

一. 实验原理

MOEA/D 算法，即一种基于分解的多目标优化算法，它将一个多目标优化问题分解成若干个标量优化子问题，并同时对其进行优化。每一个子问题都是通过使用来自它的几个相邻子问题的信息来优化的，这使得 MOEA/D 每一代的计算复杂度比 NSGA-II 要低。

1.1 多目标优化问题

一个多目标优化问题（MOP）可以被描述如下：

$$\begin{aligned} & \text{maximize } F(x) = (f_1(x), \dots, f_m(x))^T \\ & \text{subject to } x \in \Omega \end{aligned}$$

这里 Ω 是变量（决策）空间， $F: \Omega \rightarrow \mathbb{R}^m$ 包含 m 个实值目标函数， \mathbb{R}^m 被叫做目标空间。可实现的目标集合被定义为 $\{F(x) | x \in \Omega\}$ 。

如果 $x \in \mathbb{R}^n$ ，所有的目标都是连续的，那么 Ω 被描述为：

$$\Omega = \{x \in \mathbb{R}^n | h_j(x) \leq 0, j = 1, \dots, m\}$$

这里 h_j 是连续的函数，我们称上式为一个连续的多目标优化问题。

1.2 Pareto 支配关系

Pareto 支配关系：对于最小化多目标优化问题，对于 n 个目标分量 $f_i(x), i = 1 \sim n$ ，任意给定两个决策变量 X_a, X_b ，如果有以下两个条件成立，则称 X_a 支配 X_b 。

1. 对于 $\forall i \in \{1, \dots, n\}$ ，都有 $f_i(X_a) \leq f_i(X_b)$ 成立。
2. $\exists i \in \{1, \dots, n\}$ ，使得 $f_i(X_a) < f_i(X_b)$ 成立。

如果对于一个决策变量，不存在其他决策变量能够支配他，那么就称该决策变量为非支配解。

1.3 多目标优化分解

有几种方法可以将 PF 的近似问题转化为许多标量优化问题。在下面，我们介绍了两种方法。

1.3.1 权重和方法

这种方法考虑了不同目标的凸组合。令 $\lambda = (\lambda_1, \dots, \lambda_m)^T$ 为一个权重向量。即对于所有的 $i = 1, \dots, m$, $\lambda_i \geq 0$ and $\sum_{i=1}^m \lambda_i = 1$ 。然后问题转换为以下标量问题的最优解：

$$\begin{aligned} & \text{maximize } g^{ws}(x|\lambda) = \sum_{i=1}^m \lambda_i f_i(x) \\ & \text{subject to } x \in \Omega \end{aligned}$$

为了生成一组不同的帕累托最优向量，可以在上面的标量优化问题中使用不同的权重向量。如果 PF 是凹的（在最小化的情况下是凸凹的），这种方法可以很好地工作。然而，并不是每个帕累托最优向量都可以通过这种方法得到非凹面 PFs。通过理论和多次实践，此方法在非凸函数上表现不佳，因此本次对比实验并不采用此方法。

1.3.2 切比雪夫方法（Tchebycheff）

此方法将问题转化为如下标量问题：

$$\begin{aligned} & \text{minimize } g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i |f_i(x) - z_i^*|\} \\ & \text{subject to } x \in \Omega \end{aligned}$$

这里 $z^* = (z_1^*, \dots, z_m^*)^T$ 是参考点。对于任意 $i = 1, \dots, m$, $z_i^* = \max\{f_i^*(x) | x \in \Omega\}$ ³ 对于每一个 pareto 最优解 x^* 就存在一个权重向量 λ 使得其为此问题的 pareto 最优解。因此，通过改变权重向量，可以获得不同的帕累托最优解。这种方法的一个缺点是它的聚合函数对于连续的多目标优化问题来说不是平滑的。但是，它仍然可以在本 EA 框架中使用，因为此算法不需要计算聚合函数的导数。

1.4 基于分解的多目标进化框架（MOEA/D）

以 Tchebycheff 方法为例说明此框架。令 $\lambda^1, \dots, \lambda^N$ 为一组均匀分布的权重向量并且 z^* 是参考点。通过使用 Tchebycheff 方法，可以将 PF 近似的问题分解为标量优化子问题，而第 j 个子问题的目标函数是

$$\begin{aligned} & g^{te}(x|\lambda^j, z^*) = \max_{1 \leq i \leq m} \{\lambda_i^j |f_i(x) - z_i^*|\} \\ & \text{subject to } x \in \Omega \end{aligned}$$

这里 $\lambda^j = (\lambda_1^j, \dots, \lambda_m^j)^T$ 。在一次运行中，MOEA/D 同时最优化所有的 N 个目标。

g^{te} 是关于 λ 连续的，因此当 λ^j 和 λ^i 彼此接近时， $g^{te}(x|\lambda^j, z^*)$ 和 $g^{te}(x|\lambda^i, z^*)$ 也应该非常接近。因此任何关于接近 λ^i 的这些 g^{te} 权重向量的信息都应有助于优化 $g^{te}(x|\lambda^i, z^*)$ 。这也是 MOEA/D 工作的机理。

在 MOEA/D 中，一个权重向量的邻域被定义为它的几个最近的权重向量的集合。第 i 个子问题的邻域包括来自 λ^i 邻域的所有的子问题的权重向量。种群是由迄今为止每个子问题找到的最好的解决方案组成的。只有当前对其相邻子问题的解决方案被利用来优化 MOEA/D 中的子问题。在每代中，MOEA/D 以 Tchebycheff 方法保持：

- N 个点的种群 $x^1, \dots, x^N \in \Omega$ ，这里 x^i 是当前第 i 个子问题；
- FV^1, \dots, FV^N ，这里 FV^i 是 x^i 的 F-value，即对每个 $i = 1, \dots, N$, $FV^i = F(x^i)$ ；
- $z = (z_1, \dots, z_m)^T$ ，这里 z_i 是发现的最好的值。
- 外部人口（EP），用于在搜索过程中储存的非支配解。

此算法流程如下：

Input:

- MOP(1);
- 一个停止准则;
- N: 子问题的数量;
- N 个均匀分布的权向量;
- T: 每个权向量领域内权向量的个数。

Output: EP。

步骤 1) 初始化:

步骤 1.1) 设置 EP 为空集 (或者放在 1.3 后从初始化的权向量中选出非支配解初始化 EP)

步骤 1.2) 计算任意两个权向量之间的欧式距离, 然后计算每个权向量最近的 T 个权向量。对于 $i = 1, \dots, N$, 设置 $B(i) = \{i_1, \dots, i_T\}$, 这里 $\lambda^{i_1}, \dots, \lambda^{i_T}$ 是 T 个接近 λ^i 的权重向量。

步骤 1.3) 随机或通过问题的特定方法生产一个初始种群。设置 $FV^i = F(x^i)$ 。

步骤 1.4) 用问题特定的方法初始化 $z = (z_1, \dots, z_m)^T$ 。

步骤 2) 更新:

步骤 2.1) 复制: 随机从 $B(i)$ 中选择两个索引 k, l , 然后对 x^k 和 x^l 使用遗传算子产生一个新的解 y 。

步骤 2.2) 改善: 应用一个启发式方法修理/改进 y 来生产 y' 。

步骤 2.3) 更新 z: 对于所有 $j = 1, \dots, m$, 如果 $z_j < f_j(y')$, 那么设置 $z_j = f_j(y')$ 。

步骤 2.4) 更新邻域解: 对于 $j \in B(i)$, 如果 $g^{te}(y' | \lambda^j, z) \leq g^{te}(x^j | \lambda^j, z)$, 那么设置 $x^j = y'$ 和 $FV^j = F(y')$ 。

步骤 2.5) 更新 EP:

✓ 从 EP 中移除被 $F(y')$ 支配的所有向量。

✓ 如果 EP 中没有向量支配 $F(y')$, 就将 $F(y')$ 加入到 EP 中。

步骤 3) 停止准则: 如果停止准则满足, 停止并输出 EP。否则, 转向步骤 2。

1.5 实数编码的交叉操作 (SBX)

模拟二进制交叉:

$$\tilde{x}_{1j}(t) = 0.5 \times [(1 + \gamma_j)x_{1j}(t) + (1 - \gamma_j)x_{2j}(t)]$$

$$\tilde{x}_{2j}(t) = 0.5 \times [(1 - \gamma_j)x_{1j}(t) + (1 + \gamma_j)x_{2j}(t)]$$

其中

$$\gamma_j = \begin{cases} (2u_j)^{\frac{1}{\eta+1}} & u_j < 0.5 \\ (\frac{1}{2(1-u_j)})^{\frac{1}{\eta+1}} & \text{其他} \end{cases}$$

并且 $0 \leq u_j \leq 1$ 。

1.6 多项式变异 (polynomial mutation)

多项式变异:

$$\tilde{x}_{1j}(t) = x_{1j}(t) + \Delta_j$$

其中

$$\Delta_j = \begin{cases} (2u_j)^{\frac{1}{\eta+1}} - 1 & u_j < 0.5 \\ 1 - (2(1 - u_j))^{\frac{1}{\eta+1}} & \text{其他} \end{cases}$$

并且 $0 \leq u_j \leq 1$ 。

1.7 评价指标

1.7.1 C-metric (覆盖率)

令 A 和 B 是一个 MOP 中两个接近 PF 的集合, 定义 C(A,B)如下:

$$C(A, B) = \frac{|\{u \in B | \exists v \in A: v \text{ dominates } u\}|}{|B|}$$

C(A,B)不等于 1-C(B,A)。C(A,B)=1 意味着 B 中所有的解都被 A 中的某些解支配了, C(A,B)=0 意味着 B 中没有解被 A 中的解支配。

1.7.2 D-metric (距离度量)

令 P^* 为一组均匀分布在 PF 上的点的集合。A 是一个接近 PF 的集合。 P^* 到 A 的平均距离定义为:

$$D(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}$$

这里 $d(v, A)$ 是 v 和 A 中的点的最小欧式距离。如果 P^* 足够大说明其可以很好的代表 PF。 $D(A, P^*)$ 可以从某种意义上评估 A 的收敛性和多样性。为了让 $D(A, P^*)$ 的值很低, 必须设置 A 非常接近 PF, 并且不能缺失整个 PF 的任何部分。

二. 在连续的 MOP 中与 NSGA-II 对比

2.1 连续多目标的测试函数

本实验采用的是两目标的 ZDT 函数和三目标的 DTLZ 函数, 具体函数如下:

➤ ZDT1

$$f_1 = x_1$$

$$g = 1 + 9 \left(\frac{\sum_{i=2}^n x_i}{n-1} \right)$$

$$f_2 = g \left(1 - \left(\frac{f_1}{g} \right)^{0.5} \right). n = 30, x_i \in [0,1]$$

➤ ZDT2

$$f_1 = x_1$$

$$g = 1 + 9 \left(\frac{\sum_{i=2}^n x_i}{n-1} \right)$$

$$f_2 = g \left(1 - \left(\frac{f_1}{g} \right)^2 \right). n = 30, x_i \in [0,1]$$

➤ ZDT3

$$f_1 = x_1$$

$$g = 1 + 9 \left(\frac{\sum_{i=2}^n x_i}{n-1} \right)$$

$$f_2 = g \left(1 - \left(\frac{f_1}{g} \right)^{0.5} - \left(\frac{f_1}{g} \right) \sin(10\pi f_1) \right). n = 30, x_i \in [0,1]$$

➤ ZDT4

$$f_1 = x_1$$

$$g = 1 + 9 * 10 + \sum_{i=2}^n ((x_i)^2 - 10 \cos(4\pi x_i))$$

$$f_2 = g \left(1 - \left(\frac{f_1}{g} \right)^{0.5} \right). n = 10, x_1 \in [0,1]; x_{2...6} \in [-5,5]$$

➤ ZDT6

$$f_1 = 1 - e^{-4x_1} \sin^6(6\pi x_1)$$

$$g = 1 + 9 \left(\frac{\sum_{i=2}^n x_i}{n-1} \right)^{0.25}$$

$$f_2 = g \left(1 - \left(\frac{f_1}{g} \right)^2 \right). n = 10, x_i \in [0,1]$$

➤ DTLZ1

$$\begin{aligned} f_1 &= (1 + g)x_1x_2 \\ f_2 &= (1 + g)x_1(1 - x_2) \\ f_3 &= (1 + g)(1 - x_1) \\ g &= 100(n - 2) + 100 \sum_{i=3}^n \{(x_i - 0.5)^2 - \cos[20\pi(x_i - 0.5)]\} \end{aligned}$$

➤ DTLZ2

$$\begin{aligned} f_1 &= (1 + g)\cos(0.5\pi x_1)\cos(0.5\pi x_2) \\ f_2 &= (1 + g)\cos(0.5\pi x_1)\sin(0.5\pi x_2) \\ f_3 &= (1 + g)\sin(0.5\pi x_1) \\ g &= \sum_{i=3}^n (x_i)^2 \end{aligned}$$

2.2 MOEA/D 的变体

为了一个公平的比较，使用以下 MOEA/D 的变体进行对比实验。

- 没有额外的种群 EP。即最终的代数返回作为 PF 的近似。因此，步骤 2.5 删除。
- 没有修理和改进，即删除步骤 2.2。
- 使用切比雪夫方法。

2.3 MOEA/D 和 NSGA-II 的复杂度比较

➤ 空间复杂度

由于 NSGA-II 没有外部人口，所以在这个过程中，我们没有保持外部人口的数量。与 NSGA-II 相比，MOEA/D 中唯一的额外内存需求是存储 z 。 z 的大小是 $O(m)$ 。与种群规模相比，它的大小是非常小的。

➤ 时间复杂度

在上面的 MOEA/D 的变体中，主要的计算成本在步骤 2 中。在 MOEA/D 中，步骤 2 MOEA/D 产生了 N 个试验解决方案，每一代的 NSGA-II 也是如此。注意步骤 2.2 和 2.5 已经从这种变体移除，步骤 2.1 利用遗传算子随机选择两个解，步骤 2.3 进行 $O(m)$ 次比较，第 2.4 步需要 $O(mT)$ 次基本操作。因此，在上面的 MOEA/D 变体中，步骤 2 的计算复杂度是 $O(mNT)$ 。如果 MOEA/D 和 NSGA-II 都使用相同的种群规模，那么它们的计算量每一代的复杂性为

$$\frac{O(mNT)}{O(mN^2)} = \frac{O(T)}{O(N)}$$

因为 T 小于 N ，所以 MOEA/D 比 NSGA-II 复杂度更低。

三. 实验结果

本次实验分为两部分，第一部分为 Tchebycheff 的 MOEA/D 算法的实现结果，

第二部分为 Tchebycheff 的 MOEA/D 变种算法和 NSGA-II 的对比实验结果。

3.1 Tchebycheff 的 MOEA/D 实验结果

在两个多目标测试问题中，MOEA/D 的种群数目都设置为 100，在三个多目标测试问题中都设为 300。算法迭代停止在 250 代。算法均采用实数编码，交叉算子和变异算子采用模拟二进制交叉和多项式变异。交叉概率 $p_c=1$ ，变异概率 $p_m=1/n$ ， $\eta_c=2$ ， $\eta_m=5$ 。T 设为 20。每次对比实验都运行 5 次，选取 D 指标最小的实验结果如图 1 所示。

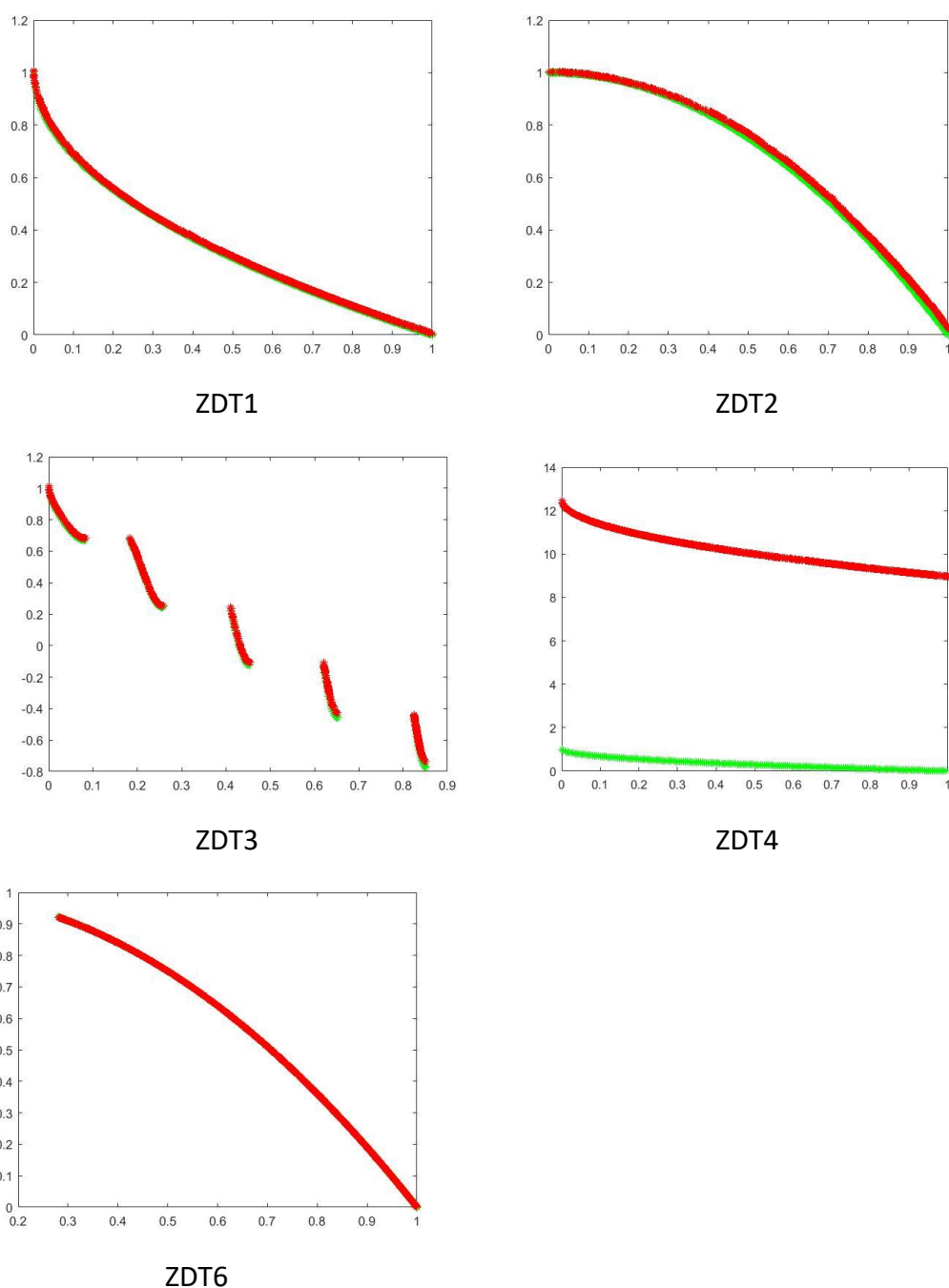
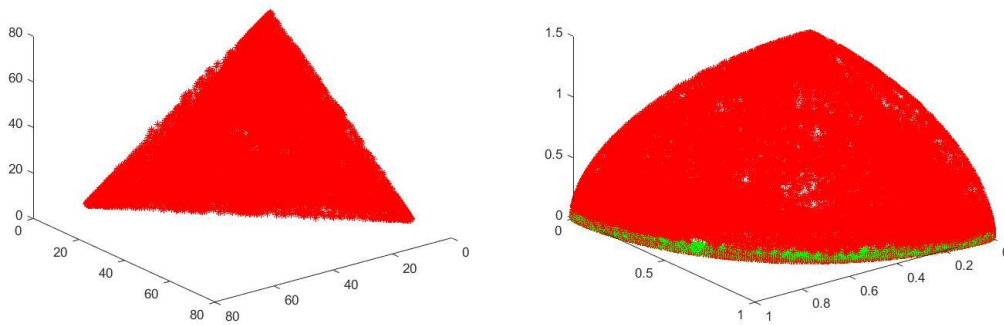


图 1 最低的 D 指标的 MOEA/D 算法在两目标问题中得到的 pareto 最优解对比

图（绿色为 PF，红色为 MOEA/D）



DTLZ1

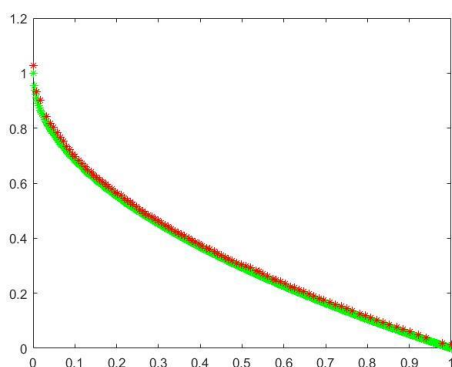
DTLZ2

图 2 最低的 D 指标的 MOEA/D 算法在三目标问题中得到的 pareto 最优解对比图（绿色为 PF，红色为 MOEA/D）

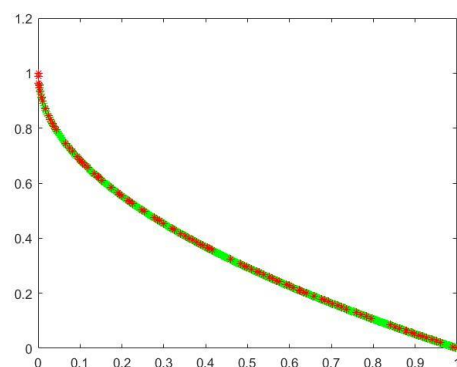
从图 1 和图 2 可以看出，对于三目标问题而言，找到的解在目标空间分布十分均匀，但离 PF 还有一定的差距。对于两目标问题而言，除 ZDT4 以外，找到的解几乎全部是 pareto 前端上的点，并且解在目标空间上分布十分均匀，该算法对于非凸非均匀的多目标函数最优解的寻找还是十分有效的。由于 ZDT4 具有许多局部 pareto 最优前沿，为了摆脱这些局部前沿，需要对决策向量进行大的改变，因此选取较强变异因子，但仍离真正的 pareto 前沿还有很大的差距。

3.2 Tchebycheff 的 MOEA/D 变种和 NSGA-II 的对比实验结果

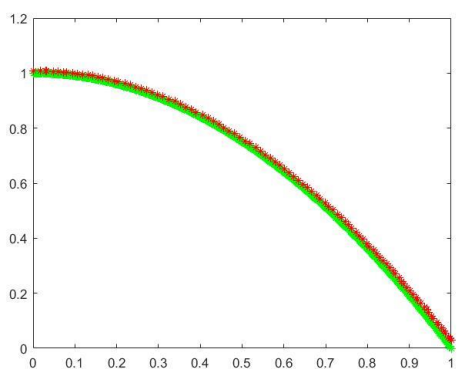
在两个多目标测试问题中，NSGA-II 和 MOEA/D 变体的种群数目都设置为 100，在三个多目标测试问题中都设为 300。算法迭代都停止在 250 代。两个算法均采用实数编码，NSGA-II 的选择算子为二进制锦标赛。交叉算子和变异算子二者均采用模拟二进制交叉和多项式变异。交叉概率 $p_c=1$ ，变异概率 $p_m=1/n$ ， $\eta_c=2$ ， $\eta_m=5$ 。T 设为 20。每次对比实验都运行 5 次，计算 C 指标和 D 指标的均值和方差，结果如表格 1-3 所示。画出最低的 D 指标的 NSGA-II 和 MOEA/D 变种算法的 pareto 最优解对比图如图 3-4 所示。



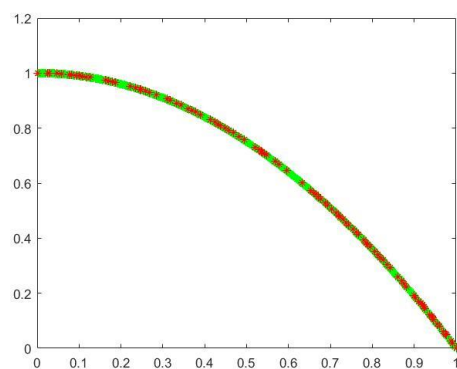
ZDT1



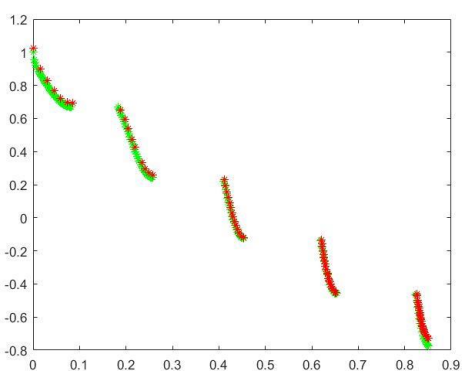
ZDT1



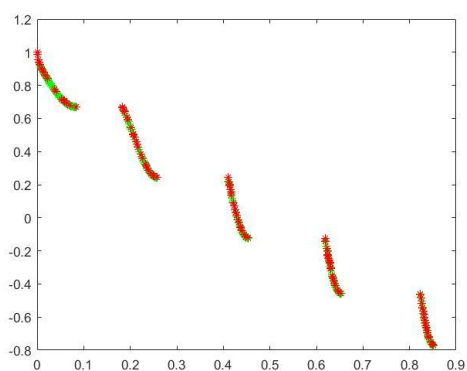
ZDT2



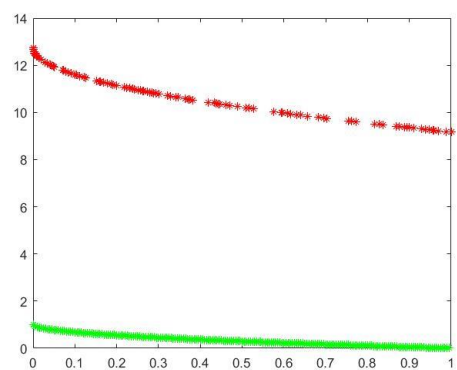
ZDT2



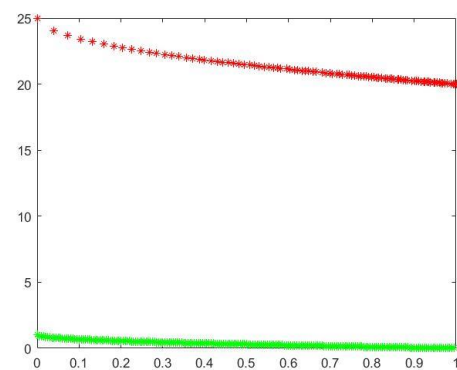
ZDT3



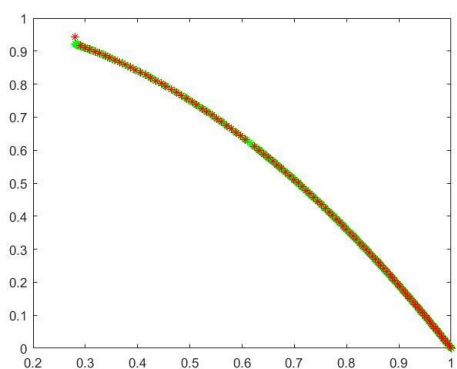
ZDT3



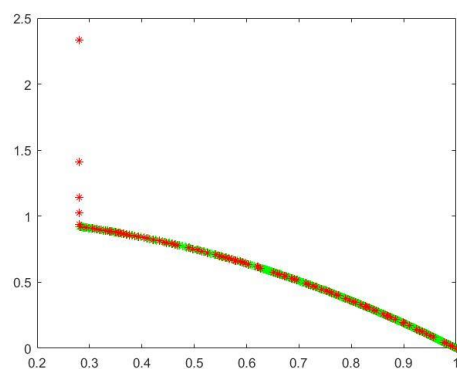
ZDT4



ZDT4



ZDT6



ZDT6

图 3 最低的 D 指标的 NSGA-II 和 MOEA/D 变种算法在两目标问题中得到的 pareto 最优解对比图（绿色为 PF，左图红色为 MOEA/D，右图红色为 NSGA-II）

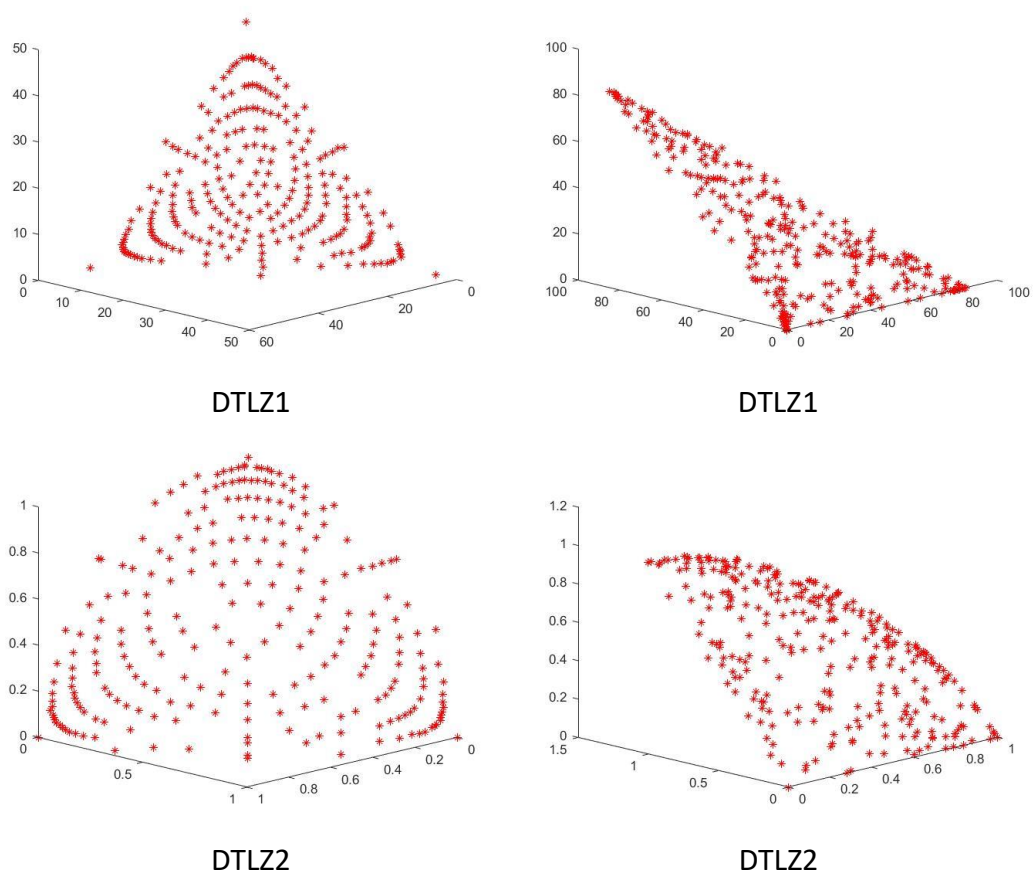


图 4 最低的 D 指标的 NSGA-II 和 MOEA/D 变种算法在三目标问题中得到的 pareto 最优解对比图（左图为 MOEA/D，右图为 NSGA-II）

图 3 和图 4 显示了在目标空间中，最低的 D 指标的 NSGA-II 和 MOEA/D 变种算法的 pareto 最优解分布，显而易见，在 ZDT1、ZDT2、ZDT6、DTLZ1 和 DTLZ2 问题中，MOEA/D 的解要优于 NSGA-II。在 ZDT4 问题中，两者的表现均不好。然而在 ZDT3 问题中，因为 MOEA/D 找到 PF 中某两段的解非常少，因此它的解明显没有 NSGA-II 好。

表 1 NSGA-II 和使用 TCHEBYCHEFF 方法的 MOEA/D 变种算法的平均运行时间（以秒为单位）

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ1	DTLZ2
NSGA-II	4.6125	7.1343	12.0886	10.8920	15.2714	70.3286	48.9496
MOEA/D	2.7837	2.7955	2.7785	2.4876	2.4900	6.8617	6.7825

表 2 NSGA-II 和使用 TCHEBYCHEFF 方法的 MOEA/D 变种算法的平均覆盖率（C 指标）

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ1	DTLZ2
NSGA-II	1.0000	1.0000	0.9540	1.0000	0.6360	0.9540	0.0347

MOEA/D	0.6240	0.6680	0.3800	1.0000	0.0820	0.8647	0.0280
--------	--------	--------	--------	--------	--------	--------	--------

表 3 NSGA-II 和使用 TCHEBYCHEFF 方法的 MOEA/D 变种算法的平均距离度量
(括号内为方差)

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ1	DTLZ2
NSGA-II	0.0061 (4.9864 e-07)	0.0057 (1.8029 e-07)	0.0065 (8.2156 e-08)	28.6043 (65.334 0)	0.0046 (7.9976 e-07)	63.4303 (1.6226e+ 2)	0.0410 (1.3114 e-05)
MOEA/D	0.0104 (1.7416 e-6)	0.1823 (0.1464)	0.0145 (2.4638 e-06)	12.6876 (9.6182)	0.0031 (1.7661 e-07)	45.0161 (256.7031)	0.0346 (1.2613 e-05)

表 1 显示了每个算法所使用的平均时间，从中可以清楚的看出，对于两目标问题而言，MOEA/D 的运行速度比 NSGA-II 快 2 倍以上，而对于三目标问题来讲，快了 8 倍以上，这一观察结果和之前的理论分析是一致的。

表 2 显示每个算法的平均覆盖率。易得，除了 ZDT4 外，MOEA/D 获得的最终解决方案都比 NSGA-II 要好。

表 3 显示每个算法的距离度量的平均值和方差，从表中可知，MOEA/D 在 ZDT4, ZDT6, DTLZ1 和 DTLZ2 上比 NSGA-II 要好，而对于其他三个问题则稍差。

综上所述，基于 Tchebycheff 的 MOEA/D 变种算法需要的运行时间比 NSGA-II 的要少，在解的质量方面，这两种算法在两个目标问题上具有差不多的性能，但是在两个 3 个目标问题上，MOEA/D 比 NSGA-II 要好。

四. 参考文献

- [1] 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.
- [2] Zhang Q, Li H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6):712-731.