·人工智能及识别技术· 文章编号：1000-3428(2019)03-0155-07 文献标志码：A 中图分类号：TP391

基于改进平衡策略的多目标分解优化算法

李智翔，李 赟，褚衍杰

（盲信号处理重点实验室，成都 610041）

摘 要：针对多目标优化算法在搜索中存在平衡解的收敛性和多样性问题，通过分析多目标分解进化算法，提出 2 种改进平衡策略。基于当前解和父代解的取值，设计繁殖算子，并与原有繁殖算子进行比较选出最优解。根据执行代数的不同，对邻居集合进行适应性调整。在此基础上，给出多目标分解进化算法。实验结果验证了 2 种平衡策略的有效性，同时该算法性能优于 MOEA/D 算法、NSGAII 算法和 IBEA 算法。

关键词：多目标优化；进化计算；分解方法；平衡策略；繁殖算子；邻居集合

中文引用格式：李智翔，李赟，褚衍杰. 基于改进平衡策略的多目标分解优化算法[J]. 计算机工程，2019，45(3)：155-161.

英文引用格式：LI Zhixiang，LI Yun，CHU Yanjie. Decomposition multi-objective optimization algorithm based on improved balancing strategy[J]. Computer Engineering，2019，45(3)：155-161.

Decomposition Multi-objective Optimization Algorithm Based on Improved Balancing Strategy

LI Zhixiang，LI Yun，CHU Yanjie

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

【Abstract】 Aiming at the problem that the Multi-objective Optimization（MOP） algorithm has the convergence and diversity problem of the equilibrium solution in the search，two improved balancing strategies have been proposed by analyzing multi-objective decomposition evolutionary algorithms. Based on the values of the current solution and the parent solution，a breeding operator is designed and compared with the original breeding operator to select the optimal solution. The set of neighbors is adaptively changed according to the execution algebra. On this basis，a multi-objective decomposition evolution algorithm is given. Experimental results verify the effectiveness of the two balancing strategies，and the algorithm is better than the MOEA/D algorithm，NSGAII algorithm and IBEA algorithm.

【Key words】 Multi-objective Optimization（MOP）；evolutionary computing；decomposition method；balancing strategy；reproduction operator；neighborhood set

DOI：10.19678/j.issn.1000-3428.0050049

0 概述

多目标优化问题（Multi-objective Optimization Problem，MOP）广泛应用于方案规划[1]、数据挖掘[2]、资源分配[3]、车间调度、企业决策等领域。根据不同的选择策略，多数多目标优化算法可以分为基于 Pareto 支配关系、基于指标和基于分解 3 种类型。基于 Pareto 支配关系的方法主要依靠 Pareto 支配关系进行选择操作，代表算法是 NSGAII 算法[4-5]。基于指标的方法使用一个预先定义的量化指标来帮助选择。基于分解的方法[6]将多目标优化问题分解成多个相互关联的简单子问题，并采用协作方式来优化，此类方法的代表是 MOEA/D 算法[7]。近年来，国内外学者相继提出多种改进算法[8]，包括 MOEA/D-DRA[9]、MOEA/D-STM[10]、MOEA/DD[11]、NSGAIII[12]等。

收敛性和多样性是多目标优化算法面临的相互制约难题[13]，多目标优化算法需要在收敛性和多样性之间找到平衡[14]。国内外研究者在平衡收敛性和多样性方面做出了一些研究，并形成相应算法。文献[15]从优化分解方法的角度提出一种新思路，该方法将目标空间以参考向量为基础分解成若干子空间，每个子空间分别得到一个最优解。文献[16]从计算资源动态分配的角度对多目标分解进化算法进行改进，显著提升了多目标分解进化算法的性能。文献[17]从自适应算子选择的角度进

作者简介：李智翔（1989—），男，博士研究生，主研方向为进化算法；李 赟，工程师；褚衍杰，助理研究员。

收稿日期：2018-01-10 **修回日期：**2018-02-26 **E-mail：**lzxest@126.com

行分析,通过实验表明自适应算子选择与 MOEA/D 算法框架的高度兼容性。文献[18]指出原始 MOEA/D 框架中定义的权重向量空间中的邻居关系可能对算法产生误导,给出多目标分解进化算法的改进方向。

本文对基于分解的多目标进化算法 MOEA/D 进行分析,针对其特点设计新的平衡策略,根据需求选用 2 种不同的繁殖算子,利用算法迭代次数对邻居集合的大小进适应性调整,从而提出 MOEA/D-BS 算法。

1 多目标分解进化算法

多目标优化问题表示如式(1)所示。

$$\begin{aligned}&\min \boldsymbol{F}(\boldsymbol{x})=(f_1(\boldsymbol{x}),f_2(\boldsymbol{x}),\cdots,f_m(\boldsymbol{x}))\\&\text{s.t. } x\in\boldsymbol{\Omega}\end{aligned}\tag{1}$$

其中,$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in\mathbb{R}^n$ 表示决策变量,$\boldsymbol{\Omega}$ 为决策空间可行域,$\boldsymbol{F}:\boldsymbol{\Omega}\to\mathbb{R}^m$ 包含 m 个目标函数 $f_1(\boldsymbol{x})\sim f_m(\boldsymbol{x})$。令 $\boldsymbol{x},\boldsymbol{y}\in\Omega$,若对任意 $i\in1,2,\cdots,m$ 有 $f_i(\boldsymbol{x})\leqslant f_i(\boldsymbol{y})$,且存在 $j\in1,2,\cdots,m$ 使得 $f_j(\boldsymbol{x})<f_j(\boldsymbol{y})$,则称 $\boldsymbol{x}$ 支配 $\boldsymbol{y}$。对于解 $\boldsymbol{x}^*\in\boldsymbol{\Omega}$ 而言,若不存在 $\boldsymbol{x}\in\boldsymbol{\Omega}$ 使得 $\boldsymbol{x}$ 支配 $\boldsymbol{x}^*$,则解 $\boldsymbol{x}^*$ 称作 Pareto 最优解[4],所有的 Pareto 最优解组成的集合称作 Pareto 集,简称 PS,PS 中的解对应在目标空间的值组成的集合被称作 Pareto 前沿,简称 PF。

在多目标分解进化算法 MOEA/D 中,一个多目标优化问题被分解成若干个单目标优化问题,这些单目标优化问题的解均为原多目标优化问题的 Pareto 最优解,它们共同构成对其 PF 的近似。适用于 MOEA/D 算法的分解方法有多种,其中切比雪夫法由于其能够得到均匀分布于 PF 的解集,成为常用的分解方法之一[19]。切比雪夫法可以表示为:

$$\min g^{te}(\boldsymbol{x}\mid\boldsymbol{\lambda}_i,\boldsymbol{z}^*)=\max_{1\leqslant j\leqslant m}\{\lambda_i^j\mid f_j(\boldsymbol{x})-z_j^*\mid\}\tag{2}$$

其中,$\boldsymbol{z}^*=(z_1^*,z_2^*,\cdots,z_m^*)^{\mathrm{T}}$ 表示理想参考点,有:

$$z_j^*<\min\{f_j(\boldsymbol{x})\mid\boldsymbol{x}\in\boldsymbol{\Omega}\},j=1,2,\cdots,m\tag{3}$$

通常由于问题的限制或出于方便计算的考虑,将式(3)的小于号取为等号。MOEA/D 是一个迭代算法,在每一代,算法维护大小为 N 的种群 $\boldsymbol{x}^1,\boldsymbol{x}^2,\cdots,\boldsymbol{x}^N$,其中每一个 $\boldsymbol{x}^i$ 表示子问题的当前解。每一个子问题对应一个权重向量,根据权重向量之间的欧式距离,算法定义了邻居,即子问题的邻居集合是那些相对应权重向量距离最小的子问题构成的集合。相应地,子问题对应的解的邻居集合也由此定义。MOEA/D 算法步骤具体描述如下:

步骤 1 确定繁殖邻居集合大小 T_m 及替换邻居集合大小 T_r。

步骤 2 对每一个子问题 i,执行以下 2 步操作,分别为:

1)繁殖操作。对子问题 i 对应的解 $\boldsymbol{x}^i$,在其 T_m 个邻居中选择对象执行繁殖算子,得到下一代解 $\boldsymbol{x}_{\text{new}}^i$。

2)替换操作。对子问题 i 对应的解 $\boldsymbol{x}^i$,观察其 T_r 个邻居中的任意一个 $\boldsymbol{x}^j$,若 $\boldsymbol{x}_{\text{new}}^i$ 优于 $\boldsymbol{x}^j$,则用 $\boldsymbol{x}_{\text{new}}^i$ 替换 $\boldsymbol{x}^j$。

步骤 3 若达到算法停止代数,停止算法;否则执行步骤 2。

2 MOEA/D-BS 算法

2.1 混合繁殖算子

在 MOEA/D 算法的繁殖操作中,交叉算子使用一点交叉法。所谓一点交叉法,即将解表示成序列的形式,随机选取序列中一点,交换该点之后的子序列,从而得到 2 个新解。对任意 2 个解 $\boldsymbol{x}^1$、$\boldsymbol{x}^2$,使用一点交叉法得到新解的可行范围不小于父代解 $\boldsymbol{x}^1$、$\boldsymbol{x}^2$ 围成的空间。采用这样的方法得到的下一代解能够覆盖的搜索范围较大,在迭代过程中能够充分保持解的多样性,防止陷于局部最优。鉴于此,如果已知下一代解比当前解在搜索空间更靠近最优解,那么继续沿着搜索方向前进更有可能快速逼近最优解,但采用一点交叉法进行繁殖操作可能导致搜索范围过大,解的进化方向不能稳定朝向最优解方向,影响算法效率。为此,本文设计一种父代学习法,利用父代解和当前解的优劣关系,方法如下:

$$\boldsymbol{x}_{\text{new}}^i=\boldsymbol{x}^i+(1-2\times rand)\times(\boldsymbol{x}^i-\boldsymbol{x}_{\text{parent}}^i)+\boldsymbol{\varepsilon}\tag{4}$$

其中,$\boldsymbol{x}_{\text{new}}^i$ 表示新解,$\boldsymbol{x}_{\text{parent}}^i$、$\boldsymbol{x}^i$ 分别表示父代解和当前解,$\boldsymbol{\varepsilon}$ 为随机扰动因子。

图 1 为 2 种繁殖算子对下一代解的搜索范围。当前解 $\boldsymbol{x}^i$ 用三角形表示,父代解 $\boldsymbol{x}_{\text{parent}}^i$ 用加号表示,一点交叉法用到的当代解 $\boldsymbol{x}^j$ 同样用三角形表示。在图 1 中,圆形点为采用一点交叉法随机生成的下一代解,可见其搜索范围涵盖了由 $\boldsymbol{x}^i$ 和 $\boldsymbol{x}^j$ 覆盖的区域,正方形为采用父代学习法随机生成的下一代解,这些解沿着父代解 $\boldsymbol{x}_{\text{parent}}^i$ 和当前解 $\boldsymbol{x}^i$ 进化的方向分布,并且其分布范围与采用一点交叉法得到的下一代解相比较小,有利于加速收敛。

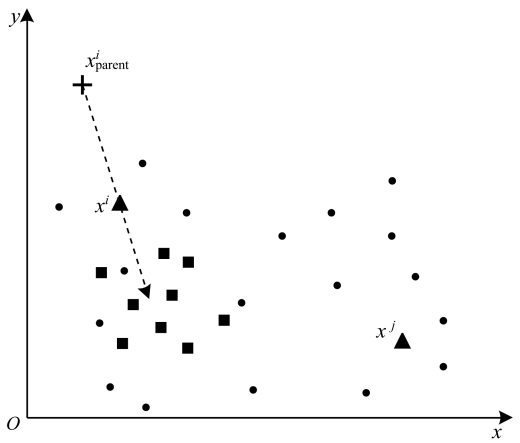


图 1　2 种繁殖算子搜索范围示意图

父代学习法固定父代解到当前解的变化方向,在当前解一定范围内计算新解。当需要进行繁殖操作时,计算当前解与父代解的目标函数取值的比值。假设优化问题为最小化问题,当目标函数取值的比值小于一定数值(小于 1)时,说明当前解较父代解提升较大,沿着父代解到当前解的变化方向继续搜索可能得到更加优秀的新解,此时算法选择使用父代学习法求得新解;否则,当目标函数取值的比值大于一定数值时,说明当前解较父代解并没有明显优越性,当前搜索方向并不一定是最优的搜索方向,应该在较大范围内继续寻找优秀解,此时算法选择使用原来的一点交叉法求得新解。根据目标函数取值的比值对繁殖算子进行选择,相比单纯使用一点交叉法作为繁殖算子,算法能够进一步平衡解的收敛性和多样性,从而提升算法性能。

2.2　动态邻居集合

对一个多目标优化算法,在算法执行的初期应多注重解的多样性,使得搜索的范围应尽可能大,降低陷入局部最优的概率。在算法执行的中后期应多注重解的收敛性,在保证解的分布多样的前提下加速算法收敛[20]。在 MOEA/D 算法中,邻居集合的概念是算法的核心思想之一,在繁殖操作和选择操作时都会使用到邻居集合,算法也分别为其维护了相应的邻居集合 T_m 和 T_r。

繁殖邻居集合大小影响解在繁殖操作时可选择的解数量。繁殖邻居集合的大小越大,可选择的解的数量就越多,从而下一代解的多样性就越强;反之,下一代解的收敛性就越强。因此在算法执行的初期增大繁殖邻居集合的大小,在算法执行过程中逐步减小繁殖邻居集合的大小,使得算法能够在收敛性和多样性之间取得较好的平衡,从而提高算法性能。

选择邻居集合的大小影响算法在选择操作时能够替换的子问题对应的解的数量。选择邻居集合越大,子问题对应的解在选择操作时能够替换的解就越多,生成的下一代解越容易重复,进而算法的收敛性越强,而多样性越弱。因此应在算法执行的初期减小邻居集合的大小,在算法执行过程中逐步增大邻居集合的大小,使得算法能够在收敛性和多样性之间取得较好的平衡,从而提高算法性能。

在本文算法中,2 个邻居集合 T_m 和 T_r 并不像 MOEA/D 算法中那样取固定值,当繁殖邻居集合取一个较大的初始值时,算法的迭代动态减小;当选择邻居集合取一个较小的初始值时,算法的迭代动态增大。具体计算公式如下:

$$T_m = \max\left\{T_{m_\min}, T_{m0} - \left\lfloor \frac{k \times (T_{m0} - T_{m_\min})}{K} \right\rfloor\right\} \quad (5)$$

$$T_r = \min\left\{T_{r_\max}, T_{r0} + \left\lfloor \frac{k \times T_{r_\max}}{K} \right\rfloor\right\} \quad (6)$$

其中,T_{m0}、T_{r0}分别为初始状态时的繁殖邻居集合和选择邻居集合的大小,$T_{m_\min}$、$T_{r_\max}$分别为繁殖邻居集合最小和选择邻居集合最大时的大小,k 和 K 分别表示当前进化代数和总的进化代数。

2.3　算法流程

以式(1)给出的优化问题为例,本文提出的 MOEA/D-BS 算法具体步骤描述如下:

步骤 1　确定初始的繁殖邻居集合大小 T_{m0}、替换邻居集合大小 T_{r0}、繁殖控制因子 δ。

步骤 2　对每一个子问题 i,执行以下 2 步操作,分别为:

1)繁殖操作。对子问题 i 对应的解 $\boldsymbol{x}^i$,若$f(\boldsymbol{x}^i)/f(\boldsymbol{x}^i_{\text{parent}}) < \delta$,则使用式(4)计算下一代解 $\boldsymbol{x}^i_{\text{new}}$;否则在其 T_m 个邻居中选择对象执行一点交叉法进行计算,得到下一代解 $\boldsymbol{x}^i_{\text{new}}$。

2)替换操作。对子问题 i 对应的解 $\boldsymbol{x}^i$,观察其 T_r 个邻居中的任意一个 $\boldsymbol{x}^j$,若 $\boldsymbol{x}^i_{\text{new}}$ 优于 $\boldsymbol{x}^j$,则用 $\boldsymbol{x}^i_{\text{new}}$ 替换 $\boldsymbol{x}^j$。

步骤 3　根据式(5)、式(6)更新邻居集合 T_m 和 T_r。

步骤 4　代数加 1。若算法达到停止条件,则停

止并返回结果;否则执行步骤 2。

3 仿真结果与分析

3.1 参数设置

为说明本文 MOEA/D-BS 算法的有效性,选择 4 种多目标优化算法进行对比测试,分别为 MOEA/D、MOEA/D-DRA、NSGAII 以及 IBEA[21]。其中,MOEA/D、NSGAII、IBEA 分别是基于分解的多目标优化算法、基于 Pareto 支配关系的多目标优化算法以及基于指标的多目标优化算法,而 MOEA/D-DRA 算法是一种改进的基于分解的多目标进化优化算法,该算法在进化中根据反馈对每个子问题的计算资源进行动态分配,从而优化算法性能。

本文选用测试函数集 DTLZ[22] 中的测试函数 DTLZ1 ~ DTLZ7 以及测试函数集 UF[23] 中的测试函数 UF1 ~ UF10 对上述算法进行对比测试。测试函数属性如表 1 所示。与 DTLZ1 ~ DTLZ7 测试函数相比,UF1 ~ UF10 测试函数的决策空间更加复杂。

表 1 测试函数属性

测试函数	目标空间维度	决策空间维度	特征
DTLZ1	3	7	非凸、多峰
DTLZ2	3	12	非凸
DTLZ3	3	12	非凸、多峰
DTLZ4	3	12	非凸
DTLZ5	3	12	非凸、降维
DTLZ6	3	12	非凸、降维
DTLZ7	3	22	混合、不连续
UF1	2	30	非凸、多峰
UF2	2	30	非凸、多峰
UF3	2	30	非凸、多峰
UF4	2	30	非凸、多峰
UF5	2	30	线性、多峰
UF6	2	30	线性、不连续、多峰
UF7	2	30	线性、多峰
UF8	3	30	非凸、多峰
UF9	3	30	线性、不连续、多峰
UF10	3	30	非凸、多峰

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。其中,DTLZ1 ~ DTLZ7 测试函数设置为 100 000,UF1 ~ UF10 测试函数设置为300 000,繁殖邻居集合和选择邻居集合的初始大小分别为 100 和 10,每个算法在每个测试函数上独立运行 20 次。

为了对不同算法的性能进行对比,本文选择常用的性能度量方法 IGD[24] 对实验结果进行分析,令 P^* 表示 PF 上均匀分布的点,S 表示通过算法得到的最优解集,则相应的 IGD 值计算公式为:

$$IGD(S,P^*)=\frac{\sum_{x\in P^*} dist(x,S)}{|P^*|} \tag{7}$$

其中,$dist(x,S)$ 表示点 x 到集合 S 中最近点的欧式距离,$|P^*|$ 指参考最优解集合 P^* 的大小。在实验中,目标空间为二维的测试函数的 $|P^*|$ 设置为 1 000,目标空间为三维的测试函数的 $|P^*|$ 设置为 10 000。IGD 指标能够同时对解集的收敛性和多样性进行度量,IGD 指标取值越小,得到的解集 S 越接近 PF。

3.2 性能分析

为了说明本文算法 MOEA/D-BS 中 2 种平衡策略的有效性,设计了其与 MOEA/D 算法的对比实验。将 MOEA/D 算法中的繁殖算子改成本文提出的混合繁殖算子,称为 MOEA/D-BSa。将 MOEA/D 算法中的邻居集合改成本文提出的动态变化的邻居集合,称为 MOEA/D-BSb。将这 2 种算法分别与 MOEA/D 算法在测试函数 UF1 ~ UF4 上进行对比,结果如表 2 所示。

表 2 不同平衡策略的有效性测试

测试函数	MOEA/D 算法	MOEA/D-BSa 算法	MOEA/D-BSb 算法
UF1	2.67e-04	1.78e-04	1.66e-04
UF2	9.20e-04	8.78e-04	8.22e-04
UF3	2.09e-03	1.76e-03	1.55e-03
UF4	2.23e-03	2.03e-03	1.94e-03

从表 2 可以看出,对于 4 个测试函数 UF1 ~ UF4,分别添加了混合繁殖算子以及添加了动态邻居集合的 2 种平衡策略下的改进算法的结果均优于 MOEA/D 算法,相比较而言添加了动态邻居集合策略的 MOEA/D-BSb 算法的表现最好。结果表明,本文提出的 2 种平衡策略是有效的,且能够有效提升基于分解的多目标进化优化算法的性能。

本文提出算法 MOEA/D-BS 算法与其他 4 种算法在测试函数上的 IGD 指标均值及标准差仿真结果如表 3所示。

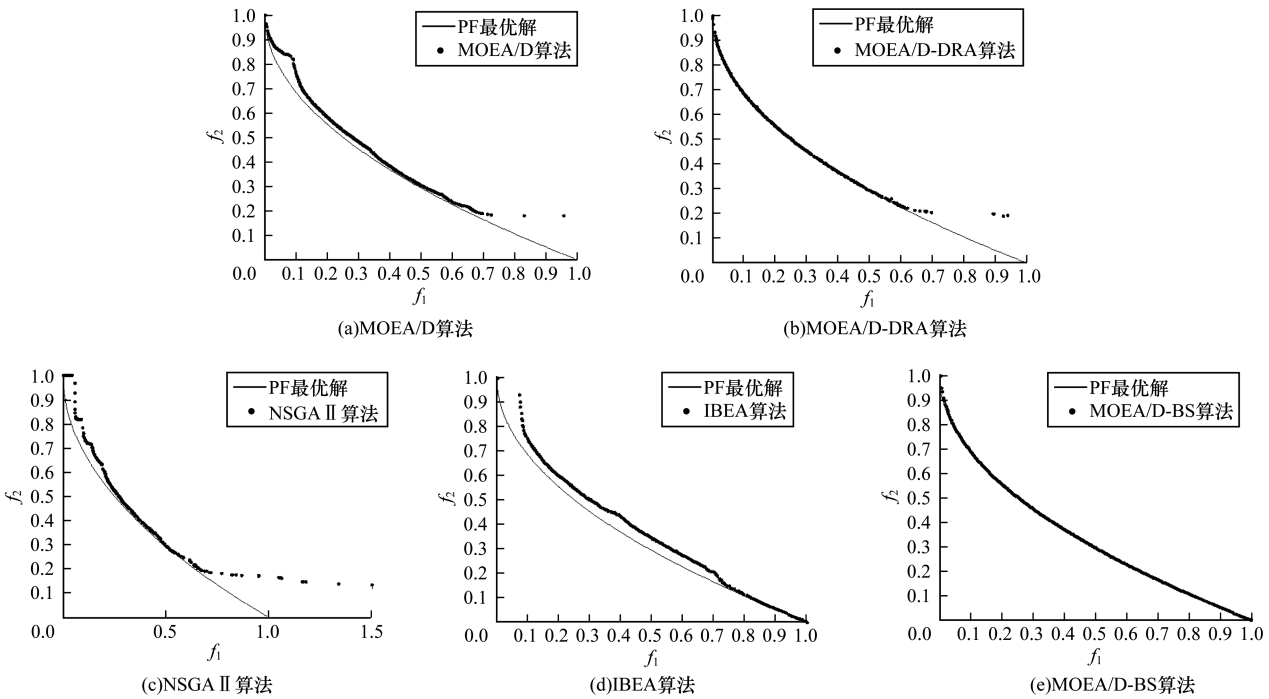
表 3　不同算法在测试函数上的 IGD 均值及标准差对比结果

测试函数	IGD 指标	MOEA/D 算法	MOEA/D-DRA 算法	NSGAII 算法	IBEA 算法	MOEA/D-BS 算法
DTLZ1	均值	3.49e－04	3.49e－04	3.24e－04	4.14e－04	3.11e－04
	标准差	2.30e－06	2.50e－06	8.10e－05	4.10e－05	2.10e－06
DTLZ2	均值	4.33e－04	4.23e－04	4.32e－04	9.85e－04	4.28e－04
	标准差	2.60e－06	2.30e－06	3.30e－05	2.70e－05	2.50e－06
DTLZ3	均值	8.67e－04	7.21e－04	7.29e－04	3.80e－03	6.23e－04
	标准差	6.90e－04	7.50e－04	8.20e－05	1.20e－04	1.30e－05
DTLZ4	均值	7.21e－04	6.90e－04	1.25e－03	2.21e－03	5.76e－04
	标准差	4.70e－05	6.80e－05	2.50e－03	3.10e－03	5.50e－05
DTLZ5	均值	1.56e－05	1.52e－05	2.03e－05	3.59e－05	1.29e－05
	标准差	4.20e－07	5.20e－07	1.20e－06	2.30e－06	2.20e－07
DTLZ6	均值	3.52e－05	3.45e－05	6.65e－05	1.63e－04	3.01e－05
	标准差	3.20e－08	4.40e－08	1.60e－06	1.50e－05	8.20e－09
DTLZ7	均值	4.70e－03	4.10e－03	4.65e－03	1.25e－02	3.68e－03
	标准差	3.00e－03	4.20e－03	6.10e－03	9.40e－03	4.40e－03
UF1	均值	2.67e－04	1.09e－04	4.00e－04	4.86e－04	7.87e－05
	标准差	7.40e－05	3.40e－05	1.00e－03	1.50e－03	5.40e－06
UF2	均值	9.20e－04	9.13e－04	1.90e－03	2.37e－03	7.34e－04
	标准差	1.20e－04	8.20e－04	7.70e－04	1.10e－03	1.30e－04
UF3	均值	2.09e－03	1.93e－03	6.42e－03	8.15e－03	1.04e－03
	标准差	1.20e－04	1.40e－03	1.80e－03	1.30e－03	4.80e－04
UF4	均值	2.23e－03	2.17e－03	3.07e－03	1.67e－03	1.48e－03
	标准差	1.50e－04	1.90e－04	3.30e－05	1.70e－04	1.60e－04
UF5	均值	1.16e－01	1.11e－01	7.93e－02	1.06e－01	7.50e－02
	标准差	2.90e－02	2.40e－02	2.70e－02	3.30e－02	2.70e－02
UF6	均值	8.51e－03	9.81e－03	1.27e－02	1.31e－02	6.70e－03
	标准差	8.20e－03	9.30e－03	6.10e－03	5.60e－03	2.50e－03
UF7	均值	3.36e－04	3.19e－04	7.15e－03	1.95e－02	2.51e－04
	标准差	2.60e－04	3.00e－04	5.10e－03	6.80e－03	5.70e－05
UF8	均值	1.19e－03	1.19e－03	2.78e－03	5.63e－03	1.09e－03
	标准差	2.90e－04	2.70e－04	7.70e－04	1.00e－05	2.50e－04
UF9	均值	1.66e－03	1.21e－03	2.93e－03	3.13e－03	1.48e－03
	标准差	7.30e－04	7.30e－04	8.50e－04	1.10e－03	6.80e－04
UF10	均值	4.98e－03	4.98e－03	5.11e－03	6.51e－03	4.76e－03
	标准差	5.00e－04	7.50e－04	1.10e－03	1.20e－03	6.10e－04

从表 3 可以看出,与 MOEA/D 算法、NSGAII 算法和 IBEA 算法相比,本文 MOEA/D-BS 算法优势较大,在上述 17 个测试函数中的 15 个测试函数上的 IGD 指标均值优于其他 4 种算法,而在 DTLZ2 和 UF9 这 2 个测试函数上 MOEA/D-DRA 算法结果最佳,但本文 MOEA/D-BS 算法与其差距较小。本文 MOEA/D-BS 算法在目标空间为二维或三维的各个测试函数上与其他 4 种算法相比表现更加稳定,尤其在 UF1 ~ UF10 这些决策空间维度高、搜索难度大的测试函数上,本文提出的 MOEA/D-BS 算法优势较大。此外,本文提出的 MOEA/D-BS 算法不仅与 MOEA/D 算法、NSGAII 算法以及 IBEA 算法相比具有较大优势,其综合表现也优于 MOEA/D-DRA 改进算法。对比 4 种算法在不同测试函数上 IGD 指标结果的标准差可以看出,本文提出的 MOEA/D-BS 算法对应的标准差与 MOEA/D 算法和 MOEA/D-DRA 算法相当,在大多数测试函数上优于 NSGAII 算法和 IBEA 算法,表明本文提出的 MOEA/D-BS 算法在多次仿真中最佳结果与最差结果的差距较小,其性能较为稳定。

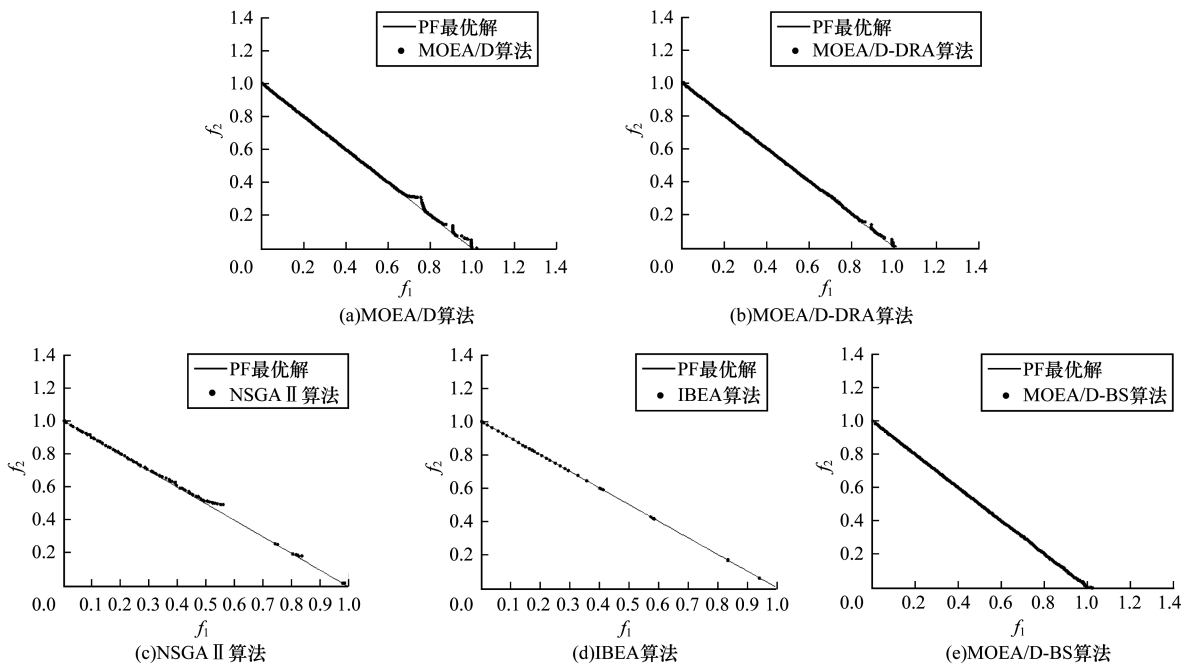
为进一步分析本文算法与其他 4 种算法的性能,以测试函数 UF3 和 UF7 为例,给出5 种算法分别

在这 2 个测试函数上得到的结果,如图 2 和图 3 所示。其中,f_1、f_2 分别表示目标值 1 和目标值 2。从图 2可以看出,IBEA 算法的结果在 PF 的左端和中部收敛性不佳,MOEA/D、MOEA/D-DRA、NSGAII 这 3 种算法的结果在 PF 的左端均与 PF 有一定的差距,在 PF 的右端均有缺失,NSGAII 算法的结果有野点存在,而本文提出的 MOEA/D-BS 算法无野点,解贴近 PF 并且分布均匀。在图 3 中,IBEA 算法在 PF 中部和右端缺失较多,NSGAII 算法在 PF 右端缺失较多且在 PF 中部与 PF 有一定差距,MOEA/D 算法及 MOEA/D-DRA 算法均在 PF 的右端与 PF 存在一定的差距,而本文 MOEA/D-BS 算法能够均匀逼近 PF 且无野点存在。综上,本文提出的 MOEA/D-BS 算法在测试函数 UF3 和 UF7 上能够对整个 PF 进行均匀逼近,且无野点存在,其表现优于其他 4 种算法。



(a)MOEA/D算法 (b)MOEA/D-DRA算法 (c)NSGAⅡ算法 (d)IBEA算法 (e)MOEA/D-BS算法

图 2 5 种算法在测试函数 UF3 上的对比结果



(a)MOEA/D算法 (b)MOEA/D-DRA算法 (c)NSGAⅡ算法 (d)IBEA算法 (e)MOEA/D-BS算法

图 3 5 种算法在测试函数 UF7 上的对比结果

4　结束语

本文在基于分解的多目标进化算法 MOEA/D 基础上,设计 2 种改进的平衡策略,并由此提出 MOEA/D-BS 算法。实验结果验证了 MOEA/D-BS 算法中 2 种平衡策略的有效性,且该算法测试结果优于 MOEA/D、NSGAII 等算法。下一步将对 MOEA/D-BS 算法的权重向量在进化过程中根据收敛情况进行动态调整,以适应不同的优化目标。此外,还将对算法中的资源分配策略进行研究。

参考文献

[1] HANNE T, NICKEL S. A multiobjective evolutionary algorithm for scheduling and inspection planning in software development projects[J]. European Journal of Operational Research,2005,167(3):663-678.

[2] SÁNCHEZ L,OTERO J,COUSO I. Obtaining linguistic fuzzy rule-based regression models from imprecise data with multiobjective genetic algorithms [J]. Soft Computing,2009,13(5):467-479.

[3] BELFARES L, KLIBI W, LO N, et al. Multi-objectives Tabu search based algorithm for progressive resource allocation[J]. European Journal of Operational Research, 2007,177(3):1779-1799.

[4] GIAGKIOZIS I,PURSHOUSE R C,FLEMING P J. An overview of population-based algorithms for multi-objective optimisation [J]. International Journal of Systems Science,2015,46(9):1572-1599.

[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] TRIVEDI A,SRINIVASAN D,SANYAL K,et al. A survey of multiobjective evolutionary algorithms based on decomposition [J]. IEEE Transactions on Evolutionary Computation,2017,21(3):440-462.

[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] 刘超,贺利军,朱光宇. 基于熵和隶属度函数的高维多目标优化问题求解[J]. 计算机工程,2016,42(6): 185-190.

[9] ZHANG Q,LIU W, LI H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances[C]//Proceedings of the 7th Conference on Congress on Evolutionary Computation. Washington D. C., USA:IEEE Press,2009:203-208.

[10] LI K, ZHANG Q, KWONG S, et al. Stable matching-based selection in evolutionary multiobjective optimization [J]. IEEE Transactions on Evolutionary Computation,2014,18(6):909-923.

[11] 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.

[12] 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.

[13] GIAGKIOZIS I,PURSHOUSE R C,FLEMING P J. An overview of population-based algorithms for multi-objective optimisation [J]. International Journal of Systems Science,2015,46(9):1572-1599.

[14] WU M, LI K, KWONG S, et al. Matching-based selection with incomplete lists for decomposition multiobjective optimization [J]. IEEE Transactions on Evolutionary Computation,2017,21(4):554-568.

[15] LIU H L, GU F, ZHANG Q. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3):450-455.

[16] ZHOU A,ZHANG Q. Are all the subproblems equally important? resource allocation in decomposition-based multiobjective evolutionary algorithms [J]. IEEE Transactions on Evolutionary Computation, 2016, 20(1):52-64.

[17] GONCALVES R A, ALMEIDA C P, POZO A. Upper confidence bound algorithms for adaptive operator selection in MOEA/D [C]//Proceedings of the 8th International Conference on Evolutionary Multi-Criterion Optimization. Berlin, Germany: Springer,2015:411-425.

[18] GIAGKIOZIS I, PURSHOUSE R C, FLEMING P J. Generalized decomposition and cross entropy methods for many-objective optimization [J]. Information Sciences,2014,282:363-387.

[19] MA X, ZHANG Q, YANG J, et al. On tchebycheff decomposition approaches for multi-objective evolutionary optimization [J]. IEEE Transactions on Evolutionary Computation,2017,22(2):226-244.

[20] LI Y L, ZHOU Y R, ZHAN Z H, et al. A primary theoretical study on decomposition-based multiobjective evolutionary algorithms [J]. IEEE Transactions on Evolutionary Computation,2016,20(4):563-576.

[21] ZITZLER E, KÜNZLI S. Indicator-based selection in multiobjective search [C]//Proceedings of the 8th International Conference on Parallel Problem Solving from Nature. Berlin, Germany: Springer,2004:832-842.

[22] DEB K, THIELE L, LAUMANNS M, et al. Scalable multi-objective optimization test problems [C]// Proceedings of Congress on Evolutionary Computation. Washington D. C., USA:IEEE Press,2002:825-830.

[23] ZHANG Q, ZHOU A, ZHAO S, et al. Multiobjective optimization test instances for the CEC 2009 special session and competition [EB/OL]. [2017-12-20]. https://dces.essex.ac.uk/staff/zhang/MOEAcompetition/cec09testproblem0904.pdf .

[24] BOSMAN P A N, THIERENS D. The balance between proximity and diversity in multiobjective evolutionary algorithms [J]. IEEE Transactions on Evolutionary Computation,2003,7(2):174-188.

编辑　赵　辉