

doi: 10.11918/j.issn.0367-6234.2015.11.012

# 结合分解技术的多目标引力搜索算法

毕晓君<sup>1</sup>, 刁鹏飞<sup>1</sup>, 王艳娇<sup>2</sup>, 肖 婧<sup>3</sup>

(1.哈尔滨工程大学 信息与通信工程学院, 150001 哈尔滨; 2.东北电力大学 信息工程学院, 132012 吉林 吉林;

3.辽宁省交通高等专科学校 信息工程系, 110122 沈阳)

**摘要:** 针对基于分解的多目标遗传算法在解决多目标问题时无法有效解决前沿面非均匀、不连续的问题, 提出一种基于分解技术的多子群串行搜索的多目标引力搜索算法(MOGSA/D). 为充分利用算法优化分解出的目标函数所得到的进化信息、提高收敛速度, 采取多种群串行的搜索方式; 针对理想前沿面为非超平面的情况, 提出一种预测理想前沿面形状的方法, 并针对预测结果选择适合的权重系数生成方式; 为提高解集的整体质量, 提出一种基于目标权值的策略删减种群. 通过标准测试函数的实验验证, 所提算法与其他多目标进化算法相比在解集的收敛性以及分布性上均有较大提高, 验证了算法的有效性.

**关键词:** 引力搜索算法; 多目标优化; 分解; 多种群策略

中图分类号: TP18

文献标志码: A

文章编号: 0367-6234(2015)11-0069-07

## Multi-objective gravitational search algorithm based on decomposition

BI Xiaojun<sup>1</sup>, DIAO Pengfei<sup>1</sup>, WANG Yanjiao<sup>2</sup>, XIAO Jing<sup>3</sup>

(1.College of Information and Communication Engineering, Harbin Engineering University, 150001 Harbin, China;

2. College of Information Engineering, Northeast Dianli University, 132001 Jilin, Jilin, China;

3. Dept. of Information Engineering, Liaoning Provincial College of Communications, 110122 Shenyang, China)

**Abstract:** When the ideal frontier is discontinuous or inhomogeneous, the multi-objective evolutionary algorithm can't solve multi-objective problems effectively by decomposition. In order to improve this situation, a novel multi-objective gravitational search algorithm based on decomposition (MOGSA/D) is proposed. In MOGSA/D, the multi-population serial strategy is good for the population study evolutionary information. According to shape prediction of ideal frontier, a suitable generation method of weight coefficient is selected. A pruning strategy is adopted to prune the solution set. Experimental results show that MOGSA has a good performance to solve multi-objective problems in comparison with other multi-objective optimization algorithms.

**Keywords:** gravitational search algorithm (GSA); multi-objective optimization; decomposition; multi-population strategy

多目标优化问题 MOP (multi-objective optimization problem) 需要同时优化 2 个或 2 个以上的目标函数, 且各个优化目标之间通常是相互冲突的, 因此不存在能使所有目标都达到最优的解.

多目标进化算法 MOEA (multi-objective evolutionary algorithm) 是目前广泛采用的一种解决 MOP 的方法, 主要可以分为 3 类: 1) 基于 Pareto 占优关系的 MOEA 算法, 如  $\epsilon$ -占优机制<sup>[1]</sup>、E-占优机制<sup>[2]</sup>; 2) 基于评估指标的 MOEA 算法, 如

HypE<sup>[3]</sup>等算法; 3) 基于分解技术的 MOEA 算法, 如 MOEA/D 算法<sup>[4]</sup>. 其中基于分解的 MOEA 算法因其计算简单、复杂度小和收敛性好受到广泛关注, 但当理想前沿面为不连续、非均匀等形状时, 所得到的优化结果的收敛精度及稳定性还有待进一步提高.

引力搜索算法 GSA (gravitational search algorithm) 是 2010 年提出的群智能优化算法, 具有设置参数少、全局搜索能力强、计算简单、收敛速度快等优点<sup>[5]</sup>. 本文将 GSA 与分解技术相结合, 提出一种基于分解技术的多目标引力优化算法. 主要创新点: 1) 采用分解技术, 将多目标问题分解成单目标问题, 然后采用多子群串行搜索的策略对各子问题展开优化, 提高进化搜索的效率; 2) 提出一种理想前沿面的预测策略, 并根据预测结果选取适合的权重系数计算方式, 一定程度上保证了解集分布的

收稿日期: 2014-11-30.

基金项目: 国家自然科学基金(61175126); 中央高校基本科研业务费专项资金(HEUCFZ1209); 高等学校博士学科点专项科研基金(20112304110009).

作者简介: 毕晓君(1964—), 女, 教授, 博士生导师.

通信作者: 毕晓君, 398317196@qq.com.

均匀性;3) 针对解在各目标所得结果的线性加权和, 提出一种种群删减策略, 提高了解集相对于理想前沿面的逼近性及分布的均匀性。

## 1 引力搜索算法及基于分解的多目标算法

### 1.1 引力搜索算法

可以把种群中的个体想象成是一组在空间中运动的粒子, 它们在万有引力的作用下, 相互运动, 其质量的大小是评价解优劣的标准, 质量较大粒子的位置对应较优解。粒子通过相互作用、相互运动实现进化信息的共享, 从而向最优区域展开搜索。

设空间中含有  $N$  个粒子, 则第  $i$  个粒子的位置表示为

$$X_i = (x_i^1, \dots, x_i^k, \dots, x_i^n); i = 1, 2, \dots, N.$$

式中:  $n$  表示搜索空间的维数;  $x_i^k$  为第  $i$  个粒子在第  $k$  维上的位置信息。在  $t$  时刻, 两个粒子间的作用力表示为

$$f_{ij}^k(t) = G(t) \frac{M_{pi}(t) \times M_{qj}(t)}{R_{ij}(t) + \varepsilon} (x_j^k(t) - x_i^k(t)).$$

式中:  $M_{qj}(t)$  和  $M_{pi}(t)$  分别为粒子  $j$  和粒子  $i$  的质量;  $\varepsilon$  是一个极小的常量;  $G(t)$  表示在  $t$  时刻的万有引力常数, 具体定义为

$$G(t) = G_0 \times e^{-at/T}.$$

式中:  $T$  为设置的最大迭代次数,  $G_0$  为初始时刻的引力常数,  $R_{ij}(t)$  为粒子  $i$  与粒子  $j$  之间的欧氏距离。则在  $t$  时刻, 粒子  $i$  在  $k$  维上受到的其他粒子的合力为

$$f_i^k(t) = \sum_{j=1, j \neq i}^N \text{rank}_j f_{ij}^k(t).$$

式中:  $\text{rank}_j$  为变化区间在  $[0, 1]$  之间的随机数;  $f_{ij}^k(t)$  为粒子  $j$  对粒子  $i$  在第  $k$  维空间上的作用力。依据牛顿第二定律, 定义  $t$  时刻粒子  $i$  在  $k$  维上的加速度为

$$a_i^k(t) = f_i^k(t) / M_{ii}(t).$$

在进化过程中, 粒子的速度和位置的更新方式为

$$v_i^k(t+1) = \text{rank}_i \times v_i^k(t) + a_i^k(t),$$

$$x_i^k(t+1) = x_i^k(t) + v_i^k(t+1).$$

粒子的质量与适应度值有关, 质量越大的粒子越接近最优, 并且它对其他粒子的作用力相应会更大, 但移动速度较慢。粒子质量的计算方式为

$$m_i(t) = \frac{f_i(t) - w(t)}{b(t) - w(t)},$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)}.$$

式中:  $f_i(t)$  为粒子  $i$  的适应度值;  $w(t)$  为质量最小粒子的适应度值;  $b(t)$  为质量最大粒子的适应度值。

图 1 给出了 GSA 算法的操作流程图。

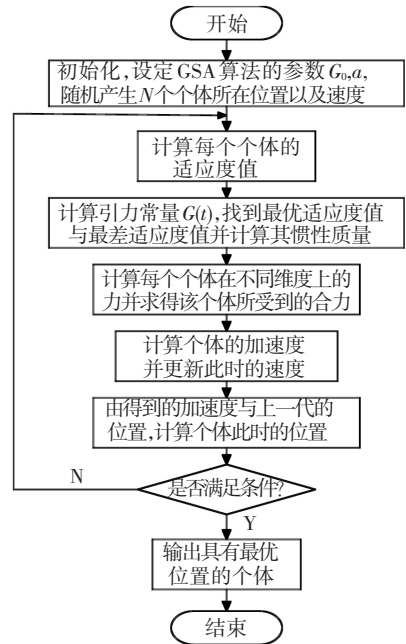


图 1 GSA 算法流程图

### 1.2 基于分解的多目标算法

基于分解的多目标优化算法结合了多目标局部搜索和多个单目标 Pareto 抽样的优点, 而切比雪夫法又是其中解决效果最好的方法。在切比雪夫分解模型下, 对于一个给定的权重向量, 对应的单目标子问题可描述为

$$g^{te}(x | \lambda, z^*) = \max_{1 \leq i \leq m} \{ |\lambda_i (f_i(x) - z_i^*)| \}. \quad (1)$$

式中: 权重向量  $\lambda = (\lambda_1, \dots, \lambda_m)$  满足  $\lambda_k \geq 0$  且  $\sum_{i=1}^m \lambda_i = 1$ ;  $z^* = (z_1^*, z_2^*, \dots, z_m^*)$  为参考点, 是由解集在各个目标函数上取得的最优值构成的虚拟点, 对于最小化问题有  $z_i^* = \min \{ f_i(x) | x \in \Omega \}$ ,  $1, 2, \dots, m$ 。在满足条件的情况下, 对于任意一个给定的权重向量, 存在一个使式(1)达到最小的最优解, 该最优解就是原多目标优化问题的一个 Pareto 最优解。所以, 可以通过设定  $N$  个不同的权重向量获得  $N$  个最优解来构成 Pareto 最优解集。

## 2 基于分解的多目标引力搜索算法

提出一种基于分解技术的多目标引力搜索算法 (MOGSA/D)。首先选取几组具有偏好性的权重系数展开优化搜索, 并以此判别理想前沿面的形状。然后基于预测结果选取适合的权重系数计算方式。种群在对分解后的优化问题展开搜索时, 相邻子问题的解具有相似性, 因此为充分利用这种近似性, 提高求解能力尤其是非均匀不连续问题的优化效率, 采用多子群串行搜索的策略。最后为提高解集的收敛精

度并兼顾粒子分布的均匀性, 提出一种基于目标权重的策略对种群进行删减。

### 2.1 权重的自适应选取

MOEA/D 通过构造多组权重系数将一个 MOP 分解为一组单目标优化问题, 并通过子问题之间的合作来同时优化这组子问题, 从而获得对理想最优解集的一个逼近。但该方法存在 2 个问题:

1) 传统 MOEA/D 算法采用的权重系数生成方法是基于理想面是超平面的情况。针对理想面非均匀的情况, 单一的权重系数生成方式会影响解集分布的均匀性, 如图 2 所示。

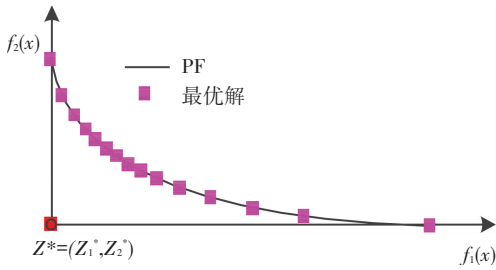


图 2 分布非均匀前沿面

2) 当前前沿面不连续时, 若算法对处在断开区域的子目标问题进行优化求解, 则由于进化压力容易得到相同的解, 势必会浪费计算资源, 降低算法的整体优化性能, 如图 3 所示。

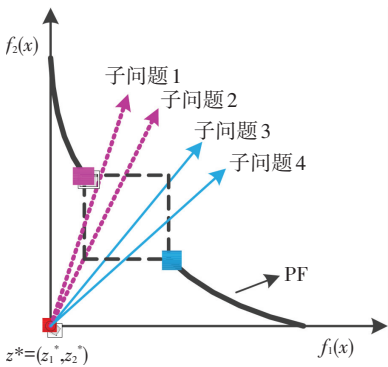


图 3 分布不连续前沿面

为解决上述问题, 文献 [6] 提出权重向量根据其自身的最优解进行自适应调整, 文献 [7] 提出一种加强的权重向量调整方法。但这些方法所适用的前提条件是理想前沿面是超平面的情况, 当前前沿面为非超平面或不连续时, 上述方法均不能保证所得解集分布的均匀性, 且会浪费计算资源。因此, 权重系数的生成必须针对不同的前沿面形状自适应产生。

基于以上分析, 首先对理想前沿面的形状进行预测。通过选择具有明显偏好性的权值系数可以得到理想前沿面的边界点, 若理想前沿面为超平面, 则对 2 个目标问题有判别方式为

$$\begin{cases} \frac{f_1^1 + f_2^1}{2} - f_{\text{mean}}^1 = \kappa, \\ \frac{f_1^2 + f_2^2}{2} - f_{\text{mean}}^2 = \kappa. \end{cases}$$

式中:  $f_1^1, f_2^1$  分别为对偏重于目标函数 1 的权重系数构建出的子问题解分别在 2 个目标上结果的归一化后取值;  $f_1^2, f_2^2$  分别为优化偏重于目标函数 2 的权重系数构建出的子问题解分别在 2 个目标上结果的归一化后取值;  $f_{\text{mean}}^1, f_{\text{mean}}^2$  分别为优化不偏重于任何目标函数的权重系数构建出的单子问题解分别在 2 个目标上的结果归一化后取值;  $\kappa$  理论上应该为零, 但考虑到算法搜索得到的结果与实际相比会有一些的误差, 因此这里表示一个范围。

理论证明: 当理想前沿面是超平面时, 则对 2 个目标问题是一条直线, 在极端情况下, 当权重向量的取值为 (1, 0) 时, 所得的最优解在 2 个目标函数上的解  $f_1^1, f_2^1$  是这条前沿面上的一个端点; 同理, 当权重向量取值为 (0, 1) 时,  $f_1^2, f_2^2$  为前沿面的另一个端点; 对一个等腰直角三角形, 角分线也是中位线, 即取权重向量为 (0.5, 0.5) 时, 所得的解  $f_{\text{mean}}^1, f_{\text{mean}}^2$  应位于前沿面的中点位置上。

对 3 个目标问题, 可判断点和面的位置关系, 即通过求解 4 组权重系数构成的子问题的最优值, 判断理想前沿面的形状。设 4 组偏好权重系数构成的子问题的最终优化结果在 3 个目标上的解为  $(f_1^1, f_2^1, f_3^1)$ 、 $(f_2^2, f_2^2, f_2^2)$ 、 $(f_3^3, f_3^3, f_3^3)$  和  $(f_4^4, f_4^4, f_4^4)$ , 其中  $f_1^1, f_2^1, f_3^1$  表示偏好目标函数 1 的权重系数构成的子问题在 3 个目标上得到的优化结果, 且  $f_1^1$  为式 (1) 中的  $z_1^*$ ;  $f_2^2, f_2^2, f_2^2$  表示偏好目标函数 2 的子问题在 3 个目标上得到的优化结果, 且  $f_2^2$  为式 (1) 中的  $z_2^*$ ;  $f_3^3, f_3^3, f_3^3$  表示偏好目标函数 3 的子问题在 3 个目标上得到的优化结果, 且  $f_3^3$  为式 (1) 中的  $z_3^*$ ;  $f_4^4, f_4^4, f_4^4$  表示随机选取权重系数构成的子问题在 3 个目标上得到的优化结果。若实际前沿面是超平面, 则 4 个点应在一个平面上, 若实际前沿面是凹平面, 则  $(f_4^4, f_4^4, f_4^4)$  应在前 3 个点构成的平面下方; 若实际前沿面为凸平面, 则  $(f_4^4, f_4^4, f_4^4)$  应在前 3 个点构成的平面上方。3 组有极端偏好权重系数构成的子问题的解, 所组成的平面方程可表示为

$$\begin{aligned} a &= (f_2^2 - f_1^1) * (f_3^3 - f_1^1) - (f_3^3 - f_1^1) * (f_2^2 - f_1^1), \\ b &= (f_2^2 - f_3^3) * (f_3^3 - f_1^1) - (f_3^3 - f_3^3) * (f_1^1 - f_1^1), \\ c &= (f_2^2 - f_1^1) * (f_3^3 - f_2^2) - (f_3^3 - f_1^1) * (f_2^2 - f_2^2), \\ a(x - f_1^1) + b(y - f_1^1) + c(z - f_1^1) &= 0. \end{aligned}$$

根据得到的平面方程, 通过比较点和面的位置关系得到前沿面的预测结果, 具体描述如下:

若

$$a(f_4^1 - f_1^1) + b(f_4^2 - f_1^2) + c(f_4^3 - f_1^3) = \kappa,$$

则点 $(f_4^1, f_4^2, f_4^3)$ 在平面里,实际前沿面为超平面。

若

$$a(f_4^1 - f_1^1) + b(f_4^2 - f_1^2) + c(f_4^3 - f_1^3) > \kappa,$$

则点 $(f_4^1, f_4^2, f_4^3)$ 在平面上,实际前沿面为凸平面。

若

$$a(f_4^1 - f_1^1) + b(f_4^2 - f_1^2) + c(f_4^3 - f_1^3) < \kappa,$$

则点 $(f_4^1, f_4^2, f_4^3)$ 在平面下,实际前沿面为凹平面。

当预测到前沿面形状后,针对不同的预测结果,为有效解决非均匀不连续前沿面问题,必须保证权重系数构成的向量与前沿面的焦点均匀分布在前沿面上。提出一种权重系数的生成方式为

$$\sum_{k=1}^m \lambda_k^p + \dots + \lambda_k^p + \dots + \lambda_m^p = 1,$$

$$\lambda_i \in \left\{ \frac{1}{H}, \frac{2}{H}, \dots, \frac{H-1}{H} \right\}, i = 1, 2, \dots, m.$$

式中: $H$ 为人为设定的常数,且每一组权重系数都是从 $\left\{ \frac{1}{H}, \frac{2}{H}, \dots, \frac{H-1}{H} \right\}$ 中不同的取值组合; $m$ 表示多目标优化问题目标函数的个数; $p$ 表示根据不同预测结果选取的参数; $\lambda_k^p$ 为第 $k$ 个目标的权重系数, $p$ 的取值原则参见文献[8]。

1) 当前沿面为超平面时,则 $p$ 取值为1,即 $\lambda$ 构成的面也为超平面;

2) 当前沿面为凸平面时,则 $p$ 取值为2,即 $\lambda$ 构成的面也为凸平面;

3) 当前沿面为凹平面时,则 $p$ 取值为0.5,即 $\lambda$ 构成的面也为凹平面。

## 2.2 多子群串行搜索

对于非连续理想前沿面,由于进化压力的驱使,必然会导致几组相似权重系数所构建出的子问题具有相同的解。为解决前沿面不连续所带来的资源浪费问题,结合GSA的收敛速度快、求解精度高、进化步长可控等特性,采取多子群串行搜索的策略对构造出的子问题进行优化。相邻权重系数所对应的解也具有相似性,根据这种相似性,可以为下一个种群粒子的初始化设定一个较小的范围,相比于在整个空间随机初始化种群缩小了搜索区域,提高了种群搜索最优解的速度。因此,当一个子种群完成对其优化问题的求解时,参考此前最优解初始化种群,粒子生成方式为

$$X_{ij}^{\text{new}}(t) = X_{\text{best},j}(t-1) \times (1+r_3) + (X_{r_{1j}}(t-1)). \quad (2)$$

式中: $X_{ij}^{\text{new}}(t)$ 为第 $t$ 个子种群第 $i$ 个个体在第 $j$ 维上的位置信息; $X_{\text{best},j}(t-1)$ 为第 $t-1$ 个子种群中最好个体在第 $j$ 维上的位置信息; $X_{r_{1j}}(t-1)$ 和

$X_{r_{2j}}(t-1)$ 为第 $t-1$ 个子种群中随机选取的2个个体在第 $j$ 维上的位置信息; $r_3$ 为最优粒子的随机扰动因子,通过实验仿真,当 $r_3$ 随机取值在区间 $[-0.1, 0.1]$ 范围内时,所得结果较好。

## 2.3 删减策略

为了利于决策者的选择,提高解集中解的代表性,需要对优化得到的最优解集进行筛选,以提高其分布的均匀性及收敛程度。以最小化问题为例,能够使各目标值都取较小的点,将更容易支配其他的解也即更靠近理想前沿面,但对互不支配的2个解则较难判断其优劣。对此,通过比较2个相似非支配解在各目标上整体的逼近程度加以删减。对非支配解尤其是较为相似的2个解,可通过计算其归一化后的各目标值的加权和进行筛选,具体描述如下:

将解集中各粒子按照欧式距离进行排序,对于相对距离小于阈值 $u$ 的2个粒子,先分别将各目标值归一化,然后计算其各目标上的加权和,即

$$f_i^l + \dots + f_i^l + \dots + f_i^m > f_j^l + f_j^l + \dots + f_j^m.$$

式中: $f_i^l$ 表示第 $i$ 个粒子在第 $l$ 个目标上的值; $f_j^l$ 表示第 $j$ 个粒子在第 $l$ 个目标上的值。则第 $i$ 个解优于第 $j$ 个解,删除第 $j$ 个解。若

$f_i^l + \dots + f_i^l + \dots + f_i^m = f_j^l + \dots + f_j^l + \dots + f_j^m$ ,  
则说明此时2个解相同,任意删除一个。若

$f_i^l + \dots + f_i^l + \dots + f_i^m < f_j^l + \dots + f_j^l + \dots + f_j^m$ ,  
则第 $j$ 个解优于第 $i$ 个解,删除第 $i$ 个解。

基于目标加权和的判定方式,可以较好地对比相似非支配解进行删减,使最终得到的解集更佳逼近前沿面。对于拥挤区域的删减,在一定程度上保证了分布的均匀性,同时,降低了决策者的选择压力。

## 2.4 算法实现流程

**步骤1** 算法初始化,设定问题维数 $D$ ,种群规模 $N_1$ 和 $N_2$ 以及2个种群各自的最大迭代次数,问题的函数评价次数,以及 $\kappa, u, G_0, a, H$ 。

**步骤2** 对多目标问题的个数,选择相应的几组偏好权值系数(若目标个数为2,则选择3组带有偏好性的权值系数,若目标个数为3,则选择4组带有偏好性的权值系数)构成目标函数。

**步骤3** 计算每个个体的适应度值,并求出其相应的惯性质量。

**步骤4** 根据个体的惯性质量,求得每个个体所受的合力,进而求得其加速度。

**步骤5** 判断是否满足终止条件,若不满足转到步骤3。

**步骤6** 当完成对各组偏好权重系数构成的单目标函数的求解后,采用式2.1节描述的方法判断前沿面的形状。若前沿面形状是凹的,则 $p = 0.5$ ;若

前沿面形状为凸的, 则  $p = 2$ ; 否则  $p = 1$ , 具体的权重系数生成方式如 2.1 节所示。

**步骤 7** 根据得到的一系列权重系数, 按步骤 3、步骤 4 进行优化搜索。

**步骤 8** 当满足终止条件时, 重新初始化新的子种群, 初始化方式采用式(2)所示方式进行。

**步骤 9** 当达到最终的函数评价次数后, 采用删减策略对解集进行删减, 即将解集中各粒子按照欧式距离进行排序, 对于相对距离小于阈值  $u$  的 2 个粒子, 先分别将各目标值归一化, 然后计算并比较粒子各目标上的加权和, 保留较优粒子。

**步骤 10** 输出最终的解集。

### 3 仿真实验与结果分析

所有实验在硬件配置为 Intel® Core™ 2 Duo CPU P7570 2.26 GHz、2 G 内存、2.27 GHz 主频的计算机上进行, 开发环境为 Matlab2 010.b。

#### 3.1 测试问题

实验选用二目标 ZDT 和三目标 DTLZ 两个系列的多目标测试函数<sup>[8]</sup>, 具体包括 ZDT1、ZDT2、ZDT3、ZDT4、ZDT6、DTLZ1 和 DTLZ2 等 7 个函数。这些测试函数具有理想前沿面形状为凹形、凸形、不连续以及多模等特点, 可验证所提算法的有效性。

采用世代距离指标 (generational distance metric, GD) 和空间度量指标 (spacing metric, SP)<sup>[9-10]</sup> 两种评价标准测试所提算法的性能。GD 和 SP 的定义为

$$GD = \frac{1}{N} \sqrt{\sum_{i=1}^N d_i^2},$$

$$SP = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\bar{d} - d_i)^2}.$$

式中:  $N$  表示通过算法求得的解集中解的个数;  $d_i$  为求得的解集中第  $i$  个解到理想最优解集的最小欧几里得距离;  $\bar{d}$  为算法求得所有解  $d_i$  的平均值。

GD 的值越小, 表明算法求得的解集与问题的理想最优解集越接近, 算法收敛性越好。SP 的值越小, 表明算法求得的解集中的解分布的越均匀, 算法的均匀性越好。

#### 3.2 参数设置

实验结果与目前解决多目标问题效果较好的 NSGAI<sup>[11]</sup>, MOEA/D-AWA<sup>[7]</sup>, Adaptive-MOEA/D<sup>[12]</sup> 和 HMOEA<sup>[9]</sup> 进行了比较, 相关参数如表 1 所示。

选取的二目标偏好权重向量个数为 3, 即分别偏好 2 个目标的向量, 以及任意选取的向量分别为

(0.99, 0.01)、(0.01, 0.99)、(0.5, 0.5); 选取的三目标偏好权重系数个数为 4, 即分别偏好 3 个目标的权重向量以及任意选取的一组向量, 分别为 (0.90, 0.05, 0.05)、(0.05, 0.90, 0.05)、(0.05, 0.05, 0.90)、(0.33, 0.33, 0.34)。

表 1 参数设置

参数	数据
ZDT 的函数评价次数	30 000
DTLZ 的函数评价次数	50 000
$N_1$	30
$N_2$	5
$u$	$1.0e - 04$
$\kappa$	$(-1.0e - 3, 1.0e - 3)$
$G_0$	10
$a$	20
子种群 $N_1$ 迭代次数	200
子种群 $N_2$ 迭代次数	50
$H$	100

#### 3.3 多子群串行搜索策略验证

为验证多子群串行搜索策略在解决多目标问题时的有效性, 通过观察算法对测试问题搜索的渐变过程进行实验, 验证多子群策略的有效性。函数 ZDT3 具有前沿面不连续的特点, 因此选取该函数作为实验函数。实验结果如图 4 所示。

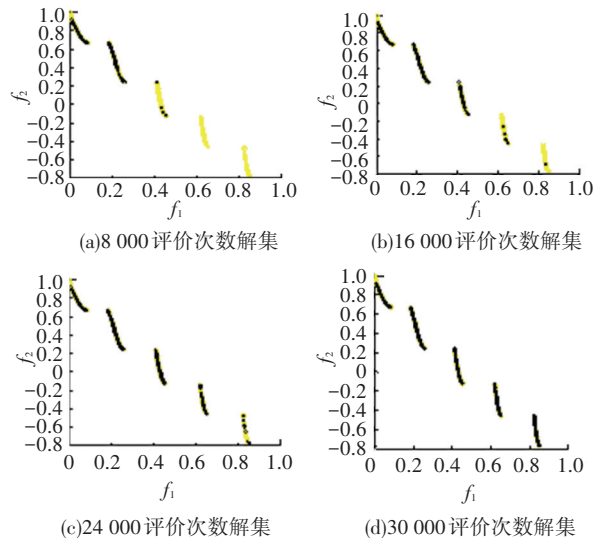


图 4 随迭代得到解集的渐进变化趋势

从图 4 可以看出, 随着函数评价次数的不断增加, 所得到的解集逐渐遍布了整个理想前沿面, 取得了较好的逼近效果, 说明多子群串行搜索的有效性。图中出现离散点, 这是因为对子问题的搜索会产生一些非支配解, 这些解对于当前子问题不是最优解但对于其他子问题来说则是最优解, 体现了多种群串行搜索策略在优化多目标问题时, 不会漏掉得到

的较优解,具有较好的全局搜索性能.

### 3.4 删减策略有效性验证

为验证删减策略在解决多目标问题中的有效性,选取测试函数 ZDT1 进行验证,实验结果如图 5 所示,指标 SP 和 GD 的对比如表 2 所示.

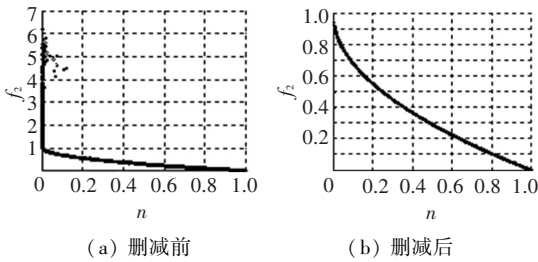


图 5 删减前后算法所得前沿面对比

表 2 解集在删减前后的 GD 值和 SP 值

策略	GD	SP
删减前	0.0040 (9.1346e-4)	7.6528e-2 (3.2408e-3)
删减后	7.2254e-5 (5.306e-6)	6.4357e-4 (2.154e-5)

从图 5 可以看出,在算法未对解集进行删减时,所得到的前沿面存在较多被支配解,当对其进行删减后,得到的解集则是由一群非支配解构成.由表 2

表 3 各算法的 GD 值(第一行)及其标准差(第二行)

算法	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ1	DTLZ2
MOGSA/D	8.196e-5 (9.742e-6)	4.091e-5 (3.241e-6)	9.4128e-6 (5.383e-5)	4.100e-6 (4.332e-6)	7.562e-5 (2.398e-5)	4.120e-2 (1.174e-4)	2.435e-2 (3.561e-3)
NSGA-II	3.856e-3 (2.135e-4)	5.644e-3 (1.390e-4)	4.821e-3 (1.788e-4)	3.216e-3 (2.693e-4)	5.741e-3 (3.575e-4)	3.620e-2 (2.226e-3)	5.826e-2 (2.645e-3)
MOEA/D-AWA	6.245e-3 (1.726e-4)	2.174e-3 (5.647e-4)	1.306e-3 (2.948e-4)	2.168e-3 (5.137e-4)	2.443e-3 (2.521e-4)	1.403e-2 (1.11e-3)	3.147e-2 (2.145e-4)
Adaptive-MOEA/D	2.574e-3 (1.103e-4)	2.165e-3 (3.874e-4)	1.248e-2 (2.963e-4)	3.620e-3 (1.250e-4)	5.331e-3 (5.899e-4)	1.605e-2 (3.266e-3)	3.296e-2 (1.130e-3)
HMOEA	3.374e-4 (7.015e-4)	5.714e-4 (1.122e-4)	7.338e-4 (1.240e-4)	1.511e-3 (4.336e-4)	1.269e-2 (9.136e-3)	2.941e-2 (3.589e-3)	3.254e-2 (1.124e-2)

表 4 各算法的 SP 值(第一行)及其标准差(第二行)

算法	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ1	DTLZ2
MOGSA/D	5.4597e-4 (4.248e-5)	3.3129e-4 (6.987e-5)	1.601e-3 (9.874e-4)	1.124e-3 (5.268e-4)	3.963e-3 (2.147e-4)	2.154e-3 (6.874e-4)	2.435e-2 (3.845e-3)
NSGA-II	7.798e-3 (7.401e-3)	7.416e-3 (6.578e-3)	8.423e-3 (1.756e-3)	6.3450e-3 (3.967e-3)	4.261e-3 (2.255e-4)	3.687e-1 (1.166e-0)	8.514e-2 (5.765e-2)
MOEA/D-AWA	4.541e-3 (2.531e-4)	4.152e-3 (1.354e-4)	6.116e-3 (3.276e-4)	4.114e-3 (2.172e-4)	3.608e-3 (2.819e-4)	2.137e-2 (1.954e-3)	2.187e-2 (1.193e-3)
Adaptive-MOEA/D	4.533e-3 (1.232e-4)	3.544e-3 (1.784e-4)	1.943e-2 (1.576e-3)	2.864e-3 (2.652e-4)	4.378e-3 (5.899e-4)	1.605e-2 (3.266e-3)	3.189e-2 (1.045e-3)
HMOEA	7.596e-1 (1.896e-2)	3.742e1 (6.987e-2)	3.174e-1 (6.996e-2)	2.874e-1 (3.069e-2)	5.625e-1 (2.396e-2)	1.336e-1 (2.648e-1)	3.254e-1 (1.124e-2)

可以看出,采取删减策略对解集进行删减后,指标 GD 和 SP 均得到较大的提升,即得到的解集无论是收敛性还是均匀性均得到了提高,验证了删减策略的有效性.

### 3.5 求解性能对比

主要考察 MOGSA/D 算法与对比算法在标准测试函数上的性能比较,测试结果如表 3 及表 4 所示.其中表 3 给出各算法在优化各问题时所得 GD 指标的均值及方差,表 4 给出各算法在优化各测试问题时所得 SP 指标的均值及方差.

从表 3 中可以看出,对于 ZDT1、ZDT2、ZDT3 和 ZDT4 等 4 个测试函数,MOGSA/D 算法在指标 GD 以及指标 SP 上都取得了优于其他对比算法的性能,也即收敛性以及分布性都是最好的;对函数 ZDT6 以及函数 DTLZ2,MOGSA/D 算法的 GD 是最小的,也即收敛性是最好的,分布性指标也仅次于 MOEA/D-AWA;对函数 DTLZ1,MOGSA/D 算法在收敛性指标上没有得到较好的效果,但分布性仍优于其他对比算法.综合来看,在求解 ZDT 和 DTLZ 问题上,本文所提算法无论在收敛性、鲁棒性还是分布性上均优于其他同类算法.

## 4 结 语

基于分解的多目标引力搜索算法 MOGSA/D, 通过对理想前沿面的估计, 自适应选取与之匹配的权重系数生成方式; 采用多子群串行搜索各子目标问题方法, 为充分利用子问题的相似性, 提出一种扰动生成新种群的策略; 根据所得解集中粒子间的拥挤度, 提出一种基于各目标加权的方式删减较劣解, 以保证解集更均匀地逼近理想前沿面. 通过对测试函数集 ZDT 和 DTLZ 上的实验证明, MOGSA/D 算法与现有的 MOEAs 相比, 具有更好地收敛性及和分布的均匀性.

## 参考文献

- [1] HERNÁNDEZ-DÍAZ A, SANTANA-QUINTERO L, COELLO COELLO C, et al. Pareto-adaptive  $\epsilon$ -dominance [J]. *Evolutionary Computation*, 2007, 15(4): 493–517.
- [2] 郭思涵, 龚小胜. 正交设计的 E 占优策略求解高维多目标优化问题研究[J]. *计算机科学*, 2012, 39(2): 276–279, 310.
- [3] BADER J, ZITZLER E. HypE: an algorithm for fast hypervolume-based many-objective optimization [J]. *Evolutionary Computation*, 2011, 19(1): 45–76.
- [4] LI Hui, ZHANG Qingfu. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 284–302.
- [5] RASHEDI E, SARYAZDI S. GSA: a gravitational search algorithm [J]. *Information Sciences*, 2009, 179(13): 2232–2248.
- [6] WANG R, PURSHOUSE R C, FLEMING P J. Preference-inspired co-evolutionary algorithm using weights for many-objective optimization [C]//*Proceeding of the fifteenth Annual Conference Companion on Genetic and Evolutionary Computation Conference Companion*. New York: ACM, 2013: 101–102.
- [7] QI Yutao, MA Xiaoliang, LIUFang, et al. MOEA/D with adaptive weight adjustment [J]. *Evolutionary Computation*, 2014, 22(2): 231–264
- [8] TAN Yanyan, JIAO Yongchang, LI Hong, et al. MOEA/D+uniform design: a new version of MOEA/D for optimization problems with many objectives [J]. *Computers & Operations Research*, 2013, 40(6): 1648–1660.
- [9] TANG L, WANG X. A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems [J]. *IEEE Trans on Evolutionary Computation*, 2013, 17(1): 20–45.
- [10] SIMON D. Biogeography-based optimization [J]. *IEEE Transactions on Evolutionary Computation*, I2008, 12(6): 702–713.
- [11] 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.
- [12] SUCIU M, PALLEZ D, CREMENE M, et al. Adaptive MOEA/D for QoS-based web service composition [M]. Berlin: Springer, 2013.

(编辑 王小唯)