

基于角度邻域的多目标差分进化算法

赵志伟^{1,2,3}, 杨景明^{1,3†}, 呼子宇^{1,3}, 车海军^{1,3}

(1. 燕山大学 电气工程学院 工业计算机控制工程河北省重点实验室, 河北 秦皇岛 066004;

2. 唐山学院 计算机科学与技术系, 河北 唐山 063000;

3. 燕山大学 国家冷轧板带装备及工艺工程技术研究中心, 河北 秦皇岛 066004)

摘要: 针对如何实现差分进化算法求解多目标优化问题, 提出了一种基于角度邻域的多目标差分进化算法, 通过在选择操作中引入弱支配概念, 实现了对多目标优化问题的求解. 该算法通过计算目标空间中个体与权重向量的夹角来确定每个个体的邻域, 并在此基础上引入了基于角度邻域的变异策略, 使个体的变异在邻域内进行, 保证进化方向. 此外, 该算法创建了一个外部存档用来保存进化过程中的非支配解, 并定期对外部存档进行维护, 大大改善了解集的分佈性. 大量的数值仿真实验结果表明通过角度确定邻域的方法比通过欧氏距离确定邻域的方法更加有效, 算法所得解集的收敛性和分佈性也均明显优于基于分解的差分多目标进化算法(multiobjective evolutionary algorithm based on decomposition and differential evolution, MOEA/D-DE)和非支配排序算法 II (nondominated sorting genetic algorithm II, NSGA).

关键词: 差分进化; 角度邻域; 外部存档; 多目标优化

中图分类号: TP18 文献标识码: A

Multiobjective differential evolution algorithm based on angle neighbourhood

ZHAO Zhi-wei^{1,2,3}, YANG Jing-ming^{1,3†}, HU Zi-yu^{1,3}, CHE Hai-jun^{1,3}

(1. Key Laboratory of Industrial Computer Control Engineering of Hebei Province, School of Electrical Engineering, Yanshan University, Qinhuangdao Hebei 066004, China;

2. Department of Computer Science and Technology, Tangshan University, Tangshan Hebei 063000, China;

3. National Engineering Research Center for Equipment and Technology of Cold Strip Rolling, Yanshan University, Qinhuangdao Hebei 066004, China)

Abstract: To solve the multiobjective optimization problem by differential evolution algorithm, a multiobjective differential evolution algorithm based on angle neighborhood is proposed. The weak domination is introduced to obtain the capacity of solving the multiobjective optimization problem. The neighbourhood of each individual is determined by computing the angle between each individual and weight vector in the objective space. To ensure the evolutionary direction of individual, the mutation strategy based on angle neighbourhood is introduced to execute the mutation operation in angle neighborhood. Additionally, an external archive is established to save the non-dominated solutions obtained in evolutionary process. The archive is maintained regularly, and the distributivity of the approximate set has been greatly improved. A large amount of experimental results show that the neighbourhood determined by angle is more effective than the neighbourhood determined by Euclidean distance, and the convergence and distribution of the approximate set obtained by the proposed algorithm are obviously superior to multiobjective evolutionary algorithm based on decomposition and differential evolution (MOEA/D-DE) and nondominated sorting genetic algorithm II (NSGA II).

Key words: differential evolution; angle neighbourhood; external archive; multiobjective optimization

1 引言(Introduction)

差分进化算法(differential evolution, DE)是由 Storn和Price开发的一种基于种群的随机搜索技术.

目前, 差分进化算法被应用到了很多领域, 比如机械工程、通信、模式识别、信号处理等^[1-3].

由于DE算法在求解单目标优化问题上表现出来

收稿日期: 2016-01-25; 录用日期: 2016-09-30.

†通信作者. E-mail: yangjm6188@sina.com; Tel.: +86 335-8060089.

本文责任编辑: 胡跃明.

国家自然科学基金项目(U1260203), 河北省高等学校创新团队领军人才培养计划项目(LJRC013), 国家冷轧板带装备及工艺工程技术研究中心开放课题(2012005)资助.

Supported by National Natural Science Foundation of China (U1260203), Cultivation Program Project for Leading Talent of Innovation Team in Colleges and Universities of Hebei Province (LJRC013) and Opening Foundation of National Engineering Research Center for Equipment and Technology of Cold Strip Rolling (2012005).

的优异性能, 很多研究者对其进行了改进, 使其能够求解多目标优化问题. Kukkonen等人将弱约束支配的概念引入DE算法的选择操作, 提出了广义差分进化算法(generalized differential evolution algorithm, GDE), 用于求解多目标优化问题^[4]. Kukkonen等人在GDE算法的基础上提出了GDE2算法, 当两个个体均为可行解且互不支配时, 该算法将拥挤度作为选择依据, 有效地改善了种群的多样性和Pareto前沿的分布性^[5]. 之后, Kukkonen等人提出了GDE3算法, 将DE/rand/l/bin策略扩展到了多目标有约束问题上^[6]. Abbass等人将Pareto排序和DE算法结合, 提出了一个Pareto差分进化算法, 实验证明其性能要优于SPEA算法^[7]. Li等人将MOEA/D算法^[8]和DE算法结合, 采用切比雪夫分解方法将多目标优化问题转化为单目标优化问题, 实验结果表明该算法要优于NSGA II算法^[9]. 刘志君等人提出了一种变邻域分解多目标自适应差分进化算法, 将多目标优化问题分解成多个单目标优化问题, 邻域种群规模根据概率自适应选择, 并采用概率匹配方法自适应选择变异策略^[10]. Rakshit等人使用新设计的策略扩展标准DE算法的选择操作, 并引入自适应采样机制来求解带有噪声的多目标优化问题^[11]. Shim等人将基于非支配排序和基于分解的两个多目标优化框架组合, 并引入了进化梯度局部搜索^[12]. Wang等人提出了一个基于Pareto支配的多目标自适应差分进化算法, 并引入了拥挤熵策略来保持Pareto最优解的多样性^[13]. Qiu等人在研究收敛性和参数敏感性的基础上, 提出了一个多目标自适应差分进化算法, 引入了基于邻域的跨代变异策略和参数自适应机制^[14].

本文提出了一种基于角度邻域的多目标差分进化算法(angle neighbourhood multiobjective differential evolution algorithm, ANMODE). ANMODE算法将标准DE算法与弱支配概念相结合实现了对多目标优化问题的求解. 此外, 为改善Pareto前沿近似解集的分布性, 引入了角度邻域的概念, 提出了基于角度邻域的变异策略和外部存档维护机制. 最后使用CEC2009标准测试函数^[15]中的UF1-UF10对ANMODE算法进行数值仿真实验.

2 ANMODE算法的结构(Structure of ANMODE algorithm)

2.1 角度邻域的确定(Determination of the angle neighborhood)

为了改善Pareto前沿近似解集的分布性, ANMODE算法在每一代的进化中引入了角度邻域的操作, 操作流程如下:

1) 根据种群大小 NP 生成数列 $A = \{1/NP, 2/NP, \dots, NP/NP\}$.

2) 为每个个体 $\mathbf{x}_{i,G}$ 生成一个权重向量 $\lambda_i = [\eta_i^t]$ ($i \in \{1, 2, \dots, NP\}$, $t \in \{1, 2, \dots, m\}$, m 为目标函

数的个数), 且权重向量 λ_i 的每一维 η_i^t 均来自数列 A , 并满足如下公式:

$$\sum_{t=1}^m \eta_i^t = 1. \quad (1)$$

3) 根据式(2)计算每个个体 $\mathbf{x}_{j,G}$ 在目标函数空间 $F(\mathbf{x}_{j,G})$ ($j \in \{1, 2, \dots, m\}$)与权重向量 λ_i 夹角 $\theta_{i,j}$ 的余弦值 $C_{i,j}$, 如图1所示. 式(2)如下:

$$C_{i,j} = \frac{\sum_{t=1}^m f_t(\mathbf{x}_{j,G}) \times \eta_i^t}{\sqrt{\sum_{t=1}^m (f_t(\mathbf{x}_{j,G}))^2} \times \sqrt{\sum_{t=1}^m (\eta_i^t)^2}}. \quad (2)$$

4) 选择与权重向量 λ_i 夹角 $\theta_{i,j}$ 最小(即余弦值 $C_{i,j}$ 最大)的 ω 个个体作为个体 $\mathbf{x}_{i,G}$ 的邻域 NB_i .

5) 重复3)和4)直到完成确定每个个体 $\mathbf{x}_{i,G}$ 的邻域.

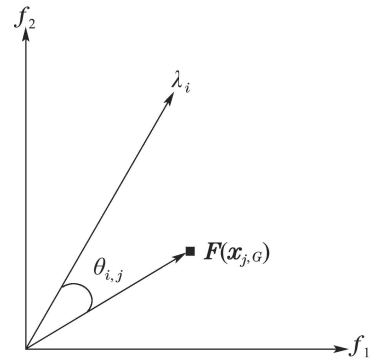


图1 $F(\mathbf{x}_{j,G})$ 与 λ_i 的夹角示意图

Fig. 1 Schematic of the angle between $F(\mathbf{x}_{j,G})$ and λ_i

2.2 基于角度邻域的变异策略(Mutation strategy based on the angle neighborhood)

ANMODE算法在角度邻域的基础上提出了一种新的变异策略——基于角度邻域的变异策略, 具体的变异过程如下:

1) 根据式(3)确定集合 L , 式(3)如下:

$$L = \begin{cases} NB_i, & \text{rand} < \delta, \\ P_G, & \text{其他}, \end{cases} \quad (3)$$

式中: rand为[0,1]间随机数; δ 为概率; P_G 为第 G 代种群.

2) 将集合 L 中的非支配个体存入集合 K .

3) 引入DE/rand/1和DE/best/1, 按照式(4)对每个个体 $\mathbf{x}_{i,G}$ 进行变异操作:

$$\bar{\mathbf{v}}_{i,G} = \begin{cases} \mathbf{x}_{r1,G} + F \times (\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}), & \text{rand} < \tau, \\ \mathbf{x}_{\text{best},G} + F \times (\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}), & \text{其他}, \end{cases} \quad (4)$$

式中: $\mathbf{x}_{r1,G}$, $\mathbf{x}_{r2,G}$ 和 $\mathbf{x}_{r3,G}$ 是从集合 L 中随机选择的个体; $\mathbf{x}_{\text{best},G}$ 是集合 K 中的任意一个个体; τ 为概率.

4) 对 $\bar{v}_{i,G} = [\bar{v}_{i,G}^k] (k \in \{1, 2, \dots, D\}, D$ 为决策变量的维数) 进行多项式变异生成 $v_{i,G} = [v_{i,G}^k]$, 公式如下:

$$v_{i,G} = \begin{cases} \bar{v}_{i,G}^k + \sigma_k \times (x_{\max}^k - x_{\min}^k), & \text{rand} < \beta, \\ \bar{v}_{i,G}^k, & \text{其他,} \end{cases} \quad (5)$$

式中: x_{\max}^k 和 x_{\min}^k 分别为决策变量第 k 维的上界和下界; $\beta = 1/D$; σ_k 由式(6)计算得到:

$$\sigma_k = \begin{cases} (2 \times \text{rand})^{\frac{1}{\omega+1}} - 1, & \text{rand} < 0.5, \\ 1 - (2 - 2 \times \text{rand})^{\frac{1}{\omega+1}}, & \text{其他,} \end{cases} \quad (6)$$

式中 ω 为邻域中个体的数量.

由于ANMODE算法将式(3)中的 δ 设为0.9, 所以变异操作大多是在邻域 NB_i 中进行的. 因邻域内个体具有较大的相似性, 有利于保证个体进化方向. 将式(4)中的 τ 设为0.8, 使得变异操作更多的采用DE/rand/1, 提高了ANMODE算法的全局探索能力, 有利于保持种群的多样性. 由于DE/best/1整合了最好个体的信息, 虽然能加快算法的收敛速度, 但容易导致个体进化方向过于集中, 从而出现群集现象, 这必然影响Pareto前沿近似解集的分布性. 为此, ANMODE算法以较小概率使用DE/best/1, 在不过度影响个体分布性的前提下, 加快算法的收敛速度.

2.3 基于弱支配的选择操作 (Selection based on weak domination)

ANMODE算法沿用标准DE算法的交叉操作生成试验向量 $u_{i,G}$. 为增加中后期进化的种群选择压力, ANMODE算法在选择操作中引入弱支配概念, 按照式(7)完成选择操作:

$$x_{i,G+1} = \begin{cases} u_{i,G}, & F(u_{i,G}) \leq F(x_{i,G}), \\ x_{i,G}, & \text{其他,} \end{cases} \quad (7)$$

式中 $x_{i,G+1}$ 为第 $G+1$ 代个体.

2.4 外部存档的维护 (Maintenance of external archive)

每一代进化完成之后, ANMODE算法将种群中的非支配解加入外部存档 R 中. 每经过一个维护周期 (maintenance cycle, MC) 就对外部存档 R 进行一次非支配排序, R 中只保留非支配个体. 如果非支配个体的数量超过种群大小 NP , 则进行如下操作:

- 1) 将外部存档 R 赋给集合 H , 并清空外部存档 R .
- 2) 计算集合 H 中的每个非支配个体与权重向量 λ_i 的夹角.
- 3) 将与权重向量 λ_i 夹角最小的非支配个体加入外部存档 R 中, 并在 H 中删除该个体.
- 4) 对每个权重向量重复2)和3)的操作.

ANMODE算法的外部存档维护机制有效地改善了非支配个体的多样性, 从而保证了Pareto前沿近似解集具有良好的分布性.

2.5 算法描述 (Description of the algorithm)

ANMODE算法的主要流程描述如下:

步骤 1 参数初始化: 种群大小为 NP , 函数评价次数计数器为 FES , 最大函数评价次数为 MAX_FES , 代计数器为 G , 外部存档为 R , 概率为 σ , τ 和 β , 邻域大小为 ω , 比例因子为 F , 交叉率为 CR , 维护周期为 MC .

步骤 2 初始化种群 P_0 并评估每个个体.

步骤 3 生成权重向量 λ_i .

步骤 4 确定每个个体 $x_{i,G}$ 的邻域 NB_i .

步骤 5 对种群中的个体按照第2.2节和第2.3节进行变异、交叉和选择操作.

步骤 6 将当前种群 P_G 中的非支配个体加入外部存档 R , 若 $\text{MOD}(G, 10) = 0$, 则对 R 进行维护.

步骤 7 判断终止条件: 若满足, 则对 R 进行维护并输出最优解集; 否则转至步骤5.

3 数值仿真实验与结果分析 (Numerical experiments and result analysis)

为验证 ANMODE 算法的性能, 本文将选择 MOEA/D-DE 和 NSGA II^[16] 算法作为比较算法. 对于 ANMODE 算法, σ 设置为0.9, τ 设置为0.8, 邻域 ω 设置为20, 比例因子 F 设置为0.5, 交叉率 CR 设置为0.4, 维护周期 MC 设置为10. 对于 MOEA/D-DE 算法, 邻域 T 设置为20, 权重向量个数与种群大小一致, 其他参数设置均遵循原始文献. NSGA II 算法的参数设置同样参照原始文献. 3个算法的种群大小 NP 设置均相同, 对于两目标优化问题设置为100, 对于三目标优化问题设置为150. 3个算法的最大函数评价次数 MAX_FES 均设置为300000, 针对所有函数都独立运行30次.

为验证角度邻域的有效性, 在实验中增加了使用欧氏距离 (MOEA/D-DE 确定邻域的方法) 确定邻域的 ANMODE 版本作为比较算法, 该算法被命名为基于欧氏距离邻域的多目标差分进化算法 (Euclidean distance neighbourhood multiobjective differential evolution algorithm, ENMODE). 除邻域的确定方法不同外, 其他结构和参数均保持一致. 此外, 为分析基于角度邻域的变异策略的有效性, 比较算法还增加了 ANMODE2 算法, 该算法与 ANMODE 算法唯一的区别在于式(3)中的集合 L 为整个种群, 即变异在整个种群中进行.

本文选择 CEC2009 标准测试函数^[15] 中的 UF1-UF10 进行数值仿真实验, 选择反向世代距离 (inverted generational distance, IGD)^[9] 作为性能评价指标. 反向

世代距离IGD的计算公式如下:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}, \quad (8)$$

式中: P^* 是真实Pareto前沿上均匀分布的点集; P 是真实Pareto前沿的一个近似解集; $d(v, P)$ 是点 v 到 P 中所有点的最短欧氏距离.

如果 P^* 中的点足够多, 能很好地表征真实的 Pareto 前沿, 则 $IGD(P^*, P)$ 既能测量 P 的多样性, 也能测量 P 的收敛性. 一个很小的 IGD 值表示 P 一定非常接近真实的 Pareto 前沿, 并且覆盖了整个前沿. 对于两目标测试函数, 本文在真实的 Pareto 前沿上均匀选择 1000 个点作为 P^* , 而对于三目标测试函数则选择 10000 个点作为 P^* .

3.1 基于IGD的比较(Comparisons based on IGD)

针对所有标准测试函数, 每个算法都独立运行 30

次, 并采集了每次运行求解到的 Pareto 前沿近似解集. 表 1 列出了每个算法 30 次运行的 IGD 均值和标准差, 表中每个函数的各项最小值被加粗显示.

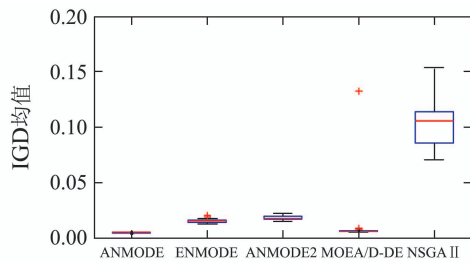
除 UF3 外, ANMODE 算法获得了所有函数的最小 IGD 均值, 这说明 ANMODE 算法求解到的 Pareto 前沿近似解集具有较好的收敛性和分布性, 性能明显优于比较算法. 除 UF10 外, ANMODE 算法获得了所有函数的 IGD 最小标准差, 这意味着 ANMODE 算法具有较强的鲁棒性, 其稳定性明显强于比较算法. NSGA II 算法是所有算法中表现最差的, 没有获得任何函数的 IGD 最小均值和标准差.

图 2 绘制了所有算法 30 次运行的 IGD 盒图. 相对于比较算法, ANMODE 算法的中位数和四分位距均较小, 说明其求解的精度和鲁棒性较好. 此外, ANMODE 算法产生的异常点也明显少于比较算法, 同样证明了其具有良好的鲁棒性.

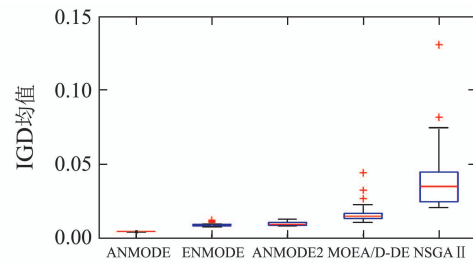
表 1 30 次运行的 IGD 的统计结果

Table 1 Statistical results of IGD for 30 runs

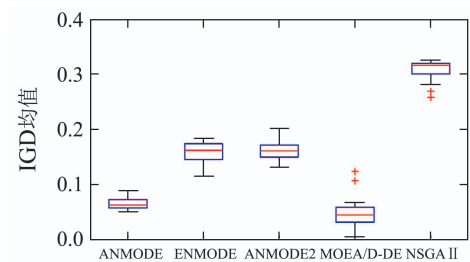
函数名	均值(标准差)				
	ANMODE	ENMODE	ANMODE2	MOEA/D-DE	NSGA II
UF1	4.93E-03 (1.76E-04)	1.54E-02(1.60E-03)	1.82E-02(1.85E-03)	1.04E-02(2.30E-02)	1.01E-01(1.95E-02)
UF2	4.25E-03 (7.81E-05)	8.74E-03(1.09E-03)	9.48E-03(1.18E-03)	1.65E-02(6.90E-03)	4.12E-02(2.37E-02)
UF3	6.49E-02(1.07E-02)	1.56E-01(1.86E-02)	1.61E-01(1.83E-02)	4.65E-02 (2.42E-02)	3.07E-01(1.77E-02)
UF4	2.55E-02 (9.25E-04)	3.65E-02(1.16E-03)	3.55E-02(1.00E-03)	7.14E-02(6.02E-03)	4.62E-02(2.98E-03)
UF5	1.97E-01 (2.41E-02)	8.32E-01(1.22E-01)	6.17E-01(1.77E-01)	3.85E-01(1.59E-01)	2.34E-01(1.15E-01)
UF6	2.14E-01 (5.18E-02)	2.46E-01(7.46E-02)	2.69E-01(8.70E-02)	4.73E-01(2.26E-01)	3.12E-01(1.24E-01)
UF7	1.25E-02 (2.46E-03)	1.59E-02(6.37E-03)	2.48E-02(6.97E-03)	4.96E-02(1.48E-01)	9.18E-02(1.26E-01)
UF8	1.11E-01 (1.22E-02)	1.70E-01(2.64E-02)	2.12E-01(4.39E-02)	1.76E-01(4.94E-02)	2.60E-01(1.80E-02)
UF9	9.24E-02 (1.33E-02)	2.00E-01(3.88E-02)	2.36E-01(3.96E-02)	1.74E-01(3.52E-02)	2.43E-01(6.58E-02)
UF10	3.86E-01 (2.41E-01)	3.02E+00(6.52E-01)	1.97E+00(4.01E-01)	5.00E-01(7.48E-02)	4.74E-01(1.99E-01)



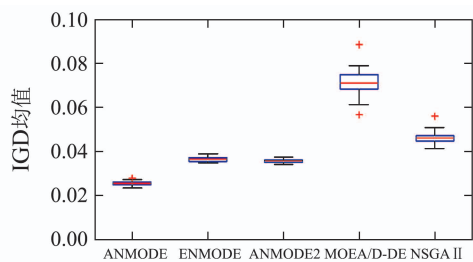
(a) UF1



(b) UF2



(c) UF3



(d) UF4

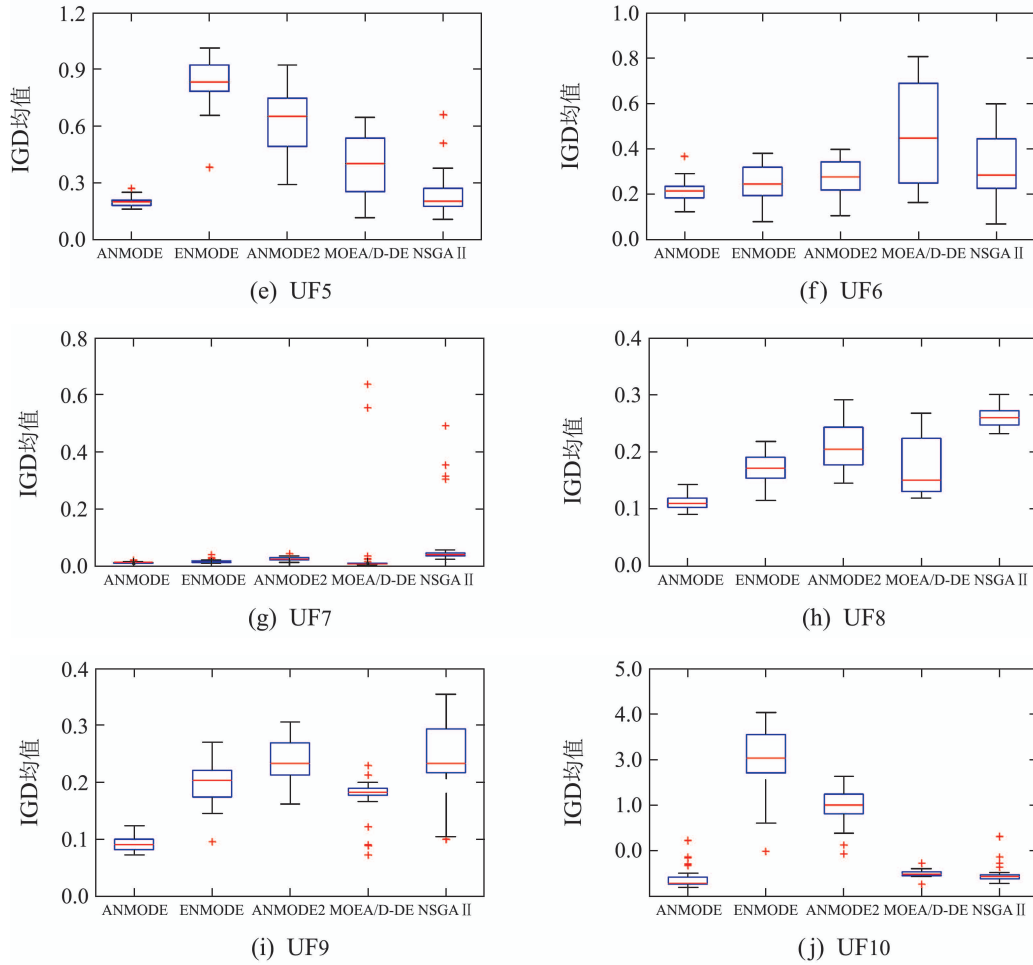


图2 30次运行的IGD盒图
Fig. 2 Box plots of IGD with 30 runs

3.2 基于P-value的比较(Comparisons based on P-value)

本文对ANMODE算法与比较算法进行了置信度为5%的Wilcoxon符号秩检验,并将得到的P-value在表2中列出.

表2 30次运行的IGD的统计结果
Table 2 Statistical results for 30 runs of IGD

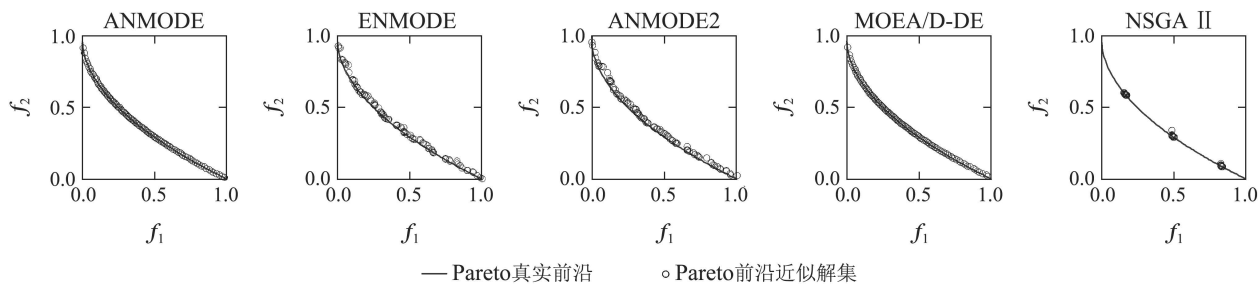
函数名	vs. ENMODE	vs. ANMODE2	vs. MOEA/D-DE	vs. NSGA II
UF1	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
UF2	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
UF3	1.7344E-06	1.7344E-06	6.1564E-04	1.7344E-06
UF4	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
UF5	1.7344E-06	1.7344E-06	1.2381E-05	3.9333E-01
UF6	4.0702E-02	6.8359E-03	6.3198E-05	8.9443E-04
UF7	1.0444E-02	1.9209E-06	1.7791E-01	1.7344E-06
UF8	1.9209E-06	1.7344E-06	3.5152E-06	1.7344E-06
UF9	1.9209E-06	1.7344E-06	2.6033E-05	1.7344E-06
UF10	1.7344E-06	1.7344E-06	1.1079E-02	3.0010E-02
+/-/-	10/0/0	10/0/0	8/1/1	9/1/0

如果 P -value 大于等于 0.05, 说明在统计学意义上 ANMODE 算法和比较算法没有明显差异, 即两者性能相似; 否则, 说明两者在统计学意义上存在显著差异. 表中加粗的表示两者性能相似, 而加粗倾斜的表示该算法的性能明显优于 ANMODE 算法. 表格最后一行的 “+/-/-” 分别表示 ANMODE 算法优于/相似于/劣于比较算法的个数. ANMODE 算法在 UF3 上的表现明显不如 MOEA/D-DE 算法, 而两者在 UF7 上的表现相似. 除 UF3 和 UF7 外, ANMODE 算法在其他 8 个函数上的表现要明显优于 MOEA/D-DE 算法. NSGA II 算法在 UF5 上的表现

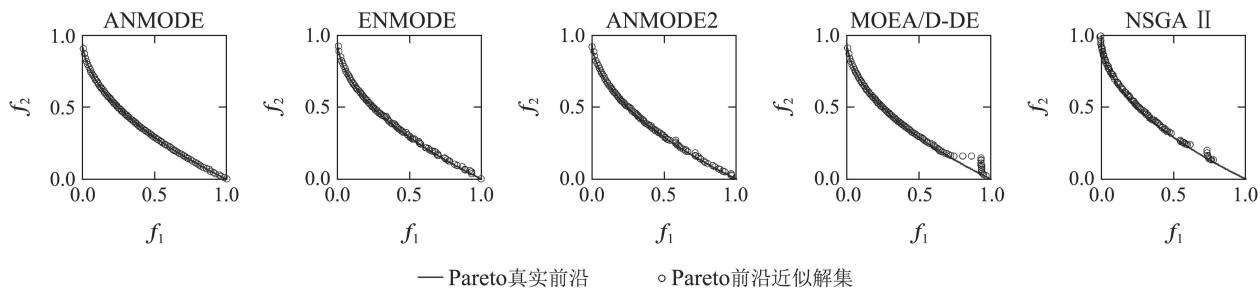
与 ANMODE 算法相似, 没有显著差异. 除 UF5 外, ANMODE 算法在其他 9 个函数上的表现均明显优于 NSGA II 算法.

3.3 收敛性和分布性比较(Comparisons on convergence and distribution)

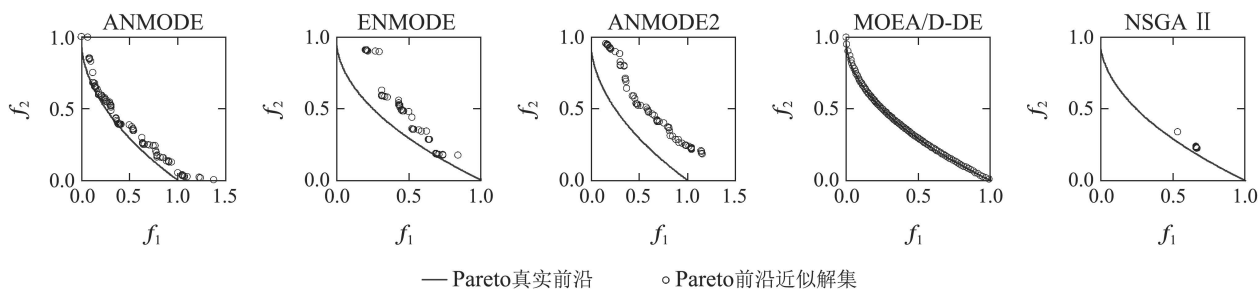
为了更形象地说明算法的性能, 使用 30 次运行中 IGD 最小的那次实验数据绘制了图 3 和图 4. 图 3 是算法求解到的 Pareto 前沿近似解集与 Pareto 真实前沿的对比图, 而图 4 是求解到的 Pareto 近似最优解集与 Pareto 真实最优解集的对比图. 图 4 是使用决策变量 x 的前三维绘制的, 即 x_1, x_2 和 x_3 .



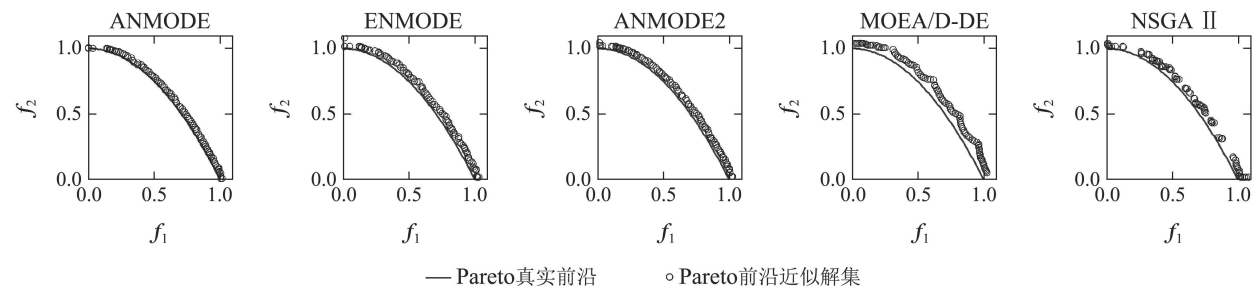
(a) UF1



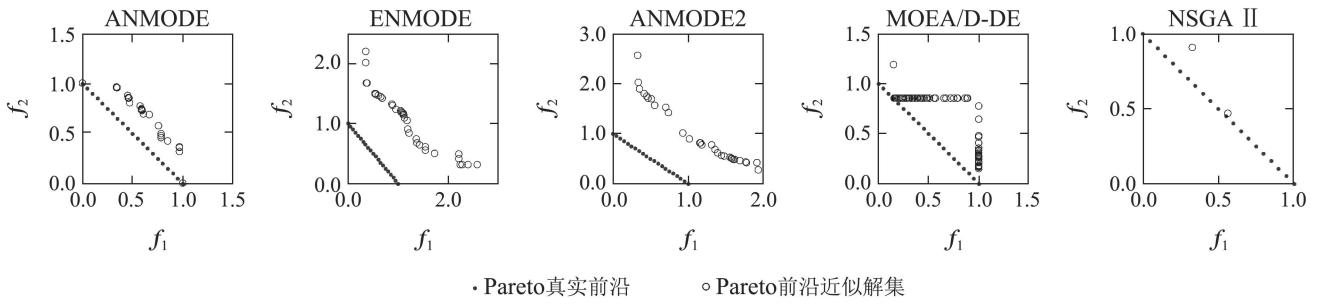
(b) UF2



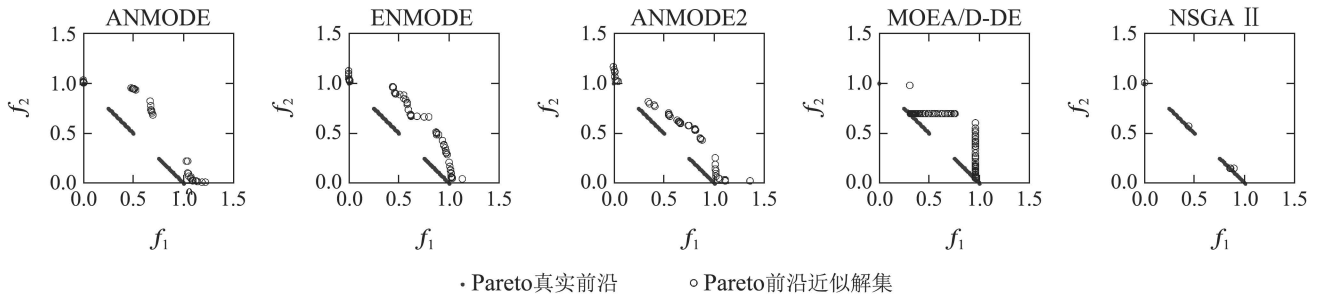
(c) UF3



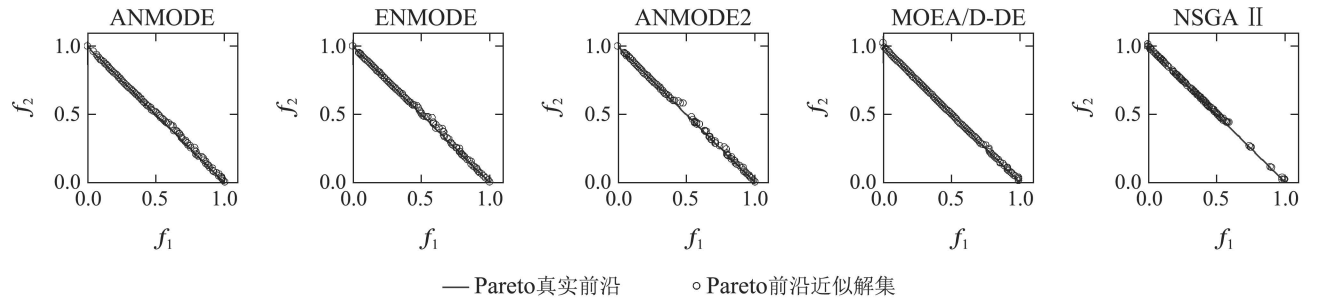
(d) UF4



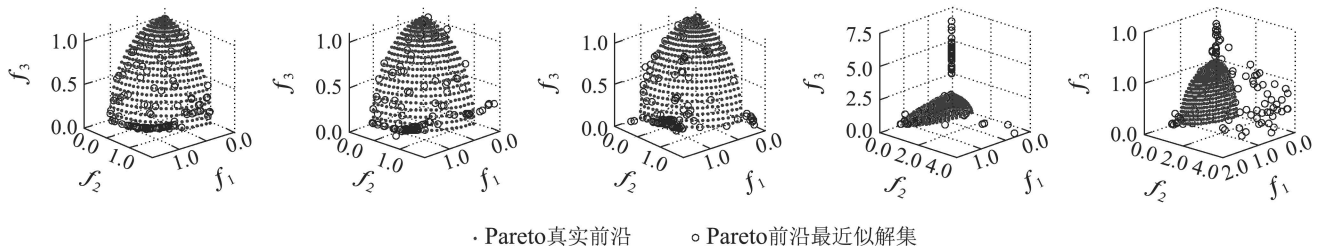
(e) UF5



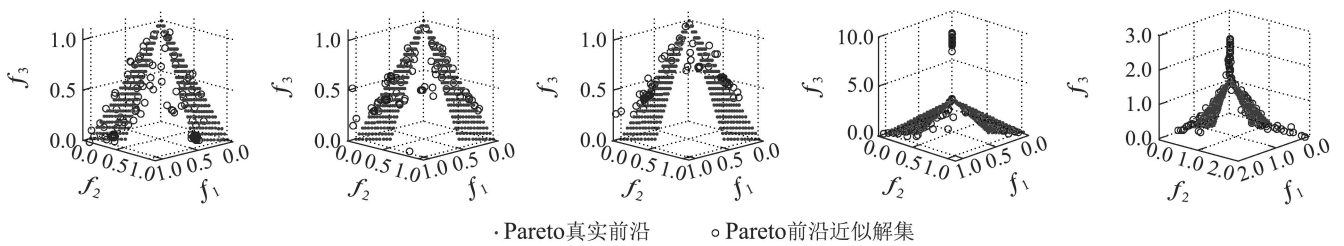
(f) UF6



(g) UF7



(h) UF8



(i) UF9

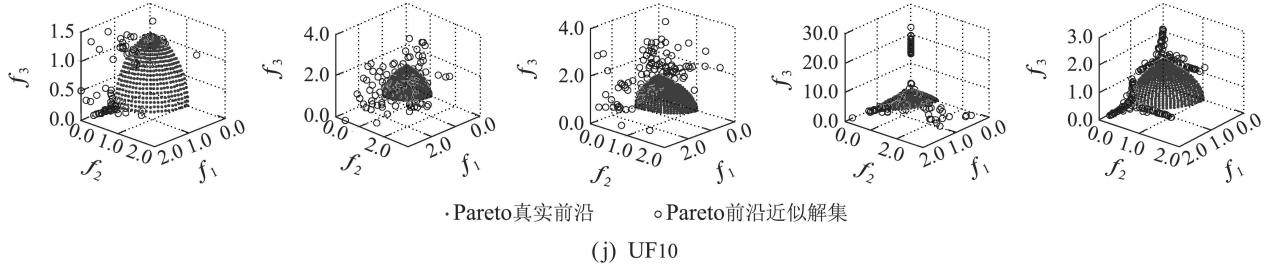
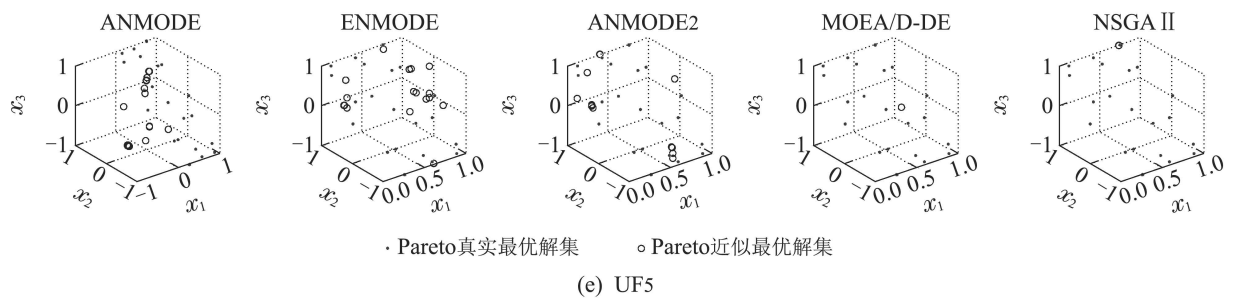
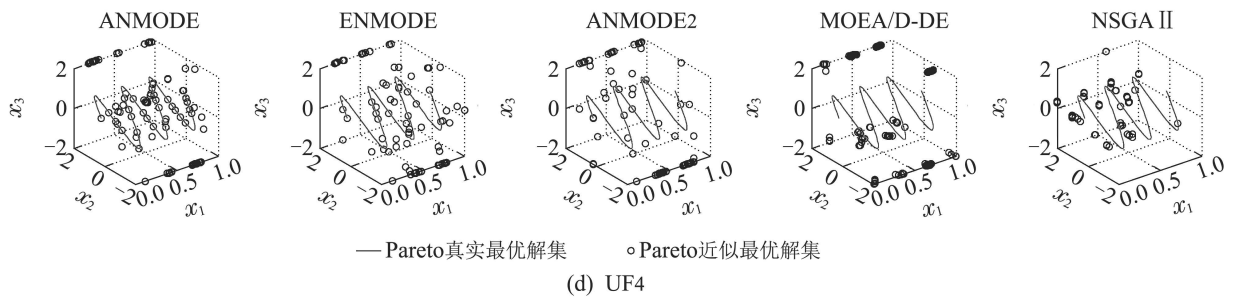
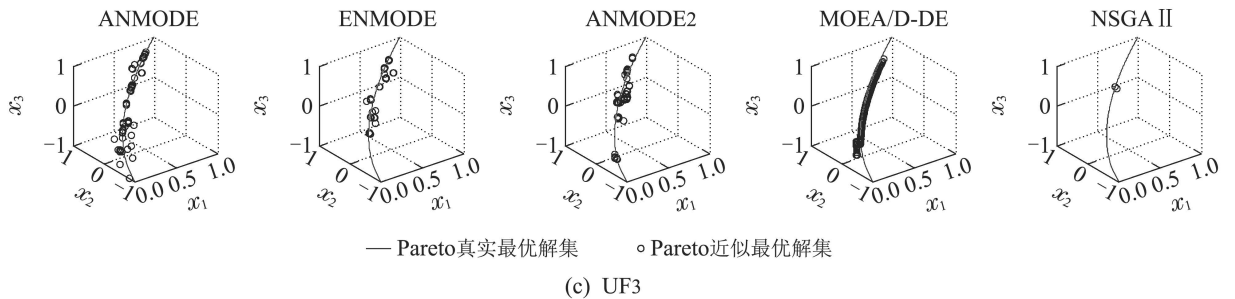
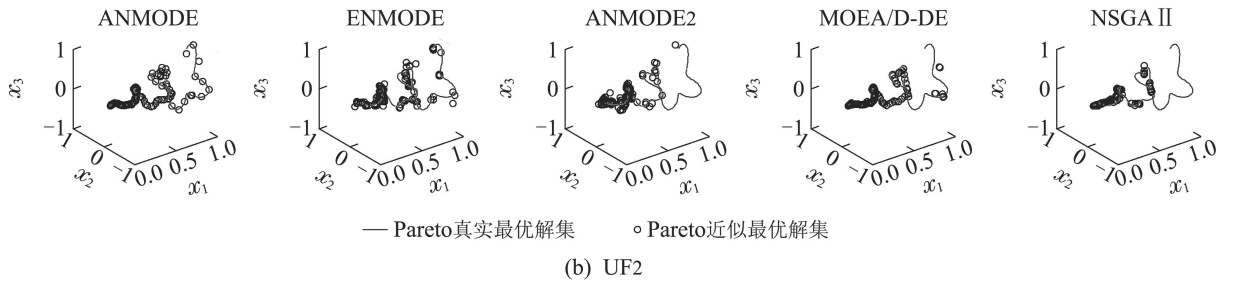
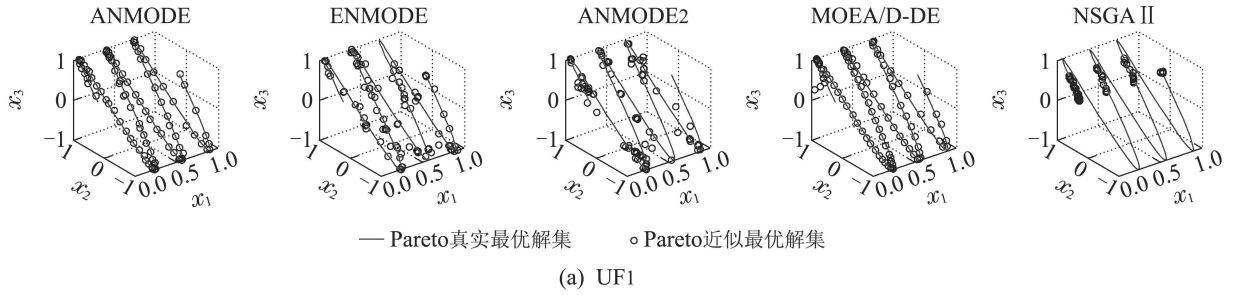


图 3 UF1–UF10的Pareto前沿近似解集
Fig. 3 The approximate set of Pareto front on UF1–UF10



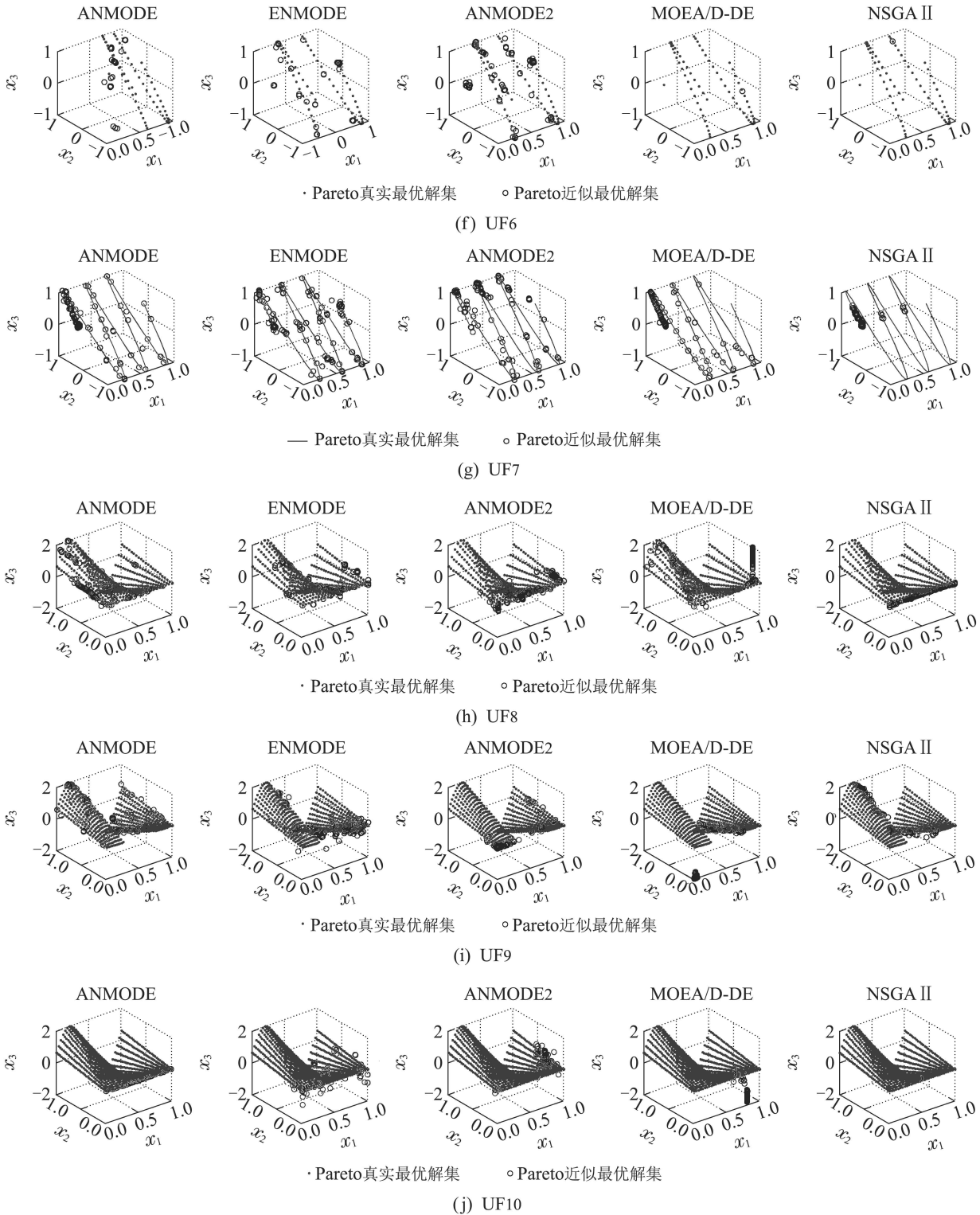


图4 UF1-UF10的Pareto近似最优解集

Fig. 4 The approximate set of Pareto optimal set on UF1-UF10

对于UF1, Pareto前沿近似解集的收敛性和分布性最好的是ANMODE和MOEA/D-DE算法, 其次是ENMODE和ANMODE2算法, 最差的是NSGAI I 算法. 在决策变量空间, ANMODE和MOEA/D-DE算法的Pareto近似最优解集的分布性同样明显优于其他3个算法. 对于UF2, ANMODE算法的Pareto前

沿近似解集的分布性在整个目标函数空间都比较均匀, ENMODE和ANMODE2算法的分布性稍差, 而MOEA/D-DE和NSGAI I 算法在 f_1 函数的 $[0.8, 1]$ 和 $[0.6, 1]$ 上分布性较差. 在决策变量空间, ANMODE算法的Pareto近似最优解集的分布性同样明显优于其他算法. 对于UF3, MOEA/D-DE算法的Pareto前

沿近似解集的收敛性和分布性均明显优于其他算法. NSGAII算法的分布性最差, 其Pareto前沿近似解集几乎集中于一点. 在决策变量空间, Pareto近似最优解集的分布性由好到坏依次为MOEA/D-DE, ANMODE, ENMODE, ANMODE2和NSGAII算法, 而NSGAII算法的Pareto近似最优解集非常集中. 对于UF4, ANMODE, ENMODE和ANMODE2算法的Pareto前沿近似解集的分布性和收敛性明显优于MOEA/D-DE和NSGAII算法. 在决策变量空间, ANMODE, ENMODE和ANMODE2算法的Pareto近似最优解集的一部分出现在Pareto真实最优解集的附近, 一部分出现在了决策变量空间的边界上. 而MOEA/D-DE算法的Pareto近似最优解集大多集中在决策变量空间的边界上. 对于UF5和UF6, 很难定性分析Pareto前沿近似解集的分布性和收敛性. 在决策变量空间, ANMODE, ENMODE和ANMODE2算法的Pareto近似最优解集的分布性要明显好于MOEA/D-DE算法. 而NSGAII算法不管是在目标函数空间还是在决策变量空间, Pareto前沿近似解集和Pareto近似最优解集的分布都过于集中. 对于UF7, ANMODE, ENMODE和MOEA/D-DE算法的Pareto前沿近似解集的收敛性和分布性在整个目标函数空间都比较理想. ANMODE2和NSGAII算法在 f_1 函数的 $[0.4, 0.5]$ 和 $[0.6, 1]$ 上存在分布性较差的情况. 在决策变量空间, ANMODE, ENMODE, ANMODE2和MOEA/D-DE算法的Pareto近似最优解集的分布性相似, 都明显优于NSGAII算法. 对于三目标函数UF8, UF9和UF10, ANMODE算法的Pareto前沿近似解集的收敛性和分布性都明显优于其他算法. 在决策变量空间, ANMODE算法的Pareto近似最优解集的收敛性和分布性也明显好于其他算法. 在UF8, UF9和UF10的 f_3 函数上, MOEA/D-DE算法表现出了较差的收敛能力, 而NSGAII算法在UF8的 f_2 函数上的收敛性较差.

综上, 除UF3外, ANMODE算法的Pareto前沿近似解集和Pareto近似最优解集的收敛性和分布性均好于比较算法, 表现出了很强的求解能力.

3.4 角度邻域的有效性分析(Efficiency analysis of angle neighborhood)

从表1和表2中不难发现, ANMODE算法的所有IGD均值都小于ENMODE算法的, 并且在统计学意义上性能也明显优于ENMODE算法. 图3和图4也定性地表明了: ANMODE算法的Pareto前沿近似解集和Pareto近似最优解集的分布性和收敛性要优于ENMODE算法. 上述实验结果表明通过角度确定邻域的方法比通过欧氏距离确定邻域的方法更能提高算法的性能, Pareto前沿近似解集的收敛性和分布

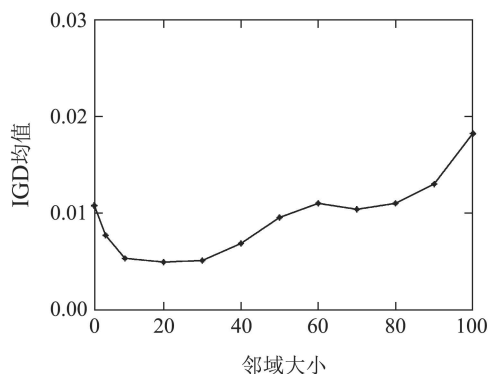
性都得到了改善. 此外, ANMODE算法的各项实验结果均优于ANMODE2算法, 这表明基于角度邻域的变异策略对改善算法性能也起到了一定的作用.

3.5 邻域大小的敏感性分析(Sensitivity analysis of neighborhood size)

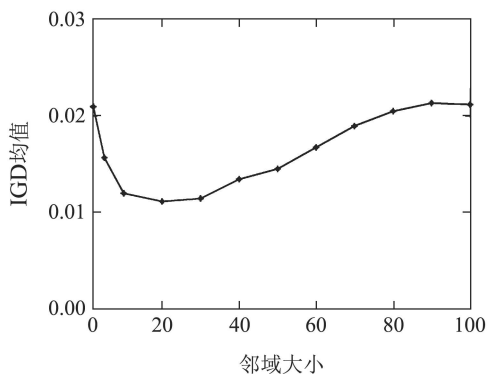
邻域大小 ω 是ANMODE算法的一个重要参数. 为分析邻域大小 ω 对算法性能的影响, 使用不同 ω 的ANMODE算法对UF1和UF8进行优化, 绘制了30次独立运行的IGD均值随 ω 变化的曲线, 如图5所示. 当邻域大小 ω 在10~30之间时, 算法性能最优. 随着 ω 的增加, 邻域越来越接近整个种群规模, 出现了算法性能退化的迹象. 而邻域 ω 非常小时, 算法性能同样表现不佳, 这有可能是邻域过小而导致的算法开发能力不足.

3.6 计算复杂度的分析(Analysis of computational complexity)

文献[8]指出MOEA/D-DE和NSGA-II算法的计算复杂度分别为 $O(mTNP)$ 和 $O(mNP^2)$. ANMODE算法的计算复杂度主要集中在邻域计算上(即步骤4), 整个算法的计算复杂度为 $O(NP^2)$. 显然, 在本文中, ANMODE算法的计算复杂度小于NSGA II算法, 略大于MOEA/D-DE算法. 因此, ANMODE算法性能的改善并没有带来计算代价的显著增加.



(a) UF1



(b) UF8

4 结论(Conclusions)

针对多目标优化问题, 本文提出了一种基于角度邻域的多目标差分进化算法(ANMODE). 在标准DE算法的基础上, 将弱支配概念融入选择操作中实现了由单目标优化算法向多目标优化算法的转变. ANMODE算法根据目标函数空间中个体与权重向量的夹角大小确定个体的邻域, 并在此基础上以较大的概率在邻域内对个体进行变异操作来引导进化方向. 为改善Pareto前沿近似解集的分布性, ANMODE算法将进化过程中的非支配解加入外部存档中, 并根据非支配解与权重向量的夹角对外部存档进行维护. 为验证所提算法的有效性和可行性, 采用CEC2009标准测试函数对其进行数值仿真实验. 实验结果表明ANMODE算法的Pareto前沿近似解集更逼近真实的Pareto前沿, 解集的分布性也更好, 性能明显优于MOEA/D-DE算法和NSGA II算法.

参考文献(References):

- [1] ROGALSKY T, KOCABIYIK S, DERKSEN R. Differential evolution in aerodynamic optimization [J]. *Canadian Aeronautics and Space Journal*, 2000, 46(4): 183 – 190.
- [2] ILONEN J, KAMARAINEN J K, LAMPINEN J. Differential evolution training algorithm for feed-forward neural networks [J]. *Neural Processing Letters*, 2003, 17(1): 93 – 105.
- [3] STORN R. Differential evolution design of an IIR-filter [C] // *Proceedings of IEEE International Conference on Evolutionary Computation*. Nagoya: IEEE, 1996, 5: 268 – 273.
- [4] KUKKONEN S, LAMPINEN J. Comparison of generalized differential evolution algorithm to other multi-objective evolutionary algorithms [C] // *Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering*. Jyvaskyla: Bollettino del CILEA, 2004: 445.
- [5] KUKKONEN S, LAMPINEN J. An extension of generalized differential evolution for multi-objective optimization with constraints [C] // *Proceedings of Parallel Problem Solving from Nature-PPSN VIII*. Berlin: Springer, 2004: 752 – 761.
- [6] KUKKONEN S, LAMPINEN J. GDE3: The third evolution step of generalized differential evolution [C] // *Proceedings of 2005 IEEE Congress on Evolutionary Computation*. Edinburgh: IEEE, 2005: 443 – 450.
- [7] ABBASS H A, SARKER R. The pareto differential evolution algorithm [J]. *International Journal on Artificial Intelligence Tools*, 2002, 11(4): 531 – 552.
- [8] ZHANG Q F, LI H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition [J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712 – 731.
- [9] LI H, ZHANG Q F. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 284 – 302.
- [10] LIU Zhijun, GAO Yakui, ZHANG Weiguo, et al. Decomposition with ensemble neighborhood size multi-objective adaptive differential evolutionary algorithm [J]. *Control Theory & Applications*, 2014, 31(11): 1492 – 1501.
(刘志君, 高亚奎, 章卫国, 等. 变邻域分解多目标自适应差分进化算法 [J]. 控制理论与应用, 2014, 31(11): 1492 – 1501.)
- [11] RAKSHIT P, KONAR A. Extending multi-objective differential evolution for optimization in presence of noise [J]. *Information Sciences*, 2015, 305(1): 56 – 76.
- [12] SHIM V A, TAN K C, TANG H. Adaptive memetic computing for evolutionary multiobjective optimization [J]. *IEEE Transactions on Cybernetics*, 2015, 45(4): 610 – 621.
- [13] WANG Y N, WU L H, YUAN X F. Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure [J]. *Soft Computing*, 2010, 14(3): 193 – 209.
- [14] QIU X, XU J, TAN K C, et al. Adaptive cross-generation differential evolution operators for multi-objective optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(2): 232 – 244.
- [15] ZHANG Q F, ZHOU A M, ZHAO S Z, et al. Multiobjective optimization test instances for the CEC 2009 special session and competition [R]. Singapore: Nanyang Technological University, 2008.
- [16] 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.

作者简介:

赵志伟 (1980–), 男, 博士, 副教授, 目前研究方向为进化算法和轧制规程优化, E-mail: wzzwzz@sina.com;

杨景明 (1957–), 男, 博士, 教授, 博士生导师, 目前研究方向为板形板厚智能控制, E-mail: yangjm6188@sina.com;

呼子宇 (1989–), 男, 博士研究生, 目前研究方向为多目标进化算法, E-mail: whoziyu@163.com;

车海军 (1974–), 男, 博士研究生, 副教授, 目前研究方向为冶金自动化, E-mail: hjche@ysu.edu.cn.