

基于多种群的多目标免疫算法

徐建伟, 黄辉先, 彭 维

XU Jian-wei, HUANG Hui-xian, PENG Wei

湘潭大学 信息工程学院, 湖南 湘潭 411105

College of Information Engineering, Xiangtan University, Xiangtan, Hunan 411105, China

E-mail: xujianwei_520@sina.com

XU Jian-wei, HUANG Hui-xian, PENG Wei. Multi-objective immune algorithm based on multi-population. *Computer Engineering and Applications*, 2009, 45(13): 47-50.

Abstract: A new multi-objective immune algorithm based on multi-population is proposed by the use of genetic algorithms parallel mechanism to avoid local convergence. Each population interacts mutually information by the shift of excellent individual. This can effectively promote the diversity of population, and accelerate convergence rate. The performance comparison has been made between the new method and the traditional multi-objective evolutionary algorithm NSGA-II by using several typical testing functions in algorithms simulation experiment. Experimental results suggest that the proposed approach can resolve the problem of multi-objective effectively and has better performances.

Key words: immune algorithm; multi-population; multi-objective optimization; Pareto optimal solution

摘 要: 借鉴遗传算法中采用并行机制避免局部收敛的思想, 提出了一种基于多种群的多目标免疫算法。在该算法中建立多个子种群分别进行免疫操作, 子种群之间通过优秀个体转移进行信息交换, 可有效地提高种群的多样性, 加速收敛。采用几种典型的多目标优化函数进行实验, 并同常用的多目标遗传算法 NSGA-II 进行比较, 仿真结果表明了该算法能有效解决多目标优化问题且具有一定的优越性。

关键词: 免疫算法; 多种群; 多目标优化; Pareto 最优解

DOI: 10.3778/j.issn.1002-8331.2009.13.014 文章编号: 1002-8331(2009)13-0047-04 文献标识码: A 中图分类号: TP18

1 引言

在科学和工程实践中, 许多问题具有多目标的特性, 它们需要同时满足几个不同的目标, 而这些目标之间往往是相互约束、相互冲突的, 这类问题被统称为多目标优化问题^[1] (multi-objective optimal problem, 简称 MOP)。同其他盲目搜索方法相比, 进化算法具有隐含并行性、对问题的依赖性小等优点, 在多目标优化领域具有优越性, 但目前大多数的多目标进化算法是基于 Pareto 优于关系进行排序的, 个体的多样性减少得很快, 常丢失多个其他的非劣解而收敛至单个解。

生物免疫系统是一个高度并行、自适应和自组织的分布式系统, 具有很强的学习、识别、记忆和特征提取能力, 具有多样性、耐受性和天然鲁棒性等特点。根据 Burnet 的克隆选择学说、Jerne 的免疫网络学说和反向选择机制构造了免疫算法 (Immune Algorithm, IA), 它把要解决的问题和约束条件当作抗原, 把问题的解当作抗体, 通过免疫操作使抗体在解空间不断搜索进化, 按照亲和度对抗体与抗原之间的匹配程度以及抗体之间的相似程度进行评价, 直至产生最优解。免疫算法在解决大空间、非线性、全局寻优等复杂问题时具有独特的优越性,

已经应用于信息安全、模式识别、优化设计、故障诊断、机器学习和控制等领域。

本文借鉴遗传算法中采用并行机制避免局部收敛的思想, 提出了一种基于多种群的多目标免疫算法 (MOIA-MP)。通过对多个常用的典型多目标进化算法问题的仿真实验, 并同常用的多目标进化算法 NSGA-II^[2] 进行比较, 验证了该算法的有效性。

2 多目标问题相关概念

最小化与最大化问题可以相互转化, 因此, 仅以最小化多目标问题为研究对象。多目标问题的一般描述为:

给定决策向量 $X=(x_1, x_2, \dots, x_n)$, 它满足下列约束:

$$g_i(X) \geq 0 \quad i=1, 2, \dots, k \quad (1)$$

$$h_i(X)=0 \quad i=1, 2, \dots, l \quad (2)$$

设有 r 个优化目标, 且这 r 个优化目标是相互冲突的, 优化目标可表示为:

$$f(X)=(f_1(X), f_2(X), \dots, f_r(X)) \quad (3)$$

寻求 $X^*=(x_1^*, x_2^*, \dots, x_n^*)$, 使 $f(X^*)$ 在满足约束(1)和(2)的

基金项目: 湖南省自然科学基金 (the Natural Science Foundation of Hunan Province of China under Grant No.06JJ5112)。

作者简介: 徐建伟 (1981-), 男, 硕士研究生, 主要研究方向: 智能信息处理与智能控制; 黄辉先 (1957-), 男, 教授, 硕士生导师, 主要研究方向: 智能控制, 智能交通系统; 彭维 (1982-), 男, 硕士研究生, 主要研究方向: 智能信息处理与智能控制。

收稿日期: 2008-03-20 修回日期: 2008-05-23

同时达到最小。

定义 1(个体的支配关系) 设 p 和 q 是进化群体 Pop 中的任意两个不同的个体, 称 p 支配(dominate) q , 则必须满足下列二个条件:

- (1)对所有的子目标, p 不比 q 差, 即 $f_k(p) \leq f_k(q) (k=1, 2, \dots, r)$;
- (2)至少存在一个子目标, 使 p 比 q 好。即 $\exists l \in \{1, 2, \dots, r\}$ 使 $f_l(p) < f_l(q)$ 。

其中 r 为子目标的数量。此时称 p 为非支配的(non-dominated), q 为被支配的(dominated)。表示为 $p > q$, 其中“ $>$ ”是支配关系(dominate relation)。

定义 2(Pareto 最优解) 给定一个多目标优化问题 $\text{Min } f(X)$, 称 $X^* \in \Omega$ 是最优解, 若 $\forall X \in \Omega$, 满足下列条件:

$$\bigwedge_{i \in I} (f_i(X) = f_i(X^*)) \tag{4}$$

或者至少存在一个 $j \in I, I = \{1, 2, \dots, r\}$, 使:

$$f_j(X) > f_j(X^*) \tag{5}$$

其中 Ω 是满足式(1)和式(2)的可行解集, 即:

$$\Omega = \{X \in R^n | g_i(X) \geq 0, h_j(X) = 0; (i=1, 2, \dots, k; j=1, 2, \dots, l)\}$$

定义 3(Pareto 最优面) 给定一个多目标优化问题 $\text{Min } f(X)$ 和它的最优解集 $\{X^*\}$, 它的 Pareto 最优面定义为:

$$PF^* = \{f(X) = (f_1(X), f_2(X), \dots, f_r(X)) | X \in \{X^*\}\} \tag{6}$$

3 基于多种群的多目标免疫算法(MOIA-MP)

借鉴遗传算法中采用并行机制避免局部收敛的思想, 在免疫算法中建立多个子种群分别进行免疫操作, 并且当每一代进化结束后, 通过交换种群之间优秀个体所携带的优良信息来打破种群内的平衡状态进入更高的平衡状态, 每进化若干代之后再把当前最优个体分配到各个子群体中。从本质上说, 多种群免疫算法是一种并行免疫算法, 能加快进化速度, 防止早熟。

3.1 免疫算法基本理论

将抗原定义为优化问题, 抗体定义为问题的解, 抗原个数即为优化子目标的个数。在此基础上, 可定义如下概念并进行理论分析。

3.1.1 亲和度的计算

定义 4 假设有 N 个抗体, 每个抗体有 M 个基因位, 每个基因位有 S 个符号可供选择, 即 k_1, k_2, \dots, k_s , 则这 N 个抗体的信息熵为

$$H(N) = \frac{1}{M} \sum_{i=1}^M H_i(N) \tag{7}$$

其中: $H_i(N) = -\sum_{j=1}^S p_{ij} \log p_{ij}$, $H_i(N)$ 为 N 个抗体第 i 位的信息熵。

定义 5(抗体的亲和度) 用于表明两抗体之间的相似度。

第 v 号抗体和第 w 号抗体之间的亲和度定义为

$$a_{vw} = \frac{1}{1+H(2)} \tag{8}$$

式中, $H(2)$ 是第 v 号抗体和第 w 号抗体的信息熵, $H(2)$ 等于 0 时说明两抗体所有基因都相同, a_{vw} 的值介于 0 和 1 之间。

同样, 抗体和抗原之间的亲和度定义为

$$a_v = \frac{1}{1+d_v} \tag{9}$$

这里, d_v 表示抗体与抗原之间的差异程度, 采用解空间距离度量, 如果是二进制编码, 一般使用海明距离来度量。 a_v 的值介于

0 和 1 之间。当 $d_v=0$ 时, $a_v=1$, 说明抗体和抗原非常匹配, 即这个抗体是最优解。

定义 6(抗体的浓度) 用于表示某个抗体以及与其很相似的抗体群的规模。第 v 号抗体的浓度 c_v 为 $c_v = \frac{1}{N} \sum_{w=1}^N b_{vw}$ 。式中, $b_{vw} = \begin{cases} 1, & \alpha a_v \leq a_{vw} \leq \beta a_v \\ 0, & \text{其它} \end{cases}$, α 和 β 为调节系数, α 比 1 稍小, β 比 1 稍大。

定义 7 抗体的期望繁殖率 e_v 为

$$e_v = \frac{a_v \prod_{w=1}^N (1-h_{vw})}{c_v \sum_{i=1}^N a_i} \tag{10}$$

式中, $h_{vw} = \begin{cases} a_{vw}, & \alpha a_v \leq a_{vw} \leq \beta a_v \\ 0, & \text{其它} \end{cases}$ 。

3.2 算法描述

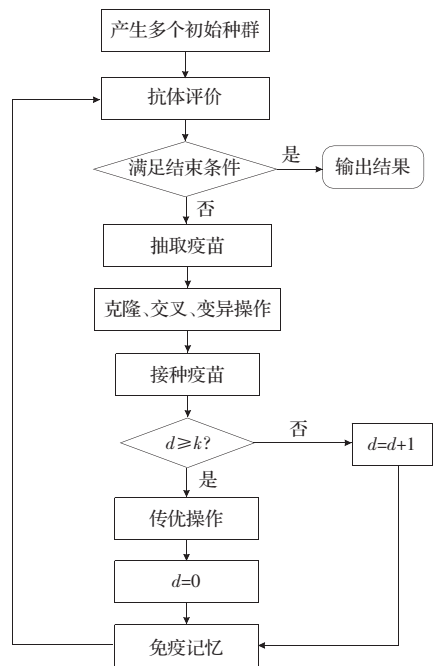


图 1 基于多种群的免疫算法的流程

3.2.1 算法流程图(如图 1)。

3.2.2 算法的实现

步骤 1 初始化种群

随机产生第一代子群体, 设子群体群规模为 n , 采用全局随机取样机制产生各子群体的第一代。模拟自然界丰富多彩的生物群落, 将算法的各子群体按其交叉、变异概率 P_c 和 P_m 分成四类。I 类子群体的 P_c, P_m 最高, 从而使其更易探测到新的解空间, 增大找到最优个体的可能性, 克服早熟。III 类子群体的 P_c, P_m 较低, 这有利于保持其中个体的稳定性, 并强化局部搜索。II 类子群体为“探测开发”子群体, 其 P_c, P_m 介于上述两类子群体之间, 模拟免疫记忆功能。将 IV 类子群体对应于记忆细胞库, 故又称其为“记忆”子群体。“记忆”子群体在进化过程中保存前三类种群进化得到的优秀个体, 使之不遭受破坏, 同时其自身也在进化, 但其 P_c, P_m 是四种子群体中最小的。在种群初始化后, 将初始个体按其适应度大小进行排序, 适应度最大的一些个体分配给记忆子群体, 较大的分配给 III 类子群体, 最

小的分配给 I 类子群体, 其余的分配给 II 类子群体。

步骤 2 抗体评价

算法采用基于浓度的选择机制, 既鼓励适应值高的解, 又抑制浓度高的解, 从而保证了算法的收敛及解群体的多样性。以信息熵作为度量亲和度的指标, 以期望繁殖率作为评价抗体的标准^[3]。

步骤 3 判断是否满足结束条件

当算法得出最优解或者达到事先设定的最大进化代数, 算法就结束, 并输出求解结果, 否则继续往下执行。

步骤 4 抽取疫苗

受到生物免疫系统的启发, 把疫苗的概念引入算法来促进收敛。种群经过前面的进化, 最优解的个别基因片段已经出现在一些抗体中, 这些基因片段就可以当作疫苗, 应该尽早把它们抽取出来, 并接种到其他抗体。抽取疫苗按下面方法进行:

每个抗体的每个基因位有 S 个符号可供选择: k_1, k_2, \dots, k_s , 在该种群中第 i 等位基因上为 k_j 的概率 p_{ij} :

$$p_{ij} = \frac{1}{N} \sum_{j=1}^N a_j \quad (11)$$

其中, $a_j = \begin{cases} 1, & g(i) = k_j \\ 0, & \text{其它} \end{cases}$, $g(i)$ 为种群中第 i 等位基因上的符号, N 为种群的个体数量。

将该等位基因上最大概率大于某个设定阈值的 k_j 作为该等位基因上的疫苗片段, 从而最终提取的疫苗为

$$B = (b_1, b_2, \dots, b_n) \quad (12)$$

其中, $b_j = \begin{cases} k_j, & \max(p_{ij}) \geq T_a \\ 0, & \text{其它} \end{cases}$, T_a 为阈值。

步骤 5 克隆、交叉、变异操作

依据生物免疫系统的免疫克隆原理提出了克隆操作, 把一个抗体复制成更多抗体。

根据交叉概率, 随机地从种群中选取两个染色体 A 和 B 作为父代, 在 A 中随机选取两个非零且不等基因位 x 和 y , 从而得到交叉区间 $[x, y]$ 或 $[y, x]$; 在 B 上找到对应的交叉区间, 将这两个区间内基因互换, 再经过基因修复, 就产生了两个新的子代个体 A' 和 B'。

根据变异概率, 随机从父代种群中选择新个体 S。在 S 中随机选择两个基因位 x_1 和 y_1 形成变异区间 $[x_1, y_1]$ 或 $[y_1, x_1]$, 在此区间上产生一个最大值为 $|x_1 - y_1|$ 的整数随机排列, 按照该排列重新组合该区间上的基因, 从而形成新的个体 S'。

步骤 6 接种疫苗

从父代群体中选择要进行接种的个体, 按照接种概率对选中的个体, 将疫苗的基因片段依次接入, 形成新免疫种群。

步骤 7 判断是否传优操作

算法要求每经过 k 代, 就执行传优操作, 所以判断当前的代数 d 是否够要求的代数 k , 如果不够, 则不执行传优操作, 让代数 d 加 1; 否则, 通过传优算子执行传优操作, 并让 d 置 0。

步骤 8 免疫记忆

在每代进化后, 把最优的几个抗体作为记忆细胞保存到记忆库中, 从记忆子群体中分别选取若干个体迁移到 I、II、III 类的各子群体中, 替换掉它们的较差个体。同时, 算法进化到一定的代数之后, 将所有子群体的个体归并排序, 再按其适应度的大小将个体重新分配到各类子群体中去。然后算法转到步骤 2 继续循环执行。

4 仿真实验与结果分析

4.1 评价方法

评价一个多目标进化算法的性能可以从三个方面去考虑:

- (1) 收敛性: 评价所求解与非劣最优解趋近程度;
- (2) 分布性: 评价所求解在目标空间分布是否均匀;
- (3) 时间效率。

针对上述三个标准, 分别采用一种评价方法去比较三个算法的相关性能。

(1) Generational distance (GD): 用来估计算法的最终解集与全局非劣最优区域的趋近程度, 计算如下:

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (13)$$

n 是解集中个体的数目, d 是每个个体到全局非劣最优解的最小欧几里得距离。GD 的值越小就说明解集越靠近全局非劣最优区域, 如果 $GD=0$ 就说明算法的解都在全局非劣最优区域上, 这是最理想的情况。

(2) Spacing (SP): Schott 提出通过计算解集中每个个体与邻居个体的距离变化来评价解集在目标空间的分布情况, 其函数定义如下:

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (14)$$

$d_i = \min_j (|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|)$, $(i, j = 1, 2, \dots, n)$, n 是解集中个体的数目, \bar{d} 是所有 d_i 的平均值。如果 $SP=0$ 说明解

表 1 实验测试函数

测试问题	目标函数	约束条件及特征
ZDT1	$f_1(x) = x_1$	$m=30; 0 \leq x_i \leq 1$ 真实的 Pareto 凸
	$f_2(x) = g(1 - \sqrt{f_1/g})$	
	$g(x) = 1 + 9 \sum_{i=2}^m x_i / (m-1)$	
ZDT3	$f_1(x) = x_1$	$m=30; 0 \leq x_i \leq 1$ 真实的 Pareto 凸且非连续
	$f_2(x) = g(1 - \sqrt{f_1/g}) - (f_1/g) \sin(10\pi f_1)$	
	$g(x) = 1 + 9 \sum_{i=2}^m x_i / (m-1)$	
ZDT6	$f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$	$m=10; 0 \leq x_i \leq 1$ 真实的 Pareto 面 凹且非均匀
	$f_2(x) = g(1 - (f_1/g)^2)$	
	$g(x) = 1 + 9 \left(\sum_{i=2}^m x_i / (m-1) \right)^{0.25}$	
DTLZ1	$f_1(X) = \frac{1}{2} x_1 x_2 \cdots x_{M-1} (1 + g(X_M))$	$n=10; 0 \leq x_i \leq 1$ 真实的 Pareto 面 线性超平面 $M=3, 4, 6$
	$f_2(X) = \frac{1}{2} x_1 x_2 \cdots (1 - x_{M-1}) (1 + g(X_M))$	
	...	
	$f_{M-1}(X) = \frac{1}{2} x_1 (1 - x_2) (1 + g(X_M))$	
	$f_M(X) = \frac{1}{2} (1 - x_1) (1 + g(X_M))$	
DTLZ2	$g(X_M) = 100 \left[X_M + \sum_{x_i \in X_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right]$	$n=10; 0 \leq x_i \leq 1$ 真实的 Pareto 面 凹 $M=3, 4, 6$
	$f_1(X) = (1 + g(X_M)) \cos(x_1\pi/2) \cdots \cos(x_{M-2}\pi/2) \cos(x_{M-1}\pi/2)$	
	$f_2(X) = (1 + g(X_M)) \cos(x_1\pi/2) \cdots \cos(x_{M-2}\pi/2) \sin(x_{M-1}\pi/2)$	
	$f_3(X) = (1 + g(X_M)) \cos(x_1\pi/2) \cdots \sin(x_{M-2}\pi/2)$	
	...	
	$f_M(X) = (1 + g(X_M)) \sin(x_1\pi/2)$	
	$g(X_M) = \sum_{x_i \in X_M} (x_i - 0.5)^2$	

集中所有个体之间的距离都相等,分布均匀,SP 的值越小说明解集分布越均匀。

(3)CPU-Time:算法运行时间(单位秒)。

4.2 测试函数及实验结果分析

在实验中选用的标准测试函数为 ZDT-1,3,6 (2 个目标)以及 DTLZ-1,DTLZ-2(3 个目标),并与典型的多目标进化算法 NSGA-II 进行了比较研究。测试函数如表 1 所示。其中,ZDT-1 是一个具有连续凸最优面的测试函数,ZDT-6 是一个具有连续凹最优面的测试函数,ZDT-3 是一个具有不连续最优面的测试函数,另外,DTLZ-1 是一个较难收敛的测试函数,而 DTLZ-2 容易收敛,这样就可以用来观察算法的分布性。

在实验中,两种算法均采用实数编码,NSGA-II 的参数设置如下:种群规模为 200,交叉概率 0.8,变异概率 0.01,运行代数 200 代。MOIA-MP 中 I、II、III、IV 类子种群规模均为 50,免疫参数交叉率 P_c 、变异概率 P_m 、疫苗接种概率 P_b 依次为 (0.95,0.3,0.35), (0.8,0.2,0.2), (0.7,0.1,0.1), (0.45,0.05,0.05)。运行代数 200,两种算法对每个测试函数独立运行 10 次,结

果取平均值。

表 2 显示了两种算法在五个测试函数上的运行结果,每个指标的第一行为均值,第二行为标准差。从表 2 可以看出在收敛度方面,无论是均值还是标准差,MOIA-MP 要明显优于 NSGA-II。

图 2、图 3 和图 4、图 5 分别反映了两种算法对测试问题 ZDT1 和 ZDT6 的解集分布情况,从图中可以看出,MOIA-MP 的解集分布情况明显优于 NSGA-II,个体的分布基本均匀。

在 3 维目标的 DTLZ1、DTLZ2 测试问题中,从图 6、图 7 和图 8、图 9 的比较可知,MOIA-MP 的收敛性要好于 NSGA-II 算法,在分布性方面,MOIA-MP 分布得比较好,另外,MOIA-MP 的时间耗费也是较少的。

图 10、图 11 为两种算法在 50 代时解集的分布情况。从中可以发现,对 DTLZ2,NSGA-II 在进化的前期分布度没有得到好的保持,在 50 代时,解集收敛于 Pareto 前沿上的一点;而 MOIA-MP 的分布性得到了较好的保持,在 50 代时,解集的形状已经接近于最优面。

表 2 各项指标的计算结果

算法		ZDT1	ZDT3	ZDT6	DTLZ1	DTLZ2	ZDT1	
MOIA-MP	GD	Average	0.000 124 50	0.000 476 22	0.004 316 98	0.000 528 11	0.004 835 26	0.000 124 50
		Std.Dev.	3.152e-005	4.221e-005	0.001 738 62	4.221e-005	0.001 738 62	3.152e-005
	SP	Average	0.007 002 1	0.009 377 3	0.006 430 8	0.008 265 4	0.006 891 7	0.007 002 1
		Std.Dev.	0.000 652 1	0.002 959 9	0.000 912 2	0.002 959 9	0.000 912 2	0.000 652 1
	TIME/s	Average	2.246	2.411	2.007	2.499	2.063	2.246
		Std.Dev.	0.044	0.021	0.038	0.021	0.038	0.044
NSGA-II	GD	Average	0.000 364 51	0.000 575 91	0.052 445 96	0.000 575 91	0.052 445 96	0.000 364 51
		Std.Dev.	0.000 284 17	8.031e-006	0.018 029 94	8.031e-006	0.018 029 94	0.000 284 17
	SP	Average	0.007 264 1	0.008 792 4	0.007 924 8	0.008 792 4	0.007 924 8	0.007 264 1
		Std.Dev.	0.009 857 1	0.001 150 1	0.003 241 1	0.001 150 1	0.003 241 1	0.009 857 1
	TIME/s	Average	2.455	2.836	2.196	2.836	2.196	2.455
		Std.Dev.	0.051	0.089	0.031	0.089	0.031	0.051

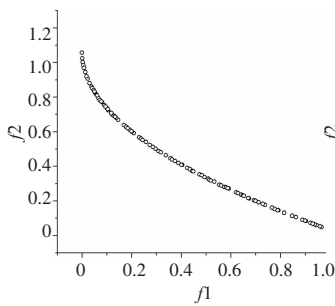


图 2 NSGA-II 对 ZDT1 的最终解集分布

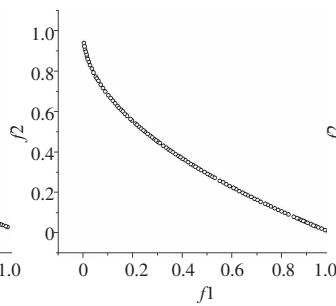


图 3 MOIA-MP 对 ZDT1 的最终解集分布

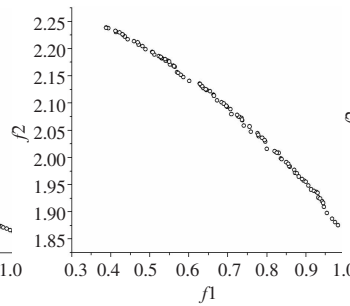


图 4 NSGA-II 对 ZDT6 的最终解集分布

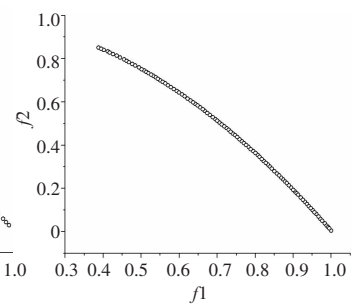


图 5 MOIA-MP 对 ZDT6 的最终解集分布

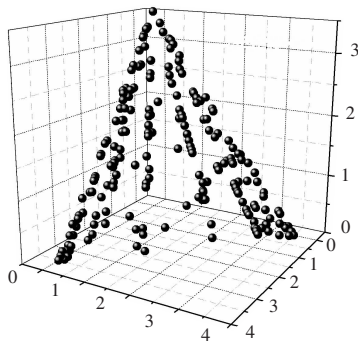


图 6 NSGA-II 对 DTLZ1 的最终解集分布

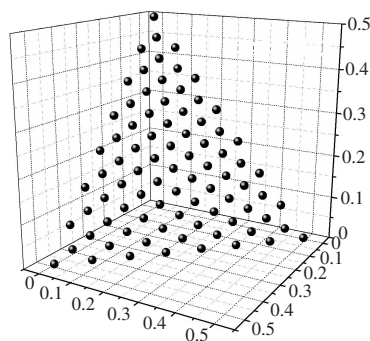


图 7 MOIA-MP 对 DTLZ1 的最终解集分布

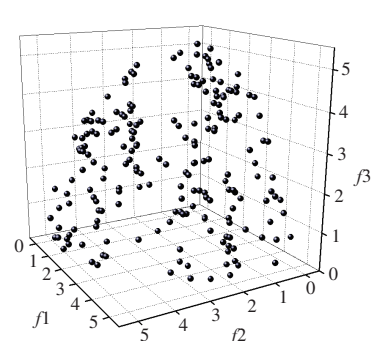


图 8 NSGA-II 对 DTLZ2 的最终解集分布