

**优化算法与实现**

**课程学习报告**

**题 目：基于分解的多目标优化算法（MOEA/D）**

**姓 名：**  **沈正圆**

**专业班级：**  **计算机技术1802**

**学 号：** 2111812118

2018年12月11日

**优化算法与实现报告**

1. **文献综述**

所谓的目标优化问题一般是指通过某种优化算法得到的目标函数的最优解。 当优化的目标函数是一个时，它被称为单目标优化。 当优化的目标函数有两个或更多时，它被称为多目标优化。与单目标优化不同，多目标优化的解决方案通常是一组均衡解。显然，多目标优化问题更接近工程实践，比单目标优化问题更复杂。工程实践中的许多优化问题最终可以转化为多目标优化问题。因此，深入研究多目标优化问题对实际应用更有价值。通常，一些算法可以解决多目标优化问题。

多目标优化问题首先由法国经济学家V. Pareto提出，当他研究经济平衡时，引入并推广了Pareto最优解。实际中优化问题大多数是多目标优化问题 ,一般情况下 ,多目标优化问题的各个子目标之间是矛盾的 ,一个子目标的改善有可能会引起另一个或者另几个子目标的性能降低 , 也就是要同时使多个子目标一起达到最优值是不可能的 , 而只能在它们中间进行协调和折中处理 , 使各个子目标都尽可能地达到最优化。其与单目标优化问题的本质区别在于 ,它的解并非唯一 ,而是存在一组由众多 Pareto最优解组成的最优解集合 ,集合中的各个元素称为 Pareto最优解或非劣最优解。

多目标优化问题中决策变量与目标函数、约束条件是函数关系。在非劣解集中决策者只能根据具体问题要求选择令其满意的一个非劣解作为最终解。多目标优化问题的数学形式可以如下描述：

其中为决策空间，一共有个优化目标。多目标优化问题不存在唯一的全局最优解 ,过多的非劣解是无法直接应用的 ,所以在求解时就是要寻找一个最终解。求最终解主要有三类方法 :

a)生成法 ,即先求出大量的非劣解 ,构成非劣解的一个子集 ,然后按照决策者的意图找出最终解 ;

b)交互法 ,不先求出很多的非劣解 ,而是通过分析者与决策者对话的方式逐步求出最终解 ;

c)事先要求决策者提供目标之间的相对重要程度 ,算法以此为依据 ,将多目标问题转换为单目标问题进行求解。而这些主要是通过算法来实现的 ,一直以来很多专家学者采用不同算法解决多目标优化问题 ,如多目标进化算法、多目标粒子群算法和蚁群算法、模拟退火算法及人工免疫系统等。

1. **优化算法的优劣**

进化多目标优化的主要算法：

1. MOGA

Fonseca 和Fleming 在1993 年提出了MOGA。该方法对每个个体划分等级，所有非支配个体的等级定义为1，其他个体的等级为支配它的个体数目加1。具有相同等级的个体用适应度共享机制进行选择。其适应度分配方式按如下方式执行：首先，种群按照等级排序，然后，对所有个体分配适应度，方法是用Goldberg 提出的线性或非线性插值的方法来分配，具有相同等级个体的适应度值是一样的。通过适应度共享机制采用随机采样进行选择。MOGA 过于依赖共享函数的选择，而且可能产生较大的选择压力，从而导致未成熟收敛。

1. NSGA

NSGA 也是基于Goldberg 的非支配排序的思想设计的。非支配解首先被确定，然后被分配一个很大的虚拟适应度值。为了保持种群的多样性，这些非支配解用它们的虚拟适应度值进行共享。这些非支配个体暂时不予考虑。从余下的种群中确定第二批非支配个体，然后它们被分配一个比先前非支配个体共享后最小适应度值还要小的虚拟适应度值。这些非支配个体也暂时不予考虑，从余下的种群中确定第三批非支配个体。该过程一直持续到整个种群都被划分为若干等级为止。NSGA 采用比例选择来复制出新一代。NSGA 的计算复杂度为O()，其中，m 是目标个数，N是种群大小。其计算复杂度较高，而且需要预先确定共享参数。

1. NSGA-II

NSGA-II是2002 年Deb 等人对其算法NSGA 的改进，它是迄今为止最优秀的进化多目标优化算法之一，NSGA-II 具有以下优点：1)新的基于分级的快速非支配解排序方法将计算复杂度由O()降到O()，其中，m 表示目标函数的数目、N 表示种群中个体的数目。2) 为了标定快速非支配排序后同级中不同元素的适应度值，同时使当前Pareto 前沿面中的个体能够扩展到整个Pareto 前沿面，并尽可能地均匀遍布。该算法提出了拥挤距离的概念，采用拥挤距离比较算子代替NSGA 中的适值度共享方法，拥挤距离的时间复杂度为O(m(2N)log(2N))。3)引入了精英保留机制，经选择后参加繁殖的个体所产生的后代与其父代个体共同竞争来产生下一代种群，因此有利于保持优良的个体，提高种群的整体进化水平。

1. MOEA/D

对多目标优化问题的研究也更多地集中于对各种算法的研究。目前多目标优化算法归结起来有传统优化算法和智能优化算法两大类。基于分解的多目标优化算法将MOP分解为N个标量的子问题。它通过进化出一个解的种群来同时解决所有子问题。对于每一代种群，种群是从所有代中选出的每一个子问题的最优解的集合。相邻两个子问题键的关联程度是由它们的聚合系数向量问的距离所决定的。对于两个相邻子问题来说，最优解应该是非常相似的。对于每一个子问题来说，只是用与其相邻的子问题的信息来优化它。该算法具有以下特性。1) 该算法提供了一个简单有效的方法，即将分解的方法引入多目标进化计算中。对于常常在数学规划领域发展的分解方法，它可以真正并人EA中，通过使用MOEA/D框架来解决MOP问题。2)该算法的适应度分配和多样性控制的难度得到降低。因为MOEA/D算法是同时优化N标量子问题，而不是直接将MOP问题作为一个整体来解决。3) 与其他优化算法相比有一个较低的计算复杂度。总体来说，在MOGLS和MOEA/D同时解决0-1背包问题测试样例中，两者使用相同的分解方法，MOEA/D在解的质量上表现得更为出色，可以产生一组种群数量少的分布均匀的解。

1. **算法框架**

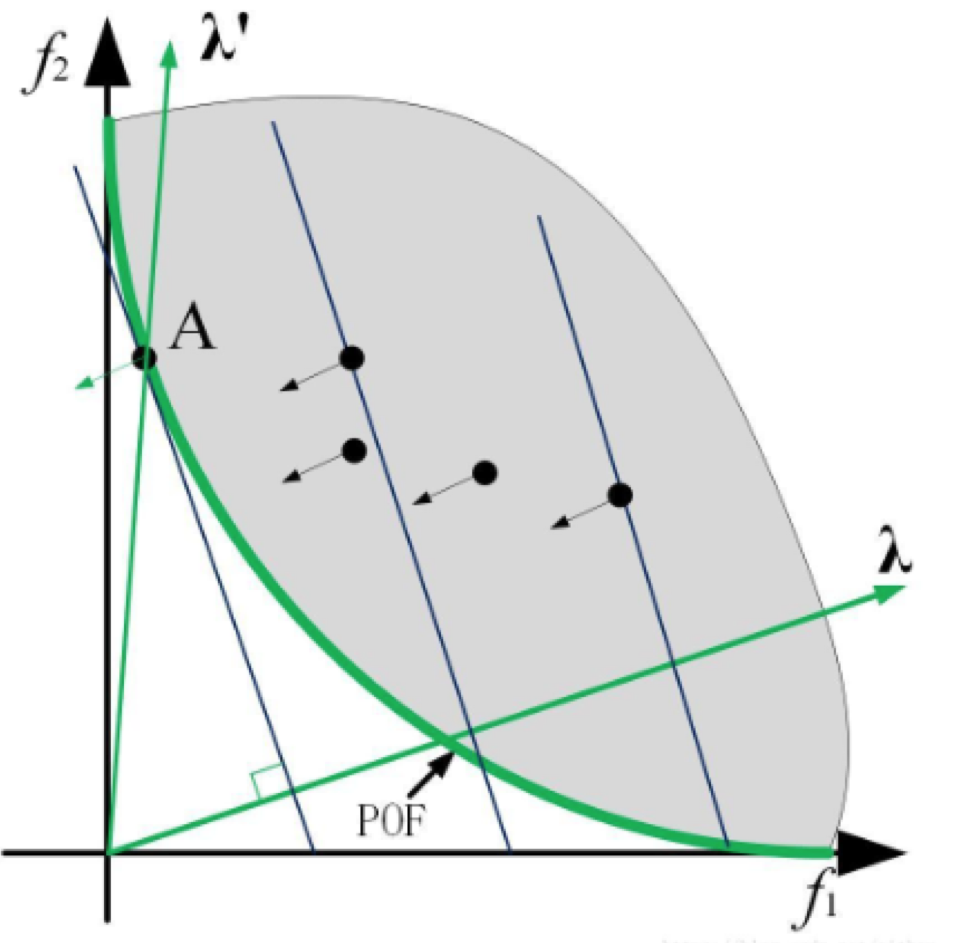
基于分解的多目标优化算法的分解方法有权重求和方法、切比雪夫聚合方法、边界交叉聚合方法。在MOEA／D中，可将这种只有更好才能取代的策略应用到所有的个体中。这种取代策略可以看做是以一种保优策略。

1. 分解问题的方法
2. 权重求和法

常用的权重求和公式为（最小化）：

其中，待优化的目标分量有个，是权重向量，每个权重分量分别对应第个目标分量；它的最优解就是MOP的一个Pareto最优解，说明了x是待优化的变量，是这个目标函数的一个系数向量。在上述标量优化问题中，如果需要一组不同的Pareto最优向量，可以使用不同的权重向量去产生他们。

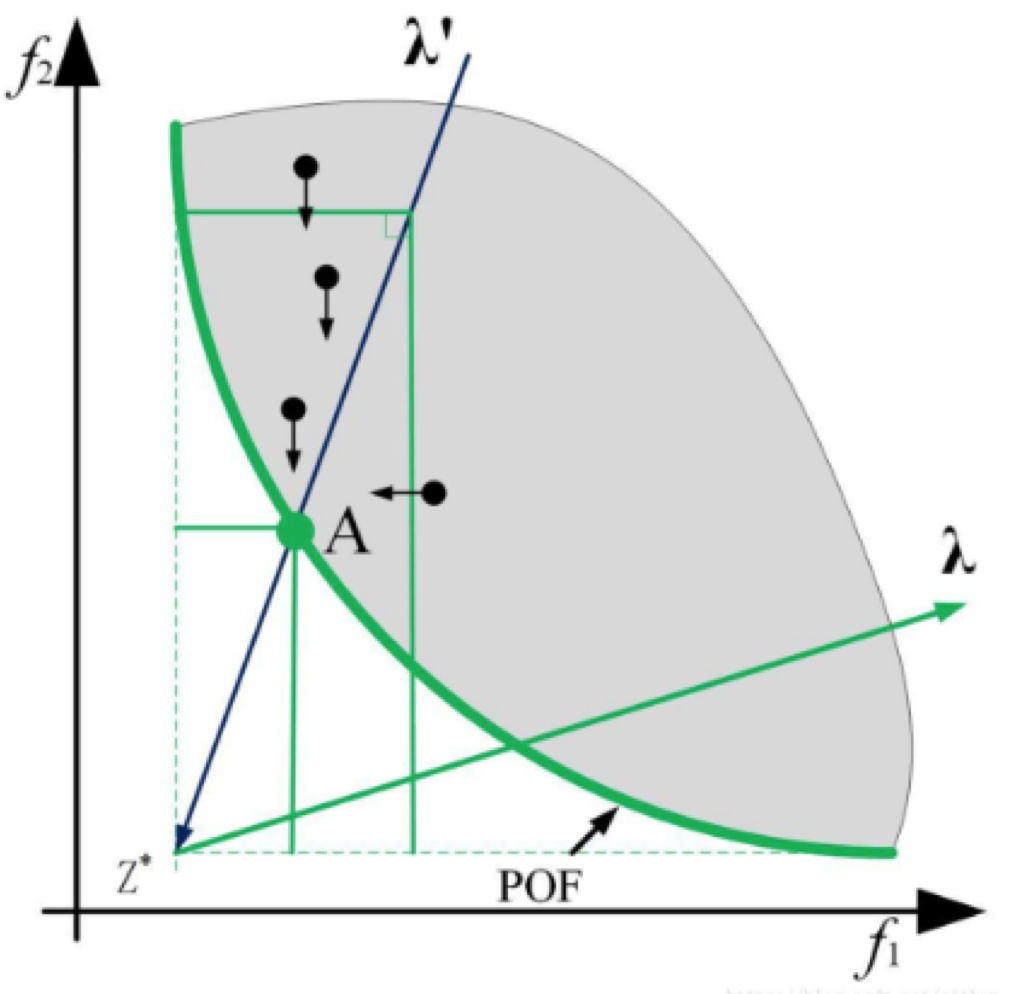
但加权求和法不能处理非凸问题，由下图可知，对于非凸问题，每个参照向量的垂线与其前沿不可能相切。



权重求和法分解子问题

1. 切比雪夫聚合法

其中，，对于每一个目标分量，，即每个目标分量最小值组成的坐标。对于每个Pareto最优解,总存在一个权重向量 ，使得上式的解是一个Pareto最优解，并且该解对于着原MOP的Pareto最优解。如果想要获得不同的Pareto最优解，可以不断地修改权重向量。

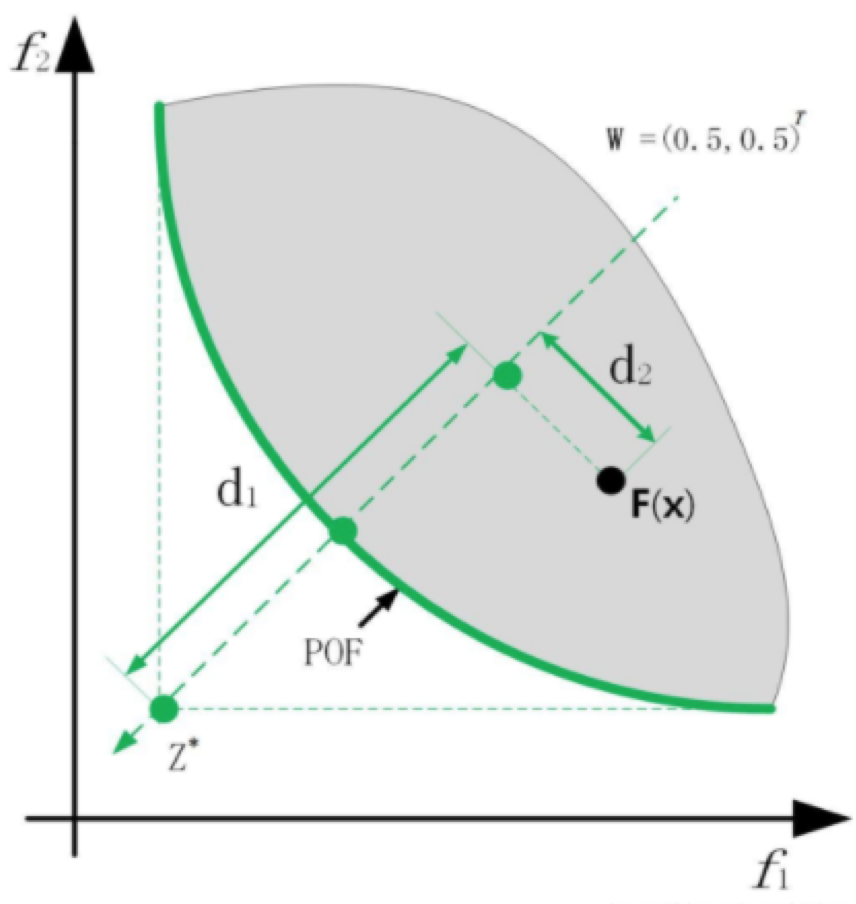


切比雪夫聚合法分解子问题

1. 边界交叉聚合法

边界交叉法旨在找到最上边界和一组线的交点。如果这些线在某种意义上均匀分布，则可以预期所得到的交叉点提供了对整个PF的良好近似。数学上，我们考虑标量优化子问题如下：

其中，,和上个方法切比雪夫中的定义一致，约束条件，需要保证与在同一直线上，这个限制条件不太现实，所以做出来改进。其中，为标量，表示在方向上的距离，越小表示越优。



边界交叉聚合法分解子问题

1. MOEA/D算法的基本流程

|  |
| --- |
| Step 1 初始化：  Step 1.1:设置。  Step 1.2:设置任意两个加权向量的欧式距离，为每个权重向量选出最近的个向量作为它的邻居。设，其中为距离最近的个权重向量。  Step 1.3:初始化种群，设。  Step 1.4:采用基于问题的特定方法初始化。  Step 2 更新：  Step 2.1:从中随机选出两个邻居使用进化算子生成新解。  Step 2.2:应用问题特定的修正或启发式的改进策略作用于生成。  Step 2.3:若，则。  Step 2.4:若，则,。  Step 2.5:将EP中所有被支配的解移出EP；若不被EP中的任意解支配，则将移入EP。  Step 3 停止判断：若满足停止准则，则算法停止，输出EP，否则返回Step 2 |

1. 关键步骤
2. 如何生成矩阵

计算权重向量之间的欧式距离，对于每个权重向量得到离它最近的个权重向量存在（相邻集合）中。注意的取值范围()，所以向量终端一定在橙线（）上，所以欧式距离越小，表示越相邻；



生成矩阵

1. 如何初始化和更新参考点：

a.初始化参考点：

其中，即每个目标分量上的最大值或最小值（视优化问题而定，这里是最小化优化）

b.更新参考点：

在交叉变异后产生新解，需要比新解较的目标各个分量上的值和参考点上各个分量上的值。视优化问题而定，如果是最小化优化的话，目标分量更小，那么相应位置的分量就更新为这个值。

1. 如何交叉变异产生新解：

a.编码方式：

若其中一个解分量x=0.5213412，先把它乘上10000然后转化为二进制作为染色体。

b.产生新解：

从邻集中随机取两个序号，根据索引找到解向量和。和随机取一个交叉位交叉产生新解。和中随机取一个解变异产生新解。然后从和取处于支配地位的解作为遗传重组算子产生的新解，如果和互不支配那就随机取一个作为新解。

1. 如何更新领域：

遍历领域中的每个权重向量，如果新解在邻居算出来的目标离基准点比子问题原来的解更近，则更新原来的解。

1. **算法性能评价指标**

对多目标进化算法的性能进行评价，需考虑两方面：1)需要有一套能够客观地反应MOEA优劣的评价工具或方法；2)需要选取一组比较有代表性的测试问题，通常选取有已知解的问题作为测试用例。

MOEA的评价主要考虑两个指标：1)MOEA的效果：指它所求得的Pareto最优解集的质量，主要是指MOEA的收敛效果和分布效果。2)MOEA的效率：指它在求取一个多目标优化问题的Pareto最优解集时所需要的CPU时间，以及它所占的空间资源。

因此，研究者们提出了许多MOEA性能评价工具或方法，可以分为三大类：1)收敛性：评价所求Pareto最优解集与真实Pareto前沿趋近程度；2)分布性：评价所求Pareto最优解集在目标空间分布是否均匀；3)综合性：是一种能够同时衡量算法收敛性和多样性的综合指标。

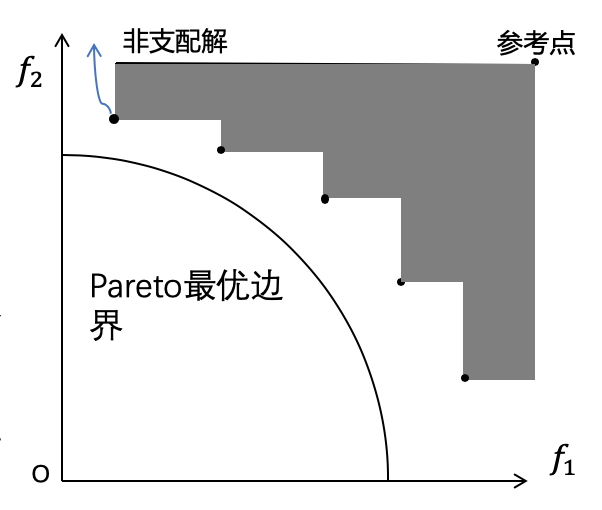
1. 世代距离（generational distance，GD）

世代距离是描述算法收敛性的一个指标值。世代距离是Coello等人在2000年提出的一种评价算法所求得的近似Pareto前沿与理想Pareto前沿间隔距离的方法。

其中是近似Pareto前沿面中的向量个数，表示算法得到的第个个体与Pareto前沿的最小欧几里得距离。值越小，说明解集离真实Pareto前言越近，趋近程度越高，解集的收敛性越好。

1. 超体积指标（hypervolume, HV）

通过计算非分配解集与参考点围成的空间的超体积的值实现对MOEA综合性能的评价。如图所示，对于2维问题来说，非支配解集与参考点构成的区域为灰色阴影部分。



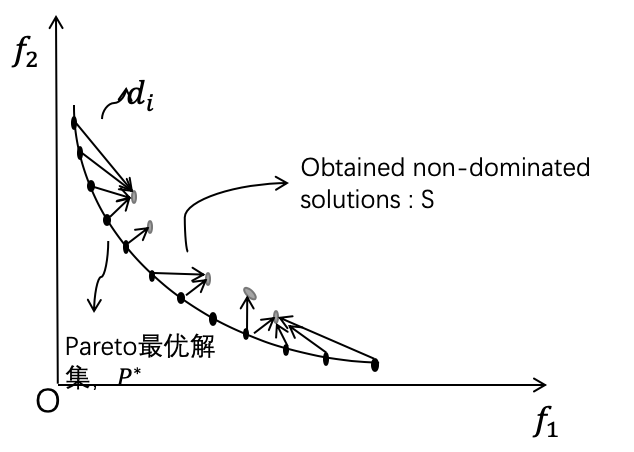
超体积指标面积

超体积指标的计算公式为：

其中，为勒贝格测度；代表参考点，一般参考点选取最低点。表示非支配解集覆盖的目标空间区域大小，HV指标的值越大，算法求得解集质量越高。超体积指标具有以下优势：1)超体积指标值是严格遵守Pareto支配原则的。若个体A支配个体B，则个体A的超体积指标一定大于个体B的超体积指标值；2)超体积指标的计算过程不需要知道Pareto最优面，所以具有很好的实用性。超体积指标的缺陷：1)超体积指标的计算时间非常大；2)参考点的选择在一定程度上决定超体积指标值得准确性。

1. 反向世代距离指标（IGD）

反世代距离是世代距离的逆向映射，它采用Pareto最优解集中的个体到算法所求得非支配解集的平均距离。



反向世代距离指标

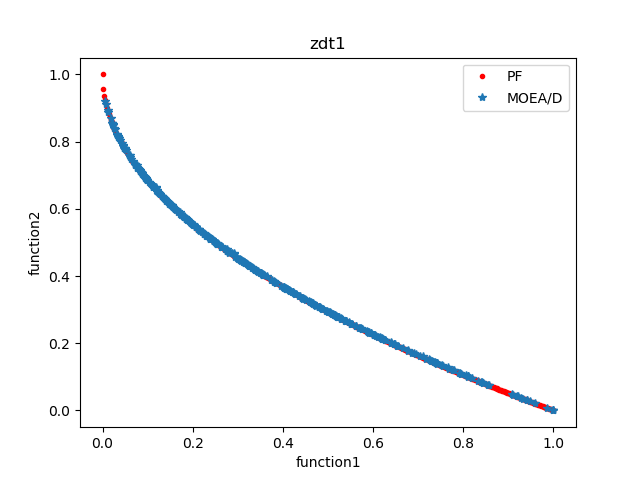
反向世代距离指标的计算公式如下：

其中，表示真实的Pareto前沿，表示算法求得的Pareto前沿，表示到的最小欧氏距离。因此，要想得到较小的值，那么算法求得的Pareto前沿必须无限逼近整个真实Pareto前沿，并且不能丢失任一部分。

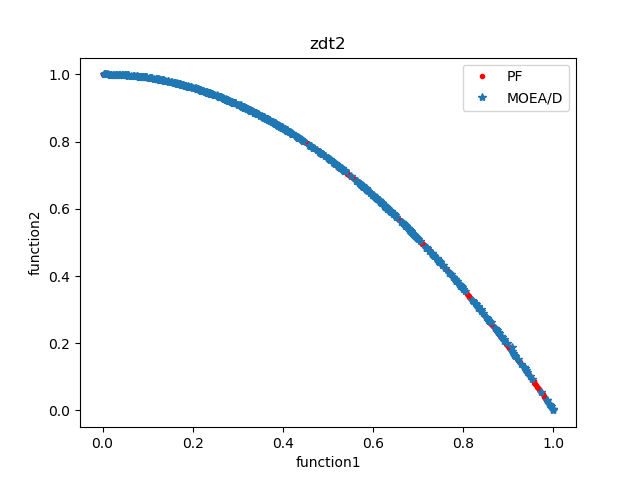
算法求得的Pareto前沿越逼近真实Pareto前沿，所得的值就越小，算法的综合性能也就越优。

1. **仿真实验与结果分析**
2. 仿真实验结果

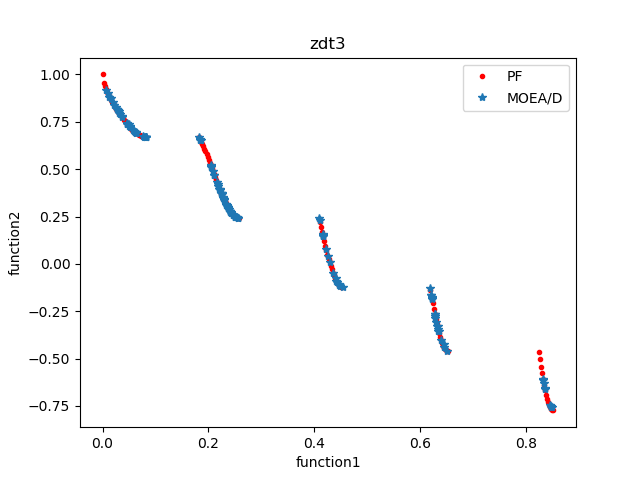
本实验一共选用4个测试函数来观察MOEA/D的性能。这四个测试函数分别为：ZDT1、ZDT2、ZDT3和ZDT4。具体的测试结果如下图所示，其中蓝线为MOEA/D算法计算得到的Pareto前沿，红线为真实Pareto前沿：



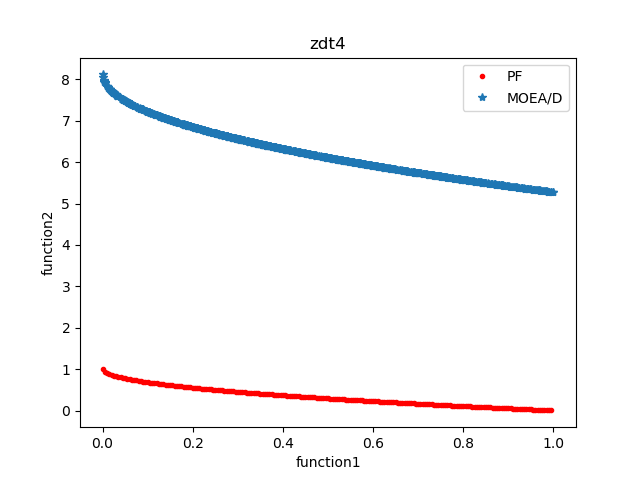
ZDT1



ZDT2



ZDT3



ZDT4

1. 实验结果分析
2. MOEA/D算法在凹函数和凸函数（ZDT1和ZDT2）上有较好的收敛性。解集分布如肉眼可见比较均匀。
3. 在ZDT3上MOEA/D算法分布不够均匀，收敛效果不是很理想。但大部分还是落到了Pareto前沿上。与原论文中描述一致。
4. 在ZDT4上MOEA/D算法陷入了局部最优解，多做几次实验还是这样，不过其它同学也有这样的情况。但原文中MOEA/D是收敛到了Pareto前沿上，可能我的实验超参数设置与论文中不一致，或者基因操作与原因不一致导致的。
5. **总结**
6. 学会了如何使用Python写MOEA/D算法
7. 学会了进化算法，多目标进化算法的基本流程，了解了多目标粒子群算法。
8. 学习了多目标优化算法的评价指标，测试函数等相关知识
9. 由于时间有限，我只实现了大体的算法流程，计算IGD，GD，HV等性能指标只做了了解。
10. 由于论文中没有写如何进行基因操作的细节，我参照了网上的做法，可能这是导致实验结果与论文略有出入的原因。