## 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：

这里Ω是变量（决策）空间，F：Ω→Rm包含m个实值目标函数，Rm被叫做目标空间。可实现的目标集合被定义为。

如果，所有的目标都是连续的，那么Ω被描述为：

这里hj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 任意给定两个决策变量,如果有以下两个条件成立，则称。

1. 对于，都有成立。
2. ，使得成立。

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。令为一个权重向量。即对于所有的。然后问题转换为以下标量问题的最优解：

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：

这里是参考点。对于任意,

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。令为一组均匀分布的权向量并且是参考点。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

这里。在一次运行中，MOEA/D同时最优化所有的N个目标。

是关于连续的，因此当和彼此接近时，和也应该非常接近。因此任何关于接近的这些权重向量的信息都应有助于优化。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邻域的所有的子问题的权向量。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，这里xi是当前第i个子问题；
* FV1, …, FVN，这里FVi是xi的F-value，即对每个，FVi=F(xi)；
* ，这里是发现的最好的值。
* 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，设置，这里是T个接近的权重向量。

**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FVi=F(xi)。

**步骤1.4）**用问题特定的方法初始化。

**步骤2）更新：**

**步骤2.1）复制**：随机从B(i)中选择两个所引k，l，然后对xk和xl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，如果，那么设置。

**步骤2.4）更新邻域解**：对于，如果，那么设置和FVj=F(yi)。

**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：

其中

并且。

**1.6 多项式变异（polynomial mutation）**

多项式变异：

其中

并且。

**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)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：

这里是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。可以从某种意义上评估A的收敛性和多样性。为了让的值很低，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
* ZDT2
* ZDT3
* ZDT4
* ZDT6
* DTLZ1
* DTLZ2

**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2MOEA/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

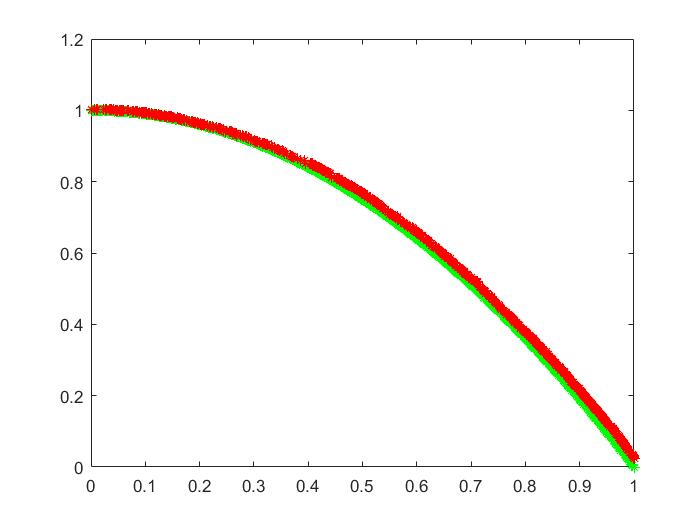
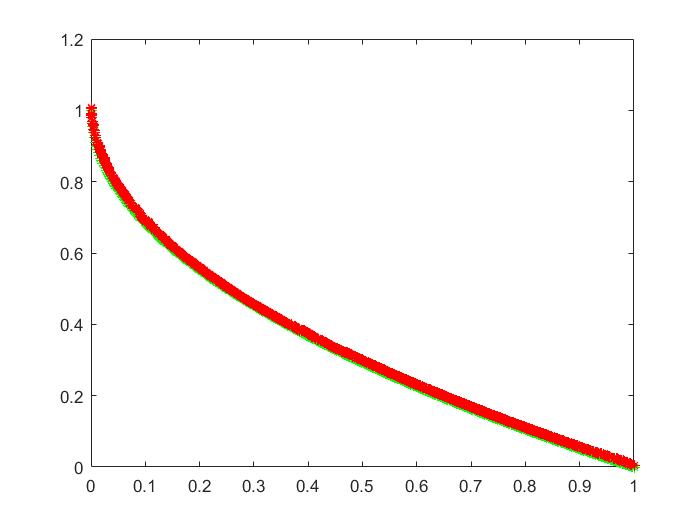
因为T小于N，所以MOEA/D比NSGA-II复杂度更低。

## 三．实验结果

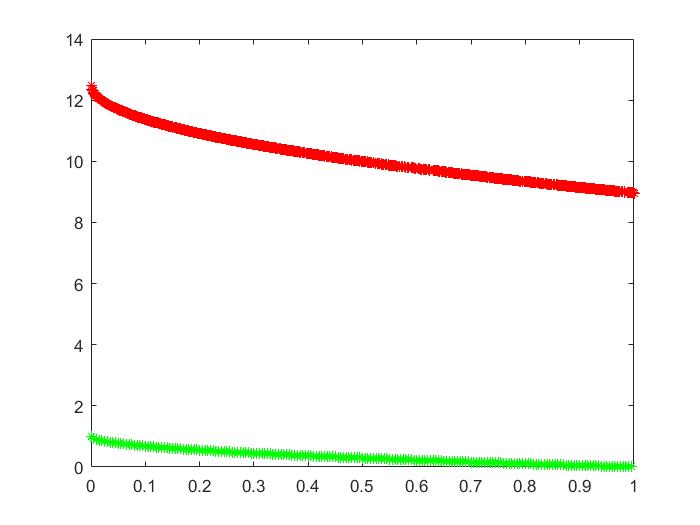
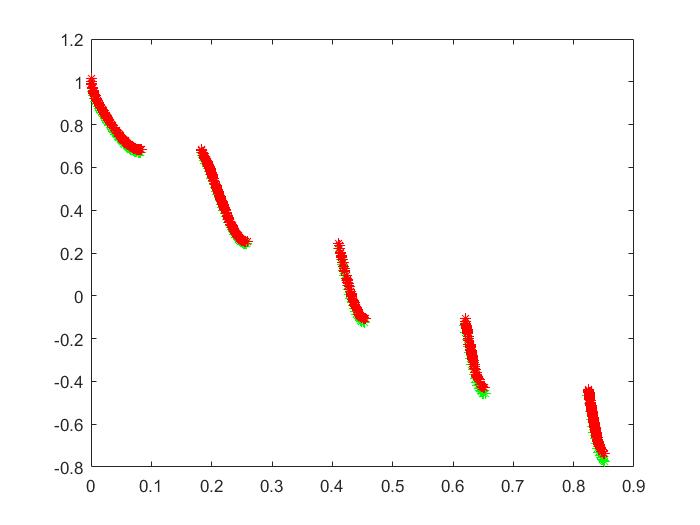
本次实验分为两部分，第一部分为Tchebycheff的MOEA/D算法的实现结果，第二部分为Tchebycheff的MOEA/D变种算法和NSGA-II的对比实验结果。

**3.1** **Tchebycheff的MOEA/D实验结果**

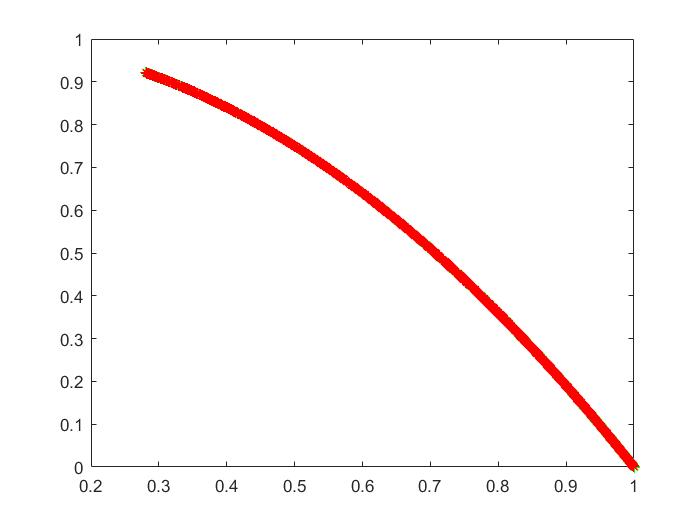
在两个多目标测试问题中，MOEA/D的种群数目都设置为100，在三个多目标测试问题中都设为300。算法迭代停止在250代。算法均采用实数编码，交叉算子和变异算子采用模拟二进制交叉和多项式变异。交叉概率pc=1，变异概率pm=1/n，ηc=2，ηm=5。T设为20。每次对比实验都运行5次，选取D指标最小的实验结果如图1所示。



ZDT1 ZDT2

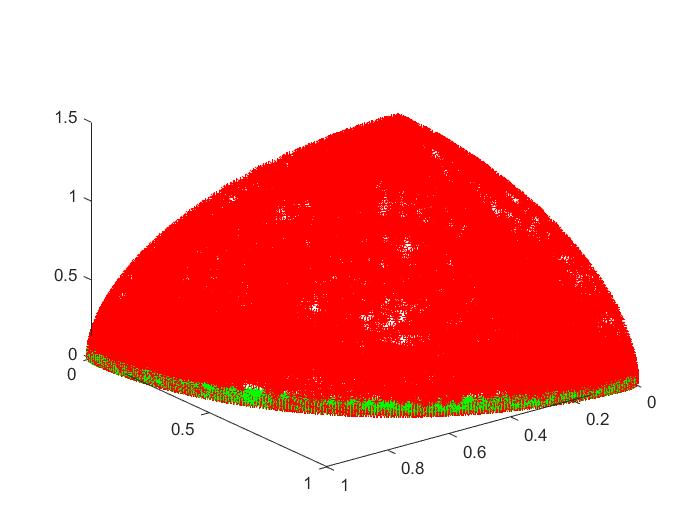
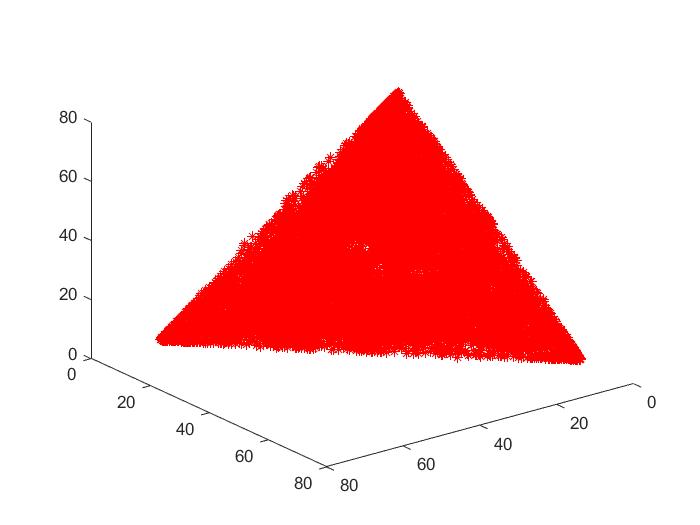


ZDT3 ZDT4



ZDT6

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）



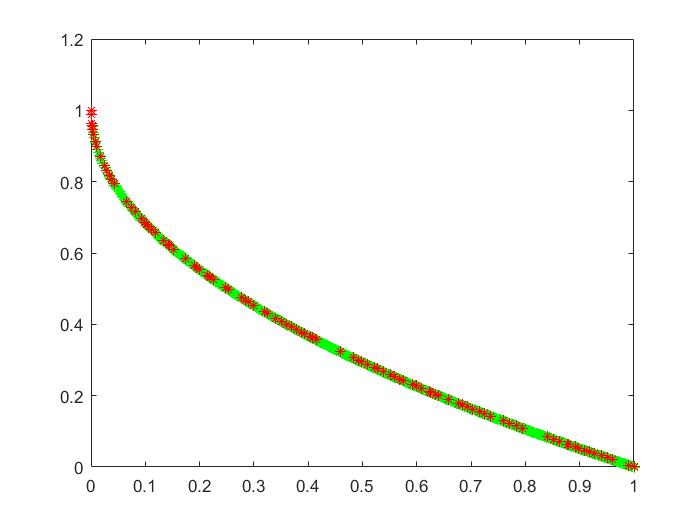
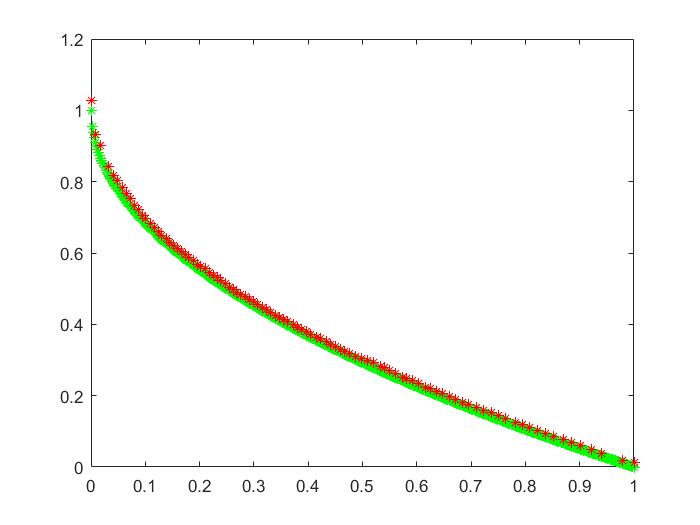
DTLZ1 DTLZ2

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）

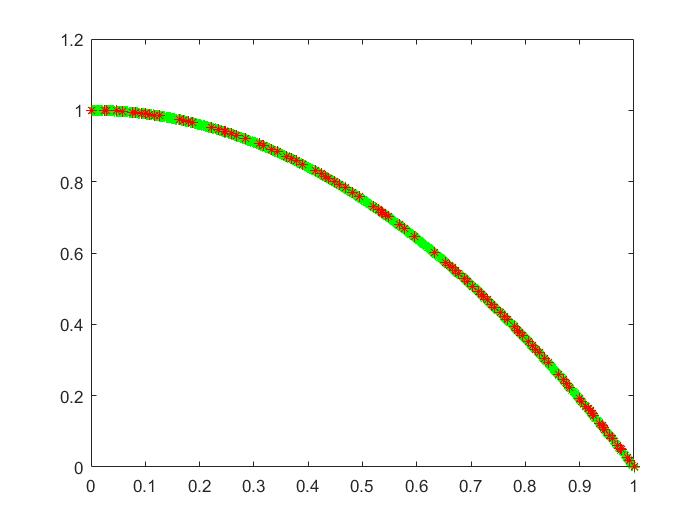
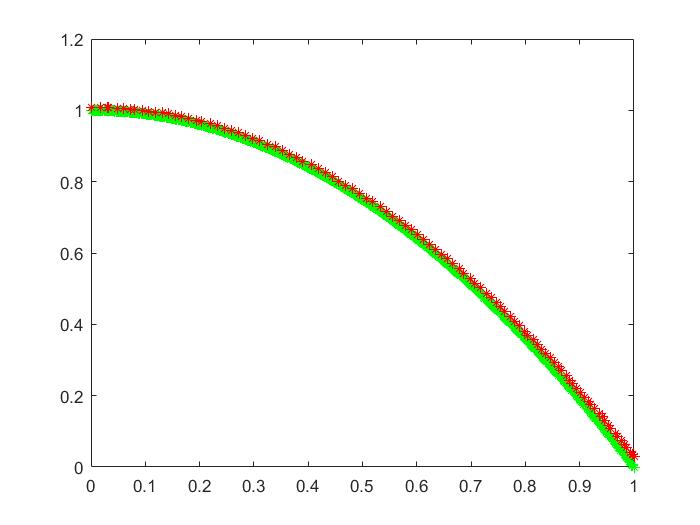
从图1和图2可以看出，对于三目标问题而言，找到的解在目标空间分布十分均匀，但离PF还有一定的差距。对于两目标问题而言，除ZDT4以外，找到的解几乎全部是pareto前端上的点，并且解在目标空间上分布十分均匀，该算法对于非凸非均匀的多目标函数最优解的寻找还是十分有效的。由于ZDT4具有许多局部pareto最优前沿，为了摆脱这些局部前沿，需要对决策向量进行大的改变，因此选取较强变异因子，但仍离真正的pareto前沿还有很大的差距。

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

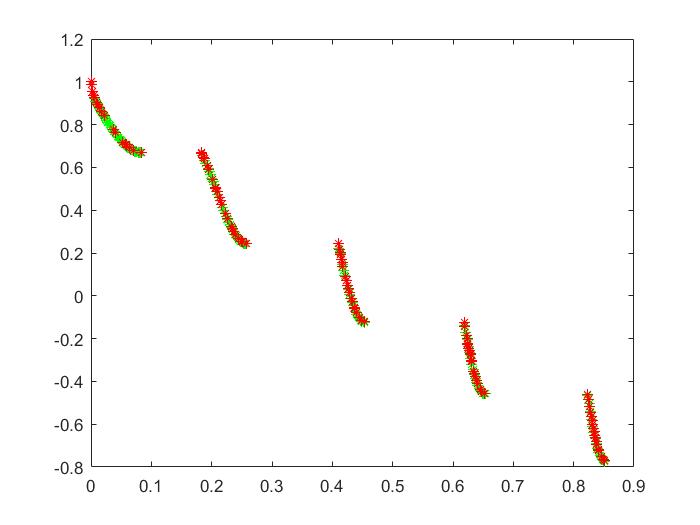
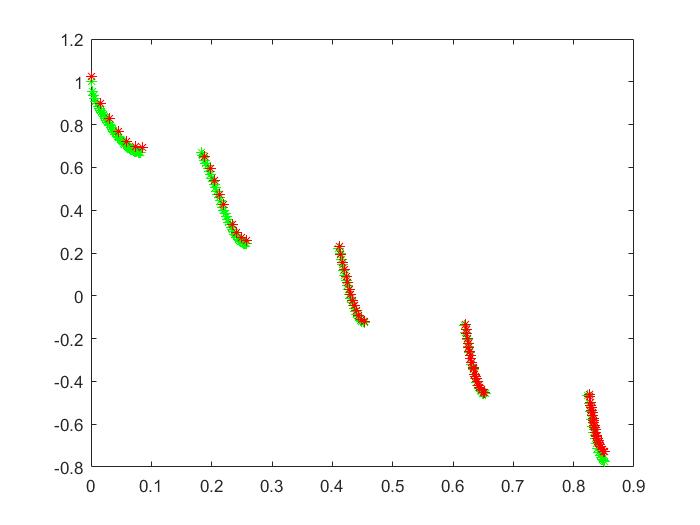
在两个多目标测试问题中，NSGA-II和MOEA/D变体的种群数目都设置为100，在三个多目标测试问题中都设为300。算法迭代都停止在250代。两个算法均采用实数编码，NSGA-II的选择算子为二进制锦标赛。交叉算子和变异算子二者均采用模拟二进制交叉和多项式变异。交叉概率pc=1，变异概率pm=1/n，ηc=2，ηm=5。T设为20。每次对比实验都运行5次，计算C指标和D指标的均值和方差，结果如表格1-3所示。画出最低的D指标的NSGA-II和MOEA/D变种算法的pareto最优解对比图如图3-4所示。



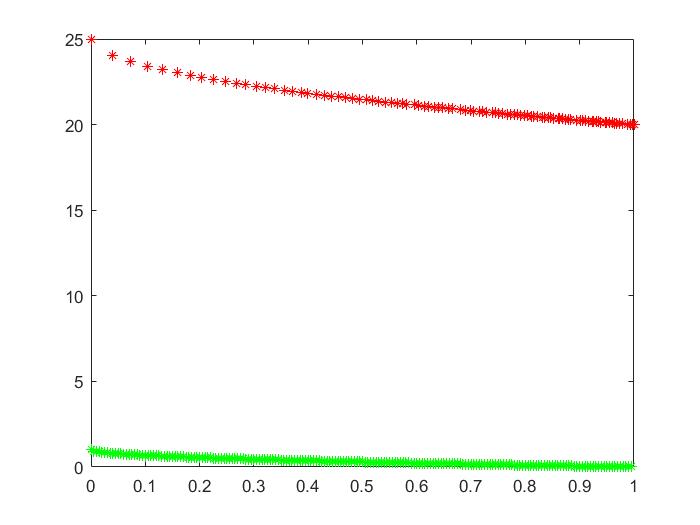
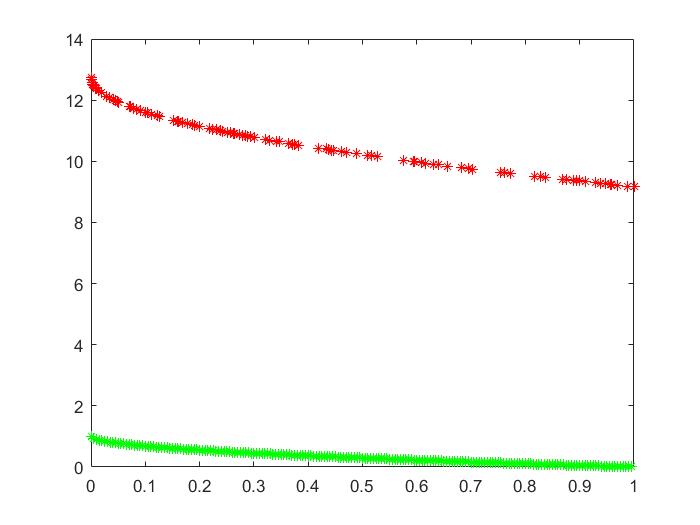
ZDT1 ZDT1



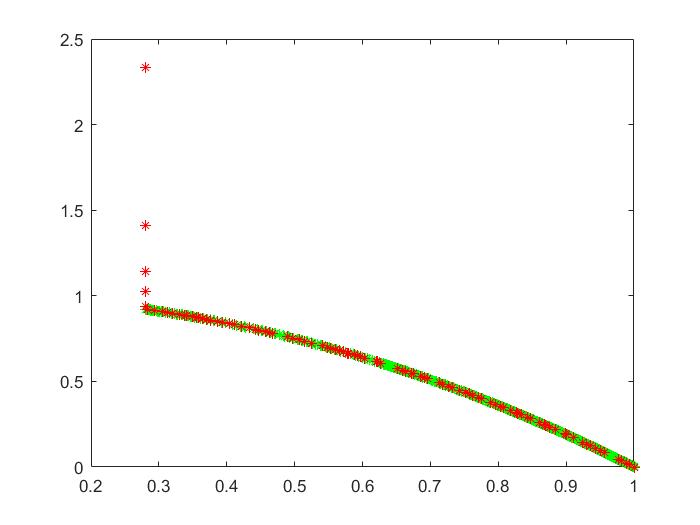
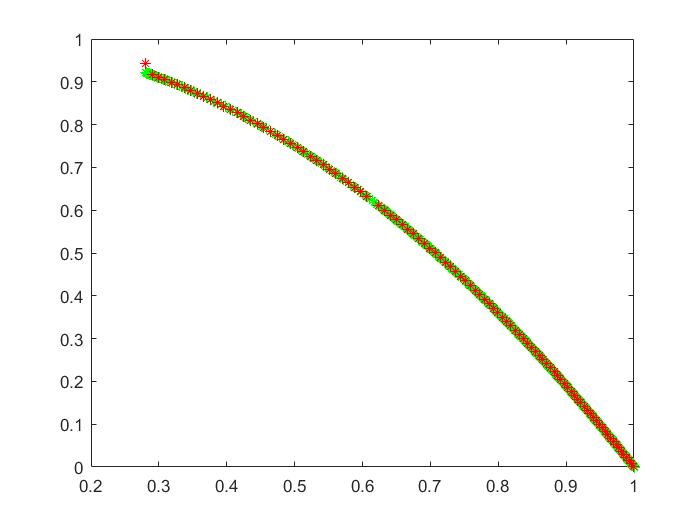
ZDT2 ZDT2



ZDT3 ZDT3

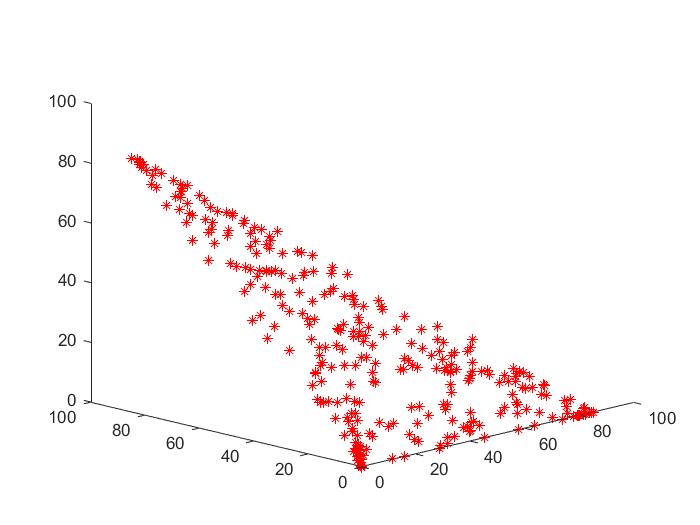
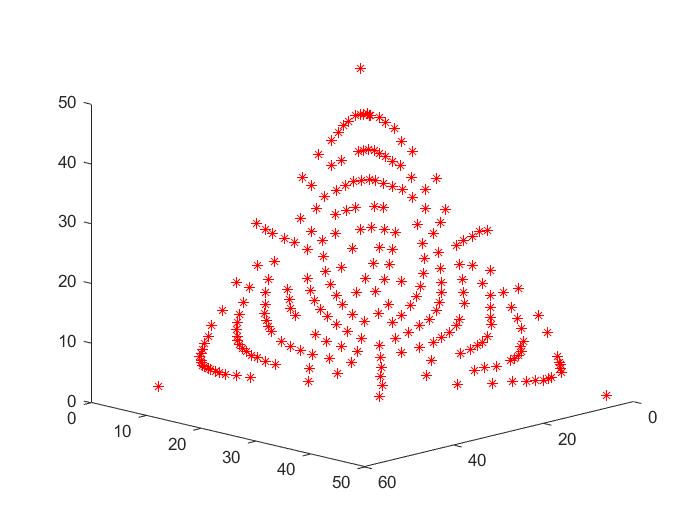


ZDT4 ZDT4

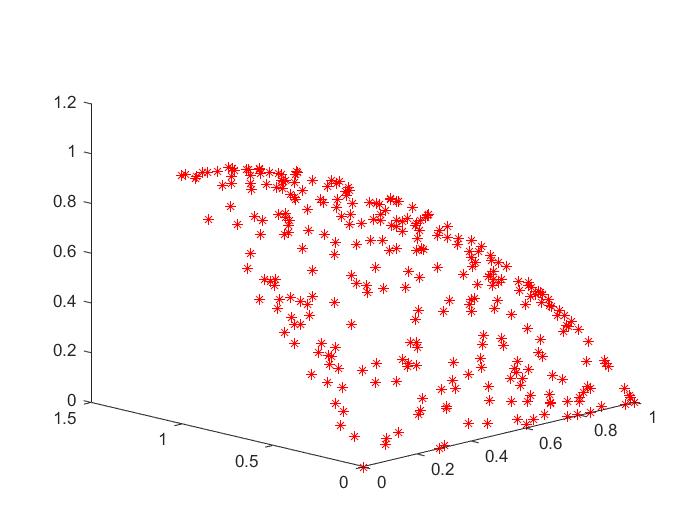
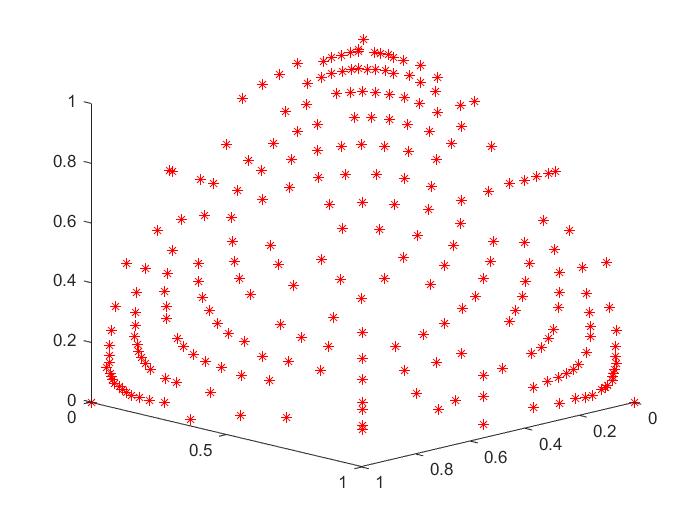


ZDT6 ZDT6

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）



DTLZ1 DTLZ1



DTLZ2 DTLZ2

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  (4.9864e-07) | 0.0057  (1.8029e-07) | 0.0065  (8.2156e-08) | 28.6043  (65.3340) | 0.0046  (7.9976e-07) | 63.4303  (1.6226e+2) | 0.0410  (1.3114e-05) |
| MOEA/D | 0.0104  (1.7416e-6) | 0.1823  (0.1464) | 0.0145  (2.4638e-06) | 12.6876  (9.6182) | 0.0031  (1.7661e-07) | 45.0161  (256.7031) | 0.0346  (1.2613e-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.