
文献阅读：
基于参考点的非支配排序方
法的多目标进化优化算法，
第一部分：解决有框约束的
问题

2022.7.28

摘 要

人们使用进化优化方法开发了多目标优化算法，主要涉及两目标和三目标问题，并在各种实际问题中证明了它们的优越性，现在越来越需要开发进化多目标优化（EMO）算法来处理多目标（有四个或更多目标）优化问题。在本文中，我们认识到最近的一些工作，并探讨了一些可行的方向，以开发一种求解多目标优化问题的潜在的 EMO 算法。此后，我们提出了一种基于参考点的多目标 NSGA-II（我们称之为 NSGA-III），它强调非支配但接近一组给定的参考点的种群个体。我们将提出的 NSGA-III 应用于许多具有 2 到 15 个目标的多目标测试问题，并与最近提出的两个版本的 EMO 算法（MOEA/D）进行比较。虽然这两种 MOEA/D 方法在不同类型的问题上都有很好的表现，但我们提出的 NSGA-III 在本研究涉及的所有问题上都产生了令人满意的结果。本文介绍的是关于无约束问题的结果，后续论文会介绍考虑约束和处理多目标优化问题的其他特点。

关键字：多目标优化；进化计算；高维度, NSGA-III；非支配性排序；多标准优化

目 录

摘 要	2
目 录	3
第一章 引言	4
第二章 多目标问题	6
2.1 处理多目标问题的困难	6
2.2 处理多目标问题的 EMO 方法	7
2.3 多目标 EMO 的两个设想	7
第三章 现有的多目标优化算法	9
第四章 提出的算法：NSGA-III	11
4.1 将种群划分为非支配性级别	11
4.2 确定超平面上的参考点	12
4.3 种群个体的自适应归一化	14
4.4 关联操作	16
4.5 小生境保留操作	18
4.6 生成子代种群的遗传操作	19
4.7 一代 NSGA-III 的计算复杂度	20
4.8 NSGA-III 的无参数特性	20
第五章 结论	22
5.1 标准化测试问题	24
5.2 经典的生成方法	36
5.3 比例测试问题	38

第一章 引言

20 世纪 90 年代以来，进化多目标优化（EMO）方法在不同的二目标和三目标优化问题中寻找一组收敛性好和多样化的非支配性解方面充分显示了其优势。然而，在大多数涉及多个利益相关者和函数的现实世界问题中，往往存在许多涉及四个或更多目标的优化问题，有时要求有 10 至 15 个目标[1],[2]。因此，处理大量的目标成为过去几年 EMO 的主要研究活动之一也就不足为奇了。多目标问题对任何优化算法都提出了许多挑战，包括 EMO。首先，随着目标数量的增加，随机选择的一组目标向量中的非支配性解的比例呈指数增加。由于非支配性解占据了大部分种群位置，任何保留精英的 EMO 都面临着在种群中容纳足够数量的新解的困难。这大大降低了搜索过程的速度[3],[4]。第二，多样性保护算子（如拥挤距离算子[5]或聚类算子[6]）的实现计算成本十分高昂。第三，高维前沿的可视化成为一项困难的任務，从而给后续的决策任务和评估算法的性能带来困难。为此，性能指标（如超体积测量[7]或其他指标[3],[8]）要么计算成本过高，要么可能没有意义。

那么要问的一个重要问题是“EMO 对多目标优化问题有用吗？”。尽管上面提到的与可视化和性能评估有关的第三个困难无法避免，但对现有的 EMO 算法进行一些算法上的改变，也许可以解决前两个问题。在本文中，我们回顾了过去在设计多目标 EMO 方面的一些努力[9],[10],[11],[12],[13],[14]，并概述了一些设计高效多目标 EMO 方法的一些可行方向。此后，我们提出了一种新的方法，该方法使用 NSGA-II 程序[5]的框架，但使用一组给定的或预定义的参考点，并证明其在求解 2 至 15 个目标的优化问题中的有效性。在本文中，我们介绍了该框架，并限制了解决各种类型的无约束问题，如具有归一化、缩放、凸、凹、脱节，以及侧重于部分帕累托最优前沿。实验可能会给定一些问题中的这样的属性。因此，针对这些可能性对算法进行充分测试仍然是一项重要的任务。我们将提出的 NSGA-III 的性能与现有的多目标 EMO（MOEA/D[10]）的两个版本进行比较，因为该方法与提出的方法有些相似。关于 MOEA/D 和 NSGA-III 这两个版本的工作，我们发现了有趣的见解。本文还评估了提出的 NSGA-III 在其他一些有趣的多目标优化和决策任务中的应用。在本文的后续部分，我们提出了 NSGA-III 算法的扩展，以处理多目标约束优化问题和一些其他特殊的、具有挑战性的多目标问题。

在本文的其余部分，我们首先讨论了求解多目标优化问题的困难，然后尝试回答上面提出的关于 EMO 算法在处理多目标时的有效性的问题。此后，在第三章中，我们对过去一些关于多目标优化的研究进行了回顾，包括最近提出的方法 MOEA/D[10]。然后，在第四章，我们详细介绍了我们提出的 NSGA-III 程序。第 5.1 节展示了使用 NSGA-III 和两个版本的 MOEA/D 对多达 15 个目标的标准化 DTLZ 测试问题的结果。接下来显示了这里建议的 DTLZ 问题的缩放版本的结果。此后的章节中，在不同类型的多目标优化问题上测试了 NSGA-III 程序。最后，NSGA-III 在第七章中被应用于两个涉及三个和九个目标的实际问题。在第八章中，我们得出了这一广泛研究的结论。

第二章 多目标问题

宽泛地说，多目标问题被定义为具有四个或更多目标的问题。两个和三个目标的问题不属于这一类，因为在大多数情况下，产生的帕累托最优前沿可以用图形方式完全地可视化。尽管对多目标优化问题的目标数量的严格上限不是很明确，但除了少数情况[15]，大多数从业者对最多 10 到 15 个目标感兴趣。在本章中，我们首先讨论了现有的 EMO 算法在处理多目标问题时可能面临的困难，并研究了 EMO 算法在处理大量目标时的有效性。

2.1 处理多目标问题的困难

在别处已经讨论过[4],[16]，在支配原则下工作的目前最先进的 EMO 算法[17]可能面临以下困难：

- 1) 种群的大部分个体是非支配性的。众所周知[3],[16]，随着目标数量的增加，随机生成的种群中非支配性个体的比例越来越大。由于大多数 EMO 算法强调种群中的非支配性解，在处理多目标问题时，在一代中没有太多空间来生成新解。这就减慢了搜索过程，因此整个 EMO 算法的效率变得很低。
- 2) 多样性指标的评估的计算成本变得昂贵：为了确定种群中解的拥挤程度，在高维空间中识别邻居的计算成本变得昂贵。为了加快计算速度，对多样性估计的任何妥协或近似都可能导致不可接受的最终解的分布。
- 3) 重组操作可能是低效的：在一个多目标问题中，如果在一个高维空间中只需要找到少数几个解，那么这些解很可能彼此相距很远。在这样的种群中，重组算子（被认为是 EMO 中的关键搜索算子）的效果变得可疑。两个相距很远的父代解可能会产生与父代解同样相距很远的子代解。因此，特殊的重组算子（交配限制或其他方案）可能是有效处理多目标问题所必需的。
- 4) 权衡面的表示是困难的：显然，要表示一个较高维度的权衡面，需要指数级增加的更多点。因此，需要一个很大的种群规模来表示所产生的帕累托最优前沿。这使得决策者难以理解并作出适当的决定来选出一个优选解。
- 5) 性能指标的计算成本很高：因为要将较高维度的点集相互比较来确定一种算法对另一种算法的性能，因此需要较大的计算量。例如，计算超体积指标需要随着目

标数量的增加而呈指数增加的计算[18], [19]。

- 6) 可视化是困难的：最后，虽然这不是一个与优化直接相关的问题，但最终对多目标问题来说，高维的权衡前沿的可视化可能是困难的。

前三个困难只能通过对现有 EMO 方法的某些修改来缓解。第四、五、六个困难是所有多目标优化问题所共有的，我们在此不做讨论。

2.2 处理多目标问题的 EMO 方法

在我们讨论上述三个困难的可能的补救措施之前，在此强调两个不同的多目标问题类别，现有的 EMO 方法仍然适用于这些问题。

首先，现有的 EMO 算法在从完整的帕累托最优集合中寻找一个优选解子集（部分集）时仍然是有用的。尽管优选子集仍将是多维的，因为目标解集中在帕累托最优前沿的一个小区域，因此上述大多数困难将通过这一原则得到缓解。一些基于 MCDM 的 EMO 方法已经被设计出来，结果显示在多达 10 个目标的问题上的表现良好[20], [21], [22], [23]。

其次，实践中的许多问题，尽管有许多目标，但通常退化到低维的帕累托最优前沿[4], [11], [24], [25]。在这样的问題中，冗余目标的识别可以与 EMO 结合起来，找到低维的帕累托最优前沿。由于最终的前沿低至二维或三维，现有的 EMO 方法应该可以很好地处理这类问题。以前对 NSGA-II 和基于主成分分析（PCA）程序的研究[4]能够解决多达 50 个目标的问题，这些问题具有两个目标的帕累托最优前沿。

2.3 多目标 EMO 的两个设想

考虑到与基于支配的 EMO 程序相关的前三个困难，可以考虑采用两种不同的策略来缓解这些困难：

- 1) 使用特殊的支配原则：上面提到的第一个困难可以通过使用特殊的支配原则来缓解，该原则将自适应地离散帕累托最优前沿并找到一组分布良好的点。例如，使用 ϵ 支配原则[26], [27]将使所有与一组帕累托最优点距离在 ϵ 以内的点都被支配，因此该过程将产生有限数量的帕累托最优点作为目标。这样的考虑也将缓解多样性保护的第二个困难。第三个困难可以通过使用交配限制方案或强调近亲解的特殊重组操作来解决（如具有大分布指数的 SBX[28]）。其他特殊的支配原则[29],

[30]也可用于此目的。Aguirre 和 Tanaka[31]以及 Sato 等人[32]建议使用一个目标子集进行支配性检查，并在每一代中使用不同的目标组合。也可以尝试使用固定锥支配[33],[34]或可变锥支配[35]原则。这些研究是在低维问题的背景下进行的，它们在求解多目标优化方面的成功还有待证实。

- 2) 使用预定义的多目标搜索：越来越清楚的是，在一个高维问题中，期望从一个基于种群的优化算法中获得其种群在帕累托最优前沿附近的收敛，同时使其在大的种群规模问题中均匀分布在整個前沿周围，这太过分了。处理这种多目标优化问题的一个方法是通过一些外部手段来帮助多样性维护。这一原则可以直接解决上面提到的第二个困难。与其在整个搜索空间中搜索帕累托最优解，不如由一个算法设定多个预定义的目标搜索。由于找到了对应于每个目标搜索任务的最优点，所以处理大的非支配集合的第一个困难也得到了缓解。重组问题可以通过使用交配限制方案来解决，其中来自相邻目标的两个解参与了重组操作。我们提出的算法（NSGA-III）就是基于这一原则，因此我们对这方面进行了较为详细的讨论。我们提出了两种不同的方法来实现预定义的多目标搜索原则：

- a) 可以事先指定一组横跨整个帕累托最优前沿的预定义搜索方向，并且可以沿着每个方向进行多次搜索。由于搜索方向分布广泛，在大多数问题中，获得的最优点也可能广泛分布在帕累托最优前沿上。最近提出的 MOEA/D 过程[10]使用了这个概念。
- b) 为此，可以指定多个预定义的参考点，而不是多个搜索方向。此后，可以强调对应于每个参考点的点，以找到一组广泛分布的帕累托最优点集。最近提出了一些这样的实现方法[36],[37],[38],[14]，本文提出了另一种方法，对第一篇参考文献[36]中提出的算法进行了扩展和完善。

第三章 现有的多目标优化算法

Garza-Fabre, Pulido 和 Coello [16]通过使用两个相互竞争亲本之间的个体目标值的差异，提出了三个单目标测量方法，并表明在 5 到 50 个目标的 DTLZ1, DTLZ3 和 DTLZ6 问题中，与包括一般的基于帕累托优势的 EMO 方法在内的许多现有方法相比，收敛性得到增强。Purshouse 和 Fleming[39]明确地表明，多样性保护和在帕累托最优前沿附近实现收敛是两个相互矛盾的目标，通常的遗传算子不足以同时实现这两个目标，特别是对于多目标问题。另一项研究[40]通过增加多样性控制算子来扩展 NSGA-II，以解决 6 到 20 个目标的 DTLZ2 问题。Koppen 和 Yoshida [41]声称 NSGA-II过程不适合多目标优化问题，并提出了一些有可能取代 NSGA-II的拥挤距离算子以获得更好的性能指标。基于对 2 到 15 个目标的 DTLZ2、DTLZ3 和 DTLZ6 问题的仿真研究，他们建议使用替代分配距离指标作为最优策略。Hadka 和 Reed[42]提出了一个基于集合的 EMO 过程，该过程使用一个合适的重组算子，根据它们在问题中的世代成功率，从一组 8 到 10 个不同的预定义算子中自适应选择。它还使用了 ϵ 优势概念和自适应种群规模的方法，据说可以成功解决多达 8 个目标的测试问题。Bader 和 Zitzler[43]提出了一个计算基于样本的超体积的快速程序，并设计了一个算法来寻找一组权衡解，以实现超体积的最大化。越来越多的关于近似超体积计算的文献[44], [18], [45]可能使这种方法在解决多目标问题时变得实用。

上述研究分析并扩展了先前提出的进化多目标优化算法，以确定其是否适合求解多目标问题。在大多数情况下，结果是乐观的，提出的算法必须在其他比标准化测试问题（如 DTLZ 问题）更具挑战性的问题上进行测试。它们也必须在现实世界的问题中被测试。在下面的段落中，我们描述了一个最近提出的算法，它与我们在第 2.3 节中给出的多目标优化算法的描述十分吻合，并且与我们提出的算法非常匹配。

MOEA/D[10]使用一组预定义的权重向量来保持一组多样化的权衡解。对于每个权重向量，产生的问题被称为子问题。首先，每个种群个体（其大小与权重向量的数量相同）与一个权重向量相关联。此后，来自相邻权重向量的两个解（通过小生境参数（ T ）定义）被交配，并产生一个子代解。然后根据性能指标将子

代与一个或多个权重向量相关联。研究中提出了两个衡量指标。一个点与理想点的惩罚距离指标是由与参考方向的垂直距离（ d_2 ）和沿参考方向的距离（ d_1 ）的加权和（权重 θ 是另一个算法参数）形成的：

$$\text{PBI}(\mathbf{x}, \mathbf{w}) = d_1 + \theta d_2 \quad (3-1)$$

我们在此称这个过程为 MOEA/D-PBI。提出的第二种方法是使用 Tchebycheff 指标，使用一个理想点 \mathbf{z}^* 和权重向量 \mathbf{w} ：

$$\text{TCH}(\mathbf{x}, \mathbf{w}, \mathbf{z}^*) = \max_{i=1}^M w_i |f_i(\mathbf{x}) - z_i^*| \quad (3-2)$$

在报告的仿真中[10]，理想点用 \mathbf{z}^* 来表示，零权重通过使用一个很小的数来处理。我们称这个过程为 MOEA/D-TCH。一个外部种群保持着非支配性解。解决前面提到的前两个困难是通过使用一组明确的权重向量来寻找点的，第三个困难是通过使用交配限制方案来缓解的。仿真结果只显示了两个和三个目标的测试问题，结论是 MOEA/D-PBI 在三个目标问题上比 MOEA/D-TCH 更好，而且 MOEA/D-TCH 的性能在使用种群最小和最大目标值的目标归一化过程中得到改善。MOEA/D 的两个版本都需要设置一个小生境参数（ T ）。基于对二目标和三目标问题的一些仿真结果，作者建议使用大部分种群规模作为 T 。此外，MOEA/D-PBI 需要适当设置一个额外的参数——惩罚参数 θ ，作者建议其值为 5。

MOEA/D 的开发人员在后来的研究中建议使用差分进化（DE）来取代重组算子和变异算子。此外，在定义特定解的邻域和用相应的子代解替换给定邻域中的子代解方面也做了进一步的改进[46]。我们在此称这种方法为 MOEA/DE。在一组主要是两个和三个目标的关联问题上的结果[47]显示，与其他算法相比，MOEA/DE 的性能更好。如前所述，MOEA/D 是一种很有前途的多目标优化方法，因为它很好地解决了上面提到的一些困难，但是上述的 MOEA/D 研究并没有很好地探索它们对大量目标的适用性。在本文中，我们将它们应用于多达 15 个目标的问题，并评估它们对真正的多目标优化问题的适用性，揭示这些算法的有趣特性。

第四章 提出的算法：NSGA-III

我们提出的多目标 NSGA-II (或 NSGA-III) 的基本框架与最初的 NSGA-II 算法[5]类似，但在选择机制上有重大变化。与 NSGA-II 不同的是，NSGA-III 通过提供和自适应更新一些分布良好的参考点，来维持种群个体之间的多样性。为了完整起见，我们首先对原始的 NSGA-II 算法进行简要的描述。

考虑 NSGA-II 算法的第 t 代。假设这一代的亲本种群为 P_t ，其种群规模为 N ，而由亲本 P_t 产生的子代种群是有 N 个个体的 Q_t 。第一步是从父代和子代组合种群 $R_t = P_t \cup Q_t$ (种群规模为 $2N$) 中选出最好的 N 个个体，从而可以保留父代种群中的精英个体。为了实现这一目标，首先，根据不同的非支配水平 (F_1 、 F_2 等) 对组合种群 R_t 进行排序。然后，从 F_1 开始，每次选择一个非支配水平来生成一个新的种群 S_l ，直到 S_l 的规模等于 N 或首次超过 N 。假设包含的最后一个级别是第 l 级。因此，从 $(l+1)$ 级开始的所有解都被拒绝纳入组合种群 R_t 中。在大多数情况下，最后接受的级别 (第 l 级) 也只是部分接受。在这种情况下，只选择那些能使第 l 级的多样性最大化的解。在 NSGA-II 中，这是通过一个计算效率高但近似的小生境保护算子来实现的，该算子将每个最后一级个体的拥挤距离计算为两个相邻解之间的客观归一化距离之和。此后，选择具有较大拥挤距离值的解。这里，我们用下面的方法取代拥挤距离算子 (第 4.1 至 4.5 节)。

4.1 将种群划分为非支配性级别

上述利用一般的支配原则[17]识别非支配前沿的程序也被用于 NSGA-III。所有从非支配前沿第 1 级到第 l 级的种群个体首先被纳入 S_l 。若 $|S_l| = N$ ，则无须进一步操作，下一代从 $P_{t+1} = S_l$ 开始。若 $|S_l| > N$ ，已经选择了从第 1 级到第 $(l-1)$ 级前沿的个体，即 $P_{t+1} = \cup_{i=1}^{l-1}$ ，剩下的 ($K = N - |P_{t+1}|$) 个种群个体从最后一个前沿 F_l 中选择。我们在下面的几个小节中描述其余的选择过程。

4.2 确定超平面上的参考点

如前所述，NSGA-III使用一组预定义的参考点来确保获得的解的多样性。所选择的参考点既可以以结构化的方式预定义，也可以由用户给定。我们将在后面的结果部分介绍这两种方法的结果。在没有任何偏好的情况下，可以采用任何预定义的结构化的参考点放置，但在本文中，我们使用 Das 和 Dennis [48] 的系统方法 1，将点放置在一个归一化的超平面上——一个 $(M-1)$ 维的单位单形面——它与所有目标轴具有都相同的倾斜度，在每个轴上的截距都为 1。如果沿每个目标划分为 p 部分，那么在 M 个目标问题中，参考点 (H) 的总数由以下几点给出：

$$H = \binom{M+p-1}{p} \quad (4-1)$$

例如，在一个三目标问题 ($M=3$) 中，参考点创建在顶点在 $(1,0,0)$ 、 $(0,1,0)$ 和 $(0,0,1)$ 的三角形上。如果为每个目标轴选择四个分区 ($p=4$)，将创建 $H = \binom{3+4-1}{4} = 15$ 个参考点。为清楚起见，这些参考点如图 4-1 所示。在提出的 NSGA-III 中，除了强调非支配解外，我们还强调在某种意义上与这些参考点相关联的种群个体。由于上述创建的参考点广泛分布在整个归一化超平面上，因此获得的解也可能广泛分布在帕累托最优前沿或附近。在用户给定一组优选参考点的情况下，理想情况下，用户可以在归一化超平面上标记 H 个点，或者为此指出任何 H, M 维的向量。所提出的算法有可能找到与给定的参考点对应的近似帕累托最优解，从而使这种方法可以更多地从决策和多目标优化的综合应用角度来使用。该程序在算法 1 中给出。

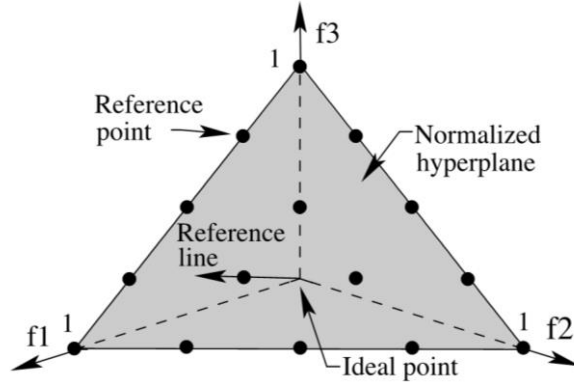


图 4-1 在一个归一化的参考平面上显示 15 个参考点，适用于 $p=4$ 的三目标问题

算法 1 NSGA-III 程序的第 t 代

输入： H 结构化参考点 Z^s 或给定的参考点 Z^a ，亲本种群 P_t

输出： P_{t+1}

- 1: $S_t = \Phi, i = 1$
- 2: $Q_t = \text{重组+变异}(P_t)$
- 3: $R_t = P_t \cup Q_t$
- 4: $(F_1, F_2, \dots) = \text{非支配性排序}(R_t)$
- 5: 重复
- 6: $S_t = S_t \cup F_i$ 且 $i = i + 1$
- 7: **until** $|S_t| \geq N$
- 8: 最后一个前沿被纳入: $F_t = F_i$
- 9: **if** $|S_t| = N$, **then**
- 10: $P_{t+1} = S_t$, **break**
- 11: **else**
- 12: $P_{t+1} = \bigcup_{j=1}^{l-1} F_j$
- 13: 将从 F_l 中选择 $K = N - |P_{t+1}|$ 个

14: 对目标进行归一化，并建立参考集 Z^r :

$$Normalize(f^n, S_i, Z^r, Z^s, Z^a)$$

15: 将 S_i 的每个个体 s 与一个参考点相关联:

$$[\pi(s), d(s)] = Associate(S_i, Z^r) \quad \% \pi(s): \text{最接近的}$$

16: 计算参考点 $j \in Z^r$ 的小生境数:

$$\rho_j = \sum_{s \in S_i / F_i} ((\pi(s) = j) ? 1 : 0)$$

17: 一次性从 F_i 中选择 K 个个体来构建

$$P_{t+1} : Niching(K, \rho_j, \pi, d, Z^r, F_i, P_{t+1})$$

18: end if

4.3 种群个体的自适应归一化

首先，对于 $\cup_{\tau=0}^t S_\tau$ 中的每个目标函数，通过确定最小值 (z_i^{\min}) 来确定种群 S_i 的理想点，并构建理想点 $\bar{z} = (z_1^{\min}, z_2^{\min}, \dots, z_M^{\min})$ $i=1, 2, \dots, M$ 。然后， S_i 的每个目标值通过用 z_i^{\min} 减去目标 f_i 来移动，因此移动后的 S_i 的理想点成为一个零向量。我们把这个移动后的目标表示为 $f'_i(\mathbf{x}) = f_i(\mathbf{x}) - z_i^{\min}$ 。此后，通过寻找使以下标量化函数最小的解 ($\mathbf{x} \in S_i$) 来确定每个目标轴上的极值点，权重向量 \mathbf{w} 为轴方向:

$$ASF(\mathbf{x}, \mathbf{w}) = \max_{i=1}^M f'_i(\mathbf{x}) / w_i, \quad \text{for } \mathbf{x} \in S_i \quad (4-2)$$

对于 $w_i = 0$ ，我们用一个很小的数字 10^{-6} 来代替它。对于第 i 个移动的目标方向 f'_i ，将产生一个极值的目标向量 $z^{i, \max}$ 。然后这些 M 个极值向量被用来构建一个 M 维的线性超平面。然后可以计算出第 i 个目标轴和线性超平面的截距 a_i (见图 4-2)，进行目标函数的归一化，如下:

$$f_i^n(\mathbf{x}) = \frac{f'_i(\mathbf{x})}{a_i - z_i^{\min}} = \frac{f_i(\mathbf{x}) - z_i^{\min}}{a_i - z_i^{\min}}, \quad \text{for } i = 1, 2, \dots, M \quad (4-3)$$

注意，每个归一化目标轴上的截点现在都在 $f_i^n = 1$ ，用这些截距点构建的超平面将使 $\sum_{i=1}^M f_i^n = 1$ 。

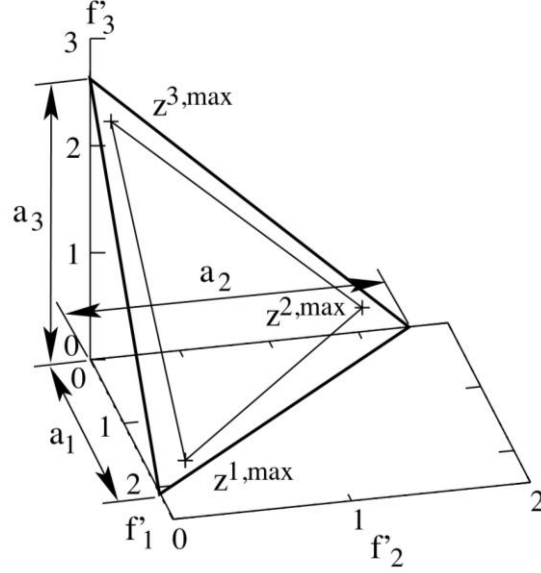


图 4-2 计算截距，然后在一个三目标问题上显示从极值点形成超平面的过程

在结构化参考点（其中有 H 个）的情况下，用 Das 和 Dennis[48]的方法计算的原始参考点已经位于这个归一化超平面上。在用户偏好参考点的情况下，使用公式 4-3 将参考点映射到上面构建的归一化超平面上。由于归一化过程和超平面的创建是在每一代使用仿真开始时发现的极值点来完成的，因此所提出的 NSGA-III 程序在每一代都能自适应地保持 S_t 个体所跨越空间的多样性。这使得 NSGA-III 能够解决具有帕累托最优前沿的问题，其目标值可能会有不同的比例。该程序也在算法 2 中进行了描述。

算法 2 $Associate(S_t, Z')$ 过程

输入： Z' , S_t

输出： f^n , Z' (归一化超平面上的参考点)

1: **for** $j=1$ **to** M **do**

2: 计算理想点: $z_j^{\min} = \min_{s \in S_t} f_j(s)$

3: 移动的目标: $f'_j(s) = f_j(s) - z_j^{\min} \quad \forall s \in S_t$

4: 计算极值点: $z^{j, \max} = s$:

$\arg \min_{s \in S_t} ASF(s, w_j)$, 其中 $w^j = (\epsilon, \dots, \epsilon)^T$

$\epsilon = 10^{-6}$, 且 $w_j^j = 1$

5: **end if**

6: 计算截距 $a_j, j=1, \dots, M$

7: 使用公式 4-3 将目标 (f^n) 归一化

8: **if** Z^a 给定, **then**

9: 使用公式 4-3 将每个 (aspiration) 点映射到归一化的超平面上,
 并将这些点保存在集合 Z' 中。

10: **else**

11: $Z' = Z^s$

12: **end if**

4.4 关联操作

在根据目标空间中 S_t 个体的范围自适应地对每个目标进行归一化后, 接下来我们需要将每个种群个体与一个参考点相关联。为此, 我们通过连接参考点和原点, 在超平面上定义一条对应于每个参考点的参考线。然后, 我们计算 S_t 的每个种群个体与每条参考线的垂直距离。在归一化目标空间中, 参考线最接近种群个

体的参考点被认为与该种群个体相关联。如图 4-3 所示。该过程在算法 3 中给出。

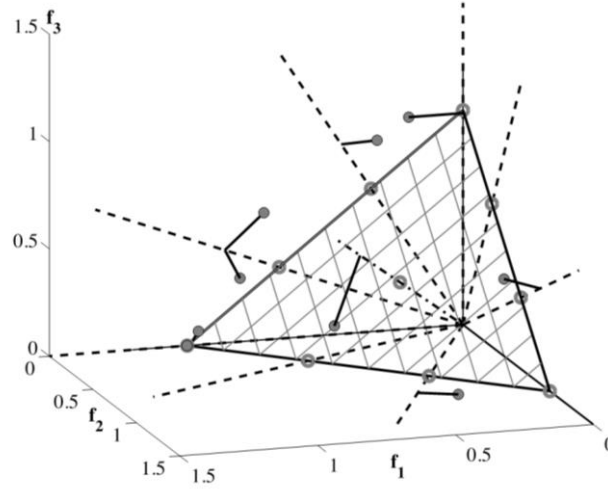


图 4-3 图示种群个体与参考点的联系

算法 3 $Associate(S_t, Z^r)$ 过程

输入: Z^r, S_t

输出: $\pi(s \in S_t), d(s \in S_t)$

- 1: **for** 每个参考点 $z \in Z^r$ **do**
 - 2: 计算参考线 $w = z$
 - 3: **end if**
 - 4: **for** 每个 $s \in S_t$ **do**
 - 5: **for** 每个 $w \in Z^r$ **do**
 - 6: 计算 $d^\perp(s, w) = \|s - w^T s / \|w\|\|$
 - 7: **end for**
 - 8: Assign $\pi(s) = w : \operatorname{argmin}_{w \in Z^r} d^\perp(s, w)$
 - 9: Assign $d(s) = d^\perp(s, \pi(s))$
 - 10: **end for**
-

4.5 小生境保留操作

值得注意的是，一个参考点可以有一个或多个种群个体与之相关联，或者没有一个种群个体与之相关联。我们计算 $P_{t+1} = S_t / F_t$ 中与每个参考点相关的种群个体的数量。让我们把第 j 个参考点的小生境计数表示为 ρ_j 。现在我们设计一个新的小生境操作如下。首先，我们确定具有最小 ρ_j 的参考点集 $J_{min} = \{j : \argmin_j \rho_j\}$ 。在有多于一个这样的参考点的情况下，随机选择一个 ($\bar{j} \in J_{min}$)。

如果 $\bar{\rho}_j = 0$ (意味着没有与参考点 \bar{j} 相关联的 P_{t+1} 个体)，那么在集合 F_t 中的 \bar{j} 可能有两种情况。第一， F_t 中存在一个或多个已经与参考点 \bar{j} 相关联的个体。在这种情况下，与参考线垂直距离最短的那个个体被添加到 P_{t+1} 。然后， $\bar{\rho}_j$ 的计数增加 1。第二，前面的 F_t 没有任何个体与参考点 \bar{j} 相关联。在这种情况下，参考点就会被排除在本代的进一步考虑之外。

在 $\bar{\rho}_j \geq 1$ 的情况下 (意味着 S_t / F_t 中已经有一个与参考点相关联的个体存在)，从前面 F_t 中随机选择一个与参考点 \bar{j} 相关联的个体 (如果存在的话) 添加到 P_{t+1} 。然后， $\bar{\rho}_j$ 的计数增加 1。小生境计数更新后，该程序共重复 K 次，以补充 P_{t+1} 的所有空的人口位置。该过程在算法 4 中给出。

算法 4 *Niching* ($K, \rho_j, \pi, d, Z^r, F_l, P_{t+1}$) 过程

输入: $K, \pi(s \in S_t), d(s \in S_t), Z^r, F_l$

输出: $P_{t+1}, d(s \in S_t)$

```

1:  $k = 1$ 
2: while  $k \leq K$  do
3:    $J_{\min} = \{j : \operatorname{argmin}_{j \in Z^r} \rho_j\}$ 
4:    $\bar{j} = \operatorname{random}(J_{\min})$ 
5:    $I_{\bar{j}} = \{s : \pi(s) = \bar{j}, s \in F_l\}$ 
6:   if  $I_{\bar{j}} \neq \Phi$  then
7:     if  $\rho_{\bar{j}} = 0$  then
8:        $P_{t+1} = P_{t+1} \cup \{s : \operatorname{argmin}_{s \in I_{\bar{j}}} d(s)\}$ 
9:     else
10:       $P_{t+1} = P_{t+1} \cup \operatorname{rand}(I_{\bar{j}})$ 
11:    end if
12:     $\rho_{\bar{j}} = \rho_{\bar{j}} + 1, F_l = F_l \setminus s$ 
13:     $k = k + 1$ 
14:  else
15:     $Z^r = Z^r \setminus \{\bar{j}\}$ 
16:  end if
17: end while

```

4.6 生成子代种群的遗传操作

生成 P_{t+1} 后，再通过应用一般的遗传算子来生成一个新的子代种群 Q_{t+1} 。在 NSGA-III 中，我们已经对解进行了仔细的精英选择，并试图通过强调最接近每个参考点的参考线的解来保持解的多样性。此外，正如我们将在第五章中描述的那

样，对于计算快速的程序，我们将 N 设定为约等于 H ，从而给予每个种群个体同等的重要性。由于所有这些原因，我们在 NSGA-III 中不采用任何明确的选择操作。种群 Q_{t+1} 是通过从 P_{t+1} 中随机挑选亲本，应用通常的交叉和变异操作来生成的。然而，为了使子代解更接近父代解（考虑到第 2.1 节中提到的第三个困难），我们建议在 SBX 算子中使用一个相对较大的分布指数值。

4.7 一代 NSGA-III 的计算复杂度

对具有 M 维目标向量的种群规模为 $2N$ 的种群进行非支配排序（算法 1 中的第 4 行）需要 $O(N \log^{M-2} N)$ 次计算[49]。算法 2 第 2 行中的理想点的识别总共需要 $O(MN)$ 次计算。目标的移动（第 3 行）需要 $O(MN)$ 次计算。然而，极值点的识别（第 4 行）需要 $O(M^2 N)$ 次计算。确定截距（第 6 行）需要一个大小为 $M \times M$ 的矩阵求逆，需要 $O(M^3)$ 次运算。此后，对最多 $2N$ 个种群个体的归一化（第 7 行）需要 $O(N)$ 次计算。算法 2 的第 8 行需要 $O(MH)$ 次计算。算法 3 中把最多 $2N$ 个种群个体与 H 个参考点相关联的所有操作需要 $O(MNH)$ 次计算。此后，在算法 4 的小生境过程中，第 3 行将需要 $O(H)$ 次比较。假设 $L = |F_t|$ ，第 5 行需要 $O(L)$ 次检查。最坏情况下的第 8 行需要 $O(L)$ 次计算。其他操作的复杂度较小。然而，在，小生境算法中的上述计算最多需要执行 L 次，从而需要较大的 $O(L^2)$ 或 $O(LH)$ 次计算。在最坏的情况下（ $S_t = F_1$ ，即第一个非支配阵线超过种群规模）， $L \leq 2N$ 。在我们所有的仿真中，我们使用了 $N \approx H$ 和 $N > M$ 。考虑到上述所有的考虑和计算，NSGA-III 一代的总体最坏情况下的复杂度为 $O(N^2 \log^{M-2} N)$ 或 $O(N^2 M)$ ，取较大值。

4.8 NSGA-III 的无参数特性

与 NSGA-II 一样，NSGA-III 算法除了通常的遗传参数如种群规模、终止参数、

交叉和变异概率及其相关参数外，不需要设置任何新的参数。参考点的数量 H 不是一个算法参数，因为这完全由用户来决定。种群规模 N 取决于 H ，因为 $N \approx H$ 。参考点的位置同样取决于用户对所获得的解中的偏好信息。

现在我们将介绍 NSGA-III对各种多目标情景的仿真结果，并将其性能与 MOEA/D 和经典方法进行比较。

第五章 结论

在本章中，我们给出了 NSGA-III 在 3 到 15 个目标的优化问题上的仿真结果。由于我们的方法有一个与 MOEA/D 类似的框架，即这两类算法都需要一组用户给定的参考点或权重向量，我们将我们提出的方法与不同版本的 MOEA/D 进行了比较（使用了 MOEA/D 网站[50]的代码）。最初的 MOEA/D 研究给出了两个程序（MOEA/D-PBI 和 MOEA/D-TCH），但没有解决四个或更多目标的问题。在这里，我们研究了这些 MOEA/D 算法与与我们提出的算法在 3 到 15 个目标问题上的性能。

作为性能指标，我们选择了逆向世代距离（IGD）指标[51], [47]作为一个单一指标，它可以提供关于所获得的解的收敛性和多样性的综合信息。由于 NSGA-III 和 MOEA/D 算法中分别给定了参考点或参考方向，并且在本章中我们展示了这些方法在测试问题上的工作，对于这些问题，确切的帕累托最优表面是已知的，我们可以在归一化目标空间中准确定位目标帕累托最优点。计算这些目标点，并称其为集合 Z 。对于任何算法，得到目标空间中最终的非支配点，并称其为集合 A 。现在，计算 IGD 指标，作为集合 Z 中的点与集合 A 中所有点的最近个体的平均欧氏距离：

$$IGD(A, Z) = \frac{1}{|Z|} \sum_{i=1}^{|Z|} \min_{j=1}^{|A|} d(z_i, a_j) \quad (5-1)$$

其中 $d(z_i, a_j) = \|z_i - a_j\|_2$ 。IGD 值越小的集合越好。如果没有找到与参考点相关联的解，则该集合的 IGD 值会很大。对于每种情况，从不同的初始种群中进行 20 次不同的运行，并报告最优、中位和最差的 IGD 性能值。对于所有的算法，呈现最后一代的种群个体并用于计算性能指标。

表 5-1 显示了不同种群规模的问题所选择的参考点（ H ）的数量。NSGA-III 的种群规模 N 被设定为比 H 大 4 倍的最小倍数。对于 MOEA/D 程序，我们使用其开发者建议的种群规模， $N' = H$ 。对于三目标问题，我们使用 $p = 12$ ，以获得

$H = \binom{3-1+12}{12}$ 或 91 个参考点（参考公式 4-1）。对于五目标问题，我们使用 $p = 6$ ，

这样就得到了 $H = 210$ 个参考点。请注意，只要不选择 $p \geq M$ ，Das 和 Dennis 的系统方法就不会产生中间点。对于八目标问题，即使我们使用 $p = 8$ （为了至少有一个中间参考点），也需要 5040 个参考点。为了避免这种情况，我们使用 p 值较小的两层参考点。在边界层使用 $p = 3$ ，这样就会生成 120 个点。在内部层使用 $p = 2$ ，这样就会生成 36 个点。然后，所有 $H = 156$ 个点都被用作八目标问题的参考点。我们在图 5-1 中说明了这种情况，对于一个三目标问题，在边界层使用 $p = 2$ ，在内部层使用 $p = 1$ 。对于 10 个目标的问题也是如此，我们对边界层和内部层分别使用 $p = 3$ 和 $p = 2$ ，因此总共需要 $H = 220 + 55$ 或 275 个参考点。同样，对于 15 个目标的问题，我们对边界层和内部层分别使用 $p = 2$ 和 $p = 1$ ，因此需要 $H = 120 + 15$ 或 135 个参考点。

表 5-1 在 NSGA-III 和 MOEA/D 算法中使用的参考点/方向的数量和相应的种群规模

目标数量 (M)	所选择的参考点的	NSGA-III 的种群规	MOEA/D 的种群规
	数量 (H)	模 (N)	模 (N')
3	91	92	91
5	210	212	210
8	156	156	156
10	275	276	275
15	135	136	135

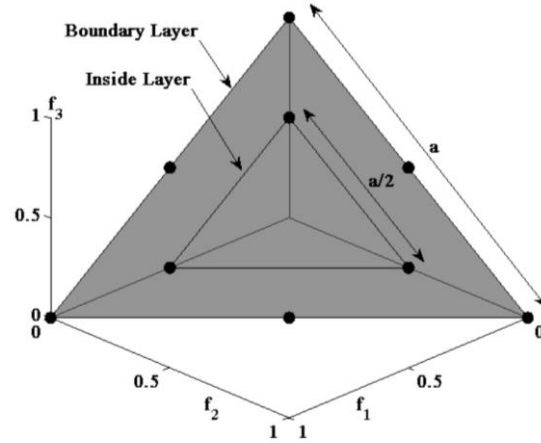


图 5-1 两层参考点的概念（边界层上有 6 个点（ $p=2$ ），内部层上有 3 个点（ $p=1$ ））是针对三目标问题显示的，但实验是针对八个或更多的目标而进行的

表 5-2 列出了本研究中使用的其他 NSGA-III 和 MOEA/D 参数。MOEA/D 需要额外的参数，我们根据其开发者给出的建议进行了设置。两种方法的邻域大小 T 都被设定为 20，另外，MOEA/D-PBI 方法的惩罚参数 θ 被设定为 5。

表 5-2 NSGA-III 和 MOEA/D 的两个版本中使用的参数值

参数	NSGA-III	MOEA/D
SBX 概率[28], p_c	1	1
多项式变异概率[3], p_m	1/n	1/n
η_c [28]	30	20
η_m [28]	20	20

5.1 标准化测试问题

首先，我们使用 3 到 15 个目标的 DTLZ1、DTLZ2、DTLZ3 和 DTLZ4 问题 [53]。变量的数量是 $(M + k - 1)$ ，其中 M 是目标的数量，对于 DTLZ1， $k = 5$ ，而对于 DTLZ2、DTLZ3 和 DTLZ4， $k = 10$ 。在所有这些问题中，对于 DTLZ1 问题，相应的帕累托最优前沿位于 $f_i \in [0, 0.5]$ ，对于其他 DTLZ 问题，则位于 $[0, 1]$ 。由于它们对每个目标都有相同的取值范围，我们在本研究中称这些问题为“标准化的测试问题”。表 5-3 展示了用于每个测试问题的最大代数。

表 5-3 在 M 目标 DTLZ1、DTLZ2、DTLZ3 和 DTLZ4 问题上，获得的 NSGA-III 和 MOEA/D 的两个版本的最优、中间和最差 IGD 值。最优性能以粗体显示

Problem	M	MaxGen	NSGA-III	MOEA/D-PBI	MOEA/D-TCH	MOEA/D-DE
DTLZ1	3	400	4.880×10^{-4}	4.095×10^{-4}	3.296×10^{-2}	5.470×10^{-3}
			1.308×10^{-3}	1.495×10^{-3}	3.321×10^{-2}	1.778×10^{-2}
			4.880×10^{-3}	4.743×10^{-3}	3.359×10^{-2}	3.394×10^{-1}
	5	600	5.116×10^{-4}	3.179×10^{-4}	1.124×10^{-1}	2.149×10^{-2}
			9.799×10^{-4}	6.372×10^{-4}	1.129×10^{-1}	2.489×10^{-2}
			1.979×10^{-3}	1.635×10^{-3}	1.137×10^{-1}	3.432×10^{-2}
	8	750	2.044×10^{-3}	3.914×10^{-3}	1.729×10^{-1}	3.849×10^{-2}
			3.979×10^{-3}	6.106×10^{-3}	1.953×10^{-1}	4.145×10^{-2}
			8.721×10^{-3}	8.537×10^{-3}	2.094×10^{-1}	4.815×10^{-2}
	10	1000	2.215×10^{-3}	3.872×10^{-3}	2.072×10^{-1}	4.253×10^{-2}
			3.462×10^{-3}	5.073×10^{-3}	2.147×10^{-1}	4.648×10^{-2}
			6.869×10^{-3}	6.130×10^{-3}	2.400×10^{-1}	4.908×10^{-2}
	15	1500	2.649×10^{-3}	1.236×10^{-2}	3.237×10^{-1}	8.048×10^{-2}
			5.063×10^{-3}	1.431×10^{-2}	3.438×10^{-1}	8.745×10^{-2}
			1.123×10^{-2}	1.692×10^{-2}	3.634×10^{-1}	1.008×10^{-1}
DTLZ2	3	250	1.262×10^{-3}	5.432×10^{-4}	7.499×10^{-2}	3.849×10^{-2}
			1.357×10^{-3}	6.406×10^{-4}	7.574×10^{-2}	4.562×10^{-2}
			2.114×10^{-3}	8.006×10^{-4}	7.657×10^{-2}	6.069×10^{-2}
	5	350	4.254×10^{-3}	1.219×10^{-3}	2.935×10^{-1}	1.595×10^{-1}
			4.982×10^{-3}	1.437×10^{-3}	2.945×10^{-1}	1.820×10^{-1}
			5.862×10^{-3}	1.727×10^{-3}	2.953×10^{-1}	1.935×10^{-1}
	8	500	1.371×10^{-2}	3.097×10^{-3}	5.989×10^{-1}	3.003×10^{-1}
			1.571×10^{-2}	3.763×10^{-3}	6.301×10^{-1}	3.194×10^{-1}
			1.811×10^{-2}	5.198×10^{-3}	6.606×10^{-1}	3.481×10^{-1}
	10	750	1.350×10^{-2}	2.474×10^{-3}	7.002×10^{-1}	2.629×10^{-1}
			1.528×10^{-2}	2.778×10^{-3}	7.266×10^{-1}	2.873×10^{-1}
			1.697×10^{-2}	3.235×10^{-3}	7.704×10^{-1}	3.337×10^{-1}
	15	1000	1.360×10^{-2}	5.254×10^{-3}	1.000	3.131×10^{-1}
			1.726×10^{-2}	6.005×10^{-3}	1.084	3.770×10^{-1}
			2.114×10^{-2}	9.409×10^{-3}	1.120	4.908×10^{-1}
DTLZ3	3	1000	9.751×10^{-4}	9.773×10^{-4}	7.602×10^{-2}	5.610×10^{-2}
			4.007×10^{-3}	3.426×10^{-3}	7.658×10^{-2}	1.439×10^{-1}
			6.665×10^{-3}	9.113×10^{-3}	7.764×10^{-2}	8.887×10^1
	5	1000	3.086×10^{-3}	1.129×10^{-3}	2.938×10^{-1}	1.544×10^{-1}
			5.960×10^{-3}	2.213×10^{-3}	2.948×10^{-1}	2.115×10^{-1}
			1.196×10^{-2}	6.147×10^{-3}	2.956×10^{-1}	8.152×10^1
	8	1000	1.244×10^{-2}	6.459×10^{-3}	6.062×10^{-1}	2.607×10^{-1}
			2.375×10^{-2}	1.948×10^{-2}	6.399×10^{-1}	3.321×10^{-1}
			9.649×10^{-2}	1.123	6.808×10^{-1}	3.923
	10	1500	8.849×10^{-3}	2.791×10^{-3}	7.174×10^{-1}	2.549×10^{-1}
			1.188×10^{-2}	4.319×10^{-3}	7.398×10^{-1}	2.789×10^{-1}
			2.083×10^{-2}	1.010	8.047×10^{-1}	2.998×10^{-1}
	15	2000	1.401×10^{-2}	4.360×10^{-3}	1.029	2.202×10^{-1}
			2.145×10^{-2}	1.664×10^{-2}	1.073	3.219×10^{-1}
			4.195×10^{-2}	1.260	1.148	4.681×10^{-1}
DTLZ4	3	600	2.915×10^{-4}	2.929×10^{-1}	2.168×10^{-1}	3.276×10^{-2}
			5.970×10^{-4}	4.280×10^{-1}	3.724×10^{-1}	6.049×10^{-2}
			4.286×10^{-1}	5.234×10^{-1}	4.421×10^{-1}	3.468×10^{-1}
	5	1000	9.849×10^{-4}	1.080×10^{-1}	3.846×10^{-1}	1.090×10^{-1}
			1.255×10^{-3}	5.787×10^{-1}	5.527×10^{-1}	1.479×10^{-1}
			1.721×10^{-3}	7.348×10^{-1}	7.491×10^{-1}	4.116×10^{-1}
	8	1250	5.079×10^{-3}	5.298×10^{-1}	6.676×10^{-1}	2.333×10^{-1}
			7.054×10^{-3}	8.816×10^{-1}	9.036×10^{-1}	3.333×10^{-1}
			6.051×10^{-1}	9.723×10^{-1}	1.035	7.443×10^{-1}
	10	2000	5.694×10^{-3}	3.966×10^{-1}	7.734×10^{-1}	2.102×10^{-1}
			6.337×10^{-3}	9.203×10^{-1}	9.310×10^{-1}	2.885×10^{-1}
			1.076×10^{-1}	1.077	1.039	6.422×10^{-1}
	15	3000	7.110×10^{-3}	5.890×10^{-1}	1.056	4.500×10^{-1}
			3.431×10^{-1}	1.133	1.162	6.282×10^{-1}
			1.073	1.249	1.220	8.477×10^{-1}

图 5-2 显示了 NSGA-III 获得的三目标 DTLZ1 问题的前沿。这个特定的运行与 IGD 性能指标的中值有关。所有 91 个点都很好地在整个帕累托最优集合上。图 5-3 显示了 MOEA/D-PBI 的结果。很明显,MOEA/D-PBI 也能找到与 NSGA-III 类似的良好点的分布。然而,图 5-4 显示,MOEA/D-TCH 无法找到一个均匀的点的分布。这样的分布在最初的 MOEA/D 研究中也有说明[10]。

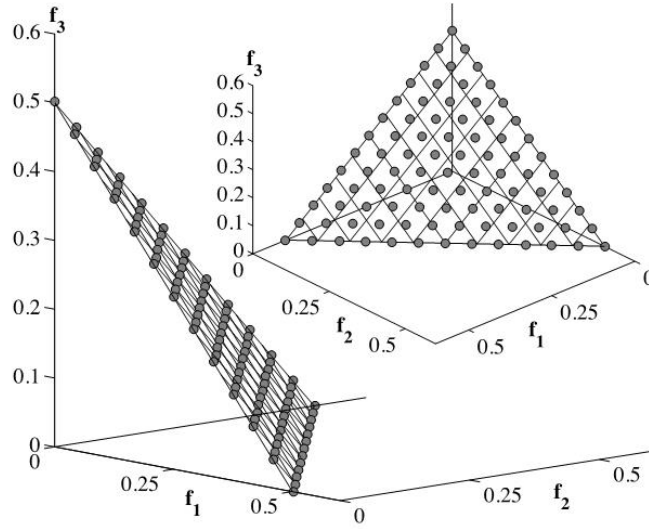


图 5-2 通过 NSGA-III 获得的 DTLZ1 的解

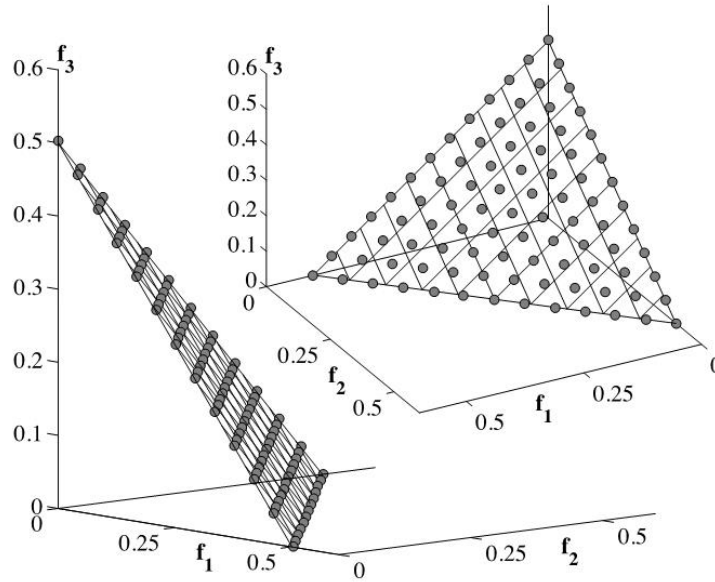


图 5-3 通过 MOEA/D-PBI 获得的 DTLZ1 的解

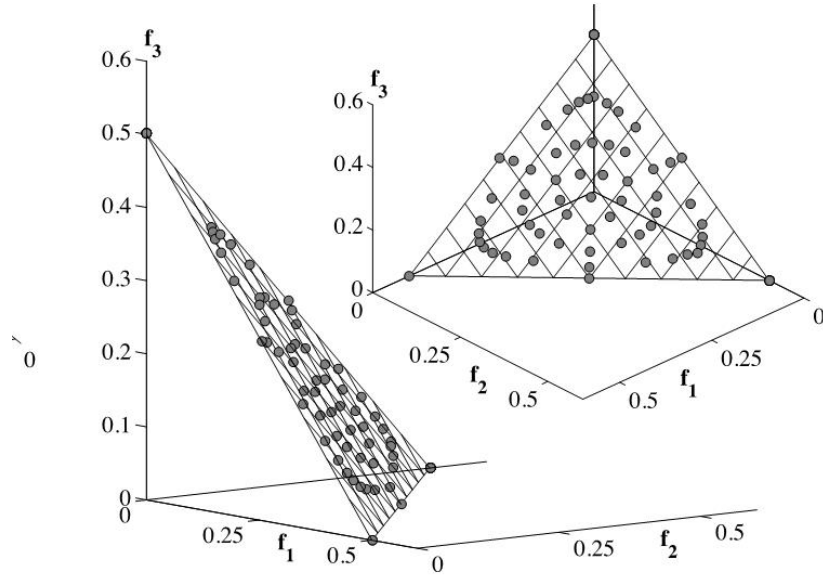


图 5-4 通过 MOEA/D-TCH 获得的 DTLZ1 的解

表 5-3 表明，对于 DTLZ1 问题，NSGA-III 在 IGD 指标方面的表现略好，其次是 MOEA/D-PBI。对于 5 个目标的 DTLZ1 问题，MOEA/D 的表现比 NSGA-III 好，但在 8、10 和 15 个目标的问题中，NSGA-III 的表现更好。MOEA/D-TCH 在所有更高维度的问题中始终表现不好。这一结论与最初基于两目标和三目标问题的 MOEA/D 研究[10]的结论相似。

对于 DTLZ2 问题，MOEA/D-PBI 的性能始终优于 NSGA-III，NSGA-III 的性能优于 MOEA/D-TCH。图 5-5、5-6 和 5-7 展示了 NSGA-III、MOEA/D-PBI 和 MOEA/D-TCH 算法在三目标 DTLZ2 问题上分别获得的点的分布。图中显示，NSGA-III 和 MOEA/D-PBI 的性能不相上下，而 MOEA/D-TCH 的性能较差。

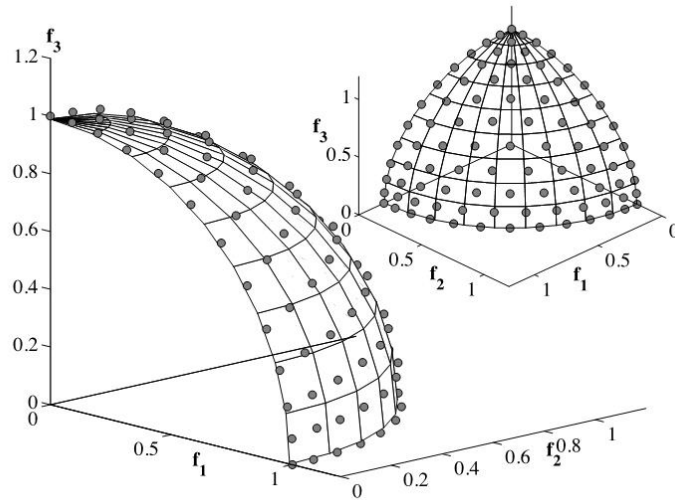


图 5-5 通过 NSGA-III 获得的 DTLZ2 的解

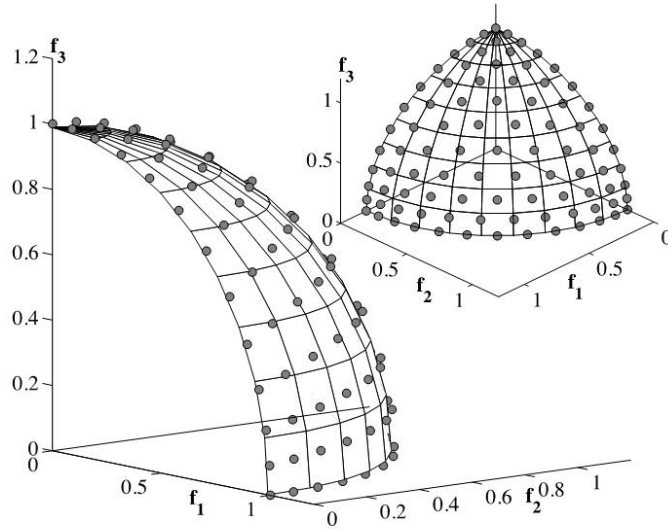


图 5-6 通过 MOEA/D-PBI 获得的 DTLZ2 的解

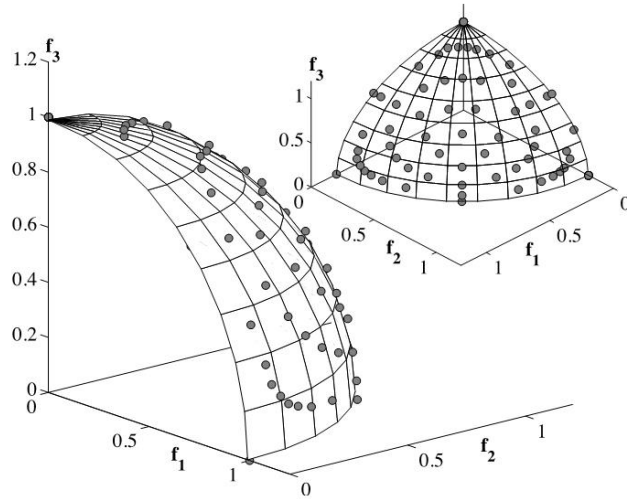


图 5-7 通过 MOEA/D-TCH 获得的 DTLZ2 的解

图 5-8 展示了在 NSGA-III 和 MOEA/D-PBI 方法下八目标 DTLZ2 问题的 IGD 指标值随函数评估的变化。图中展示了 20 次运行的平均 IGD 指标值。显然，这两种方法 IGD 值都能随着函数评估的进行而降低。

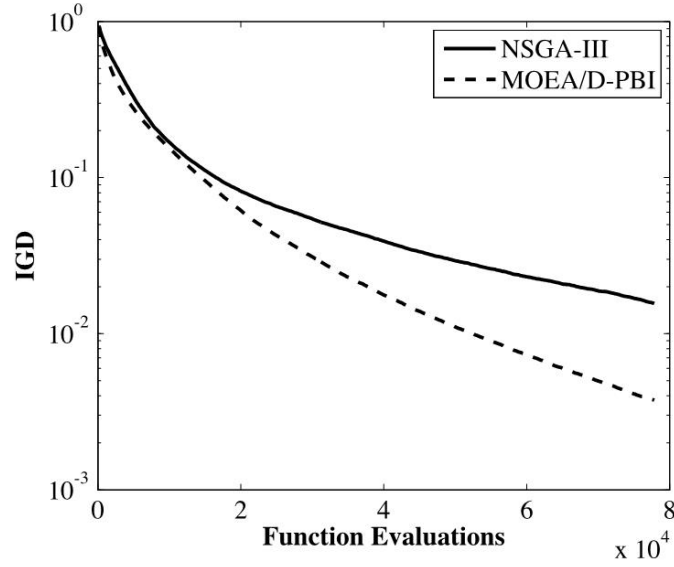


图 5-8 在 NSGA-III 和 MOEA/D-PBI 下 DTLZ2 的 IGD 指标值随着函数评估的变化

对 DTLZ3 问题也进行了类似的观察。这个问题引入了许多局部帕累托最优前沿，为算法接近全局帕累托最优前沿提出了严峻的挑战。虽然 NSGA-III 和 MOEA/D-PBI 的性能相似，MOEA/D-PBI 略胜一筹，但 MOEA/D-TCH 的性能却很差。然而，在某些运行中，MOEA/D-PBI 无法接近帕累托最优前沿，这一点从 IGD 指标值的较大值中可以看出。图 5-9 显示了 NSGA-III 和 MOEA/D-PBI 方法在 20 次运行中的平均 IGD 指标值的变化。在大约 80000 次函数评估之后，NSGA-III 成功地找到了比 MOEA/D-PBI 方法更好的 IGD 值。

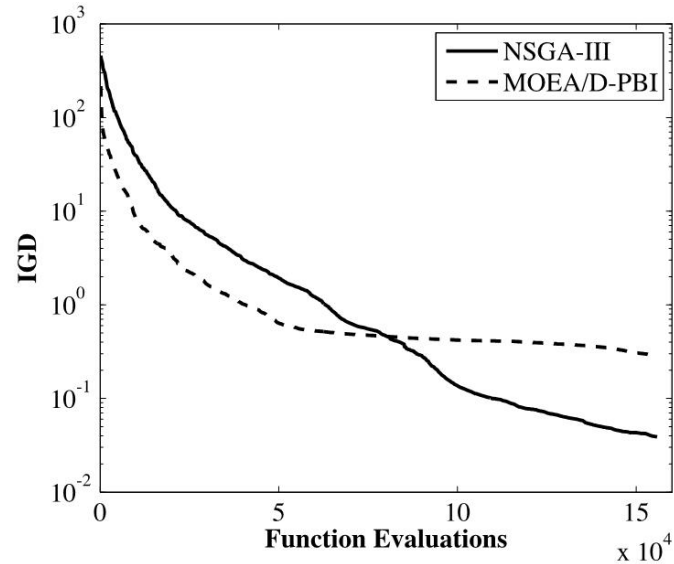


图 5-9 在 NSGA-III 和 MOEA/D-PBI 下 DTLZ3 的 IGD 指标值随着函数评估的变化

问题 DTLZ4 有一个偏离 $f_M = 0$ 的密度的点，然而帕累托最优前沿与 DTLZ2

中的相同。从这个问题可以清楚地看出 NSGA-III 和 MOEA/PBI 的性能差异。MOEA/D 两种算法都不能找到适当的点分布，而 NSGA-III 算法的表现与它在其他问题中的表现一样。图 5-10、5-11 和 5-12 显示了在三目标 DTLZ4 问题上获得的点的分布。这些算法无法找到 $f_3 = 0$ 附近的帕累托最优点，而 NSGA-III 则能够在整个帕累托最优前沿找到一组分布良好的点。这些图是使用在每种情况下运行的中位数绘制的。表 5-3 清楚地表明，NSGA-III 算法的 IGD 指标值比 MOEA/D 算法的指标值要好。图 5-13 显示了 NSGA-III 对 10 个目标的 DTLZ4 问题获得的所有解的成本路径图。所有 10 个目标的解在 $f_i \in [0,1]$ 上的分布以及它们之间的权衡从图中可以清楚地看到。相比之下，图 5-14 显示了使用 MOEA/D-PBI 获得的相同问题的成本路径图。该图清楚地表明，对于 10 个目标的 DTLZ4 问题，MOEA/D-PBI 无法找到一组广泛分布的点。MOEA/DPBI 方法没有找到具有较大目标值的点 f_1 到 f_7 。

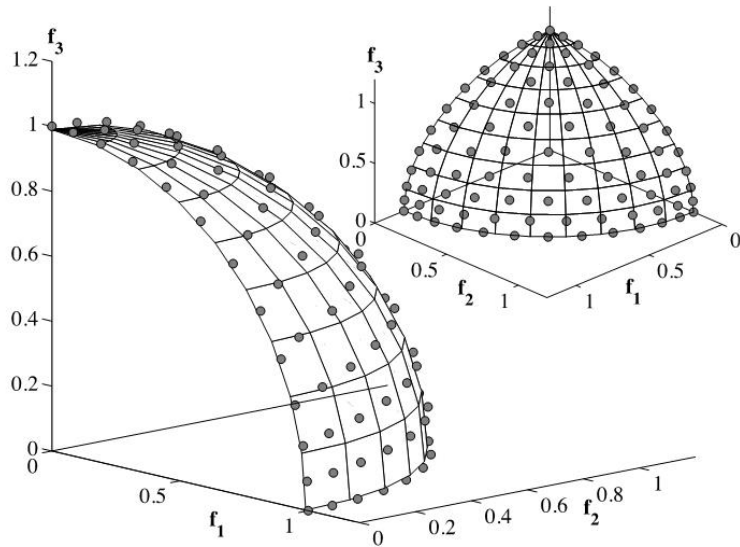


图 5-10 通过 NSGA-III 获得的 DTLZ4 的解

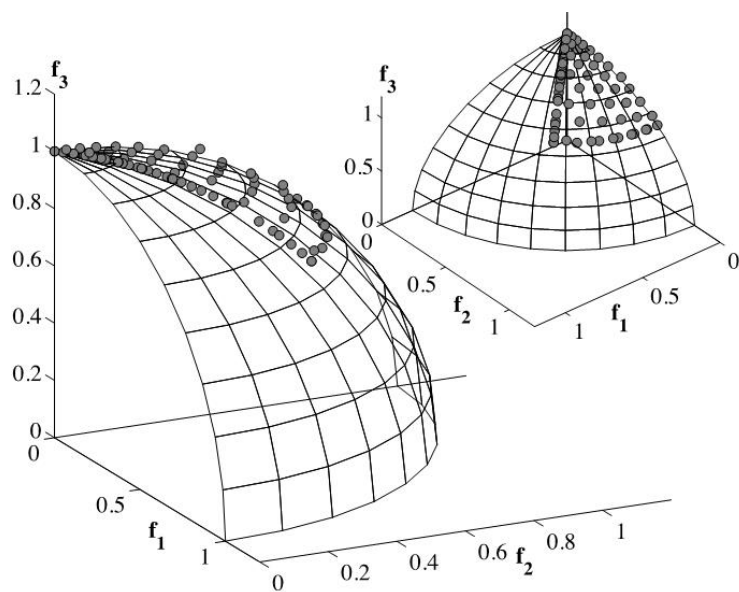


图 5-11 通过 MOEA/D-PBI 获得的 DTLZ4 的解

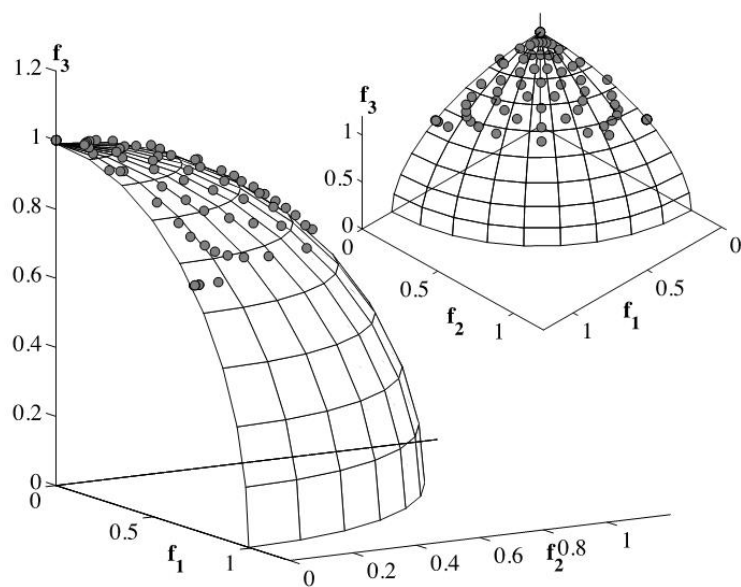


图 5-12 通过 MOEA/D-TCH 获得的 DTLZ4 的解

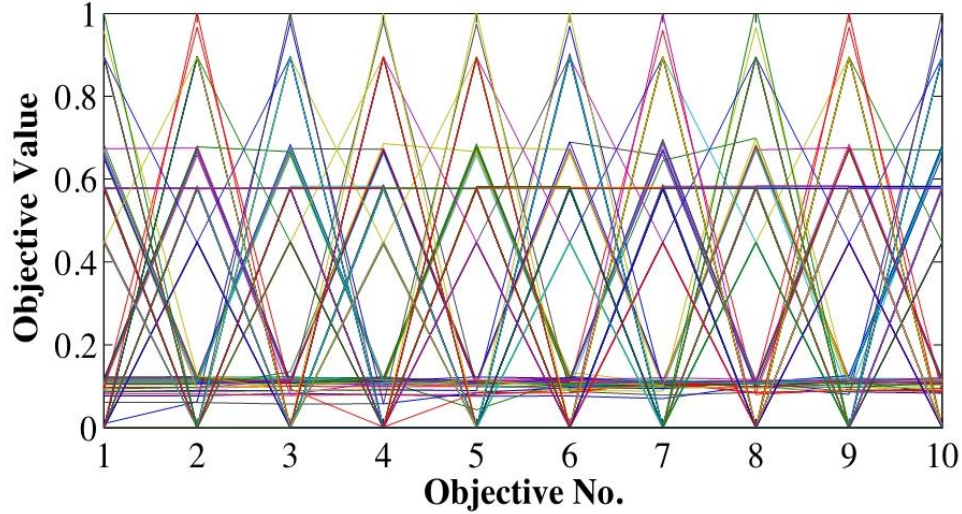


图 5-13 采用 DTLZ4 的 10 个目标值路径格式显示 NSGA-III 的解

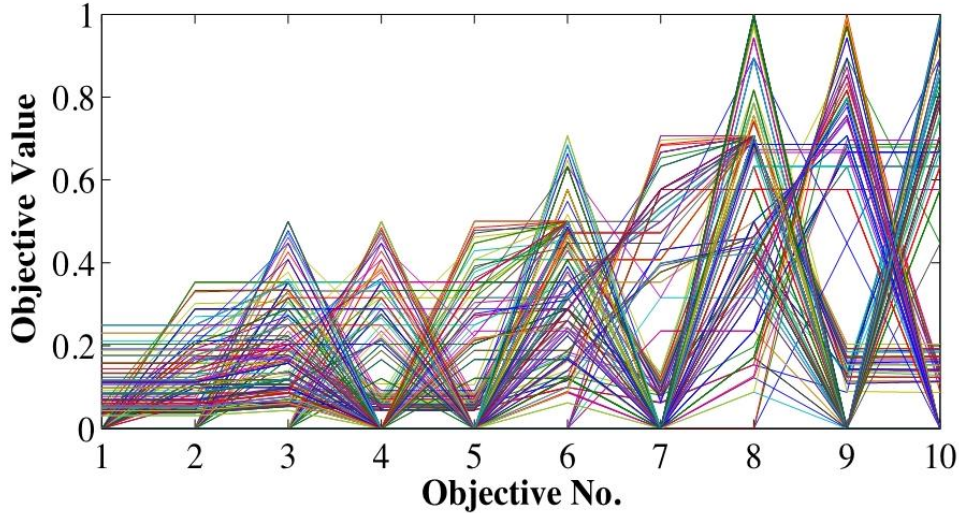


图 5-14 采用 DTLZ4 的 10 个目标值路径格式显示 MOEA/D-PBI 的解

表 5-3 的最右边一栏是最近提出的 MOEA/DE 方法的性能[46]。该方法使用差分进化 (DE) 代替 SBX 和多项式变异算子。在 MOEA/DE 程序中引入了两个额外的参数。首先, 引入一个子代可以相关联的权重向量的数量的最大界限 (n_r), 并设定为 $n_r = 2$ 。其次, 在选择亲本的交配对象时, 设置选择邻近对象的概率 (δ), 选择任何其他种群个体的概率 ($1 - \delta$)。作者建议使用 $\delta = 0.9$ 。我们在研究中使用了这些参数的相同值。表 5-3 显示, 这个版本的 MOEA/D 在标准化的 DTLZ 问题上表现不佳, 但是它在 DTLZ4 问题上的表现比 MOEA/D-PBI 好。由于它在这类问题上的总体表现不佳, 我们没有进一步应用它。

在上述 MOEA/D 的运行中，我们使用了 SBX 重组参数指数 $\eta_c = 20$ （如表 5-2 所示），主要是因为在最初的 MOEA/D 研究中选择了这个值[10]。接下来，我们研究了 MOEA/D-PBI 在 $\eta_c = 30$ 时的性能（NSGA-III 也使用了这个值）。表 5-4 显示，MOEA/D 的性能并没有因为上述 η_c 值的改变而发生很大的变化。

接下来，我们将 NSGA-III 和 MOEA/D-PBI 方法应用于两个 WFG 测试问题。参数设置与之前相同。图 5-15 和 5-16 显示了在 WFG6 问题上获得的点。NSGA-III 方法的收敛性稍好。同样地，图 5-17 和 5-18 显示了上述两种方法在 WFG7 问题上的类似性能比较。表 5-5 显示了这两种方法在 15 个目标的 WFG6 和 WFG7 问题上的表现。

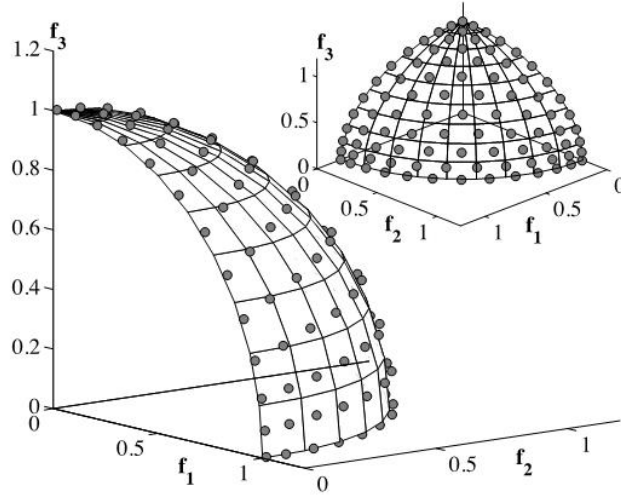


图 5-15 通过 NSGA-III 获得三目标 WFG6 问题的解

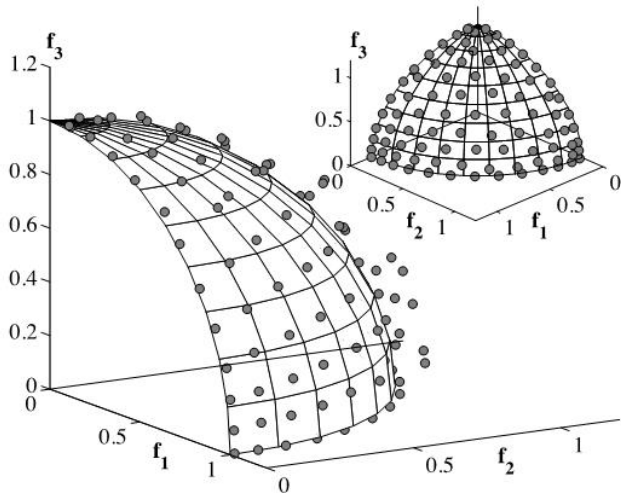


图 5-16 通过 MOEA/D-PBI 获得三目标 WFG6 问题的解

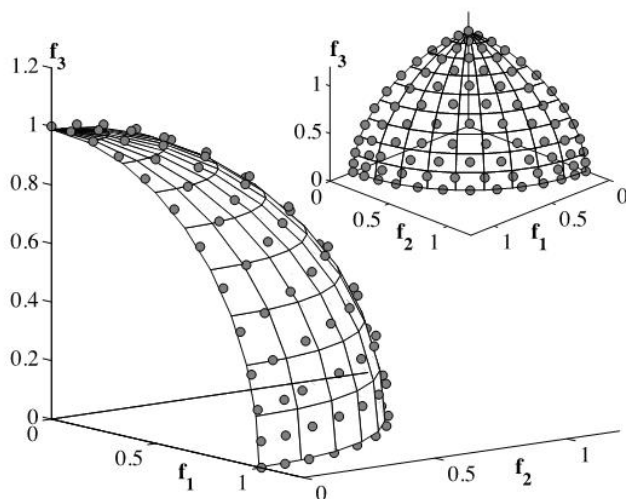


图 5-17 通过 NSGA-III 获得三目标 WFG7 问题的解

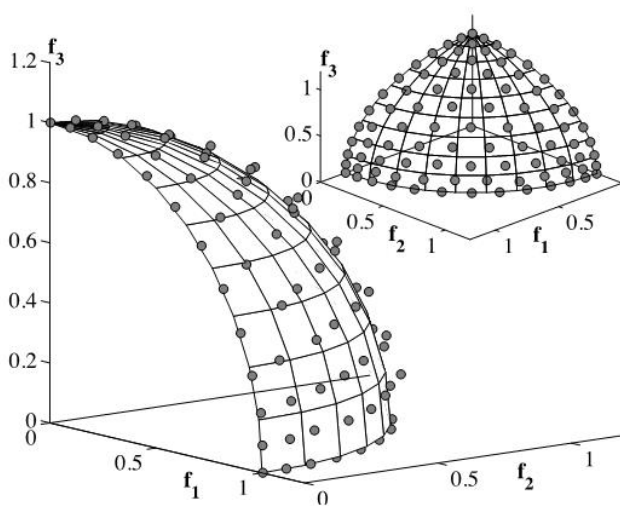


图 5-18 通过 MOEA/D-PBI 获得三目标 WFG7 问题的解

表 5-4 $\eta_c = 30$ 的 MOEA/D-PBI 方法的 IGD 值

Function	M	MaxGen	NSGA-III	MOEA/D-PBI
DTLZ1	3	400	4.880×10^{-4}	4.047×10^{-4}
			1.308×10^{-3}	1.797×10^{-3}
			4.880×10^{-3}	5.699×10^{-3}
	5	600	5.116×10^{-4}	2.849×10^{-4}
			9.799×10^{-4}	5.717×10^{-4}
			1.979×10^{-3}	1.647×10^{-3}
	8	750	2.044×10^{-3}	4.053×10^{-3}
			3.979×10^{-3}	6.839×10^{-3}
			8.721×10^{-3}	1.127×10^{-2}
	10	1000	2.215×10^{-3}	4.057×10^{-3}
			3.462×10^{-3}	5.251×10^{-3}
			6.869×10^{-3}	6.300×10^{-3}
	15	1500	2.649×10^{-3}	1.301×10^{-2}
			5.063×10^{-3}	1.653×10^{-2}
			1.123×10^{-2}	2.695×10^{-2}
DTLZ2	3	250	1.262×10^{-3}	4.535×10^{-4}
			1.357×10^{-3}	5.778×10^{-4}
			2.114×10^{-3}	8.049×10^{-4}
	5	350	4.254×10^{-3}	1.096×10^{-3}
			4.982×10^{-3}	1.208×10^{-3}
			5.862×10^{-3}	1.370×10^{-3}
	8	500	1.371×10^{-2}	2.793×10^{-3}
			1.571×10^{-2}	3.615×10^{-3}
			1.811×10^{-2}	5.231×10^{-3}
	10	750	1.350×10^{-2}	2.286×10^{-3}
			1.528×10^{-2}	2.433×10^{-3}
			1.697×10^{-2}	2.910×10^{-3}
	15	1000	1.360×10^{-2}	5.292×10^{-3}
			1.726×10^{-2}	5.952×10^{-3}
			2.114×10^{-2}	8.413×10^{-3}
DTLZ3	3	1000	9.751×10^{-4}	1.001×10^{-3}
			4.007×10^{-3}	5.080×10^{-3}
			6.665×10^{-3}	1.154×10^{-2}
	5	1000	3.086×10^{-3}	7.191×10^{-4}
			5.960×10^{-3}	2.344×10^{-3}
			1.196×10^{-2}	6.118×10^{-3}
	8	1000	1.244×10^{-2}	6.285×10^{-3}
			2.375×10^{-2}	2.032×10^{-2}
			9.649×10^{-2}	1.133
	10	1500	8.849×10^{-3}	2.848×10^{-3}
			1.188×10^{-2}	6.110×10^{-3}
			2.083×10^{-2}	4.834×10^{-1}
	15	2000	1.401×10^{-2}	4.647×10^{-3}
			2.145×10^{-2}	1.110×10^{-2}
			4.195×10^{-2}	1.271
DTLZ4	3	600	2.915×10^{-4}	1.092×10^{-1}
			5.970×10^{-4}	4.277×10^{-1}
			4.286×10^{-1}	5.235×10^{-1}
	5	1000	9.849×10^{-4}	2.342×10^{-1}
			1.255×10^{-3}	4.384×10^{-1}
			1.721×10^{-3}	7.347×10^{-1}
	8	1250	5.079×10^{-3}	3.821×10^{-1}
			7.054×10^{-3}	8.015×10^{-1}
			6.051×10^{-1}	9.686×10^{-1}
	10	2000	5.694×10^{-3}	5.083×10^{-1}
			6.337×10^{-3}	8.321×10^{-1}
			1.076×10^{-1}	1.024
	15	3000	7.110×10^{-3}	9.406×10^{-1}
			3.431×10^{-1}	1.157
			1.073	1.219

表 5-5 通过 NSGA-III 获得三目标 WFG6 问题的解

Problem	M	MaxGen	NSGA-III	MOEA/D-PBI
WFG6	3	400	4.828×10^{-3}	1.015×10^{-2}
			1.224×10^{-2}	3.522×10^{-2}
			5.486×10^{-2}	1.066×10^{-1}
	5	750	5.065×10^{-3}	8.335×10^{-3}
			1.965×10^{-2}	4.230×10^{-2}
			4.474×10^{-2}	1.058×10^{-1}
	8	1500	1.009×10^{-2}	1.757×10^{-2}
			2.922×10^{-2}	5.551×10^{-2}
			7.098×10^{-2}	1.156×10^{-1}
	10	2000	1.060×10^{-2}	9.924×10^{-3}
			2.491×10^{-2}	4.179×10^{-2}
			6.129×10^{-2}	1.195×10^{-1}
	15	3000	1.368×10^{-2}	1.513×10^{-2}
			2.877×10^{-2}	6.782×10^{-2}
			6.970×10^{-2}	1.637×10^{-1}
WFG7	3	400	2.789×10^{-3}	1.033×10^{-2}
			3.692×10^{-3}	1.358×10^{-2}
			4.787×10^{-3}	1.926×10^{-2}
	5	750	8.249×10^{-3}	8.780×10^{-3}
			9.111×10^{-3}	1.101×10^{-2}
			1.050×10^{-2}	1.313×10^{-2}
	8	1500	2.452×10^{-2}	1.355×10^{-2}
			2.911×10^{-2}	1.573×10^{-2}
			6.198×10^{-2}	2.626×10^{-2}
	10	2000	3.228×10^{-2}	1.041×10^{-2}
			4.292×10^{-2}	1.218×10^{-2}
			9.071×10^{-2}	1.490×10^{-2}
	15	3000	3.457×10^{-2}	7.552×10^{-3}
			5.450×10^{-2}	1.063×10^{-2}
			8.826×10^{-2}	2.065×10^{-2}

根据对 3 至 15 个目标的标准化 DTLZ 和 WFG 测试问题的结果，可以得出以下结论：(i) MOEA/D-PBI 的表现一直优于 MOEA/DTCH 方法，(ii) MOEA/D-PBI 在某些问题上表现最好。而我们提出的 NSGA-III 方法在其他一些问题上表现最好，(iii) 在非均匀分布的帕累托最优前沿（如 DTLZ4 问题）中，MOEA/D 方法都不能保持良好的点分布，而 NSGA-III 表现良好。

5.2 经典的生成方法

为了解决多目标优化问题，我们需要一组参考点——要么由用户给定，要么像前面讨论的那样系统地构建。人们可能会思考所提出的 NSGA-III 方法将如何与经典的生成方法相比较，在经典的生成方法中，多个标量化的单目标优化问题可以针对每个优选的或结构化的参考点制定并独立解决。为此，我们对每种情况下的 PBI 指标进行最小化。沿着将理想点 (z^*) 与由 w 向量（单位向量 $w = (\bar{z} - z^*) / \|\bar{z} - z^*\|$ ）指定的所提供的参考点 (\bar{z}) 相连接而得到的参考方向，计算

出任何点 x 沿 w 向的距离 (d_1) 和垂直于 w 向的距离 (d_2)，然后最小化这两个方向的加权和，如下所示：

$$\text{Minimize}_x \quad d_1 + \theta d_2 = w^T f(x) + \theta \|f(x) - w^T f(x)w\| \quad (5-2)$$

对于所有参考点，参数 θ 被设定为 5[10]。换句话说，如果存在交叉，上述最小化过程有可能找到参考方向 w 和帕累托最优曲面的交叉点。由于这种方法要求理想点 (z^*) 是已知的，所以我们假设原点是这个问题的理想矢量。

为了进行公平的比较，对于 H 个参考点，我们在每个优化过程中最多分配 $T = FE_{\text{NSGA-III}} / H$ (其中 $FE_{\text{NSGA-III}}$ 是 NSGA-III 需要的解的评估总数) 的函数评估。

对于三目标的 DTLZ1 和 DTLZ2 问题，需要 $FE_{\text{NSGA-III}} = 92 \times 400$ 或 36800 才能找到 91 个解。因此，我们通过 Matlab 的 `fmincon` 例程为每个优化运行分配了 36,800/91 或 405 次函数评估。一个随机的解被用于初始化。图 22 和 23 显示了通过这种生成方法获得的最终解决方案。

对于三目标 DTLZ1 和 DTLZ2 问题，需要 $F\text{ENSGA-III} = 92 \times 400$ 或 36,800 才能找到 91 个解。因此，我们为 Matlab 的 `fmincon` 例程运行的每次优化分配了 36,800/91 或 405 次函数求值。随机解决方案用于初始化。图 5-19 和 5-20 显示了用这种生成方法得到的最终解。

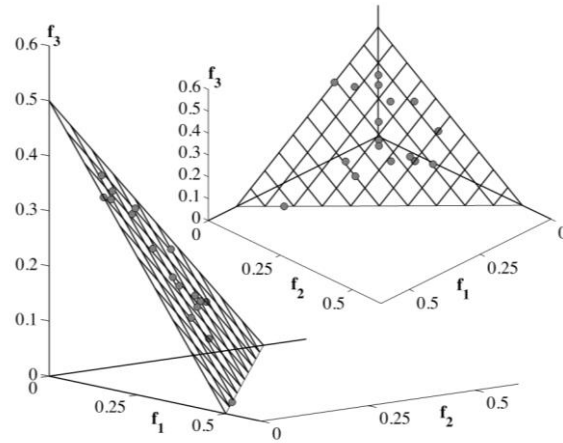


图 5-19 用经典生成法得到的 DTLZ1 的解，只显示了接近真正的帕累托最优前沿的点

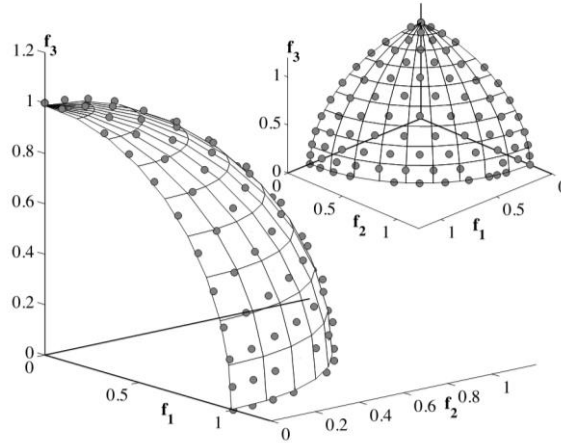


图 5-20 用经典生成法获得的 DTLZ2 的解

DTLZ1 问题有多个局部前沿，Matlab 的 `fmincon` 例程无法在规定的函数评估次数内每次都找到一个帕累托最优解。表 5-6 显示了 IGD 和 GD 指标（获得的点与最近的参考点的平均距离，一个与 IGD 指标意义相反的指标）值。对于三个目标的 DTLZ2 问题，生成方法能够找到一个分布均匀的点集，如图所示。然而，表格显示，在本研究中限制了函数评估的数量，点的分布不如 NSGA-III 得到的好。NSGA-III 固有的并行搜索构成了一个更有效的优化。

表 5-6 在三个目标的 DTLZ1 和 DTLZ2 问题上，NSGA-III 和经典生成方法获得了最佳、中位和最差的 IGD 和收敛指标值

Prob.	FE	NSGA-III		Generative Method	
		IGD	GD	IGD	GD
DTLZ1	36,400	4.880×10^{-4}	4.880×10^{-4}	6.400×10^{-2}	1.702×10^1
		1.308×10^{-3}	6.526×10^{-4}	8.080×10^{-2}	1.808×10^1
		4.880×10^{-3}	7.450×10^{-4}	1.083×10^{-1}	1.848×10^1
DTLZ2	22,750	1.262×10^{-3}	1.264×10^{-3}	1.113×10^{-3}	9.678×10^{-5}
		1.357×10^{-3}	1.270×10^{-3}	6.597×10^{-3}	1.019×10^{-4}
		2.114×10^{-3}	1.274×10^{-3}	9.551×10^{-3}	1.082×10^{-4}

5.3 比例测试问题

为了研究算法在具有不同比例目标值的问题上的表现，我们再次考虑 DTLZ1 和 DTLZ2 问题，但现在我们对它们进行如下修改。目标 f_i 被乘以一个系数 10^i 。为了说明这一点，DTLZ1 问题的三个目标值的 f_1 、 f_2 和 f_3 分别乘以 10^0 、 10^1 和 10^2 。

为了处理不同比例的目标，并使距离 (d_1 和 d_2) 沿着和垂直于 MOEA/D-PBI 方法中的参考方向，我们使用原始 MOEA/D 研究中为 MOEA/D-TCH 方法建议的程序对目标值进行标准化[10]。我们还使用 MOEA/D 网站上的代码[50]来获得这些点。图 5-21、5-22 和 5-23 分别显示了使用 NSGA-III、MOEA/D-PBI 和 MOEA/D-TCH 方法获得的点分布。很明显，带有目标归一化的两种 MOEA/D 方法都不能充分处理目标中涉及的缩放问题，而 NSGA-III 的算子、目标归一化和自适应超平面构造过程能够很好地协商目标的缩放。在按比例计算的问题中，IGD 指标的计算方法是：首先使用精确的帕累托最优前沿的理想点和最低点对目标值进行归一化，然后像以前一样使用参考点计算 IGD 值。结果 IGD 值显示在表 5-7 中。如图 5-24、5-25 和 5-26 所示，在按比例 DTLZ2 问题上也观察到类似的性能，结果见表 5-7。

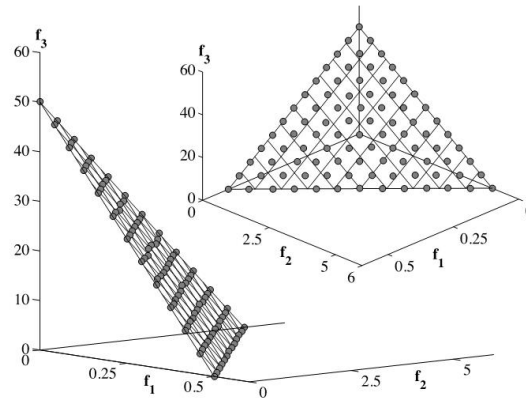


图 5-20 通过 NSGA-III 获得的比例 DTLZ1 的解

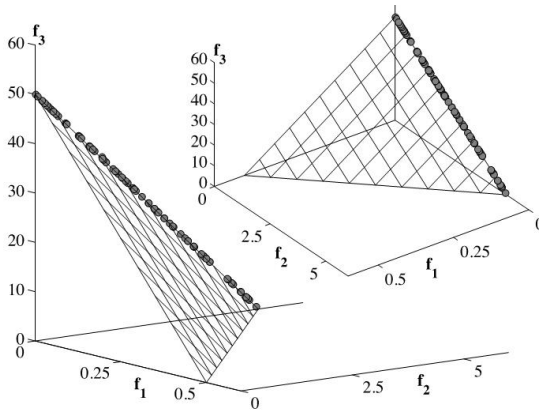


图 5-21 通过 MOEA/D-PBI 获得的比例 DTLZ1 的解

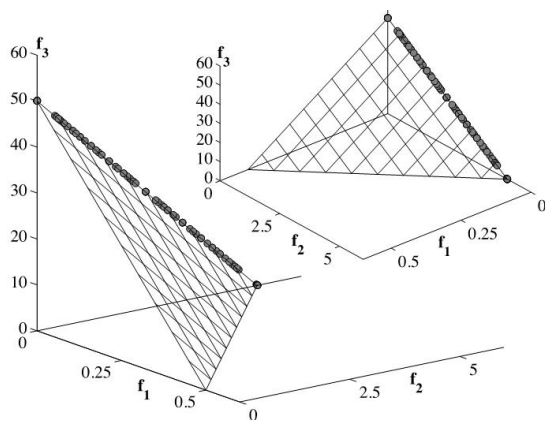


图 5-22 通过 MOEA/D-TCH 获得的比例 DTLZ1 的解

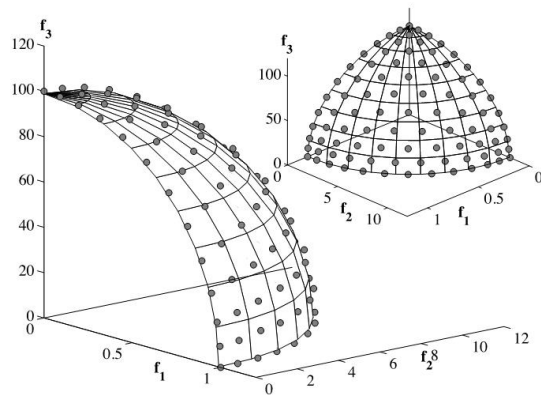


图 5-23 通过 NSGA-III 获得的比例 DTLZ2 的解

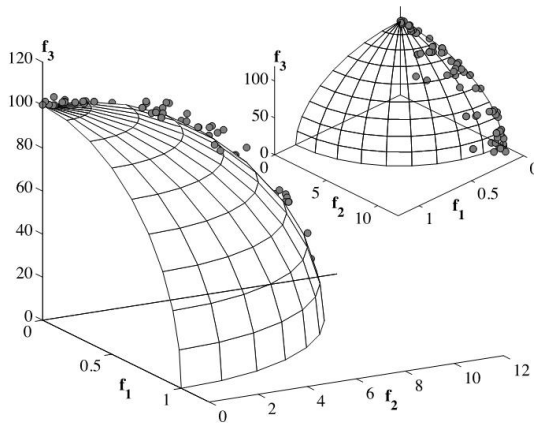


图 5-24 通过 MOEA/D-PBI 获得的比例 DTLZ2 的解

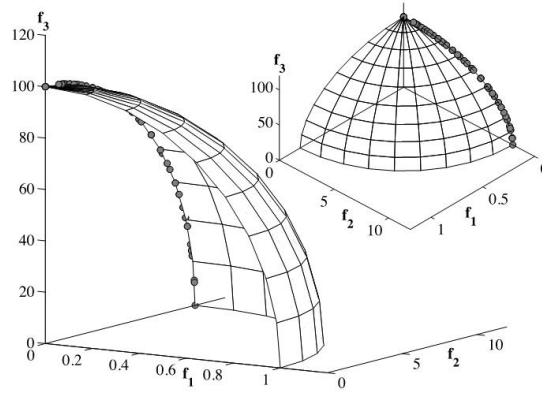


图 5-25 通过 MOEA/D-TCH 获得的比例 DTLZ2 的解

表 5-7 使用 NSGA-III 和 MOEA/D 算法对三目标缩放的比例 DTLZ1 和 DTLZ2 问题的最佳、中位和最差 IGD 值。使用了 10^i 的缩放系数, $i=1,2,\dots,M$

Prob.	MaxGen	NSGA-III	MOEA/D -PBI	MOEA/D -TCH
DTLZ1	400	3.853×10^{-4}	2.416×10^{-2}	8.047×10^{-2}
		1.214×10^{-3}	2.054×10^{-1}	2.051×10^{-1}
		1.103×10^{-2}	4.113×10^{-1}	2.704×10^{-1}
DTLZ2	250	1.347×10^{-3}	4.749×10^{-2}	2.350×10^{-1}
		2.069×10^{-3}	3.801×10^{-1}	5.308×10^{-1}
		5.284×10^{-3}	5.317×10^{-1}	5.321×10^{-1}

值得注意的是,在上一小节中,MOEA/D-PBI 算法在标准化测试问题中表现得很好,但在相同问题的比例版中却表现不佳。实际问题远远没有被归一化,而且目标通常有不同的比例。一个有效的优化算法必须像 NSGA-III 所显示的那样有效地处理目标的不同比例。

接下来,表 5-8 显示了 5、8、10 和 15 个目标比例的比例 DTLZ1 和 DTLZ2 问题中获得的解的 IGD 值。虽然难以直观地看到,但在每种情况下,小的 IGD 值指的是一个分布良好的点集。由于 MOEA/D 算法在比例问题中的表现不佳,我们没有将其进一步应用于上述更高维的比例问题。