

### 1.1 约束高维多目标优化问题的数学描述

约束高维多目标优化问题通用的数学模型<sup>[14]</sup>为

$$\begin{aligned} \min F(x) &= (f_1(x), \dots, f_M(x))^T. \\ \text{s.t. } g_j(x) &\geq 0, \quad j = 1, 2, \dots, P; \\ h_k(x) &= 0, \quad k = 1, 2, \dots, Q; \\ x &\in \Omega; \\ \Omega &= \prod_{i=1}^n [a_i, b_i] \subseteq R^n; \\ X &= (x_1, \dots, x_n)^T \in \Omega. \end{aligned} \quad (1)$$

其中: $M$ 为目标函数的个数且 $M$ 大于3, $P$ 为不等式约束的数目, $Q$ 为等式约束的数目, $\Omega$ 为决策变量空间, $x$ 为一个候选解, $F$ 为 $M$ 个相互冲突的目标函数向量, $R^M$ 为目标函数空间。

满足所有约束条件的解称为可行解。当存在一个约束条件不满足时,该解称为非可行解。高维多目标优化问题的解不是唯一的,而是一组均衡解,称为最优非劣解集或Pareto最优解集,并且这组解无差别。

### 1.2 NSGA-II基本思想

快速非支配排序遗传算法(NSGA-II)是在NSGA的基础上增加了精英保留策略。首先随机产生 $N$ 个个体作为父代 $P_t$ ,通过交叉变异得到 $N$ 个子代 $Q_t$ ,将子代与父代合并形成 $R_t = P_t \cup Q_t$ ,从 $R_t$ 中选出优先级高的 $N$ 个个体进入下一代。在选择之前,对 $R_t$ 进行非支配排序。令支配个体 $a$ 的个数为 $X_a$ ,支配集合 $S_a$ ,如果 $a \succ b$ ,则个体 $a$ 支配集合 $S_a = S_a \cup \{b\}$ ,支配个体 $b$ 的个数 $X_b = X_b + 1$ 。将 $X = 0$ 的个体放入 $F_1$ 中, $F_2$ 为进化种群中去掉 $F_1$ 的支配个体的非支配个体, $F_3$ 为进化种群中去掉 $F_2$ 的支配个体的非支配个体,以此类推。将每个解划分等级, $I = \{F_1 \cup F_2 \cup \dots \cup F_{l-1}\}$ , $T = \{F_1 \cup F_2 \cup \dots \cup F_l\}$ ,当集合 $I$ 的个数小于 $N$ 时, $T$ 的个数大于 $N$ , $F_l$ 称为临界层,使用拥挤度对临界层进行再排序,选出 $N$ 个最优个体进入下一代<sup>[8]</sup>。

NSGA-II在进行支配排序时,仅依靠适应度函数选择最优解。当目标维度增大时,这种支配关系会导致选择压力在接近最优前沿时明显降低,使种群中出现大量等价解和相同解。为此,本文提出在子代与父代合并后,剔除相同解,然后进行交叉和变异,直到产生 $2N$ 个不同的解,再进行排序。在排序中,使用分解和约束支配对Pareto支配出现的等价解进行惩罚,从

而提高算法的选择压力.

### 1.3 基于分解的多目标优化算法

基于帕累托支配的算法在处理高维多目标问题时性能较低,而基于惩罚的边界交叉方法(PBI)性能表现较优<sup>[15]</sup>. PBI主要是使用解与理想点之间的距离在权重向量的投影 $d_1$ 和解到权重向量的距离 $d_2$ 来评估解的性质. $d_1$ 、 $d_2$ 的计算公式如下:

$$d_1(x) = \frac{\|(F(x))^T R_i\|}{\|R_i\|}, \quad (2)$$

$$d_2(x) = \left\| F(x) - d_1(x) \frac{R_i}{\|R_i\|} \right\|. \quad (3)$$

其中

$$W = \begin{bmatrix} M + p - 1 \\ p \end{bmatrix}, \quad (4)$$

$$\begin{aligned} R &= [R_1, R_2, \dots, R_M], \\ R_i &= [R_i^1, R_i^2, \dots, R_i^M], \\ R_i^1 + R_i^2 + \dots + R_i^M &= 1, \end{aligned} \quad (5)$$

$M$ 为目标函数的个数, $W$ 为权重向量的个数, $R$ 为一组具有 $1/p$ 的均匀间距的向量, $R_i$ 表示 $M$ 维向量, $p$ 为沿着每个目标坐标上的划分线,而

$$F(x) = (F_1(x), F_2(x), \dots, F_M(x))^T \quad (6)$$

是解 $x$ 的归一化目标向量.

## 2 DBCDP-NSGA-II算法

### 2.1 权重向量选取

使用Das等<sup>[16]</sup>提出的系统方法来生成一组均匀分布的权重向量.在具有均匀间隔 $\delta = 1/P(P = 5)$ 的三目标优化问题中生成权重向量,使用式(4)得到231个均匀分布的权重向量.

### 2.2 种群关联权重向量

在生成权重向量之后,对 $R_t$ 种群个体的目标函数值进行规范化.首先确定要归一化的总体中每个目标函数 $f_i$ 的最小值 $f_i^{\min}$ 和最大值 $f_i^{\max}$ ,以构造向量

$$\begin{aligned} F^{\min} &= (f_1^{\min}, f_2^{\min}, \dots, f_M^{\min})^T, \\ F^{\max} &= (f_1^{\max}, f_2^{\max}, \dots, f_M^{\max})^T; \end{aligned} \quad (7)$$

然后对于每个解 $x$ ,用下面的公式计算归一化目标值 $F_i(x)$ :

$$F_i(x) = \frac{f_i(x) - f_i^{\min}}{f_i^{\max} - f_i^{\min}}. \quad (8)$$

在 $R_t$ 种群个体归一化之后,必须使每个个体关联到 $d_2$ 距离最短的权重向量,并记录匹配到权重向量的解的个数,使用 $RP_j(u)$ 表示分配给解 $u$ 的权重向量的解的个数,再根据 $RP_j(u)$ 判断解的密度.

## 2.3 分解约束支配策略

### 2.3.1 Deb约束支配

文献[17]对约束优化问题存在的问题和求解方法进行了概括,由Deb<sup>[18]</sup>提出的可行性法则的性能表现最好.但在处理不连通的可行域或者前沿复杂的可行域多目标问题时,如果 $x$ 、 $y$ 为非可行解,则仅采用约束违反度选取最优个体,将使算法易陷入局部最优;如果 $x$ 为可行解, $y$ 为非可行解,则选取可行解为较优个体,将使算法过于保护可行解. Deb约束支配准则中,每个个体仅根据Pareto支配、约束违反度两种信息来选取较优个体,这使在稀疏区域并靠近可行区域的非可行解被淘汰,从而导致种群探索未知领域的的能力丧失.

约束违反度计算方法<sup>[7]</sup>为

$$VFD(x) = \sum_{j=1}^P \langle g_j(x) \rangle + \sum_{k=1}^Q |h_k(x)|. \quad (9)$$

其中:当 $g(x) < 0$ 时, $g(x) = -g(x)$ ;否则 $g(x) = 0$ .当 $x$ 满足约束条件中任何条件,即 $x$ 在可行域内时, $VFD = 0$ ;当 $x$ 不完全满足约束条件,即 $x$ 不在可行域内时, $VFD \neq 0$ .  $VFD$ 越小, $x$ 越靠近可行域.

### 2.3.2 分解约束支配准则

针对上述分析,本文将Pareto支配、基于分解、约束支配这3种方法相融合,提出一种新的支配方式——分解约束支配(DBCDP支配).该支配先使用Pareto支配对解进行快速排序,再通过分解和约束支配对等价解进行惩罚. DBCDP支配主要是根据Pareto支配、约束违反度、解的密度3种信息来选取较优个体,提高对稀疏区域并靠近可行区域的非可行解的保护.

由于 $u$ 、 $v$ 两个解的性质不同,分解约束支配可分为以下3种情况:

1) 当 $u$ 、 $v$ 两个解均为可行解并为等价解时,如果 $u$ 、 $v$ 匹配到相同的权重向量,则根据 $d_1$ 的值判断解的收敛情况,选出距离理想点最近的解;如果 $u$ 、 $v$ 匹配到不相同的权重向量,则根据 $RP_j$ 的值判断解的密度,选出距离理想点最近和密度最小的解.

2) 当 $u$ 、 $v$ 两个解都为非可行解并互不Pareto支配时,如果 $u$ 、 $v$ 依附相同的权重向量,则根据约束违反度的大小判断解与可行域的距离,选择靠近可行域的解;否则,根据约束违反度和 $RP_j$ 的值选择稀疏区域的解.

3) 当 $u$ 为可行解, $v$ 为非可行解时,根据 $RP_j$ 的值判断解的密度,选出稀疏区域的解,从而提高种群探索未知领域的的能力.

根据以上 3 种情况的分析,分解约束支配关系定义如下:给定一组权重向量  $R$ ,  $u$ 、 $v$  是种群  $P$  中的两个解. 如果下列语句之一成立,则解  $u$  DBCDP 支配解  $v$ :

- 1)  $u$  和  $v$  同为可行解:
  - ①  $u$  Pareto 支配  $v$ .
  - ②  $u$  和  $v$  是 Pareto 等价解:
    - i)  $RP(u) = RP(v)$  并且  $d_1(u) < d_1(v)$ ;
    - ii)  $RP(u) \neq RP(v)$ ,  $d_1(u) < d_1(v)$  并且  $RP_j(u) < RP_j(v)$ .

如图 1 所示,  $u$ 、 $v$ 、 $y$  为可行解并且两两 Pareto 等价, 因  $RP(y) = RP(v)$  并且  $d_1(y) < d_1(v)$ , 故  $y$  DBCDP 支配  $v$ . 因  $RP(u) \neq RP(v)$ ,  $d_1(u) < d_1(v)$ , 并且  $RP_j(u) = 1 < RP_j(v) = 2$ , 故  $u$  DBCDP 支配  $v$ .

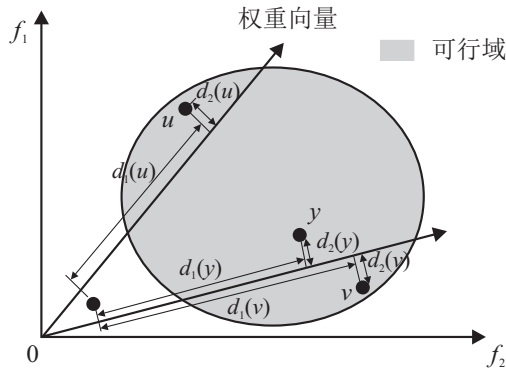


图 1  $u$ 、 $v$ 、 $y$  为可行解分布图

- 2)  $u$  和  $v$  同为非可行解:
  - ①  $u$  Pareto 支配  $v$ .
  - ②  $u$  和  $v$  是 Pareto 等价解:
    - i)  $RP(u) = RP(v)$  且  $VFD(u) < VFD(v)$ ;

ii)  $RP(u) \neq RP(v)$ ,  $VFD(u) < VFD(v)$  且  $RP_j(u) < RP_j(v)$ .

如图 2 所示,  $u$ 、 $v$ 、 $y$  为非可行解但两两 Pareto 等价, 因  $RP(y) = RP(v)$ ,  $VFD(u) < VFD(v)$ , 故  $y$  DBCDP 支配  $v$ . 因  $RP(u) \neq RP(v)$ ,  $VFD(u) < VFD(v)$ , 且  $RP_j(u) = 1 < RP_j(v) = 2$ , 故  $u$  DBCDP 支配  $v$ .

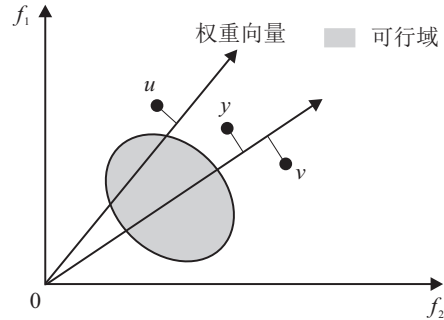


图 2  $u$ 、 $v$ 、 $y$  为非可行解分布图

3)  $u$  为可行解,  $v$  为非可行解且  $RP_j(u) < RP_j(v)$ .

为了清楚地说明 DBCDP 支配的作用, 图 3 给出了使用 Pareto 支配和 DBCDP 支配对 DTLZ8 测试问题的 19 个随机生成解进行非支配排序的结果. 对于每一个解, 在其右侧标记 DBCDP 支配排序, 左边标记帕累托支配秩(所在层数).

由图 3 可知, 构成第 1 层非支配前沿的 4 个帕累托等价解 ( $A, B, C, D$ ), 当使用 DBCDP 支配排序时, 该前沿被细分为两个 DBCDP 支配前沿: 第 1 个是  $C_1 = \{A, B, D\}$ , 包含两个极端个体  $A$  和  $D$  以及具有最小  $d_1$  距离和与最不拥挤的权重向量相关联的解  $B$ ;

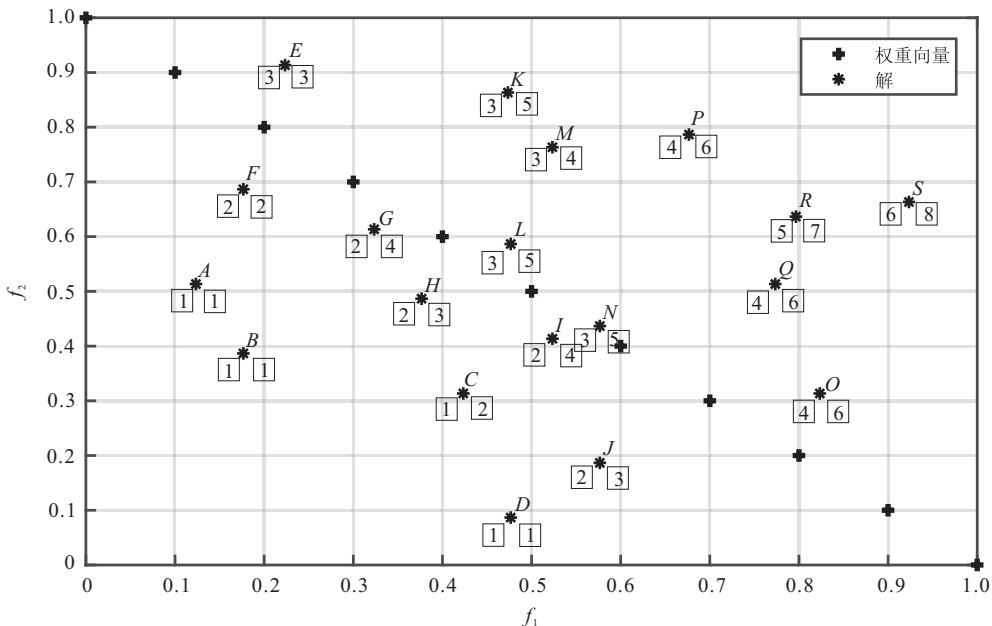


图 3 Pareto 支配和 DBCDP 支配对 DTLZ8 的 19 个随机生成解进行排序

第2个是  $C_2 = \{C\}$ , 其具有较大的距离  $d_1$  和其相关权重向量的较大密度. 根据  $d_1$  和  $RP_j$ , DBCDP 支配对某些帕累托解的秩进行了惩罚. 从图3中还可以看到 DBCDP 支配有8层解, 而 Pareto 支配有6层解, 这表明 DBCDP 支配选择压力比传统的帕累托支配更强. 从而进一步验证了 DBCDP 支配可以对帕累托等价解进行惩罚.

## 2.4 DBCDP-NSGA-II算法流程

step 1: 初始值设定. 初始种群  $P_t$ , 决策变量数  $P$ , 目标函数个数  $M$ , 种群个数  $N$ , 权重向量大小为  $H = N$ , 迭代次数为  $t = 0$ , 最大迭代次数  $D_{\max}$ . 在决策空间中随机产生  $N$  个个体, 构成初始种群  $P_t$ , 并计算其目标函数值和约束违反度. 按照 Das 提出的基于惩罚的边界交叉 (PBI) 方法生成均匀分布的权重向量集  $Z_r$ .

step 2: 对种群  $P_t$  进行交叉变异, 形成子代  $Q_t$ , 计算子代  $Q_t$  的目标函数值和约束违反度. 将父代与子代合并得到  $R_t$ , 剔除相同解个体, 再进行交叉变异, 直到产生  $2N$  个不同的个体.

step 3: 为  $R_t$  中的每个解寻找权重向量:

1) 对  $R_t$  进行归一化处理 (找出  $R_t$  中每个目标函数的最小值和最大值以形成  $F^{\min}$  和  $F^{\max}$ ), 计算  $R_t$  的理想点  $Z^*$ .

2) 计算垂直距离  $d_1$ 、 $d_2$ , 根据  $d_2$  将每个个体分配到其最近的权重向量, 并计算每个个体到权重向量的投影距离  $d_1$ . 求出每个权重向量关联解的个数.

step 4: 排序:

- 1) 根据 DBCDP 支配方式求每个个体的秩;
- 2) 根据所求的秩进行严格排序.

step 5: 锦标赛选择. 利用上述排序结果, 将  $F_1, F_2, \dots, F_{l-1}$  按顺序存储, 再对临界层进行拥挤度排序, 直到选出  $N$  个最优个体, 形成新种群  $P_{t+1}$ .

step 6: 清除种群的秩.

step 7: 如果  $t + 1$  大于最大迭代次数, 则结束运算; 否则, 返回 step 2.

## 3 实验仿真与分析

### 3.1 测试函数

本实验研究 DBCDP-NSGA-II 在约束 DTLZ<sup>[19]</sup> 测试问题上的性能. DTLZ 测试函数是用于评价高维 MOEAs 性能最广泛的测试函数之一. 本实验选取的约束 DTLZ 测试问题主要分为两类: 第1类为只有一个约束条件的 C-DTLZ1、C-DTLZ2; 第2类为有  $M$  个约束条件的 DTLZ8、DTLZ9.

C-DTLZ1: C-DTLZ1 测试函数的最优前沿值与

DTLZ1 相同都为超平面, 并且在逼近最优前沿时存在由一个约束条件导致的不可行区域障碍, 从而使算法在收敛时困难增大.

C-DTLZ2: 在 DTLZ2 问题中引入一个约束条件, 使 DTLZ2 的最优前沿变成不连通的最优前沿. 只有位于半径为  $r$  的  $(M + 1)$  超球体内的目标空间为可行域. 该函数可测试算法是否具有处理不连通前沿的能力.

DTLZ8: DTLZ8 测试函数的帕累托最优解是由一条直线和一个超平面组成, 其中直线由前  $(M - 1)$  个约束条件确定, 超平面由第  $M$  个约束条件决定.

DTLZ9: DTLZ9 测试函数的帕累托最优解是前  $(M - 1)$  个目标函数约束的交集. 在二维空间的帕累托最优解图中, 最优解是单位圆的四分之一圆弧. 大多优化算法很难找到 DTLZ9 全局最优解, 只能找到某个区域的最优解.

C-DTLZ1、C-DTLZ2、DTLZ8、DTLZ9 测试函数的决策变量均为  $[0, 1]$ , 表达式及性质见表1.

### 3.2 参数设置与算法对比

实验环境: Inter Core(TM)i5-2450M CPU, 内存为 4GB, Window10 操作系统, MatlabR2017a 版本.

参数设置: 决策变量个数 =  $10 \times$  目标函数个数, 目标函数个数  $M = 5, 8, 12, 16, 20$ , 种群大小为 100. 文献 [15] 使用覆盖 PF\* 的权重向量来保护算法多样性, 所以权重向量的数量与种群大小相同. 交叉方式选择模拟二进制, 交叉率 = 0.5, 突变率 = 0.02. 突变步长 =  $0.1 \times (\text{决策变量上限值} - \text{决策变量下限值})$ , 最大迭代次数为 100.

对比算法如下: 为了验证本文所提出的 DBCDP-NSGA-II 算法, 选择引入 Deb 约束准则的 NSGA-II (C-NSGA-II)、C-NSGA-III、C-MOEA/D、C-MOEA/DD 这 4 种约束多目标优化算法进行对比实验研究.

### 3.3 性能指标

为了比较不同算法的性能, 选择反向世代距离 (IGD)<sup>[20]</sup> 作为评价算法的性能指标. IGD 可以测量 MOEA 产生的所有近似解到真实 PF\* 的平均距离. IGD 的值越低, 表示得到的解集合越接近 PF\*. 此指标能够综合评估算法的收敛性和分布性.

### 3.4 结果分析

本文给出了 DBCDP-NSGA-II 算法在 C-DTLZ1、C-DTLZ2、DTLZ8、DTLZ9 测试问题上所得到的结果, 并与 C-NSGA-II、C-MOEA/D、C-MOEA/DD 和 C-NSGA-III 进行比较. 在结果比较中, 使用 Wilcoxon 秩和检验的方式<sup>[21]</sup> 比较与检测算法之间的差异. 在置

表 1 C-DTLZ1、C-DTLZ2、DTLZ8、DTLZ9 测试函数表达式及性质

测试函数	目标函数	约束条件	函数性质
C-DTLZ1	$\min f_1(x) = \frac{1}{2}x_1x_2 \dots x_{M-1}(1 + h(x_M))$ $\min f_2(x) = \frac{1}{2}x_1x_2 \dots (1 - x_{M-1})(1 + h(x_M))$ $\vdots$ $\min f_M(x) = \frac{1}{2}(1 - x_1)(1 + h(x_M))$ $h(x) = 100\left(10 + \sum_{i=3}^m (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right)$	$g(x) = 1 - \frac{f_M(x)}{0.6} - \sum_{i=1}^{M-1} \frac{f_i(x)}{0.5} \geq 0$	线性最优面
C-DTLZ2	$\min f_1(x) = (1 + h(X_M)) \cos\left(\frac{\pi}{2}x_1\right) \dots \cos\left(\frac{\pi}{2}x_{(M-1)}\right)$ $\min f_2(x) = (1 + h(X_M)) \cos\left(\frac{\pi}{2}x_1\right) \dots \sin\left(\frac{\pi}{2}x_{(M-1)}\right)$ $\vdots$ $\min f_M(x) = (1 + h(X_M)) \sin\left(\frac{\pi}{2}x_1\right)$ $h(x) = \sum_{i=3}^m (x_i - 0.5)^2$	$g(x) = \max \left\{ \max_{i=1}^M \left[ (f_i(x) - 1)^2 + \sum_{j=1, j \neq i}^M f_j^2 - r^2 \right], \left[ \sum_{i=1}^M (f_i(x) - 1/\sqrt{M})^2 - r^2 \right] \right\}$	凹函数
DTLZ8	$\min f_j(x) = \frac{1}{\lfloor n/M \rfloor} \sum_{\lfloor (j-1)n/M \rfloor}^{\lfloor jn/M \rfloor} x_i$ $j = 1, 2, \dots, M$	$g_j(x) = f_M(x) + 4f_j(x) - 1 \geq 0$ $j = 1, 2, \dots, M - 1$ $g_M(x) = 2f_M(x) + \min_{i,j=1, i \neq j}^{(M-1)} [f_i(x) + f_j(x)] - 1 \geq 0$	线性函数
DTLZ9	$\min f_j(x) = \sum_{\lfloor (j-1)n/M \rfloor}^{\lfloor jn/M \rfloor} (x_i)^{0.1}$ $j = 1, 2, \dots, M$	$g_j(x) = f_M^2(x) + f_j^2(x) - 1 \geq 0$ $j = 1, 2, \dots, M - 1$	非线性凹函数

信度为 95% 的情况下,用符号“+”、“-”、“=”表示统计检验的结果.“+”表示显著优于、“-”表示显著劣于、“=”表示无差异于 DBCDP-NSGA-II. 例如,表 2 中当  $M = 5$  时,C-NSGA-II 的 IGD 平均值和标准差“2.618 2e-1-”表示 DBCDP-NSGA-II 在置信度为 95% 的情况下,显著优于 C-NSGA-II.

从表 2 可知,在只有一个约束条件的 C-DTLZ1、C-DTLZ2 测试函数中,DBC DP-NSGA-II 求出 IGD

的结果相对较小,所求最优解显著优于 C-NSGA-II、C-NSGA-III、C-MOEA/D、C-MOEA/DD. 而 C-NSGA-II 和 C-MOEA/D 在 C-DTLZ1、C-DTLZ2 测试问题中所求 IGD 值较大,表明这两种算法没有准确地找到可行域的位置. 主要原因是当目标个数增加时,一个解支配另一个解的能力下降,而且这两种算法过于强调可行解支配非可行解,使算法易陷入局部最优解. 在 C-DTLZ1 的例子中当  $M = 5$  时,C-NSGA-III 显著优

表 2 C-DTLZ1、C-DTLZ2 测试函数的 IGD 平均值和标准差

测试函数	决策变量数	目标数	C-NSGA-II		C-NSGA-III		C-MOEA/D		C-MOEA/DD		DBC DP-NSGA-II	
			平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
C-DTLZ1	50	5	2.618 2e-1-	6.12e-1	<b>5.058 9e-2+</b>	<b>8.95e-1</b>	5.693 0e-2-	3.21e-1	5.156 1e-2+	7.58e-1	5.173 7e-2	5.08e-1
C-DTLZ1	80	8	2.727 8e-1-	3.87e-1	1.990 1e-1-	4.54e-1	2.893 5e-1-	6.75e-1	1.564 6e-1-	6.62e-1	<b>1.151 7e-1</b>	<b>6.92e-1</b>
C-DTLZ1	120	12	1.992 5e-1-	3.39e-1	1.600 7e-1-	7.08e-1	1.802 2e-1-	4.01e-1	1.537 4E-1-	5.33e-1	<b>1.358 9e-1</b>	<b>5.41e-1</b>
C-DTLZ1	160	16	3.153 2e-1-	5.33e-2	2.375 4e-1-	4.46e-2	2.491 3e-1-	3.74e-2	2.324 8e-1-	8.48e-2	<b>2.232 1e-1</b>	<b>4.30e-2</b>
C-DTLZ1	200	20	3.957 7e-1-	7.70e-2	2.818 0e-1-	7.09e-1	3.253 0e-1-	4.32e-1	2.438 3e-1-	7.71e-1	<b>2.393 5e-1</b>	<b>7.87e-1</b>
C-DTLZ2	50	5	4.007 1e-1-	1.33e-1	1.912 6e-1-	8.05e-1	2.569 0e-1-	2.11e-1	1.899 7e-1-	5.24e-1	<b>1.326 6e-1</b>	<b>5.99e-2</b>
C-DTLZ2	80	8	3.592 5e-1-	3.39e-1	2.400 7e-1-	7.08e-1	3.802 2e-1-	2.01e-1	2.537 4e-1-	7.33e-1	<b>2.286 7e-1</b>	<b>5.11e-2</b>
C-DTLZ2	120	12	3.727 8e-1-	3.87e-1	2.693 5e-1-	6.75e-1	2.990 1e-1-	4.54e-1	2.564 6e-1-	6.62e-1	<b>2.420 0e-1</b>	<b>3.74e-2</b>
C-DTLZ2	160	16	3.618 2e-1-	6.12e-1	2.951 9e-1-	8.95e-1	3.693 0e-1-	9.21e-1	2.136 1e-1-	6.58e-1	<b>2.108 9e-1</b>	<b>6.21e-2</b>
C-DTLZ2	200	20	5.465 7e-1-	6.98e-2	3.894 5e-1-	6.96e-2	5.044 6e-1-	4.75e-2	3.690 9e-1-	7.10e-2	<b>3.398 3e-1</b>	<b>6.62e-2</b>

于DBC DP-NSGA-II. 对于凹函数C-DTLZ2, 当  $M = 12, 16$  时, C-MOEA/DD 取得仅次于DBC DP-NSGA-II 的最优值. 主要原因是C-NSGA-III 算法过于摒弃在稀疏区域且靠近可行域的非可行解. C-MOEA/DD 虽然在锦标赛选择阶段保留了稀疏区域的非可行解, 但在支配阶段过于强调可行解支配非可行解. DBC DP-NSGA-II 在锦标赛选择、支配两个阶段都强调了保留稀疏区域的非可行解. 因此才能取得更好的结果.

从表3中可看出, 对于第2类有  $M$  个约束条件的DTLZ8、DTLZ9测试问题, DBC DP-NSGA-II 求出IGD的结果相对较小, 所求最优解显著优于C-NSGA-II、C-NSGA-III、C-MOEA/D、C-MOEA/DD. 而C-NSGA-II、C-MOEA/D 仍没取得较好的最优解. 在由一条直

线和一个超平面组成的DTLZ8例子中, 当  $M = 12$  时, C-MOEA/DD 取得最优前沿值, C-NSGA-III 结果显著优于DBC DP-NSGA-II. 当  $M = 5, 16, 20$  时, C-MOEA/DD 结果仅次于DBC DP-NSGA-II. 对于Pareto前沿由一条曲线组成的DTLZ9测试函数, 当  $M = 5, 12$  时, C-MOEA/DD 仅次于DBC DP-NSGA-II 的结果. 当  $M = 16$  时, C-NSGA-III 显著优于DBC DP-NSGA-II 的结果. C-NSGA-III、C-MOEA/DD 都通过参考点将目标空间均匀分割, 并将解与参考点一一对应. DBC DP-NSGA-II 采用了均匀分布的权重向量, 将解分配到离自己最近的区域内, 不存在一一对应关系, 从而提高了种群的均匀分布.

表3 DTLZ8、DTLZ9测试函数的IGD平均值和标准差

测试函数	决策变量数	目标数	C-NSGA-II		C-NSGA-III		C-MOEA/D		C-MOEA/DD		DBC DP-NSGA-II	
			平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
DTLZ8	50	5	4.684 6e-1	4.16e-2	2.613 2e-1	6.52e-2	3.668 6e-1	2.14e-2	2.529 3e-1	4.18e-2	<b>2.279 6e-1</b>	<b>4.20e-2</b>
DTLZ8	80	8	5.562 8e-1	4.58e-2	2.468 6e-1	7.79e-2	4.803 6e-1	3.56e-2	2.889 0e-1	5.73e-2	<b>2.420 1e-1</b>	<b>5.95e-2</b>
DTLZ8	120	12	5.759 1e-1	4.57e-2	2.192 9e-1	7.50e-2	5.237 9e-1	2.89e-2	<b>2.045 4e-1</b>	<b>5.63e-2</b>	2.660 6e-1	6.40e-2
DTLZ8	160	16	6.545 9e-1	4.65e-2	3.726 3e-1	5.71e-2	5.584 7e-1	5.60e-2	3.603 6e-1	4.38e-2	<b>3.571 5e-1</b>	<b>6.05e-2</b>
DTLZ8	200	20	7.799 6e-1	4.76e-2	4.718 1e-1	5.20e-1	7.030 3e-1	5.43e-2	4.324 1e-1	4.71e-2	<b>4.287 6e-1</b>	<b>6.51e-2</b>
DTLZ9	50	5	4.096 8e-1	1.04e-1	3.629 6e-1	6.50e-1	3.974 6e-1	2.73e-1	3.569 0e-1	6.62e-1	<b>3.013 1e-1</b>	<b>6.90e-1</b>
DTLZ9	80	8	5.545 1e-1	2.01e-1	4.498 7e-1	3.20e-1	5.143 8e-1	2.04e-1	4.594 5e-1	5.83e-1	<b>4.212 0e-1</b>	<b>7.64e-1</b>
DTLZ9	120	12	5.052 5e-1	2.77e-1	3.530 8e-1	5.13e-1	4.761 6e-1	7.79e-1	3.520 0e-1	6.35e-1	<b>3.440 7e-1</b>	<b>5.31e-1</b>
DTLZ9	160	16	5.932 9e-1	4.20e-1	<b>4.819 8e-1</b>	<b>6.21e-1</b>	5.471 4e-1	7.99e-1	4.881 3e-1	7.59e-1	4.864 5e-1	6.91e-1
DTLZ9	200	20	6.792 1e-1	6.43e-1	5.558 4e-1	7.84e-1	6.868 9e-1	4.55e-1	5.653 4e-1	7.58e-1	<b>5.265 1e-1</b>	<b>7.18e-1</b>

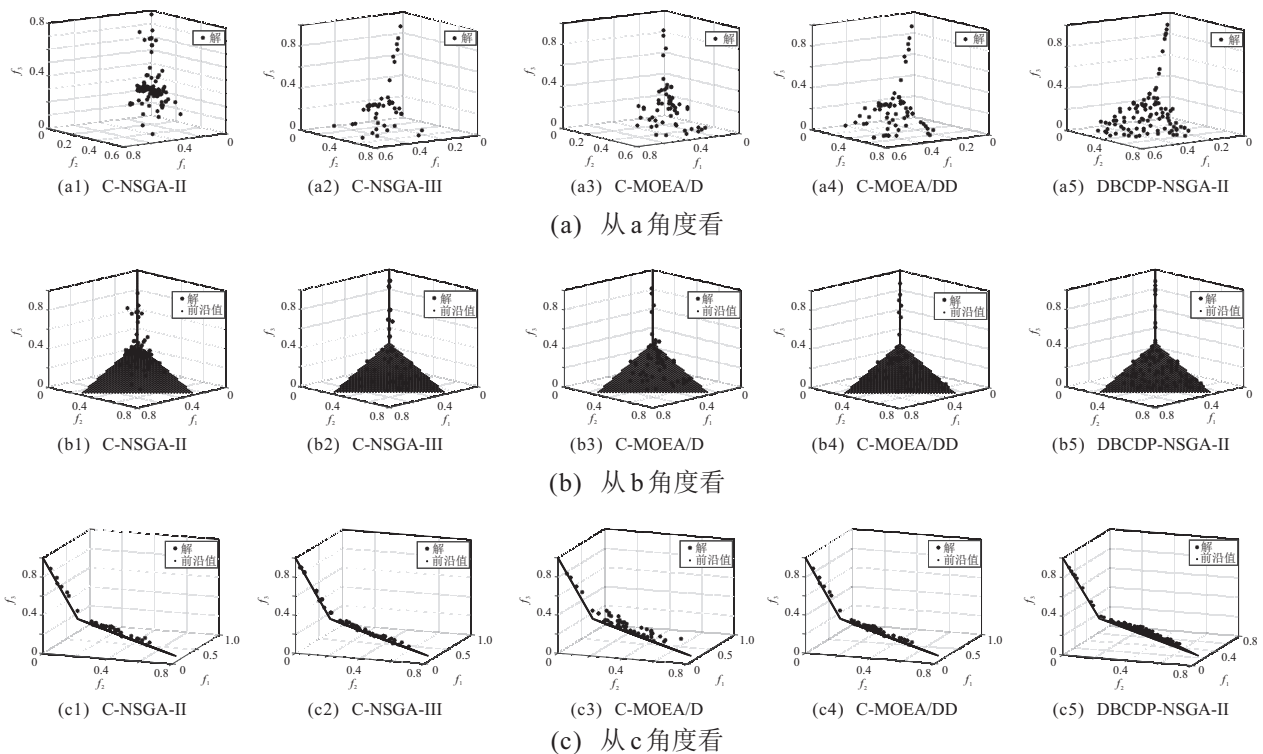


图4 C-NSGA-II、C-NSGA-III、C-MOEA/D、C-MOEA/DD、DBC DP-NSGA-II在DTLZ8测试函数中的结果

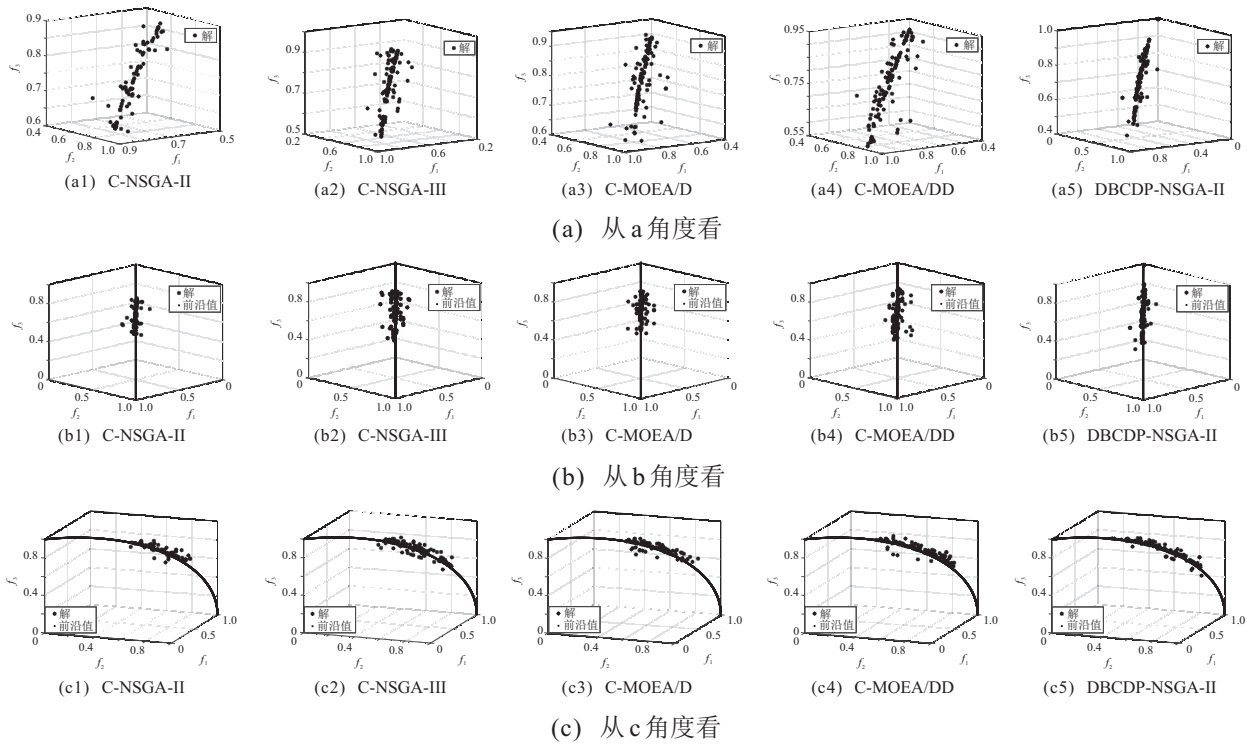


图5 C-NSGA-II、C-NSGA-III、C-MOEA/D、C-MOEA/DD、DBCDP-NSGA-II在DTLZ9测试函数中的结果

图4从a、b、c这三个角度展示了当  $M = 3$  时迭代50次C-NSGA-II、C-NSGA-III、C-MOEA/D、C-MOEA/DD、DBCDP-NSGA-II的DTLZ8测试函数结果. 这些结果显示:C-NSGA-II解的分布性差;而C-NSGA-III、C-MOEA/D、C-MOEA/DD、DBCDP-NSGA-II的解在直线和超平面上分布较均匀.

图5从a、b、c这三个角度展示了当  $M = 3$  时迭代50次C-NSGA-II、C-NSGA-III、C-MOEA/D、C-MOEA/DD、DBCDP-NSGA-II的DTLZ9测试函数结果. 这些结果显示,C-NSGA-III、C-MOEA/DD、DBCDP-NSGA-II分布较广,并且DBCDP-NSGA-II在曲线的端点处找到2个解.

从Wilcoxon秩和检验的结果中可以得出,在20组数学实验中,DBCDP-NSGA-II比C-NSGA-II优20次,DBCDP-NSGA-II比C-NSGA-III优17次,DBCDP-NSGA-II比C-MOEA/D优20次,DBCDP-NSGA-II比C-MOEA/DD优18次.由此可以看出,在C-DTLZ1、C-DTLZ2、DTLZ8、DTLZ9测试问题中,DBCDP-NSGA-II相对于C-NSGA-II、C-NSGA-III、C-MOEA/D、C-MOEA/DD而言,在IGD指标上取得了较好的结果.

从表2和表3、图4和图5关于收敛性和分布性的综合结果得知,DBCDP-NSGA-II具有较好的整体性能.仿真结果表明,本文采用Pareto支配与基于分解的方法相融合的DBCDP支配,成功地发挥了两种

方法各自的优势.

### 4 结论

本文综合考虑Pareto支配、分解和约束支配3种方法的优势,提出了一种分解约束支配NSGA-II优化算法.在支配排序中,进行了Pareto支配排序,并从解到权重向量的距离、解与理想点之间的距离在权重向量的投影、解的密度3个方面对等价解进行了惩罚和严格排序,从而保留稀疏区域的可行解和非可行解,提高了种群的收敛性和分布性.综合分析了该算法与其他4种算法在测试函数约束DTLZ上的结果比较,结果分析表明,所提出的算法能够提供整体性能最优的结果.下一步的研究方向将以新型支配关系来代替传统的Pareto支配求解约束高维多目标优化问题.

### 参考文献(References)

- [1] Li B, Li J, Tang K, et al. Many-objective evolutionary algorithms: A survey[J]. *Acm Computing Surveys*, 2015, 48(1): 1-35.
- [2] Christian von Lüken, Benjamín Barán, Brizuela C. A survey on multi-objective evolutionary algorithms for many-objective problems[J]. *Computational Optimization and Applications*, 2014, 58(3): 707-756.
- [3] Mendes J B, Maia N A, Veloso R R. A hybrid multi-objective evolutionary algorithm for truck dispatching in open-pit-mining[J]. *IEEE Latin America Transactions*, 2016, 14(3): 1329-1334.

- [4] Matrosova E S, Huskova I, Kasprzyk J R, et al. Many-objective optimization and visual analytics reveal key trade-offs for London's water supply[J]. *Journal of Hydrology*, 2015, 531: 1040-1053.
- [5] Fan Z, Li W, Cai X, et al. Push and pull search for solving constrained multi-objective optimization problems[J]. *Swarm and Evolutionary Computation*, 2019, 44(1): 665-679.
- [6] Zhang Q, Li H. MOEA/D: A multi-objective evolutionary algorithm based on decomposition[J]. *IEEE Transactions on Evolutionary Computation*, 2008, 11(6): 712-731.
- [7] Li H, Zhang Q. Multi-objective optimization problems with complicated pareto sets, MOEA/D and NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 284-302.
- [8] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [9] Takahama T, Sakai S. Constrained optimization by epsilon constrained particle swarm optimizer with epsilon-level control[C]. *Soft Computing as Transdisciplinary Science and Technology*. Berlin, Heidelberg: Springer, 2005: 1019-1029.
- [10] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2000, 4(3): 284-294.
- [11] Jain H, Deb K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 602-622.
- [12] 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.
- [13] 毕晓君, 王朝. 一种基于参考点约束支配的NSGA-算法[J]. *控制与决策*, 2019, 34(2): 369-376.  
(Bi X J, Wang C. A reference point constrained dominance-based NSGA-III algorithm[J]. *Control and Decision*, 2019, 34(2): 369-376.)
- [14] 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): 114-122.  
(Liu J C, Li F, Wang H H, et al. Survey on evolutionary many-objective optimization algorithms[J]. *Control and Decision*, 2018, 33(5): 114-122.)
- [15] Elarbi M, Bechikh S, Gupta A, et al. A new decomposition-based NSGA-II for many-objective optimization[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017, 48(7): 1191-1210.
- [16] Das I, Dennis J E. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems[J]. *Siam Journal on Optimization*, 1998, 8(3): 631-657.
- [17] 李智勇, 黄滔, 陈少森, 等. 约束优化进化算法综述[J]. *软件学报*, 2017, 28(6): 1529-1546.  
(Li Z Y, Huang T, Chen S M, et al. Overview of constrained optimization evolutionary algorithms[J]. *Journal of Software*, 2017, 28(6): 1529-1546.)
- [18] Deb K. An efficient constraint handling method for genetic algorithms[J]. *Computer Methods in Applied Mechanics and Engineering*, 2000, 186(2/4): 311-338.
- [19] Deb K, Thiele L, Laumanns M, et al. Scalable test problems for evolutionary multi-objective optimization[C]. *Evolutionary Multi-objective Optimization*. London: Springer, 2005: 105-145.
- [20] Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multi objective optimizers: An analysis and review[J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 117-132.
- [21] Derrac J, García S, Molina D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. *Swarm & Evolutionary Computation*, 2011, 1(1): 3-18.

### 作者简介

顾清华(1981—), 男, 教授, 博士生导师, 从事计算智能、群智能优化算法、复杂工业系统建模及仿真等研究, E-mail: qinghuagu@126.com;

莫明慧(1994—), 女, 硕士生, 从事计算智能算法理论的研究, E-mail: 1281800549@qq.com;

卢才武(1965—), 男, 教授, 博士生导师, 从事计算智能、群智能优化算法、复杂工业系统建模及仿真等研究, E-mail: 1114238783@qq.com;

陈露(1993—), 男, 博士生, 从事智能算法理论的研究, E-mail: 1015435316@qq.com.

(责任编辑: 李君玲)