

# 一种基于偶图匹配的多目标分解进化算法

李智翔<sup>†</sup>, 贺 亮, 韩杰思, 游 凌

(盲信号处理重点实验室, 成都 610041)

**摘 要:** 针对基于分解的多目标进化(MOEA/D)算法在选择下一代解时未考虑解和子问题之间的相对距离, 可能导致算法得到的最终解多样性较差的问题, 提出一种基于偶图匹配的多目标分解进化(MOEA/D-BM)算法. 所提算法利用偶图匹配模型对解和子问题的相互关系进行建模, 在选择下一代解的同时, 考虑收敛性和多样性, 以提高算法性能. 通过与其他 3 种经典的多目标分解进化算法在多个测试函数上进行实验, 验证了所提出算法的有效性.

**关键词:** 进化计算; 多目标优化; 选择策略; 偶图匹配

中图分类号: TP391

文献标志码: A

## A bigraph matching method for decomposition multiobjective optimization

LI Zhi-xiang<sup>†</sup>, HE Liang, HAN Jie-si, YOU Ling

(National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu 610041, China)

**Abstract:** The multiobjective evolutionary algorithm based on decomposition(MOEA/D) selects next generation without considering the distance between solutions and subproblems, which may yield a result of poor diversity. Therefore, the MOEA/D with bigraph matching(MOEA/D-BM) is proposed. The algorithm models the relationship between solutions and subproblems through bigraph matching, and thus improves the performance of the algorithm by considering the convergence and diversity simultaneously in next generation selection. The experimental results on a variety of test instances show that the proposed algorithm is effective in comparison with 3 state-of-the-art decomposition-based multiobjective evolutionary algorithms.

**Keywords:** evolutionary computing; multiobjective optimization; selection strategy; bigraph matching

### 0 引 言

在绝大多数多目标优化问题中, 不同的优化目标之间存在相互冲突, 即不存在一个最优解使得所有  $m$  维优化目标均最优. 因此, 决策者通常需要寻找在某种度量方式下能够平衡不同优化目标的最优解集. 为了解决这一问题, 有学者提出了一个针对多目标优化问题的最优解的概念——Pareto 最优解<sup>[1]</sup>.

一个多目标优化问题(Multiobjective optimization problem, MOP)可以表示为

$$\begin{aligned} \min \mathbf{F}(\mathbf{x}) &= (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T; \\ \text{s.t. } \mathbf{x} &\in \Omega. \end{aligned} \quad (1)$$

其中:  $\Omega$  表示搜索空间的可行域;  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  表示  $n$  维的决策向量; 在函数  $\mathbf{F} : \Omega \rightarrow \mathbf{R}^m$  中,  $m$  表

示目标函数  $f_1(\mathbf{x}), \dots, f_m(\mathbf{x})$  的数量,  $\mathbf{R}^m$  表示目标空间.

令  $\mathbf{x}, \mathbf{y} \in \Omega$ , 若对于任意  $i \in \{1, 2, \dots, m\}$ , 有  $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ , 且存在  $j \in \{1, 2, \dots, m\}$  使得  $f_j(\mathbf{x}) < f_j(\mathbf{y})$ , 则称  $\mathbf{x}$  支配  $\mathbf{y}$ , 记作  $\mathbf{F}(\mathbf{x}) \prec \mathbf{F}(\mathbf{y})$ . 式(1)中, 对于解  $\mathbf{x}^* \in \Omega$  而言, 若不存在  $\mathbf{x} \in \Omega$  使得  $\mathbf{F}(\mathbf{x}) \prec \mathbf{F}(\mathbf{x}^*)$ , 则解  $\mathbf{x}^*$  称作 Pareto 最优解. 所有的 Pareto 最优解组成的集合被称作 Pareto 集(Pareto set), 简称 PS. PS 中的解对应应在目标空间中的值组成的集合称作 Pareto 前沿(Pareto front), 即  $\text{PF} = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in \text{PS}\}$ .

决策者面对的实际问题往往具有较为复杂的形式, 传统的优化求解方法不能很好地解决这些问题. 而事实上, 决策者只需要得到 PF 的近似解集, 并不需要得到精确的 PF. 一般情况下, 决策者关注近似性和

收稿日期: 2017-05-29; 修回日期: 2017-08-01.

基金项目: 国家自然科学基金项目(61403301, 61221063); 中国博士后科学基金项目(2015T81032).

责任编委: 陈家伟.

作者简介: 李智翔(1989-), 男, 博士生, 从事进化优化算法及其应用的研究; 游凌(1971-), 男, 高级工程师, 博士生导师, 从事智能信息处理等研究.

<sup>†</sup>通讯作者. E-mail: lzxest@126.com

多样性两个指标,这两个指标分别衡量了近似解集与PF的距离和分布均匀情况.基于上述原因,采用启发式的优化算法处理多目标优化问题是一个好的思路,其中多目标进化算法(Multiobjective evolutionary algorithms, MOEAs)由于其天然的并行特性,在过去的20年中吸引了绝大多数研究者的目光,并形成了一系列优秀算法<sup>[2-3]</sup>.这些多目标进化算法大致可分为基于支配关系、基于标示和基于分解3类.

随着处理的优化问题日益复杂,基于支配关系和基于标示两种方式的多目标进化算法逐渐暴露出一些不足之处<sup>[4]</sup>.在过去10年左右的时间里,基于分解框架的MOEA吸引了大多数研究者的目光,并陆续提出了MOEA/D<sup>[5]</sup>以及其他优秀的改进算法,包括MOEA/D-DE<sup>[6]</sup>、MOEA/D-DRA<sup>[7]</sup>、MOEA/D-STM<sup>[8]</sup>、MOEA/DD<sup>[9]</sup>、NSGA-III<sup>[10]</sup>、MOEA/D-AGR<sup>[11]</sup>等.对于基于分解的多目标进化算法而言,选择策略是子问题分别选择适合解的过程;在选择解时,绝大多数MOEA/D改进算法只考虑了收敛性,而用子问题之间的区别来控制多样性.MOEA/D-STM在子问题与解之间建立了互倾向的评价指标,利用解对子问题的倾向值显性地控制多样性,取得了较好的效果.在文献[8]的基础上,本文从处理子问题与解的相互关系角度出发,将解的选择建模成有权偶图最优匹配问题进行求解,从而在平衡多样性和收敛性的前提下,为每个子问题分配一个适合解.在此基础上,本文提出一种新的算法MOEA/D-BM(Multiobjective evolutionary algorithm based on decomposition with bigraph matching).通过与其他算法在标准测试集函数上的对比实验,验证了本文所提出算法的有效性.

## 1 多目标分解进化算法

本节简要介绍多目标分解进化算法MOEA/D的核心思想,给出算法的基本流程,并对算法中使用的选择策略存在的缺陷进行分析.

### 1.1 多目标分解进化算法的实现

在多目标分解进化算法MOEA/D中,多目标优化问题被分解成 $N$ 个单目标优化子问题,每个子问题对应一个权重向量 $\lambda = (\lambda_1, \dots, \lambda_m)$ ,其中 $\lambda_i \geq 0, i = 1, 2, \dots, m$ ,且 $\sum_{i=1}^m \lambda_i = 1$ .MOEA/D中研究了3种不同的分解方法,分别是加权求和法(Weighted sum)、切比雪夫法(Tchebycheff)和惩罚边界法(Penalty-based boundary intersection, PBI).本文考虑使用切比雪夫法,其定义与文献[5]中稍有不同,通过修改使得该方法能够生成均匀分布的解<sup>[12]</sup>.

$$\begin{aligned} \min g^{\text{tch}}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{z}^{**}) &= \max_{1 \leq i \leq m} \{|f_i(\mathbf{x}) - z_i^{**}|/\lambda_i\}; \\ \text{s.t. } \mathbf{x} &\in \Omega. \end{aligned} \quad (2)$$

其中: $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_m)^T$ 是预先设定的均匀分布的权重向量,对于所有 $i \in \{1, \dots, m\}$ ,有 $\lambda_i \geq 0$ ,且 $\sum_{i=1}^m \lambda_i = 1$ .当 $\lambda_i = 0$ 时, $\lambda_i$ 被设置成一个很小的值 $10^{-6}$ ,使得除数不为0. $\mathbf{z}^{**} = (z_1^{**}, \dots, z_m^{**})^T$ ,对于所有 $i \in \{1, \dots, m\}$ ,有 $z_i^{**} = z_i^* - 10^{-6}$ , $z_i^* = \min_{\mathbf{x} \in \Omega} f_i(\mathbf{x})$ .当 $\boldsymbol{\lambda}$ 的值确定时,式(2)的最优解即为原多目标优化问题的一个Pareto最优解.通过改变 $\boldsymbol{\lambda}$ 的值,切比雪夫法能够找到不同的Pareto最优解.

MOEA/D的一个核心思想是构建子问题之间的邻居关系.通常来讲,由于权重向量的距离较小,子问题与对应邻居子问题拥有相似的最优解.因此,MOEA/D提出了邻居的思想,并针对繁殖交配和替换选择维护了邻居集合,在选择交配个体和替换个体时均从相应的邻居集合里选择.

在MOEA/D中,多目标优化问题被分解成 $N$ 个单目标优化子问题,每个子问题 $p^i$ 对应一个权重向量 $\lambda$ ,算法维护一个大小为 $N$ 的种群,其中每一个个体对应一个子问题.MOEA/D计算权重向量之间的距离,依照距离的大小为每个子问题构建邻居集合.每次得到下一代的新个体时,都在其对应子问题的邻居子问题上与现有个体比较,择优替换.MOEA/D的基本流程如下.

Step 1: 初始化种群、权重向量、邻居大小等参数.

Step 2: 进行繁殖交配.利用该子问题对应的个体解 $\mathbf{x}^i$ 与邻居集合 $T_m(i)$ 中随机选取的解产生下一代解 $\bar{\mathbf{x}}^i$ .

Step 3: 进行选择替换.对于邻居集合 $T_r(i)$ 中每一个个体解 $\mathbf{x}^j$ ,如果针对子问题 $p^j$ 对应的目标函数 $g_j(\mathbf{x}^j)$ ,有 $g_j(\bar{\mathbf{x}}^i) < g_j(\mathbf{x}^j)$ ,则用 $\bar{\mathbf{x}}^i$ 替换 $\mathbf{x}^j$ .

Step 4: 迭代次数加1.若算法达到终止条件,则返回结果;否则返回Step 2.

### 1.2 选择策略分析

在MOEA/D以及一些改进算法中,对下一代解的选择是基于子问题的取值,也就是说,这些算法的选择策略显式地考虑收敛性,而利用不同子问题的区别和分布对多样性进行控制.如图1所示,解 $a$ 和解 $b$ 分别是子问题 $p^1$ 和 $p^2$ 产生的解,对于子问题 $p^2$ 而言,当采用切比雪夫法作为分解方法时,由于解 $a$ 在子问题 $p^2$ 上的取值小于解 $b$ ,故解 $a$ 优于解 $b$ .然而,如果衡量解到子问题的垂直距离,则解 $a$ 接近子问题 $p^1$ ,而解 $b$ 接近子问题 $p^2$ ;如果算法选择解 $a$ 作为子问题 $p^2$

的下一代解,则子问题 $p^2$ 最终得到的最优解会接近于子问题 $p^1$ 与PF交点附近,这样的解与子问题 $p^1$ 本身的解距离较小,如此一来,算法最终得到解集的多样性会受到影响.为了得到分布均匀的解集,本文试图在考虑解的子问题取值的基础上,考虑解到子问题的距离这一关键因素,从而在解的选择上综合考虑收敛性和多样性.

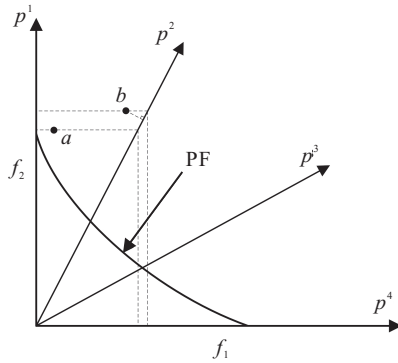


图1 下一代解的选择

## 2 基于偶图匹配的多目标分解进化算法

基于上一节对多目标分解进化算法中选择策略的分析,本文设计了一种基于有权偶图最优匹配的模式,用于在子问题和解之间建立关联关系.本文提出的选择策略同时考虑解在子问题上的适应值以及解与子问题的垂直距离两个因素,即能够平衡解的收敛性和多样性两种需求.

### 2.1 有权偶图的最优匹配

MOEA/D中的选择策略即为每一个子问题选择一个适合的解.假设对于一个多目标优化问题,有 $N$ 个子问题和 $M$ 个解,子问题与解两两之间均有一个确定的适应值,这样的关系可以用有权偶图进行建模.如图2所示,上部的点表示 $N$ 个不同的子问题,下部的点表示 $M$ 个不同的解,子问题与解之间存在赋权边的连接.

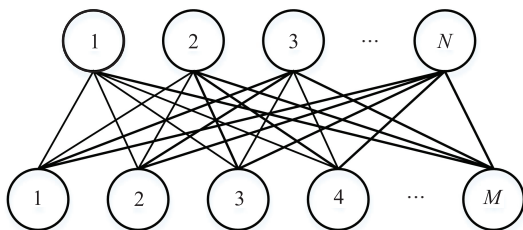


图2 由子问题和解构成的有权偶图匹配问题

试图找到 $N$ 条赋权边连接 $N$ 个不同的子问题和 $M$ 个解中的 $N$ 个不同的解,使得边的权值之和最大.为了给 $N$ 个子问题找到 $N$ 个适合的解,可以使用Kuhn-munkres(KM)算法进行求解. KM算法的实现

是基于偶图完美匹配求解的经典算法——匈牙利算法,通过给定并不断修改偶图的可行顶点标号,直到在某个相等子图中找到一个完美匹配为止. KM算法的具体步骤参见文献[13-14].对于本文构建的有权偶图而言,通常有 $M > N$ ,故需要使用KM算法找到 $N$ 组匹配. KM算法得到的最优匹配是具有最大权和的匹配.下面对边的赋权进行分析.

### 2.2 有权偶图的赋权

令 $d^\perp(x^i, p^j)$ 表示解 $x^i$ 到子问题 $p^j$ 的垂直距离,则有

$$d^\perp(x^i, p^j) = \left\| F(x^i) - \frac{\lambda^T F(x^i)}{\lambda^T \lambda} \lambda \right\|, \quad (3)$$

其中 $\lambda$ 表示子问题 $p^j$ 对应的权重向量.式(3)的结果即为解 $x^i$ 在目标空间上的取值 $F(x^i)$ 与子问题 $p^j$ 对应的权重向量 $\lambda$ 的欧氏距离.例如图1所示的二维目标空间,解 $b$ 到子问题 $p^2$ 的距离即为点到射线的垂线段长度.

对每个解 $x^i$ 依照式(3)计算与子问题的垂直距离,并将距离最小的 $K_v$ 个子问题的编号保存下来,构成长度为 $K_v$ 的数组 $\varphi_i$ ,即

$$\varphi_i = \{p^j \mid \text{rank}_{1 \leq j \leq N}(d^\perp(x^i, p^j)) \leq K_v\}. \quad (4)$$

将数组 $\varphi_i$ 每个解对应若干子问题的形式转换为每个子问题 $p^j$ 构造一个包含若干解的数组 $\psi$ ,依照子问题的取值排序,控制其大小不超过 $K_e$ ,有

$$\psi_j = \{x^i \mid p^j \in \varphi_i \wedge \text{rank}_{1 \leq i \leq N}(g^{\text{tch}}(x^i | \lambda^j, z^{**})) \leq K_e\}. \quad (5)$$

若数组 $\psi_j$ 的大小小于 $K_e$ ,则在剩下的解中选择子问题取值较优的解依次补充,直到数组大小等于 $K_e$ .于是,对连接解 $x^i$ 和子问题 $p^j$ 的边 $e_{ji}$ 赋值如下:

$$e_{ji} = \begin{cases} K_e - k, & x^i \in \psi_j, \psi_j(k) = x^i; \\ 0, & \text{Otherwise.} \end{cases} \quad (6)$$

本文首先根据解与权重向量的垂直距离将权重向量对每个解进行排序,这里只考虑选择与权重向量距离较近的若干解,体现了对多样性的要求;之后,将上述结果表示成不同解对应权重向量的形式,由于搜索空间的复杂性,可能出现不同权重向量对应解的数量差别很大的结果,算法为每个权重向量对应解的数量设定一个限值,依照子问题的取值对解进行补充或筛选,并进行排序,这里的顺序即权重向量选择解的优先次序;最后,根据上述次序得到权重向量与解之间边的赋值,应用KM算法获得最优匹配,从而得到选择的下一代解.

### 2.3 繁殖算子

差分进化算子和多项式变异算子已经被证明对于大多数多目标优化问题是高效的. 设当前解  $\mathbf{x}^j$  为第 1 个参与交配的解  $\mathbf{x}^{r_1}$ , 其他两个参与解  $\mathbf{x}^{r_2}$ 、 $\mathbf{x}^{r_3}$  均从交配池中选取, 则根据差分杂交公式得到的中间解  $\mathbf{y}^i = (y_1^i, \dots, y_n^i)$  计算方式如下:

$$y_k^i = \begin{cases} x_k^{r_1} + F \times (x_k^{r_2} - x_k^{r_3}), & \text{rand} \leq \text{CR}; \\ x_k^{r_1}, & \text{Otherwise.} \end{cases} \quad (7)$$

其中: CR 和  $F$  是控制参数, rand 是  $[0, 1]$  之间的随机数. 下一步, 依照多项式变异公式计算新解  $\bar{\mathbf{x}}^i = (\bar{x}_1^i, \dots, \bar{x}_n^i)$ , 如下所示:

$$\bar{x}_k^i = \begin{cases} \bar{y}_k^i + \sigma_k \times (b_k - a_k), & \text{rand} \leq p_m; \\ \bar{y}_k^i, & \text{Otherwise.} \end{cases} \quad (8)$$

其中

$$\sigma_k = \begin{cases} (2 \times \text{rand})^{\frac{1}{\eta+1}} - 1, & \text{rand} \leq 0.5; \\ 1 - (2 - 2 \times \text{rand})^{\frac{1}{\eta+1}}, & \text{Otherwise.} \end{cases} \quad (9)$$

$\eta$  和  $p_m$  是控制参数, rand 是  $[0, 1]$  之间的随机数,  $a_k$ 、 $b_k$  是决策变量第  $k$  维的下界和上界.

### 2.4 MOEA/D-BM 算法流程

将偶图匹配的思想引入多目标分解进化算法, 构成 MOEA/D-BM 算法. 算法流程如下.

**Step 1:** 确定多目标优化问题、算法停止条件、种群大小  $N$ 、邻居集合大小  $T$  以及偶图匹配控制参数  $K_v$  和  $K_e$ . 随机初始化种群  $P \leftarrow \{x^1, \dots, x^N\}$ , 初始化权重向量  $\lambda \leftarrow \{\lambda^1, \dots, \lambda^N\}$ , 初始化理想参考点  $z^*$ . 迭代次数初始化为 0.

**Step 2:** 依照式 (7)~(9) 对当前种群进行繁殖操作, 得到新解集合.

**Step 3:** 将新解集合  $U$  与当前种群解集合组成混合解集合, 依照式 (3)~(6) 计算子问题与混合解集合中解的连接权.

**Step 4:** 根据 KM 算法为每个子问题匹配解, 从而得到下一代解集合.

**Step 5:** 迭代次数加 1, 判断算法是否达到停止条件, 若达到, 则算法停止, 输出解集作为优化结果; 否则返回 Step 2.

## 3 实验分析

### 3.1 实验设计

为了证明本文所提出算法的有效性, 选择 3 种优秀的多目标分解进化算法进行对比测试, 这 3 种算法分别是 MOEA/D-DE、MOEA/D-DRA 以及 MOEA/D-STM. 其中: MOEA/D-DE 是在经典算法 MOEA/D 的

基础上使用差分算子进行了改进; MOEA/D-DRA 对每个子问题的计算资源进行动态分配, 并获得了 CEC2009 的最佳算法奖; 而 MOEA/D-STM 首先使用了一个针对子问题和解的匹配模型来选择下一代解, 取得了一定的效果.

本文选用测试集 DTLZ<sub>1</sub> ~ DTLZ<sub>7</sub><sup>[15]</sup>、 $F_1$  ~  $F_9$ <sup>[16]</sup> 以及 MOP<sub>1</sub> ~ MOP<sub>7</sub><sup>[17]</sup>, 这些测试函数的特征如表 1 所示.

表 1 测试函数特征

测试函数	目标空间维度	决策空间维度	特征
DTLZ <sub>1</sub>	3	7	非凸, 多峰
DTLZ <sub>2</sub>	3	12	非凸
DTLZ <sub>3</sub>	3	12	非凸, 多峰
DTLZ <sub>4</sub>	3	12	非凸
DTLZ <sub>5</sub>	3	12	非凸, 降维
DTLZ <sub>6</sub>	3	12	非凸, 降维
DTLZ <sub>7</sub>	3	22	混合, 不连续, 多峰
$F_1$	2	30	非线性, 凸, 单峰
$F_2$	2	30	非线性, 凸, 多峰
$F_3$	2	30	非线性, 凸, 多峰
$F_4$	2	30	非线性, 凸, 多峰
$F_5$	2	30	非线性, 凸, 多峰
$F_6$	3	10	非线性, 非凸, 多峰
$F_7$	2	10	非线性, 凸, 多峰
$F_8$	2	10	非线性, 凸, 多峰
$F_9$	2	30	非线性, 非凸, 多峰
MOP <sub>1</sub>	2	10	非线性, 凸, 多峰
MOP <sub>2</sub>	2	10	非线性, 非凸, 多峰
MOP <sub>3</sub>	2	10	非线性, 非凸, 多峰
MOP <sub>4</sub>	2	10	非线性, 凸, 不连续, 多峰
MOP <sub>5</sub>	2	10	非线性, 凸, 多峰
MOP <sub>6</sub>	2	10	非线性, 非凸, 多峰
MOP <sub>7</sub>	2	10	非线性, 非凸, 多峰

参照文献 [5-6], 相关算法的参数设置如下: 种群大小为 300; 分解方法选择切比雪夫法, 如式 (2) 所示; 算法停止条件为目标函数计算次数, DTLZ<sub>1</sub> ~ DTLZ<sub>7</sub> 测试函数设置为 100 000,  $F_1$  ~  $F_5$  和  $F_7$  ~  $F_9$  测试函数设置为 150 000,  $F_6$  和 MOP<sub>1</sub> ~ MOP<sub>5</sub> 测试函数设置为 300 000, MOP<sub>6</sub> 和 MOP<sub>7</sub> 测试函数设置为 900 000; 繁殖操作中, 差分算子参数设置为  $\text{CR} = 1.0$ ,  $F = 0.5$ , 多项式变异算子参数设置为  $p_m = 1/n$ ,  $\eta_m = 20$ ; 邻居大小为 20; 每个算法在每个测试函数上独立运行 20 次.

与其他 3 种算法不同, 参数  $K_v$  和  $K_e$  是本文所提出的算法特有的两个参数. 这两个参数决定了有权偶图中非零权重边的数量, 从而控制了算法在选择下一代解时每个子问题需要考虑的解集大小. 若参数  $K_v$  和  $K_e$  的取值过小, 则每个子问题对应解的个数有限, 解的多样性无法得到很好的保证; 若取值过大, 则在算法性能没有本质提升的情况下, 算法的时间复杂度进一步增大. 通过多次尝试, 发现在这两个参数

的取值增大至10和20的过程中,算法得到的最终解集的IGD指标评价是逐渐变好的;继续增大两个参数的取值,则结果变化并不明显.故本文算法的参数 $K_v$ 和 $K_e$ 分别设置为10和20.

为了对不同算法的性能进行对比,本文选择了常用的性能度量方法IGD<sup>[18]</sup>对实验结果进行分析.令 $P^*$ 表示PF上均匀分布的点的集合, $S$ 表示通过算法得到的最优解集,则相应的IGD值通过下式计算:

$$IGD(S, P^*) = \sum_{x \in P^*} \text{dist}(x, S) / |P^*|. \quad (10)$$

其中: $\text{dist}(x, S)$ 表示点 $x$ 到集合 $S$ 中最近点的欧氏距离, $|P^*|$ 表示参考最优解集合 $P^*$ 的大小.在本文实验中,对二维目标函数, $|P^*|$ 的大小设置为1000;对三维目标函数, $|P^*|$ 的大小设置为10000.IGD能够同时对收敛性和多样性进行度量,IGD的值越小, $S$ 对PF的近似度越优.

### 3.2 实验结果和分析

#### 3.2.1 DTLZ<sub>1</sub> ~ DTLZ<sub>7</sub>测试函数上的实验结果和分析

DTLZ<sub>1</sub> ~ DTLZ<sub>7</sub>测试函数的PS是线性的.4种算法在DTLZ<sub>1</sub> ~ DTLZ<sub>7</sub>上的IGD均值对比结果如表2所示.

表2 DTLZ<sub>1</sub> ~ DTLZ<sub>7</sub>测试函数上IGD均值对比

算法	MOEA/D-DE	MOEA/D-DRA	MOEA/D-STM	MOEA/D-BM
DTLZ <sub>1</sub>	3.49e-04	3.49e-04	3.48e-04	<b>2.66e-04</b>
DTLZ <sub>2</sub>	4.33e-04	4.32e-04	4.32e-04	<b>3.34e-04</b>
DTLZ <sub>3</sub>	8.67e-04	7.21e-04	7.29e-04	<b>5.82e-04</b>
DTLZ <sub>4</sub>	7.76e-04	7.76e-04	7.60e-04	<b>5.07e-04</b>
DTLZ <sub>5</sub>	1.59e-05	1.58e-05	1.54e-05	<b>1.27e-05</b>
DTLZ <sub>6</sub>	<b>3.45e-05</b>	3.45e-05	3.45e-05	4.92e-05
DTLZ <sub>7</sub>	6.28e-03	<b>4.51e-03</b>	6.41e-03	6.45e-03

由表2可以看到:在7个测试函数中,本文所提出的算法MOEA/D-BM在DTLZ<sub>1</sub> ~ DTLZ<sub>5</sub>测试函数上表现最优,而MOEA/D-DE和MOEA/D-DRA分别在DTLZ<sub>6</sub>和DTLZ<sub>7</sub>上表现最优;尽管本文提出的算法在DTLZ<sub>6</sub>和DTLZ<sub>7</sub>测试函数上表现不是最优,但与最优值差距不大;MOEA/D-STM尽管没有最优值,但表现较为稳定.由于DTLZ测试集中测试函数的PS是线性的,与 $F_1 \sim F_9$ 以及MOP<sub>1</sub> ~ MOP<sub>7</sub>相关测试函数相比,DTLZ测试集难度较小,4种算法在DTLZ测试集上并无明显差距.

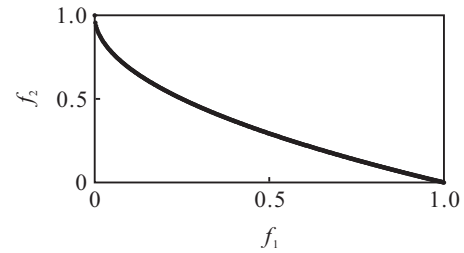
#### 3.2.2 $F_1 \sim F_9$ 测试函数上的实验结果和分析

$F_1 \sim F_9$ 是具有复杂PS的测试函数.4种算法在 $F_1 \sim F_9$ 上的IGD均值对比结果如表3所示.

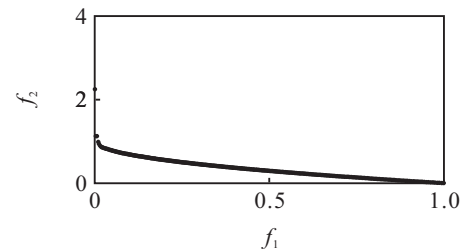
由表3可以看到:在9个测试函数中,本文算法在 $F_2, F_4 \sim F_7$ 五个测试函数上表现最优,而MOEA/D-

表3  $F_1 \sim F_9$ 测试函数上的IGD均值对比

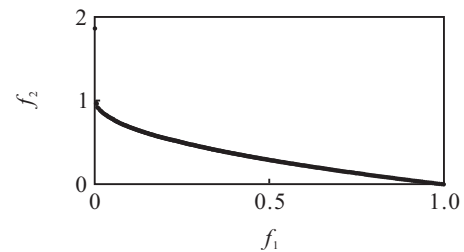
算法	MOEA/D-DE	MOEA/D-DRA	MOEA/D-STM	MOEA/D-BM
$F_1$	7.27e-05	<b>7.24e-05</b>	7.30e-05	8.36e-05
$F_2$	1.40e-04	1.51e-04	1.33e-04	<b>1.25e-04</b>
$F_3$	6.62e-04	<b>4.40e-04</b>	1.04e-03	8.69e-04
$F_4$	3.72e-04	4.20e-04	3.66e-04	<b>3.50e-04</b>
$F_5$	9.76e-04	1.11e-03	1.97e-03	<b>7.87e-04</b>
$F_6$	1.69e-03	1.84e-03	1.80e-03	<b>1.22e-03</b>
$F_7$	1.10e-03	1.53e-03	9.53e-04	<b>2.34e-04</b>
$F_8$	5.10e-03	5.31e-03	<b>4.03e-03</b>	4.13e-03
$F_9$	1.95e-04	2.17e-04	<b>1.47e-04</b>	1.84e-04



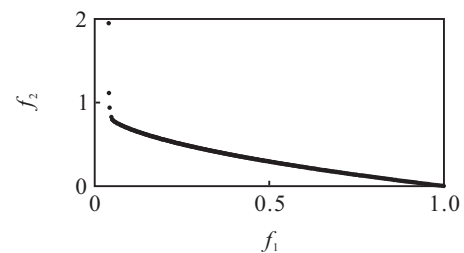
(a) PF



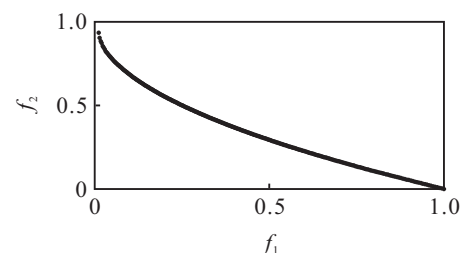
(b) MOEA/D-DE



(c) MOEA/D-DRA



(d) MOEA/D-STM



(e) MOEA/D-BM

图3 4种算法在 $F_7$ 测试函数上的结果对比

DRA在 $F_1$ 和 $F_3$ 上表现最优,MOEA-/STM在 $F_8$ 和 $F_9$ 上表现最优,本文所提出算法在测试函数 $F_5 \sim F_7$ 上占优,4种算法在其他测试函数上并无明显差距.

图3为4种算法在测试函数 $F_7$ 上运行20次后得到的最优IGD度量值结果对比.可以看出,MOEA/D-DE、MOEA/D-DRA和MOEA/D-STM均有野点存在,而本文算法MOEA/D-BM并无野点,表现较优.

### 3.2.3 MOP<sub>1</sub> ~ MOP<sub>7</sub>测试函数上的实验结果和分析

MOP<sub>1</sub> ~ MOP<sub>7</sub>测试函数同样具有较为复杂的形式.4种算法在MOP<sub>1</sub> ~ MOP<sub>7</sub>上的IGD均值对比结果如表4所示.

表4 MOP<sub>1</sub> ~ MOP<sub>7</sub>测试函数上的IGD均值对比

算法	MOEA/D-DE	MOEA/D-DRA	MOEA/D-STM	MOEA/D-BM
MOP <sub>1</sub>	1.79e-02	1.84e-02	1.84e-02	<b>6.39e-03</b>
MOP <sub>2</sub>	1.23e-02	<b>1.17e-02</b>	1.28e-02	1.22e-02
MOP <sub>3</sub>	2.26e-02	2.26e-02	<b>2.26e-02</b>	2.28e-02
MOP <sub>4</sub>	4.94e-02	4.94e-02	4.96e-02	<b>4.86e-02</b>
MOP <sub>5</sub>	1.66e-02	1.60e-02	1.61e-02	<b>2.91e-03</b>
MOP <sub>6</sub>	1.11e-02	1.02e-02	1.01e-02	<b>8.97e-03</b>
MOP <sub>7</sub>	1.31e-02	<b>1.27e-02</b>	1.33e-02	1.32e-02

由表4可以看到:本文算法在MOP<sub>1</sub>、MOP<sub>4</sub> ~ MOP<sub>6</sub>共4个测试函数上表现最优,而MOEA/D-DRA在MOP<sub>2</sub>和MOP<sub>7</sub>测试函数上最优,MOEA/D-STM在MOP<sub>3</sub>测试函数上最优.本文算法在测试函数MOP<sub>1</sub>、MOP<sub>5</sub>、MOP<sub>6</sub>上与其他3种算法相比具有优势,而4种算法在其他测试函数上并无较大差距.

图4为4种算法在测试函数MOP<sub>1</sub>上运行20次后得到的最优IGD度量值结果对比.可以看出,MOEA/D-DE、MOEA/D-DRA和MOEA/D-STM均不能对PF完全覆盖,解的多样性较差,而本文算法MOEA/D-BM基本实现了对PF的覆盖,优势明显.总体而言,MOEA/D-DE、MOEA/D-DRA以及MOEA/D-STM三种算法在不同的测试函数上得到的结果差别较小,而MOEA/D-BM能够在多个测试函数上与其他3种算法拉开差距,这说明了保持解的多样性的重要性.

### 3.2.4 4种算法的运行时间对比实验结果和分析

对4种算法在测试函数DTLZ<sub>1</sub> ~ DTLZ<sub>7</sub>上分别运行10次,对运行时间进行比较.为了消除不同性能计算机带来的影响,记录各自的总运行时间与MOEA/D-DE总运行时间的比值,结果如表5所示.

表5 算法的运行时间对比

算法	MOEA/D-DE	MOEA/D-DRA	MOEA/D-STM	MOEA/D-BM
用时比	1.00	0.97	4.48	6.91

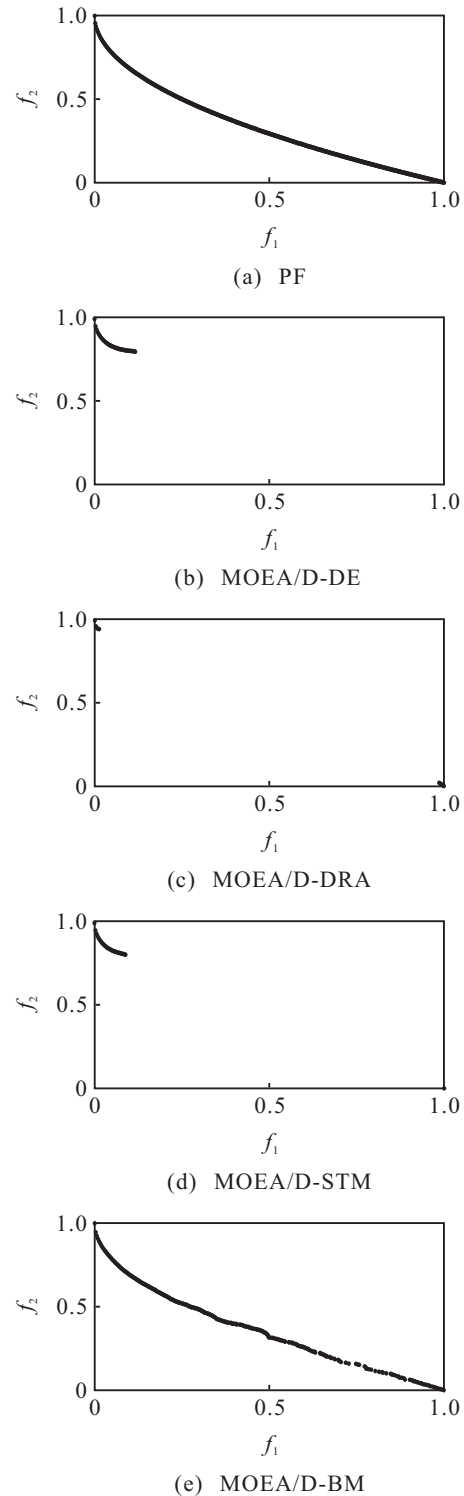


图4 4种算法在MOP<sub>1</sub>测试函数上的结果对比

从表5可以看出,MOEA/D-DE、MOEA/D-DRA的运行时间较短,MOEA/D-STM次之,本文提出的MOEA/D-BM的运行时间最长.这是由于MOEA/D-DE和MOEA/D-DRA在下一代解的选择上未考虑保持解的多样性,而后两种算法通过各自的方式考虑了在选择时保持解的多样性,在牺牲了一定时间效率的同时提升了算法的性能.与MOEA/D-STM中的匹配模型相比,本文提出的MOEA/D-BM中的有权偶图匹配模型复杂度较高,导致算法用时更长.

## 4 结论

选择策略是影响基于分解的多目标进化算法的一个重要组成部分. 目前大多数多目标分解进化算法中的选择算子并未考虑维持种群的多样性, 这可能会对最终解集的分布性产生影响. 本文从改进选择策略的角度出发, 提出了采用有权偶图最优匹配对解的选择进行建模的思路, 在解的选择过程中同时考虑了多样性和收敛性, 从种群整体最优的角度选择下一代解, 并由此提出了新的算法 MOEA/D-BM. 与其他优秀算法在公开测试集上的对比实验表明了本文所提出算法的有效性.

从实验结果可以发现, 本文提出的 MOEA/D-BM 尽管在大多数测试函数上的表现优于其他优化算法, 但是由于有权偶图匹配的时间效率较低, MOEA/D-BM 与其他算法相比耗时较长. 如何在不影响 MOEA/D-BM 性能的前提下提升算法的时间效率, 是今后研究的重点.

### 参考文献(References)

- [1] Deb K, Kalyanmoy D. Multi-objective optimization using evolutionary algorithms[M]. New York: John Wiley & Sons, 2001.
- [2] 蓝艇, 刘士荣, 顾幸生. 基于进化算法的多目标优化方法[J]. 控制与决策, 2006, 21(6): 601-605.  
(Lan T, Liu S R, Gu X S. Approaches of evolutionary multiobjective optimization[J]. Control and Decision, 2006, 21(6): 601-605.)
- [3] 公茂果, 焦李成, 杨咚咚, 等. 进化多目标优化算法研究[J]. 软件学报, 2009, 20(2): 271-289.  
(Gong M G, Jiao L C, Yang D D, et al. Research on evolutionary multi-objective optimization algorithms[J]. J of Software, 2009, 20(2): 271-289.)
- [4] Giagkiozis I, Purshouse R C, Fleming P J. An overview of population-based algorithms for multi-objective optimisation[J]. Int J of Systems Science, 2015, 46(9): 1572-1599.
- [5] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Trans on Evolutionary Computation, 2007, 11(6): 712-731.
- [6] Li H, Zhang Q. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2009, 13(2): 284-302.
- [7] Zhang Q, Liu W, Li H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances[C]. IEEE Congress on Evolutionary Computation. Trondheim: IEEE, 2009: 203-208.
- [8] Li K, Zhang Q, Kwong S, et al. Stable matching-based selection in evolutionary multiobjective optimization[J]. IEEE Trans on Evolutionary Computation, 2014, 18(6): 909-923.
- [9] Li K, Deb K, Zhang Q, et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition[J]. IEEE Trans on Evolutionary Computation, 2015, 19(5): 694-716.
- [10] 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 Trans on Evolutionary Computation, 2014, 18(4): 577-601.
- [11] Wang Z, Zhang Q, Zhou A, et al. Adaptive replacement strategies for MOEA/D[J]. IEEE Trans on Cybernetics, 2016, 46(2): 474-486.
- [12] Giagkiozis I, Purshouse R C, Fleming P J. Generalized decomposition[M]. Evolutionary Multi-Criterion Optimization. Berlin: Springer Berlin Heidelberg, 2013: 428-442.
- [13] Kuhn H W. The hungarian method for the assignment problem[M]. 50 Years of Integer Programming 1958-2008. Berlin: Springer Berlin Heidelberg, 2010: 83-97.
- [14] Munkres J. Algorithms for the assignment and transportation problems[J]. J of the Society for Industrial & Applied Mathematics, 1957, 5(1): 32-38.
- [15] Deb K, Thiele L, Laumanns M, et al. Scalable multi-objective optimization test problems[C]. Proc of the 2002 Congress on Evolutionary Computation. Honolulu: IEEE, 2002: 825-830.
- [16] Li H, Zhang Q. Comparison between NSGA-II and MOEA/D on a set of multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2009, 13(2): 284-302.
- [17] Liu H L, Gu F, Zhang Q. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems[J]. IEEE Trans on Evolutionary Computation, 2014, 18(3): 450-455.
- [18] Bosman P A N, Thierens D. The balance between proximity and diversity in multiobjective evolutionary algorithms[J]. IEEE Trans on Evolutionary Computation, 2003, 7(2): 174-188.

(责任编辑: 齐 霖)