

基于目标空间分解的自适应多目标进化算法^①郑金华^② 张作峰^③ 邹娟

(湘潭大学信息工程学院“智能计算与信息处理”教育部重点实验室 湘潭 411105)

摘 要 针对基于分解的多目标进化算法(MOEA/D)个体与子问题的匹配问题,在分析 MOEA/D 的进化规律的基础上,提出了一种基于目标空间分解的自适应多目标进化算法(MOEA/OSD)。该算法采用以测试问题的参考点为起点的均匀权重向量分解目标空间,根据个体信息动态选择适合的子问题,并使用辅助向量的方法弥补分解方法的不足。对比实验结果表明,MOEA/OSD 拥有较好的收敛性和分布性,采用不同的分解方法均能搜索到最优解,且具有较好的收敛速度。

关键词 多目标优化,目标空间分解,子问题,自适应,适合的子问题

0 引 言

现实世界中的许多实际问题是多属性的,往往需要同时优化多个目标。多目标进化算法(multi-objective evolutionary algorithms, MOEAs)由于通用性强且不依赖于函数模型,适用于处理复杂的多目标优化问题,因而在过去 10 年中得到了很大发展。根据选择机制的不同,可将 MOEAs 分成三类:基于 Pareto 支配的算法,基于群体的算法以及基于聚集函数的算法^[1]。目前,尽管基于 Pareto 的算法已成为 MOEAs 的主流^[2],同时也涌现出了许多经典的算法,如 NSGA-II^[3]、SPEA2^[4]、PAES^[5]、PESA-II^[6]等,此类算法适用于处理低维优化问题,但随着目标维数的增加,算法的搜索能力显著下降^[7],无法找到优化问题的近似 Pareto 最优面。基于群体方法的 MOEAs 往往考虑一个目标而忽视其他目标,不能对优化问题的所有目标同时优化,进而致使算法难以搜索到完整的最优 Pareto 面^[8]。基于聚合函数的 MOEAs 则采用加权法将多目标优化问题分解成单目标优化问题,并结合归档集保优策略,不断调整权重向量,进而搜索到优化问题的 Pareto 最优解,但解集的分布性无法保持^[9]。

针对现有 MOEAs 的不足,Zhang 和 Li^[10]通过

预先产生均匀分布的权值向量将多目标优化问题转换为单目标子问题,并为每个子问题分配一个个体,从而提出基于分解的多目标进化算法(MOEA based on decomposition, MOEA/D)。该算法能有效地处理高维问题及决策空间不连续问题^[11,12]。之后,涌现出多种不同版本的 MOEA/D,性能也得到显著提高,如 MOEA/D-DE^[13]引入差分交叉算子,提高了处理复杂 PS 的能力;MOEA/D-DRA^[14]根据子问题进化程度动态调整计算资源,增强了统筹计算资源的能力;MOEA/D-EGO^[15]引进高斯模型,能有效地解决昂贵问题;ENS-MOEA/D^[16]根据子问题生成优秀解的概率动态调整邻域,使收敛性能得到较大改进。尽管不同版本的 MOEA/D 有较理想的性能,但个体与子问题间的匹配问题仍旧是个难点,严重制约了其性能的进一步提高。由于 MOEA/D 随机地为子问题分配个体,无法为子问题分配合适的个体,因而难以满足邻近子问题解的相似性,也就无法通过邻域内交叉变异产生有利于特定子问题的个体。加之各子问题同时进化,容易出现重复搜索,进而直接影响了算法的运行效率。本文从不同于以往的多目标优化研究重点(进化算法的性能和搜索能力)的角度出发,分析了原有 MOEA/D 的进化规律,提出了个体与子问题的匹配方法,并在此基础上构造了一种新的算法:基于目标空间分解的自适应

① 国家自然科学基金(61070088),湖南省教育厅项目(12C0378, 11C1224)和湖南省科技厅项目(2011GK3063)资助。

② 男,1963 年生,博士,教授;研究方向:进化计算;E-mail: jhzheng@xtu.edu.cn

③ 通讯作者,E-mail: zhangzuofeng@foxmail.com

(收稿日期:2012-06-20)

多目标进化算法(MOEA based on objective space decomposition, MOEA/OSD)。该算法避免了随机指定子问题带来的各种问题。

1 基本的 MOEA/D 算法

MOEA/D 将多目标优化问题分解为一组单目标子问题并为每个子问题分配一个个体后,由各子问题上的个体组成初始种群,通过并行进化种群来优化所有子问题。由于相邻子问题之间存在对彼此进化有利的信息,MOEA/D 基于权重向量间的欧式距离建立子问题的邻域,MOEA/D 通过领域内交叉变异产生新的个体。基于邻域的进化是 MOEA/D 的有效搜索机制之一^[16]。在进化过程中,优势基因一旦被某个子问题搜索到,就会在邻域内扩散,因此通过相邻子问题的信息搜索到优秀解的可能性更高。此外,MOEA/D 成功地将数学规划中常用的分解方法引入到 MOEAs^[17]。目前常用的分解方法有: Weighted Sum、Tchebycheff 和 Penalty-based Boundary Intersection(PBI)^[10]。

Weighted Sum 分解方法可用式

$$\min g^{\text{WS}}(\tilde{x} | \tilde{\lambda}) = \sum_{i=1}^m \lambda_i f_i(\tilde{x}) \quad (1)$$

表示,其中, $\tilde{x} \in \Omega$, m 为目标个数, $\tilde{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)^T$ 是一组权重向量。对于 $i \in (1, 2, \dots, m)$, $\sum_{i=1}^m \lambda_i = 1$ 。

Tchebycheff 分解方法可用式

$$\min g^{\text{Tche}}(\tilde{x} | \tilde{\lambda}, z^*) = \max_{1 \leq i \leq m} \{ \lambda_i | f_i(\tilde{x}) - z_i^* | \} \quad (2)$$

表示,其中, $z^* = \min \{ f_i(\tilde{x}) | \tilde{x} \in \Omega \}, i \in (1, 2, \dots, m)$ 。

PBI 分解方法可用式

$$\begin{aligned} \min g^{\text{PBI}}(\tilde{x} | \tilde{\lambda}, z^*) &= d_1 + \theta d_2 \\ d_1 &= \frac{\| (z^* - F(\tilde{x}))^T \tilde{\lambda} \|}{\| \tilde{\lambda} \|} \\ d_2 &= \| F(\tilde{x}) - (z^* - d_1 \tilde{\lambda}) \| \end{aligned} \quad (3)$$

表示,其中, $\theta > 0$ 为预设参数。

2 MOEA/D 算法进化规律分析

MOEA/D 的性能在很大程度上是由分解方法决定的^[18],个体的进化方向也取决于分解方法。假设采用权重向量 $\tilde{\lambda}_i = (0.3, 0.7)$ 加权得到一个子问题,三种分解方法的适应度等高线如图 1 所示。每条等高线将目标空间划分为两个区域,只有当个体落在等高线下方时才能取代等高线上的个体,子问题将沿适应度升高的方向进化。当个体落到最优 Pareto 面附近时,子问题收敛。分解方法将整个种群向近似 Pareto 最优面(PF)的逼近转化为各子问题沿一定方向向 PF 的逼近。子问题的适应度等高线是关于 V 对称的,随着适应度上升,个体逐步向 V

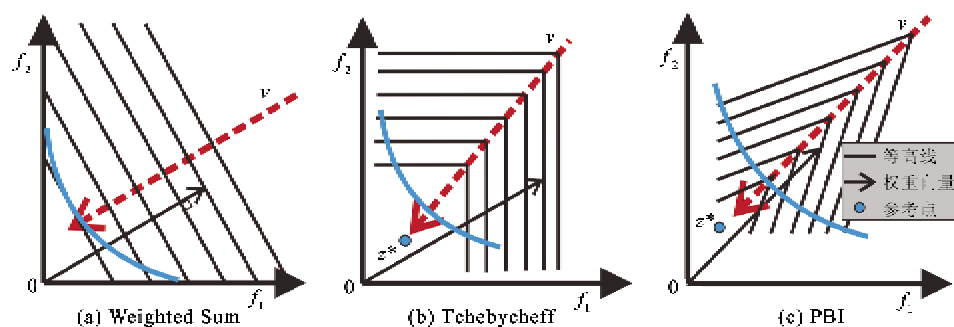


图 1 三种分解方法的等高线

靠近,同时沿 V 向参考点靠近,直至与 PF 相交。 V 即是通过分解方法给予子问题施加的选择压力方向。又由于 Tchebycheff 和 PBI 分解方法同时含有收敛性信息和分布性信息,个体的适应度越高,个体对子问题进化的贡献越大,这个子问题也就越适合这个个体。如图 2(b) 和图 2(c),任意个体 i 和两个不同的子问题 p_1 和 p_2 , V_1 和 V_2 分别是子问题 p_1 和 p_2 所对

应的 V 。个体 i 在子问题 p_1 和 p_2 上的适应度不同,当个体 i 离某一个子问题的 $V(p_2)$ 对应的 V_2 越近时,个体 i 的适应度越高 ($fit_2 > fit_1$)。因此,个体在最靠近的 V 附近时适应度最高,这个 V 所对应的子问题(图 2 中 p_2) 也就是个体 i 最适合的子问题。而 Weighted Sum 分解方法的聚合函数是线性的,不含有分布性信息。无论个体与 V 的距离是近是远,总

是在坐标轴附近的 V 对应的子问题上适应度最高 (图(2(a)))。如果选择适应度最高的子问题将造成算法只能搜索到坐标轴附近的极端解。而选择离

V 最近的子问题有助于弥补上述不足,提高分布性。基于以上分析可知:

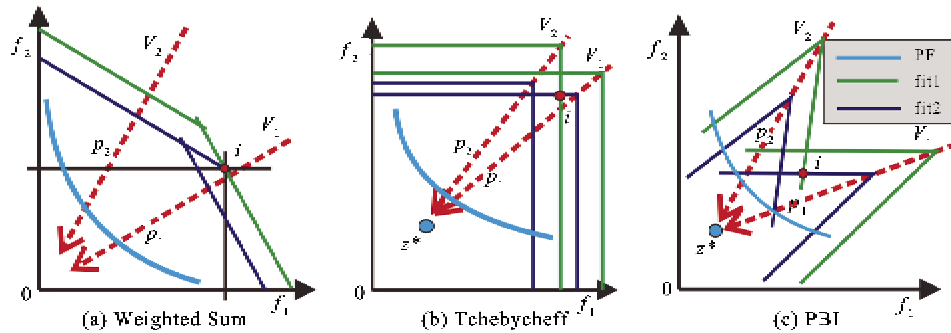


图2 个体适合子问题

性质1:距离个体 i 最近的 V 所对应的子问题即是个体 i 最适合的子问题。

为任意子问题 p 分配的权重向量是固定不变的,经过分解方法的转换,形成了沿方向 V 的选择压力,那么权重向量与 V 之间存在必然的联系。由于 V 以参考点为起点且穿过 PF,其中 V 与 PF 的交点,即是子问题 p 上个体的收敛位置。因此根据个体的收敛位置和参考点即可确定方向 V 。实验观察发现,采用 Weighted Sum 和 Tchebycheff 分解方法, $\bar{\omega}_i$ 与权重向量方向关于 $f_2 = f_1$ 对称;采用 PBI 分解方法, $\bar{\omega}_i$ 与权重向量方向一致。其中 $\bar{\omega}_i$ 是 V 的单位向量。对于三维问题,权重向量 $\bar{\omega}_i = (\omega_i^1, \omega_i^2, \omega_i^3)$ 关于 $f_1 = f_2 = f_3$ 的对称向量为 $\bar{\omega}'_i = (\lambda_i^1, \lambda_i^2, \lambda_i^3)$, 而

$$\lambda_i^j = \frac{1/(\omega_i^j - z^{*j})}{\sum_{k=1}^m \frac{1}{(\omega_i^k - z^{*k})}} \quad (4)$$

为了便于统计,对数据进行如下处理:取某个体 i 的权重向量与 $\bar{\omega}_i$ 的偏差 $\Delta\omega_i = (\sum_{j=1}^m |\omega_i^j - \bar{\omega}_i^j|)/m$, 其中 $\bar{\omega}_i^j$ 是实验假定的与 ω_i^j 对应的权重向量。 $\Delta\omega = (\sum_{i=1}^N \Delta\omega_i)/N$ 为整个种群权重向量与 $\bar{\omega}_i$ 的偏差。独立运行 MOEA/D 算法 30 次,取 $\Delta\omega$ 的算术平均值作为计算结果。结果见表 1。

由于 Weighted Sum 分解方法是线性的,无论如何调整权重系数,都难以搜索到非凸解^[19],所以在使用 Weighted Sum 分解方法处理 ZDT2、ZDT3、ZDT6 和 DTLZ 系列的非凸问题时 $\Delta\omega$ 数值较大。由表 1 可知;在一定的误差范围内假设是正确的。这也证明了 MOEA/D 算法的另一个性质:

表1 不同分解方法的 $\Delta\omega$ 值

	Weighted sum	Tchebycheff	PBI
ZDT1	0.052241	1.63E-06	2.17E-05
ZDT2	0.560958	1.36E-06	6.51E-06
ZDT3	0.573045	0.001220	6.51E-06
ZDT4	0.059845	3.73E-04	8.37E-04
ZDT6	0.657271	7.37E-05	7.65E-05
DTLZ1	0.48026	0.045989636	3.13E-06
DTLZ2	0.45673	0.045998992	8.52E-07
DTLZ3	0.45956	0.046017196	2.18E-06
DTLZ4	0.71934	0.046013184	8.47E-07
DTLZ5	0.46211	0.050687641	0.009251

性质2:采用 Weighted Sum 或 Tchebycheff 分解方法,子问题的 V 与权重向量关于 $f_2 = f_1$ 对称;采用 PBI 分解方法, V 与权重向量方向一致。

结合性质 1 和性质 2 可得:

结论1:采用 Weighted Sum 和 Tchebycheff 分解方法,适合子问题的个体相对于参考点的方向与权重向量关于 $f_2 = f_1$ 对称;采用 PBI 分解方法,适合子问题的个体相对于参考点的方向与权重向量方向一致。

通过分析 MOEA/D 的进化规律,发现了个体与适合子问题之间存在的联系。利用这种关系,使自适应地选择适合的子问题成为可能。在此基础上,提出一种新的算法——基于目标空间分解的自适应多目标进化算法(MOEA/OSD)。

3 基于目标空间分解的自适应多目标进化算法

本文构造了一组以参考点为起点的均匀权重向

量,使用这组权重向量将目标空间分解为若干均匀的区域。并为每个子问题分配一个区域,每个子问题都在各自的区域中进化,可以有效地减少子问题的重复搜索。为了给予子问题分配适合的个体,每当产生新个体时,根据结论 1 计算出适合的子问题,并将个体分配给适合的子问题。同一个区域内的个体较为接近,利用分解方法计算适应度无法准确地筛选出更好的个体,本文采用一种辅助向量方法进行弥补,以达到从在某个子问题上同等优秀的个体中选择更有利于其他子问题的个体。

3.1 目标空间的分解

预先产生的权重向量是以坐标原点为起点,终点在曲面 $f_1 + f_2 + \dots + f_m = 1$ 或 $f_1^2 + f_2^2 + \dots + f_m^2 = 1$ 上的一组均匀的权重向量。均匀的权重向量所对应的 V 也是均匀的。这些权重向量均匀地划分目标空间,其所对应的 V 是以参考点为起始点的均匀向量。当个体落在 V 与最优 Pareto 面的交点或附近(图 3 中点)时,个体能保持很好的收敛性和分布性,这正是理想的结果。又由于个体的生成是随机的,不可能为每个子问题都分配一个适合的个体。某些子问题可能有多个个体,也可能不存在个体,直到出现适合的个体。在此过程中,种群中每增加一个个体,就从个体最多的子问题中删除一个最差的个体,这样既可以维持种群大小不变,也可以保持种群分布的广泛性。以参考点作为坐标原点可以解决测试问题范围不确定的问题,提高了划分精度。

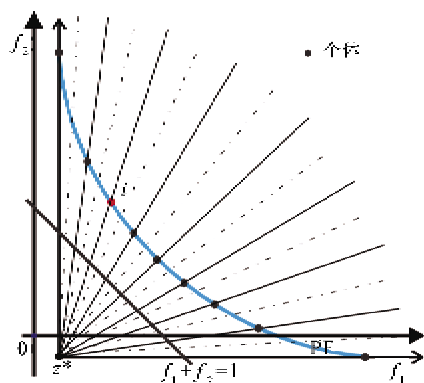


图 3 目标空间的分解

3.2 选择适合子问题的方法

根据性质 1,当产生新个体后首先根据公式(4)计算出 $\bar{\omega}$ 。按照性质 2 确定个体适合的权重向量,距离最近的子问题即是个体适合的子问题,算法伪代码见算法 1。寻找适合的子问题的时间复杂度为 $O(N + m)$,其中 N 为种群大小, m 为目标个数。由

于通常 N 远大于 m , 因此时间复杂度为 $O(N)$ 。自适应地选择适合的子问题避免了适合某一子问题的优势个体在另一个子问题中被错误淘汰,然后在子问题上使用分解方法计算适应度。根据适应度选择个体,能提供足够的选择压力。虽然在此过程中某些解会被重复搜索到,但由于在其适合的子问题中处于劣势,因而当其再次出现时也不会被保留,这样也避免了重复搜索。

算法 1: 选择适合子问题的方法

输入: x : 个体

输出: $suitableIndex$: 适合的子问题

```

1:  $sum \leftarrow 0$  {函数值相对于参考点的距离之和}
2: for  $i = 1$  to  $m$  do
3:  $sum \leftarrow sum + 1/(fun[i] - z[i])$  { $m$  为目标个数,  $fun$  为个体  $x$  的函数值,  $z$  为参考点}
4: end for
5: for  $i = 1$  to  $m$  do
6:  $\omega[i] \leftarrow (1/(fun[i] - z[i]))/sum$  { $\omega$  为个体  $x$  相对于参考点的位置}
7: end for
8:  $suitableIndex \leftarrow 1$ 
9:  $minDistance \leftarrow distance(\omega, \lambda[1])$  {计算权重向量间的距离}
10: for  $i = 2$  to  $N$  do
11:  $di \leftarrow distance(\omega, \lambda[i])$ 
12: if  $di < minDistance$  then
13:  $suitableIndex \leftarrow i$ 
14:  $minDistance \leftarrow di$ 
15: end if
16: end for
    
```

3.3 辅助向量排序

对于任意权重向量 $p(\lambda_1, \lambda_2)$ 上拥有不同适应度值的两个 n 维个体 X_1 和 X_2 很容易判断其优劣,这种情况实际上是在沿其等高线下降方向判断 $p(\lambda_1, \lambda_2)$ 条件下个体的优劣。然而,为某个子问题分配的区域个体较为接近时,容易出现权重向量相同,适应度值也相同的两个 n 维个体,此时,仅仅根据分解方法就很难判断个体的好坏。本文由此构造了一个辅助向量,将具有相同适应度值的个体分别放到 $p(\lambda_1, \lambda_2)$ 关于 $f_2 = f_1$ 对称的权重向量 $p'(\lambda'_1, \lambda'_2)$ 上去计算。这样,最终进化结果如果成功逼近 PF 的话,不仅可以在 $p(\lambda_1, \lambda_2)$ 是最优的,同时在 $p'(\lambda'_1, \lambda'_2)$ 也最优。个体 X_1 和 X_2 在同一个

子问题 p 的适应度等高线上,无法从中选择一个更好的个体,此时将两个个体同时放到子问题 p 关于 $f_2 = f_1$ 对称的子问题 p' 中,可以判断出两个个体

的好坏(图4)。通过这种方法可以从同一条等高线上的个体中选出更有利于其他子问题进化的个体,弥补了解方法的不足。

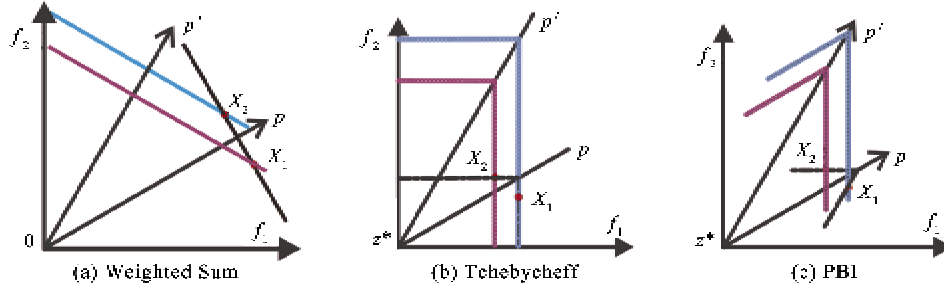


图4 辅助向量排序

3.4 算法框架

为了增强算法的搜索能力和提高对临近子问题优势基因的利用能力,在进化过程中不断调整邻域的大小,邻域初始大小为种群大小 N ,随着进化逐渐缩小,进化结束时缩小到 $N/10$,如算法2所示。

算法2: 基于目标空间分解的自适应多目标进化算法

输入: MOP :多目标优化问题; $\tilde{\lambda}$:均匀的权重向量; $\tilde{z} = (z_1, \dots, z_m)$:参考点; N :种群大小

输出: EP

- 1: 步骤1 初始化
- 2: 步骤1.1 $EP = \emptyset$
- 3: 步骤1.2 初始化种群 $\tilde{x} = (\tilde{x}^1, \dots, \tilde{x}^N)$,并将每个个体放入适合的子问题
- 4: 步骤2 更新
- 5: 调整 NS | NS 为邻域大小,随着进化逐渐缩小直至 $N/10$
- 6: **for** $i = 1$ **to** N **do**
- 7: 步骤2.1 基因重组:从邻域中随机选取两个个体 $\tilde{x}^k$ 和 $\tilde{x}^l$,交叉变异后产生 y
- 8: 步骤2.2 修正解 y 产生 y'
- 9: 步骤2.3 更新解
- 10: 寻找 y' 适合的子问题 p
- 11: **if** p 中已经存在个体 **then**
- 12: **if** y' 比 p 中最差的个体 y^p 优秀 **then**
- 13: y' 取代 y^p
- 14: **end if**
- 15: **else** 将 y' 放入 p ,并删除个体最多的子问题上的最差个体
- 16: **end if**
- 17: **end for**
- 18: 步骤3 终止条件
- 19: **if** 终止条件满足 **then**
- 20: 停止并输出 EP
- 21: **else** 转步骤2
- 22: **end if**

步骤1.2 每一个个体都要寻找适合的子问题,因此时间复杂度为: $O(N^2)$,算法2的时间复杂度主要集中在步骤2,基因重组的时间复杂度为 $O(p)$,其中 p 为决策空间维数。修正解的复杂度为 $O(1)$ 。更新解的复杂度主要集中在寻找适合子问题,复杂度为 $O(N)$ 。步骤2的时间复杂度为 $O(N \cdot (p + 1 + N))$ 。

4 算法测试及结果分析

实验从采用 Weighted Sum 分解方法处理非凸问题的能力 and 综合性能两个方面比较分析。

4.1 实验参数设置

为了说明算法的有效性并与其他算法比较,实验采用综合性能指标 IGD (inverted generational distance) 来评价算法的性能,计算如下:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (5)$$

P^* 是目标空间中均匀分布在真实 PF 上的点的集合, P 为待评价点的集合, $d(v, P)$ 是 v 到 P 的最小欧几里德距离。

二维测试函数权重向量和个体的数目设置为100,三维测试函数为105。MOEA/D 邻域大小为10。MOEA/OSD 和 MOEA/D 采用模拟二进制交叉和多项式变异。按照文献[16]的实验,模拟二进制交叉和多项式变异的分布系数为20,交叉概率为1.00,变异概率为 $1/N$, N 为决策变量个数。PBI 分解方法的 θ 取5。

4.2 实验分析

4.2.1 综合性能

实验选取 MOEA/D、经典的进化算法 NSGA-II

与 MOEA/OSD 比较,分别独立运行各算法 30 次,取算数平均值,如表 2 所示。结果表明,MOEA/OSD 具有较好的综合性能,采用 Weighted Sum 分解方法可以很好地处理非凸问题(DTLZ1、DTLZ2、DTLZ3、DTLZ4、DTLZ5、ZDT2、ZDT3 和 ZDT6),但在处理 ZDT4 这样的欺骗问题时陷入了局部最优,MOEA/D 处理非凸问题性能较差,NSGA-II 算法处理非凸问

题性能较好;采用 Techebycheff 分解方法,MOEA/OSD 综合性能好于 MOEA/D,但处理非连续问题(ZDT3)的能力较差;采用 PBI 分解方法,MOEA/OSD 性能好于 MOEA/D,部分问题(DTLZ2、DTLZ4 和 ZDT2)与 MOEA/D 接近,但处理非连续问题(ZDT3、DTLZ5)效果较差。

表 2 两种算法采用三种不同分解方法的 IGD 指标

	Weighted Sum		Techebycheff		PBI		NSGAII
	MOEA/D	MOEA/OSD	MOEA/D	MOEA/OSD	MOEA/D	MOEA/OSD	
DTLZ1	0.0063902	0.007148	0.0074593	0.0160774	0.0074785	0.0074356	0.064830493
DTLZ2	0.3960392	0.0306898	0.0979207	0.0458794	0.0324862	0.0324858	0.070861805
DTLZ3	0.6147164	0.0644202	0.0627548	0.0533993	0.0605999	0.0605658	0.071221194
DTLZ4	0.6201084	0.0642731	0.06974537	0.0679957	0.0605501	0.0605497	0.0701904
DTLZ5	0.3563042	0.2750161	0.004889	0.0048889	0.2932661	0.2932762	0.296699341
ZDT1	0.0039586	0.0044897	0.0030001	0.0029999	0.005878	0.0049933	0.005886358
ZDT2	0.100178	0.0029862	0.1682173	0.1683681	0.0030002	0.0030002	0.499414751
ZDT3	0.4774273	0.1683398	0.0054772	0.0044622	0.1690158	0.1694483	0.946239847
ZDT4	0.01384	0.10784057	0.0029343	0.0031823	0.0089345	0.0069893	0.064706967
ZDT6	0.0797432	0.0029589	0.0074593	0.0160774	0.0029204	0.0029154	0.085145964

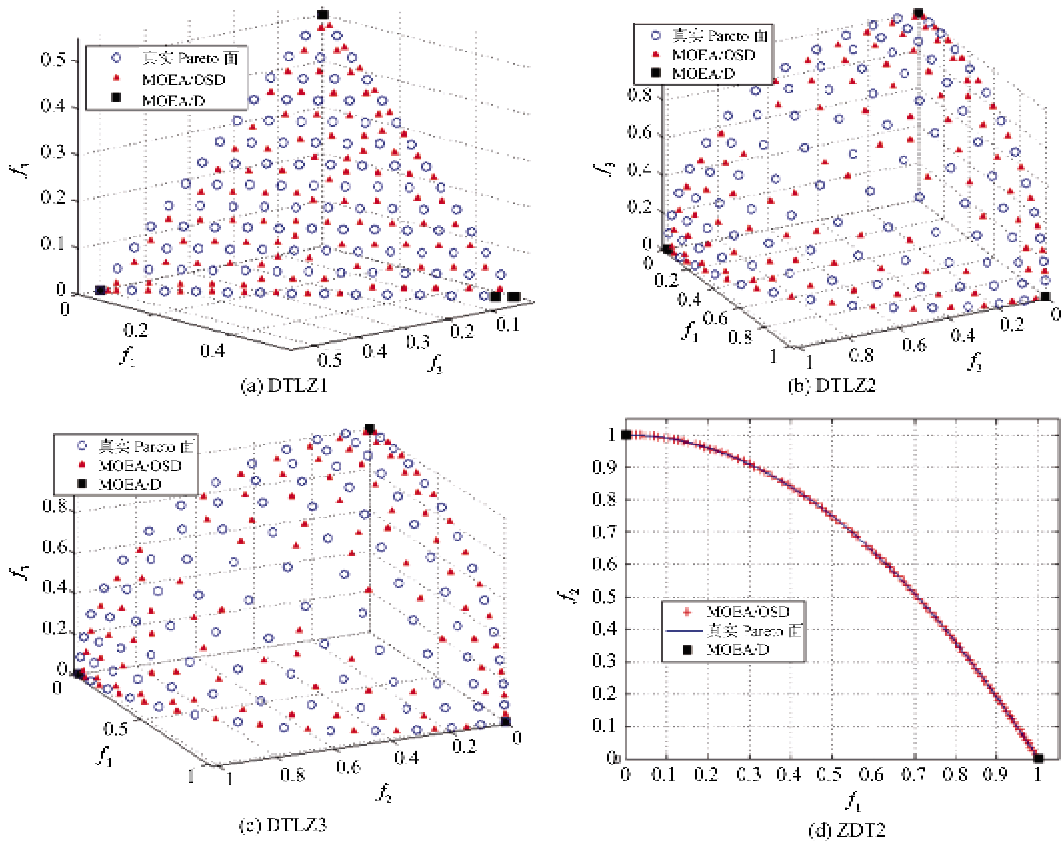


图 5 采用 Weighted Sum 处理非凸问题

4.2.2 处理非凸问题的能力

实验选取 DTLZ1、DTLZ2、DTLZ3 和 ZDT2 四种

非凸问题来测试算法采用 Weighted Sum 分解方法处理非凸问题的能力(如图 5),DTLZ1 的最优 Pare-

to 面为平面, DTLZ2、DTLZ3 和 ZDT2 问题的最优 Pareto 面为凹面, MOEA/D 算法几乎所有的个体都收敛到坐标轴附近, 分布性较差。而 MOEA/OSD 算法已收敛, 且分布性较好。由于 Weighted Sum 分解方法等高线是一组与权重向量垂直的平行线(图 1)。无论如何调整权重向量, MOEA/D 也只能搜索到等高线与 PF 的切点处的解, 因此 MOEA/D 搜索到的解集中在几个切点上。而 MOEA/OSD 以参考点为起始点均匀的分解目标空间和 PF, 为每块区域保留一个非支配的最优解, 当个体收敛到 PF 时, 最优个体能均匀地分布在 PF 上, 因此在处理非凸问题时 MOEA/OSD 能够保持较好的分布。

两组实验表明, MOEA/OSD 在处理非连续性问题和欺骗问题时性能不好, 但是 MOEA/OSD 拥有较好的综合性能。收敛速度较快且采用 Weighted Sum 分解方法也可以处理非凸问题。MOEA/OSD 能在收敛性和分布性方面有较好的改进效果, 也表明了本文提出的以 MOEA/D 的进化规律为依据, 根据个体的信息选择适合的子问题的假设是正确的。

5 结 论

本文通过对 MOEA/D 进化规律的分析, 论证了一种根据个体信息寻找适合的子问题的方法, 并以此为基础提出了基于目标空间分解的自适应多目标进化算法(MOEA/OSD)。MOEA/OSD 通过以参考点为起始点的均匀权重向量分解目标空间, 根据个体所在的位置自适应地选择适合的子问题, 并使用辅助向量弥补分解方法的不足和保持边界。MOEA/OSD 较全面地考虑了 MOEA/D 框架的特征, 探索了一种子问题与个体间匹配的思想, 通过与经典 MOEA/D 和 NSGAII 的对比分析, 可以得到如下结论: (1) 无论采用任何一种分解方法, MOEA/OSD 均能得到 PF 上分布均匀且有代表性的 Pareto 最优解。(2) 对于大部分的测试问题, MOEA/OSD 均能取得良好的收敛效果。

个体与子问题间的匹配是长期困扰 MOEA/D 的难题之一, 根据进化规律和个体信息选择适合的子问题使此难题得到很好的解决, 而且效果较好, 但对非连续问题和欺骗问题的处理效果欠佳。在接下来的研究中, 将进一步完善 MOEA/OSD 解决非连续问题和欺骗问题的能力, 并将本文的思想引入到性能评价。

参考文献

- [1] 郑金华. 多目标进化算法及其应用. 科学出版社, 2007
- [2] Ishibuchi H, Sakane Y, Tsukamoto N, et al. Simultaneous use of different scalarizing functions in MOEA/D. In: Proceedings of the Genetic and Evolutionary Computation Conference, 2010. 519-526
- [3] Deb K, Pratap A, Agarwal S. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 2: 182-197
- [4] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. In: Proceedings of the Evolutionary Methods for Design, Optimization and Control with Application to Industrial Problems, 2001. 95-100
- [5] Knowles J, Corne D. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. In: Proceedings of the Congress of Evolutionary Computation, 1999. 98-105
- [6] Corne D, Jerram N R, Knowles J, et al. PESA-II: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference, 2001. 283-290
- [7] Ishibuchi H, Tsukamoto N, Hitotsuyanagi Y. Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems. In: Proceedings of the Genetic and Evolutionary Computation Conference, 2008. 649-656
- [8] Schaffer J D. Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition): [PhD thesis], Nashville, TN, USA, 1984. AAI8522492
- [9] Jin Y, Okabe T, Sendho B. Adapting weighted aggregation for multiobjective evolution strategies. In: Proceedings of the Evolutionary Multi-Criterion Optimization, 2001. 96-110
- [10] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE transactions on evolutionary computation*, 2007. 712-731
- [11] Chang P C, Chen S H, Zhang Q, et al. MOEA/D for flowshop scheduling problems. In: Proceedings of the 2008 IEEE Congress on Evolutionary Computation. 2008. 1433-1438
- [12] Ishibuchi H, Sakane Y, Tsukamoto N, et al. Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2009
- [13] Li H, Zhang Q. Multiobjective optimization problems with

- complicated Pareto sets, MOEA/D and NSGA-II. *IEEE transactions on evolutionary computation*, 2009, 13(2)
- [14] Zhang Q, Liu W, Li H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: Proceedings of the Eleventh Conference on Congress on Evolutionary Computation, Piscataway, USA, 2009. 203-208
- [15] Zhang Q, Liu W, Tsang E. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 2010, 14(3): 456-474
- [16] Zhao S, Suganthan P N, Zhang Q. Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes. *IEEE Transactions on Evolutionary Computation*, 2011. 16(3):442-446
- [17] 公茂果, 焦李成, 杨咚咚等. 进化多目标优化算法研究. 软件学报, 2009, 20(2):271-289
- [18] Ishibuchi H, Sakane Y, Tsukamoto N, et al. Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm. In: Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization, Nantes, France, 2009. 438-452
- [19] Das I, Dennis J. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization*, 1997, 14(1): 63-69

An adaptive multi-objective evolutionary algorithm directed by objective space decomposition

Zheng Jinhua, Zhang Zuofeng, Zou Juan

(Institute of Information Engineering, Xiangtan University, Key Laboratory of Intelligent Computing & Information Processing of Ministry of Education, Xiangtan 411105)

Abstract

Aiming at solving the matching problem of individual and sub-problem of the multi-objective evolutionary algorithm based on decomposition (MOEA/D), the paper proposes an adaptive multi-objective evolutionary algorithm directed by objective space decomposition (MOEA/OSD) based on the evolution analysis of the MOEA/D. The MOEA/OSD decomposes an objective space by even spread weight vectors whose start points are the reference points, chooses a suitable sub-problem by using the information of individuals, and uses auxiliary weight vectors to compensate for the limitations of the decomposition approaches. The experimental results demonstrates that the MOEA/OSD could not only balance the convergence and diversity effectively but also approach the optimal solution by applying different decomposition approaches, and has a better convergence speed.

Key words: multi-objective optimization, objective space decomposition, sub-problem, self-adapting, suitable sub-problem