## MOEAD实验结果

### Pareto Front







以上为ZDT1、ZDT2、ZDT3测试函数分别使用NSGA-II和MOEAD得到的Pateto Front，可以明显看出MOEAD得到的解明显要比NSGA-II更加均匀，这得益于MOEAD采用把多目标优化问题分解为许多个标量优化问题的策略



上图是MOEAD得到的3变量DTLZ1、DTLZ2测试函数的PF，权重向量采用随机生成的方式，效果欠佳，目前正在学习3维权重向量的生成方法

### 主要参数

#### 1、MOEAD

交叉算子：SBX Pc=1

变异算子：多项式变异 Pm=1/n

迭代次数：200

种群数量：N=100（2目标）、N=300（3目标）

近邻数量：T=20

分解方法：Tchebycheff Approach

为了与NSGA-II比较，本实验不保存外部种群，直接返回当前种群作为最终解

#### 2、NSGA\_II

交叉算子：二进制单点交叉 Pc=0.9

变异算子：二进制位点变异 Pm=1/n

迭代次数：200

种群数量：100

### **3、**python运行时间

|  |  |  |
| --- | --- | --- |
|  | NSGA-II(s) | MOEA\_D(s) |
| ZDT1 | 77.757 | 7.241 |
| ZDT2 | 77.935 | 7.221 |
| ZDT3 | 74.510 | 7.202 |
| DTLZ1 |  | 17.276 |
| DTLZ2 |  | 17.223 |

除交叉、变异算子不同外，保持MOEAD和NSGA-II种群数量、迭代次数相同，并且MOEAD不保存外部种群，从上表中可以看出MOEAD运算开销明显小于NSGA-II。

### 4、总结

MOEAD采用分解的策略将多目标优化问题转化为多个单目标优化问题，使得最终得到的PF更加均匀，其中的每个个体的权重向量至关重要，计算开销也要比NSGA-II小。本算法中有两个难点：其一是如何产生均匀分布的权重向量，二维均匀分布的权重向量好产生，三维有些困难，目前正在学习。其二是在算法第2步中如何对交叉变异后的后代进行改善、修复，目前仅仅通过决策空间的边界进行限制，还没通过具体问题对产生的解进行改进。

在试验中，有些函数的目标空间函数值存在异比性，比如ZDT3，使用MOEAD得出的PF不够均匀，解更加偏向取值范围更大的那个目标函数值。在论文中的解决方案是使用目标归一化，聚合函数使用



使用后发现，解更加偏向取值范围小的那个目标函数值，具体原因还在寻找。