

DOI:10.11918/201905252

# 自适应交配限制概率的自组织多目标演化算法

李桂英<sup>1,2</sup>, 王述洋<sup>1</sup>, 宋申民<sup>3</sup>, 张虎<sup>3</sup>

(1. 东北林业大学 机电工程学院, 哈尔滨 150040; 2. 黑龙江大学 机电工程学院, 哈尔滨 150080;  
3. 哈尔滨工业大学 控制理论与制导技术研究中心, 哈尔滨 150001)

**摘要:** 为平衡多目标演化算法求解不同优化问题以及求解同一优化问题时不同搜索阶段的勘探与开采能力, 并考虑到减小聚类算法辅助演化算法时产生的计算开销, 提出了一种基于自适应交配限制概率的自组织多目标演化算法 (adaptive mating restriction probability based self-organizing multiobjective evolutionary algorithm, ASMEA). 首先, ASMEA 在每一代利用自组织映射 (self-organizing map, SOM) 算法建立了演化种群个体间的邻居关系, 基于此关系有利于算子实施恰当的重组操作, 并在演化算法后期产生优质解, 与此同时, 为了节省利用 SOM 建立当前种群个体之间的邻居关系时引起的计算开销, 将 SOM 与演化算法相融合, 交替地进行 SOM 训练与种群演化. 然后, 运用交配限制概率控制交配父代来源于 SOM 发现的邻居种群或者是整个种群, 以分别加强开采和勘探. 最后, 根据采用不同父代来源的重组在过去一定代数产生后代个体的效用, 自适应地调整算法的交配限制概率. 利用 ASMEA 和 5 种具有代表性的多目标演化算法对标准测试题进行求解, 求解结果表明: ASMEA 在搜索质量、搜索效率以及可视化方面优于其他 5 种算法, 从而验证了 ASMEA 算法对多目标优化问题具有良好的求解性能.

**关键词:** 多目标优化; 演化算法; 聚类算法; SOM; 交配限制概率

中图分类号: TP301 文献标志码: A 文章编号: 0367-6234(2020)12-0105-11

## Adaptive mating restriction probability-based self-organizing multiobjective evolutionary algorithm

LI Guiying<sup>1,2</sup>, WANG Shuyang<sup>1</sup>, SONG Shenmin<sup>3</sup>, ZHANG Hu<sup>3</sup>

(1. College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China; 2. School of Mechanical & Electrical Engineering, Heilongjiang University, Harbin 150080, China; 3. Center for Control Theory and Guidance Technology, Harbin Institute of Technology, Harbin 150001, China)

**Abstract:** To balance the exploration and exploitation in the searching process for different optimization problems and the same optimization problem in different search phases, as well as to reduce the computational cost, an adaptive mating restriction probability-based self-organizing multiobjective evolutionary algorithm (ASMEA) was proposed. Firstly, the self-organizing map (SOM) algorithm was used to establish the neighborhood relationships among the population individuals in ASMEA. The appropriate reproduction based on the above relationship is helpful to produce high quality solutions during the later period of the searching process. To reduce the computational cost of the cluster, the evolutionary algorithm was combined with SOM. At each generation, ASMEA alternately conducts the SOM training step and evolves the population. Secondly, the mating restriction probability was set to control the mating parents selected from both the neighbor population built by SOM and the whole population, so as to strengthen the exploitation and exploration respectively. Lastly, the mating restriction probability was self-adaptively updated in each generation by the utility of generation offspring based on different paternal sources in previous generations. ASMEA and five representative multiobjective evolutionary algorithms were experimented on a number of test instances. Results suggest that ASMEA performed better than the others on search quality, search efficiency, and visual comparison, which verified the ability of ASMEA to solve complicated multiobjective optimization problems.

**Keywords:** multiobjective optimization; evolutionary algorithm; clustering algorithm; self-organizing map; mating restriction probability

收稿日期: 2019-05-31

基金项目: 哈尔滨市青年基金(2017RAQXJ137)

作者简介: 李桂英(1974—), 女, 博士研究生;

王述洋(1958—), 男, 教授, 博士生导师;

宋申民(1968—), 男, 教授, 博士生导师

通信作者: 宋申民, songshenmin@hit.edu.cn

实际工程中存在大量的多目标优化问题 (multiobjective optimization problem, MOP). 带约束的典型 MOP 表达式如下<sup>[1]</sup>:

$$\begin{aligned} \min \mathbf{F}(\mathbf{x}) &= (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T, \\ \text{s. t. } \mathbf{x} &= (x_1, \dots, x_n)^T \in \Omega. \end{aligned}$$

式中:  $\Omega = \prod_{i=1}^n [a_i, b_i] \subseteq \mathbf{R}^n$  为变量空间;  $a_i, b_i$  分别为变量  $x_i$  取值的下边界和上边界;  $\mathbf{x} = (x_1, \dots, x_n)^T$  为  $n$  维决策变量向量;  $\mathbf{F}: \Omega \rightarrow \mathbf{R}^m$  为变量空间向目标空间的映射;  $\mathbf{F}(\mathbf{x})$  为由  $m$  个目标函数  $f_i(\mathbf{x}), i = 1, \dots, m$ , 组成的目标向量.

多目标优化问题不同于单目标优化问题, 其优化的各个目标往往是相互冲突的, 因此对于多目标问题求解有一定的难度. 目前较常用的求解方法是同时优化多个目标, 对相互冲突的目标进行折衷. MOP 存在大量的权衡所有目标的折衷解, 即 Pareto 最优解. 对于决策空间的一个解  $\mathbf{x}^i \in \Omega$ , 如果决策空间其他所有的解都不能支配  $\mathbf{x}^i$ , 则称  $\mathbf{x}^i$  为决策空间  $\Omega$  中的 Pareto 最优解. 所有 Pareto 最优解组成 Pareto 解集 (Pareto set, PS), Pareto 最优解的目标值形成 Pareto 前沿 (Pareto front, PF). 求解多目标优化问题时不可能求出其所有的 Pareto 最优解, 决策者希望搜索到的有限逼近解所对应的目标向量不仅要靠近 PF, 并且要均匀覆盖整个 PF.

为了更好地解决多目标优化问题, 学者们提出了多目标演化算法 (multiobjective evolutionary algorithm, MOEA), 该算法通过单次运算就能获得一系列非支配解, 并且具有应用领域广、并行性好、鲁棒性高等特点. 个体重组与环境选择是 MOEA 重要的组成部分. 就个体重组而言, MOEA 有运用模拟二进制交叉<sup>[2]</sup>、多项式变异<sup>[3]</sup>、差分进化<sup>[4]</sup>等针对单目标优化问题提出的重组算子. 这种算子产生后代有一定的局限性. 考虑到 Pareto 解集具有的规则特性, 学者们提出了基于模型的多目标演化算法. 其典型的代表有 Zhang 等<sup>[5-6]</sup> 在 2008 年提出的基于规则特性的 RM-MEDA 和 2009 年提出的通过建立概率模型同时逼近 PF 和 PS 的 MMEA. 也有学者提出了基于解分布结构的重组算子. 基于环境选择, 近年来的多目标演化算法主要有 3 类: 以 NSGA-II 等<sup>[7]</sup> 为代表的基于支配的多目标演化算法, 以 SMS-EMOA 等<sup>[8]</sup> 为代表的基于指标的多目标演化算法, 以 MOEA/D-DE<sup>[9]</sup> 等为代表的基于分解的多目标演化算法. 目前多目标演化算法的性能可以用不同特性的数值优化问题进行测试<sup>[10]</sup>, 并逐渐被应用于大规模优化问题<sup>[11]</sup>、动态优化问题<sup>[12]</sup> 以及实际工程问题<sup>[13]</sup>.

多目标演化算法在演化过程中应恰当地平衡开采和勘探能力有助于获得分布广泛的代表性逼近解集. 许多研究成果表明, 差异较大个体之间进行重组有利于勘探, 相似个体之间进行重组利用了 Pareto 解集的连通性和规则特性<sup>[14]</sup> 能够增强局部搜索. 目

前, 学者们已经提出了多种交配限制策略来平衡搜索过程中的开采和勘探, 但是这些限制策略往往需要设计额外的、使用者进行调节的控制参数, 或者采用复杂的判断准则来决定勘探或开采<sup>[15-16]</sup>. 因此需要设计一种交配限制概率, 使算法能够简单、自适应地确定所需要的交配限制概率. 此外, 聚类是一种能够挖掘 Pareto 解集分布结构的有效方法<sup>[17]</sup>, 根据聚类识别的种群结构信息利用交配限制概率控制父代来源, 可以平衡搜索过程中的开采和勘探. 自组织映射 (SOM) 算法作为一种无监督学习方法, 其能够对输入的高维训练点产生低维表征<sup>[18]</sup>. 即 SOM 可以保持输入数据间的近邻关系, 从而进行聚类. 受此启发, 本文提出一种基于自适应交配限制概率的自组织多目标演化算法 ASMEA. 在每一代 ASMEA 利用 SOM 聚类算法挖掘当前种群分布结构, 根据此种群结构以一定交配限制概率控制父代来源于邻居或整个种群以分别加强开采和勘探, 根据以整个种群为交配池和以邻居为交配池, 在过去一定代数产生后代个体的效用, 更新算法的交配限制概率.

## 1 自适应自组织多目标演化算法

### 1.1 SOM 聚类算法

SOM 算法能够实现高维数据特征在低维空间进行表示, 并保持拓扑结构不变. SOM 首次训练时每个神经元的初始特征向量从训练点集合中随机挑选, 随后特征向量的迭代更新由距离这些特征向量最近的训练数据点完成, 通过训练输入数据找到每个神经元的特征向量, 从而进行特征识别. 在训练中, SOM 根据更新获胜神经元及邻近神经元的特征向量, 将输入的训练数据映射到输出层的神经元上, 并保持其拓扑结构不变, 根据此结构对输入数据进行聚类划分. SOM 的具体算法见文献<sup>[18]</sup>.

### 1.2 ASMEA 算法框架

算法 1 给出了 ASMEA 的算法框架. 在 ASMEA 的每一代, SOM 首先添加一些新的训练数据到  $S$  中 (第 1 行), 随后 ASMEA 演化了  $T$  代 (第 2~24 行). 更新训练参数 (第 3~8 行). 根据学习结果实现每个个体关联一个神经元 (第 9~14 行). 以概率  $\beta$  为个体  $\mathbf{x}^u$  设置交配池由邻居种群组成, 以  $1 - \beta$  的概率设置交配池由整个种群组成 (第 17 行). 基于交配池  $Q$ , 利用重组算子为  $\mathbf{x}^u$  产生一个新解  $\mathbf{y}$  (第 18 行). 通过环境选择更新种群  $\mathcal{A}$  (第 19 行). 更新训练数据集  $S$ , 训练数据集由新添加到种群中的个体组成 (第 21 行). 基于重组效用更新交配限制概率  $\beta$  (第 22 行). 在 ASMEA 算法中, 训练数据是每一代中新产生的个体. 如果采用一个新的数据集, 并不会

用 SOM 重新进行学习, 而是使用先前训练过的数据结构信息, 这样减少了反复训练大量数据所带来的计算开销。

### 1.3 新解生成

ASMEA 利用差分进化 (differential evolution, DE) 算子和多项式变异 (polynomial mutation, PM) 算子生成新解。新解产生的具体步骤见算法 2。以配池  $Q$  作为父代来源, 从  $Q$  中随机地挑选两个个体  $p^1$  和  $p^2$  为父代个体 (第 1 行), 然后利用差分进化算子产生后代基因 (第 2 行), 并利用多项式变异算子进一步演化后代基因 (第 4 行), 为确保新解可行性, 对差分进化生成的新解和变异操作的新解分别进行修补 (第 3、5 行)。在挑选父个体时, 如果交配池为当前解的邻居且邻居数目小于 2, 则从整个种群中选择父个体。

### 1.4 环境选择

环境选择可以在每一代新解  $y$  产生后从  $\mathcal{P}'$  挑选出优秀的解作为后代演化的父代种群。ASMEA 采用快速非支配排序和超体积环境选择相结合的环境选择方法。这种环境选择方法促使种群保持收敛性和多样性。具体算法见算法 3。

ASMEA 利用快速非支配排序法衡量  $\mathcal{P}'$  中个体的收敛性, Pareto 等级最差的个体收敛性最差, 最先被舍弃。按照 Pareto 等级划分  $\mathcal{P}'$  中个体到  $L$  个不同的非支配前沿  $\{B^1, \dots, B^L\}$  中。其中  $B^1$  为最优等级,  $B^L$  为最差等级。

**算法 1** ASMEA 算法框架。

**Input:**  $N$  为种群大小;  $\tau_0$  为初始 SOM 学习率;  $\sigma_0$  为初始邻居半径;  $H$  为邻居交配池大小;  $\beta$  为交配限制概率;  $HL$  为历史时长。

**Output:** 优化种群  $\mathcal{P}$ 。

1) 初始化种群  $\mathcal{P} = \{x^1, x^2, \dots, x^N\}$ , 初始训练数据集  $S = \mathcal{P}$ , 初始化神经元权向量集合  $\mathcal{P} = \{w^1, \dots, w^N\}$ 。

2) **for**  $t = 1, \dots, T$  **do**

3) **for each**  $x^s \in S, s = 1, \dots, |s|$  **do**

4) 更新训练参数:

$$\sigma = \sigma_0 \times \left(1 - \frac{(t-1)N + s}{TN}\right), \tau = \tau_0 \times \left(1 - \frac{(t-1)N + s}{TN}\right).$$

5) 找到距  $x^s$  最近神经元:  $u' = \arg \min_{1 \leq u \leq N} \|x^s - w^u\|_2$ 。

6) 确定邻近神经元:

$$U = \{u \mid 1 \leq u \leq N \wedge \|z^u - z^{u'}\|_2 < \sigma\}.$$

7) 更新所有邻近神经元的权向量 ( $u \in U$ ):

$$w^u + \tau \cdot \exp(-\|z^u - z^{u'}\|_2) (x - w^u).$$

8) **end**

9) 设置  $A = \mathcal{P}$  以及  $\mathcal{U} = \{1, \dots, N\}$ 。

10) **while**  $A \neq \Phi$  **do**

11) 随机选择  $x \in A$  且设置  $A = A \setminus \{x\}$ 。

12) 设置  $x^u = x$ , 其中  $u = \arg \min_{u' \in U} \|x - w^{u'}\|_2$ 。

13) 设置  $\mathcal{U} = \mathcal{U} \setminus \{u\}$ 。

14) **end**

15) 设置  $A = \mathcal{P}$ 。

16) **for each**  $u \in \{1, \dots, N\}$  **do**

17) 为  $x^u$  设置交配池:

$$Q = \begin{cases} \bigcup_{k=1}^H \{x^{u^k}\}, & \text{如果 } \text{rand} < \beta; \\ \mathcal{P}, & \text{其他.} \end{cases}$$

rand 为在  $[0, 1]$  之间产生一个均匀分布的随机数,  $u^k$  是在隐空间中距离神经元  $u$  的第  $k$  个最近神经元。

18) 产生一个新个体  $y = \text{SolGen}(Q, x^i, F, CR, p_m, \eta_m)$ 。

19) 通过环境选择, 保存优良解  $\mathcal{P} = \text{EnvSel}(\mathcal{P}, y)$ 。

20) **end**

21) 更新训练数据集:  $S = \mathcal{P} \cup A$ 。

22) 基于重组效用更新交配限制概率  $\beta$ 。

23) **end**

24) **return** 种群  $\mathcal{P}$ 。

**算法 2** SolGen( $Q, x^i, F, CR, p_m, \eta_m$ ) 算法框架。

**Input:** 当前演化个体  $x^i$ ; 交配池  $Q$ ; 差分进化算子参数  $F, CR$ ; 多项式变异概率  $p_m$ ; 变异分布指数  $\eta_m$ 。

**Output:** 新解  $y = (y_1, \dots, y_n)^T$ 。

1) 从  $Q$  中挑选两个父个体  $p^1$  和  $p^2$ 。

2) 产生一个新解  $y' = (y'_1, \dots, y'_n)^T$ 。

$$y'_j = \begin{cases} x_j^i + F \times (p_j^1 - p_j^2), & \text{如果 } \text{rand}() < CR; \\ x_j^i, & \text{其他} \end{cases} \quad (j=1, \dots, n).$$

3) 修补新解:

$$y''_j = \begin{cases} a_j^i, & \text{如果 } y'_j < a_j^i; \\ b_j^i, & \text{如果 } y'_j > b_j^i; \\ y'_j, & \text{其他.} \end{cases}$$

其中  $a_j^i, b_j^i$  分别为第  $j$  维变量的下、上边界。

4) 变异新解:

$$y'''_j = \begin{cases} y''_j + \delta_j \times (b_j^i - a_j^i), & \text{如果 } \text{rand}() < p_m; \\ y''_j, & \text{其他,} \end{cases} \quad (j=1, \dots, n)$$

$$\text{且 } \delta_j = \begin{cases} \left[ 2r + (1-2r) \left( \frac{b_j^i - y''_j}{b_j^i - a_j^i} \right)^{\frac{1}{\eta_m+1}} \right]^{\frac{1}{\eta_m+1}} - 1, & \text{如果 } r < 0.5; \\ 1 - \left[ 2-2r + (2r-1) \left( \frac{y''_j - a_j^i}{b_j^i - a_j^i} \right)^{\frac{1}{\eta_m+1}} \right]^{\frac{1}{\eta_m+1}}, & \text{其他.} \end{cases}$$

$$5) \text{ 修补新解: } y_j = \begin{cases} a_j^i, & \text{如果 } y^m < a_j^i; \\ b_j^i, & \text{如果 } y^m > b_j^i; (j=1, \dots, n) \\ y^m, & \text{其他.} \end{cases}$$

**算法 3** 环境选择算法框架.

**Input:** 新解  $y$ , 种群  $\mathcal{P}$ .

**Output:** 优化种群  $\mathcal{P}$ .

- 1) 新解  $y$  添加到种群  $\mathcal{P}$ , 辅助种群  $\mathcal{P}'$ , 用快速非支配排序衡量  $\mathcal{P}$  中个体的收敛性, 划分为  $L$  个等级  $\{B^1, \dots, B^L\}$ .
- 2) 如果  $l > 1$  时
  - ① 计算  $\mathcal{P}$  中支配每个个体  $x^*$  的个体数目  $d(x, \mathcal{P})$ .
  - ② 移除  $\mathcal{P}$  中具有最大  $d(x, \mathcal{P}')$  的个体  $x^*$ .
- 3) 如果  $l = 1$ 
  - ① 计算  $\mathcal{P}'$  中每个个体  $x^*$  的超体积贡献度  $\Delta_\phi(x, \mathcal{P}')$ .
  - ② 移除  $\mathcal{P}'$  中超体积贡献度最小的个体  $x^*$ .
- 4) 将  $\mathcal{P}'$  赋给  $\mathcal{P}, \mathcal{P} = \mathcal{P}'$ .
- 5) 输出  $\mathcal{P}$ .

当  $L > 1$  时,  $\mathcal{P}'$  中包含多个等级, 则需要移除最劣等级中具有最大  $d(x, \mathcal{P}')$  的解, 其中  $d(x, \mathcal{P}')$  表示在  $\mathcal{P}'$  中支配  $x$  的点的数量. 当  $L = 1$  时,  $\mathcal{P}'$  中只包含一个等级, 删除具有最小超体积贡献度  $\Delta_\phi(x, B^1)$  的个体, 计算超体积贡献度  $\Delta_\phi$  的具体方法见文献[19].

### 1.5 交配限制概率更新

在 ASMEA 中, 用于产生新个体的父个体来源于邻居的优点是利于开采, 缺点是在进化的早期阶段, 容易丢失种群多样性. 产生新个体的父个体来源于整个种群的优点是可以提高算法的勘探能力, 产生的后代能够具有很好的多样性, 但在演化后期产生高质量解的能力较弱. 所以需要在演化过程中恰当地控制父代来源. 此外, 实际工程中为了更有效地求解多目标优化问题, 不同的优化问题需要设置不同的  $\beta$  值控制父代来源以平衡全局搜索与局部搜索, 在演化过程的前期和后期也需要不同的  $\beta$  值. 设置一个合适的  $\beta$  值对 ASMEA 是合理且有效的. 因此, ASMEA 基于聚类设计一种交配限制策略控制父个体来源, 从而平衡算法求解过程中的勘探和开采.  $\beta$  在演化过程中自动更新方法如下, 在第  $t+1$  代:

$$\beta_{t+1} = \frac{u_t^{\text{gsp}} + \varepsilon}{u_t^{\text{gsp}} + u_t^{\text{clu}} + \varepsilon},$$

式中,  $\varepsilon = 10^{-10}$  保证计算表达式的合理性.  $u_t^{\text{clu}}$  是在第  $t$  代由邻居产生新个体的重组效用, 其计算公式为

$$u_t^{\text{clu}(\text{gsp})} = \begin{cases} \frac{\sum_{k=t-HL+1}^t RF_k^{\text{clu}(\text{gsp})}}{\sum_{k=t-HL+1}^t GN_k^{\text{clu}(\text{gsp})}}, & t > HL; \\ \frac{\sum_{k=1}^t RF_k^{\text{clu}(\text{gsp})}}{\sum_{k=1}^t GN_k^{\text{clu}(\text{gsp})}}, & t < HL. \end{cases}$$

式中:  $GN_k^{\text{clu}}$  为在第  $t$  代父代来源于邻居种群所产生的新个体的数目;  $RF_k^{\text{clu}}$  为在第  $t$  代父代来源于邻居种群中产生的所有新解的粗糙适应度之和. 同理可以求解出在前  $HL$  代, 利用整个种群产生新个体的效用  $u_t^{\text{gsp}}$ . 根据  $\beta$  的计算表达式可知, 为了避免  $u_t^{\text{clu}}$  过大或者过小时导致算法仅从整个种群或者邻居产生新个体, ASMEA 在演化过程中每一代会有两次重组分别以整个种群和邻居种群为父代来源.

### 1.6 计算复杂度分析

ASMEA 中 SOM 算法和环境选择算子的计算复杂度高于重组算子与交配限制概率更新算子, 因此主要考虑 SOM 算法和环境选择算子的计算复杂度. ASMEA 运行 SOM 算法的计算复杂度为  $O(nND)$ , 其中  $N$  为种群大小,  $n$  为变量维度,  $D$  为神经元数. 环境选择算子结合了快速非支配排序和超体积, 前者的复杂度为  $O(mN^2)$ , 后者为  $O(N^m)$ . 其中,  $m$  为优化目标个数. 因此, 环境选择算子的时间复杂度为  $O(mN^2 + N^m)$ . 综上所述, ASMEA 算法的计算复杂度为  $O(nND + mN^2 + N^m)$ .

## 2 分 析

### 2.1 标准测试实例及性能指标

为了验证 ASMEA 的性能, 本文选取具有复杂 Pareto 前沿结构和 Pareto 解集的多目标优化问题 GLT1-GLT6<sup>[20]</sup> 对 ASMEA 算法进行测试. 选用反转世代距离 IGD<sup>[21]</sup> 和超体积 HV<sup>[22]</sup> 两个常用性能指标, 可以同时有效地评估解的收敛性和多样性. 并且 IGD 指标值越小逼近前沿的收敛性和多样性越好, HV 指标值越大, 表明算法搜索到的逼近前沿越好.

使用 HV 指标值评估算法性能时需要设定恰当的参考点, 本文求解 GLT1-GLT6 测试题, 在计算 HV 指标值时设置参考点分别为  $r = (2, 2)^T$ ,  $r = (2, 11)^T$ ,  $r = (2, 2)^T$ ,  $r = (2, 3)^T$ ,  $r = (2, 2, 2)^T$  和  $r = (2, 2, 2)^T$ .

### 2.2 参数设置

为了检验 ASMEA 的搜索性能, 本文选择了 5 种具有代表性的多目标演化算法与其进行对比实验.

对比算法分别是: NSGA-II、SMS-EMOA、RM-MEDA、MOEA/D-DE 和 TMOEA/D. 本文基于原始文献并通过多次实验对算法中的主要参数进行了调节优化. 各算法具体参数设置见表 1.

表 1 求解 GLT 测试题的 NSGA-II、SMS-EMOA、RM-MEDA、MOEA/D-DE、TMOEA/D 及 ASMEA 的参数值

Tab. 1 Parameter values for NSGA - II, SMS-EMOA, RM-MEDA, MOEA/D-DE, TMOEA/D, and ASMEA on GLT test instances

算法名称	参数设置
公共参数	种群大小: $N = 100 (m = 2), 7 * 15 (m = 3)$ ; 变量维度: $n = 10$ ; 最大演化代数: $T = 300$ .
NSGA-II	DE 算子控制参数 $F = 0.5, CR = 1$ ; PM 算子控制参数 $p_m = 1/n, \eta_m = 20$ .
SMS-EMOA	DE 算子控制参数 $F = 0.5, CR = 1$ ; PM 算子控制参数 $p_m = 1/n, \eta_m = 20$ .
RM-MEDA	局部主成分分析中聚类个数为 5; 扩展采样率为 0.25. 邻居大小 $NS = 5$ ;
MOEA/D-DE	父代从邻居中挑选的概率 $\beta = 0.7$ ; DE 算子控制参数 $F = 0.9, CR = 0.6$ ; PM 算子控制参数 $p_m = 1/n, \eta_m = 20$ .
TMOEA/D	邻居大小 $NS = 30$ ; 第 1 阶段演化代数 $T1 = T/10$ ; 第 2 阶段演化代数 $T2 = \alpha T$ ; $\alpha = \{0.01, 0.02, \dots, 0.10, 0.10, 0.10, 0.15\}$ ; DE 算子控制参数 $F = 0.5, CR = 1$ ; SOM 结构: $1 * 100 (m = 2), 7 * 15 (m = 3)$ ; 邻居交配池大小 $H = 5$ ; 历史时长 $HL = 15$ ;
ASMEA	初始学习率 $\tau_0 = 0.7$ ; 交配控制概率初始值 $\beta_0 = 0.5$ ; DE 算子控制参数 $F = 0.5, CR = 1$ ; PM 算子控制参数 $p_m = 1/n, \eta_m = 20$ .

表 1 中各主要参数调节方法为: 1) 选择一个被调节的参数, 根据经验把其他参数设置为固定值; 2) 调节当前被调节参数的参数值, 使算法性能最优的值被赋给当前参数. 反复调节直到任何参数的变化都无法显著提高算法的整体性能. ASMEA 中的控制参数  $F, CR, H$  和  $HL$  按照上述方法进行调节, SOM 结构是根据种群大小确定不需要调节. 根据参数灵敏度分析, 算法性能对  $\tau_0 = 0.7$  和  $\beta_0 = 0.5$  并

不敏感, 可以按照经验估计, 亦可用上述方法进行调节.

## 2.3 对比实验结果

### 2.3.1 搜索质量

分别由算法 NSGA-II、SMS-EMOA、RM-MEDA、TMOEA/D、MOEA/D-DE 以及 ASMEA 在 GLT 测试集上获得的 23 个逼近前沿的 IGD 和 HV 的均值和标准差结果见表 2. 6 种算法获得的平均 IGD 指标值按照升序, 平均 HV 指标值按照降序进行排序, 并将排序结果置于表 2 的方括号中. 在 ASMEA 和每个对比算法之间采用了显著性水平 5% 的 Wilcoxon 秩和检验. 表中“†\*”、“§”、“≈”3 个符号分别表示 ASMEA 的性能在 5% 显著水平上优于、差于和类似于比较算法. 每种算法求解所有测试题获得的平均秩见表 2.

从表 2 可以看出, 根据 5% 显著性水平的 Wilcoxon 秩和检验, ASMEA 与其他 5 种对比算法相比, 分别获得了 12、12、12、10 和 11 个显著更优的平均性能评估指标值. 按照综合排名将 6 种算法性能从优到劣的排序为 ASMEA、TMOEA/D、SMS-EMOA、RM-MEDA、NSGA-II 及 MOEA/D-DE. 虽然 TMOEA/D 在求解 GLT2 及 GLT1 时稍优于 ASMEA, 但是在对 6 道测试题的求解过程中及与 5 种对比算法的 12 次 Wilcoxon 秩和检验中, 根据最佳平均指标值及平均秩的结果, ASMEA 对于 GLT 测试集求解性能最佳.

进一步利用 box-plot 描述算法的统计性能, 6 种算法对 GLT 测试集独立运行 23 次获得的 IGD 指标值的箱线图如图 1 所示, 从图 1 中可看出 ASMEA 在求解 GLT1、GLT3-GLT6 时获得了最小的中位 IGD 指标值和四分位距, 在求解 GLT2 时获得第 2 小的中位 IGD 指标值. 结果说明 ASMEA 对于 GLT 能稳定地求解出具有良好多样性和收敛性的解.

### 2.3.2 搜索效率

从搜索效率的角度比较 ASMEA 与 5 种对比算法. 图 2 绘制出了 ASMEA、TMOEA/D、SMS-EMOA、RM-MEDA、NSGA-II 及 MOEA/D-DE 获得的 IGD 的均值和标准差的演化曲线. 图 2 中显示, 对于 GLT1、GLT3-GLT6, ASMEA 能以最高的搜索效率获得最小的 IGD 指标值, 对于 GLT2, ASMEA 与 TMOEA/D 获得了相似的 IGD 值. ASMEA 总体上能以最快搜索效率获得最小的 IGD 值.

表 2 NSGA-II、SMS-EMOA、RM-MEDA、TMOEA/D、MOEA/D-DE 以及 ASMEA 在 GLT 测试题上获得的统计结果

Tab. 2 Statistical results obtained by NSGA-II, SMS-EMOA, RM-MEDA, TMOEA/D, MOEA/D-DE, and ASMEA on GLT test instances

测试题	IGD					
	NSGA-II	SMS-EMOA	RM-MEDA	TMOEA/D	MOEA/D-DE	ASMEA
GLT1	$1.940 \times 10^{-2\dagger}$ $1.60 \times 10^{-2}[6]$	$1.850 \times 10^{-2\dagger}$ $9.20 \times 10^{-3}[5]$	$1.680 \times 10^{-2\dagger}$ $2.0 \times 10^{-2}[4]$	$4.700 \times 10^{-3\dagger}$ $6.92 \times 10^{-4}[2]$	$6.600 \times 10^{-3\dagger}$ $6.66 \times 10^{-4}[3]$	$3.500 \times 10^{-3}$ $6.80 \times 10^{-3}[1]$
GLT 2	$4.240 \times 10^{-2\dagger}$ $3.40 \times 10^{-2}[3]$	$4.350 \times 10^{-2\dagger}$ $3.10 \times 10^{-3}[4]$	$4.710 \times 10^{-2\dagger}$ $6.50 \times 10^{-2}[5]$	$3.540 \times 10^{-2}\S$ $6.51 \times 10^{-3}[1]$	$3.843 \times 10^{-1\dagger}$ $5.76 \times 10^{-2}[6]$	$3.660 \times 10^{-2}$ $1.70 \times 10^{-3}[2]$
GLT 3	$1.290 \times 10^{-2\dagger}$ $6.30 \times 10^{-3}[2]$	$2.510 \times 10^{-2\dagger}$ $1.10 \times 10^{-2}[4]$	$1.930 \times 10^{-2\dagger}$ $1.26 \times 10^{-2}[3]$	$3.350 \times 10^{-2\dagger}$ $6.51 \times 10^{-2}[5]$	$4.080 \times 10^{-2\dagger}$ $1.06 \times 10^{-2}[6]$	$1.030 \times 10^{-2}$ $8.50 \times 10^{-3}[1]$
GLT 4	$4.600 \times 10^{-2\dagger}$ $5.29 \times 10^{-2}[5]$	$6.440 \times 10^{-2\dagger}$ $7.10 \times 10^{-2}[6]$	$4.410 \times 10^{-2\dagger}$ $4.29 \times 10^{-2}[4]$	$2.080 \times 10^{-2\dagger}$ $4.74 \times 10^{-2}[2]$	$2.930 \times 10^{-2\dagger}$ $4.40 \times 10^{-2}[3]$	$9.700 \times 10^{-3}$ $1.58 \times 10^{-2}[1]$
GLT 5	$6.250 \times 10^{-2\dagger}$ $3.20 \times 10^{-3}[5]$	$3.020 \times 10^{-2\dagger}$ $4.93 \times 10^{-4}[2]$	$5.050 \times 10^{-2\dagger}$ $1.30 \times 10^{-3}[4]$	$4.470 \times 10^{-2\dagger}$ $1.40 \times 10^{-3}[3]$	$8.140 \times 10^{-2\dagger}$ $3.30 \times 10^{-3}[6]$	$2.910 \times 10^{-2}$ $2.70 \times 10^{-4}[1]$
GLT 6	$5.380 \times 10^{-2\dagger}$ $3.30 \times 10^{-3}[6]$	$2.250 \times 10^{-2\dagger}$ $4.37 \times 10^{-4}[2]$	$3.810 \times 10^{-2\dagger}$ $1.70 \times 10^{-3}[4]$	$3.270 \times 10^{-2\dagger}$ $1.10 \times 10^{-3}[3]$	$5.200 \times 10^{-2\dagger}$ $1.40 \times 10^{-3}[5]$	$2.170 \times 10^{-2}$ $3.72 \times 10^{-4}[1]$

测试题	HV					
	NSGA-II	SMS-EMOA	RM-MEDA	TMOEA/D	MOEA/D-DE	ASMEA
GLT 1	$3.302^\dagger$ $2.97 \times 10^{-2}[5]$	$3.296^\dagger$ $2.84 \times 10^{-2}[6]$	$3.313^\dagger$ $3.94 \times 10^{-2}[4]$	$3.693^\S$ $1.80 \times 10^{-3}[1]$	$3.365^\S$ $7.40 \times 10^{-3}[2]$	$3.357$ $1.45 \times 10^{-2}[3]$
GLT 2	$1.972 \times 10^{1\dagger}$ $8.30 \times 10^{-3}[4]$	$1.975 \times 10^{1\dagger}$ $1.17 \times 10^{-2}[3]$	$1.966 \times 10^{1\dagger}$ $2.14 \times 10^{-1}[5]$	$1.977 \times 10^{1\dagger}$ $1.59 \times 10^{-2}[2]$	$1.944 \times 10^{1\dagger}$ $4.81 \times 10^{-2}[6]$	$1.980 \times 10^1$ $2.40 \times 10^{-3}[1]$
GLT 3	$3.946^\dagger$ $1.20 \times 10^{-3}[2]$	$3.944^\dagger$ $2.10 \times 10^{-3}[4]$	$3.945^\dagger$ $2.10 \times 10^{-3}[3]$	$3.940^\dagger$ $2.25 \times 10^{-2}[6]$	$3.941^\dagger$ $3.23 \times 10^{-3}[5]$	$3.948$ $1.90 \times 10^{-3}[1]$
GLT 4	$4.957^\dagger$ $3.08 \times 10^{-2}[5]$	$4.895^\dagger$ $1.54 \times 10^{-1}[6]$	$4.964^\dagger$ $2.60 \times 10^{-2}[2]$	$4.962^\dagger$ $1.14 \times 10^{-1}[3]$	$4.959^\dagger$ $1.02 \times 10^{-1}[4]$	$4.989$ $8.60 \times 10^{-3}[1]$
GLT 5	$7.940^\dagger$ $2.60 \times 10^{-3}[5]$	$7.968^\dagger$ $2.22 \times 10^{-4}[2]$	$7.952^\dagger$ $1.20 \times 10^{-3}[4]$	$7.958^\dagger$ $7.90 \times 10^{-4}[3]$	$7.940^\dagger$ $1.60 \times 10^{-3}[6]$	$7.969$ $6.99 \times 10^{-5}[1]$
GLT 6	$7.933^\dagger$ $2.90 \times 10^{-3}[6]$	$7.960^\dagger$ $8.13 \times 10^{-4}[2]$	$7.948^\dagger$ $1.30 \times 10^{-3}[4]$	$7.952^\dagger$ $1.00 \times 10^{-3}[3]$	$7.940^\dagger$ $1.10 \times 10^{-3}[5]$	$7.962$ $2.29 \times 10^{-4}[1]$

注:1)NSGA-II 的平均秩为 4.500,SMS-EMOA 的平均秩为 3.833,RM-MEDA 的平均秩为 3.833,TMOEA/D 的平均秩为 2.833,MOEA/D-DE 的平均秩为 4.750,ASMEA 的平均秩为 1.250;2)NSGA-II 的 †/§/≈ 为 12/0/0,SMS-EMOA 的 †/§/≈ 为 12/0/0,RM-MEDA 的 †/§/≈ 为 12/0/0,TMOEA/D 的 †/§/≈ 为 10/2/0,MOEA/D-DE 的 †/§/≈ 为 11/1/0.

### 2.3.3 可视化比较

从表 2 中可知 ASMEA 和 TMOEA/D 是表现最好的两种算法,将 TMOEA/D 和 ASMEA 进行单独的可视化对比实验.图 3 绘制出 TMOEA/D 和 ASMEA 求解 GLT1-GLT6 时独立运算 23 次获得的全部逼近前沿和代表性逼近前沿.从图 3(a)、图 3(b)可以看出,对每道测试题 ASMEA 获得的全部逼近前沿都能够很好地覆盖整个的 Pareto 前沿并收敛,

TMOEA/D 获得的逼近前沿没有覆盖 GLT5-GLT6 整个的 Pareto 前沿,且无法全部收敛到 GLT4-GLT6 的 Pareto 前沿.从图 3(c)、图 3(d)可以看出,对于 GLT1-GLT2 以及 GLT4,TMOEA/D 和 ASMEA 获得代表性逼近前沿能收敛并且均匀覆盖整个 Pareto 前沿.对于 GLT3,ASMEA 找到更多的 Pareto 解. ASMEA 对于 GLT5-GLT6 获得的代表性逼近前沿比 TMOEA/D 具有更好的多样性.

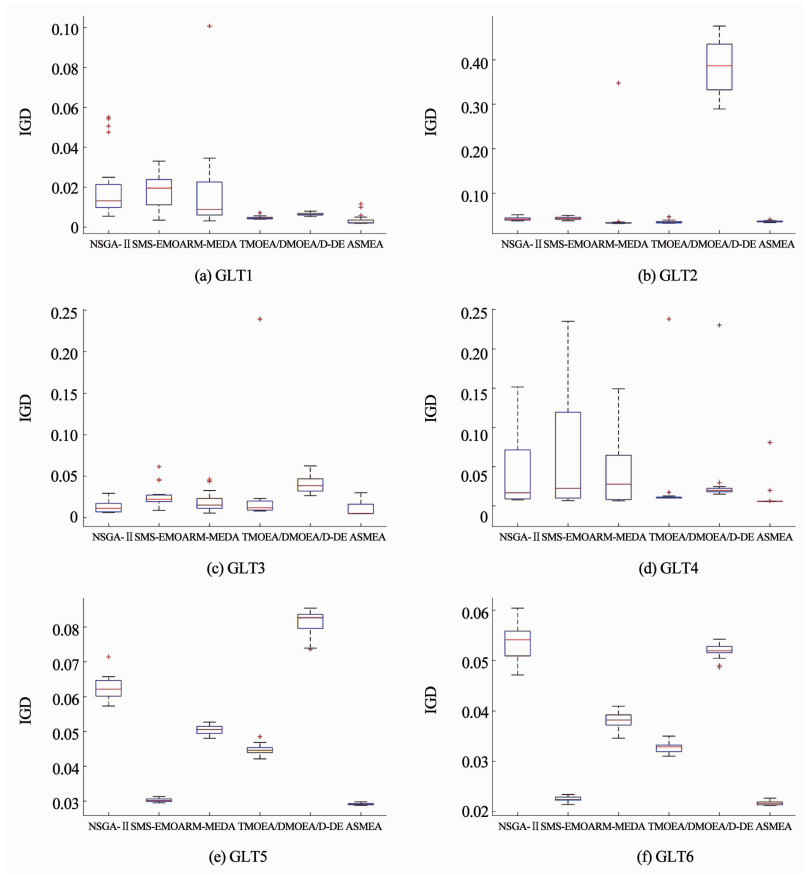


图 1 IGD 指标值箱线图

Fig. 1 IGD indicator box diagram

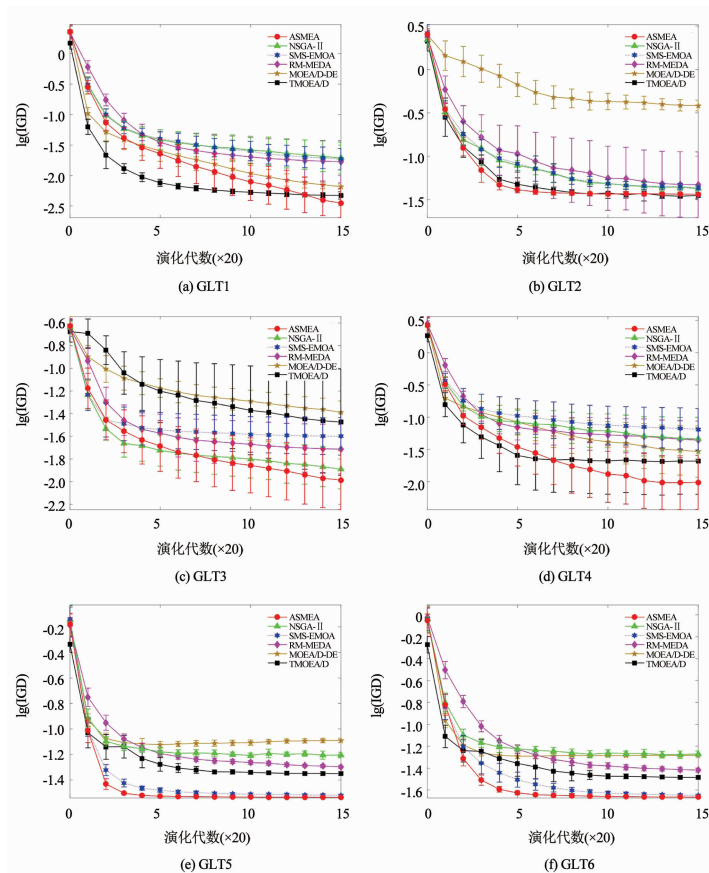


图 2 平均 IGD 值的平均值和标准差演化曲线

Fig. 2 Evolution curves of mean IGD values and corresponding standard deviations

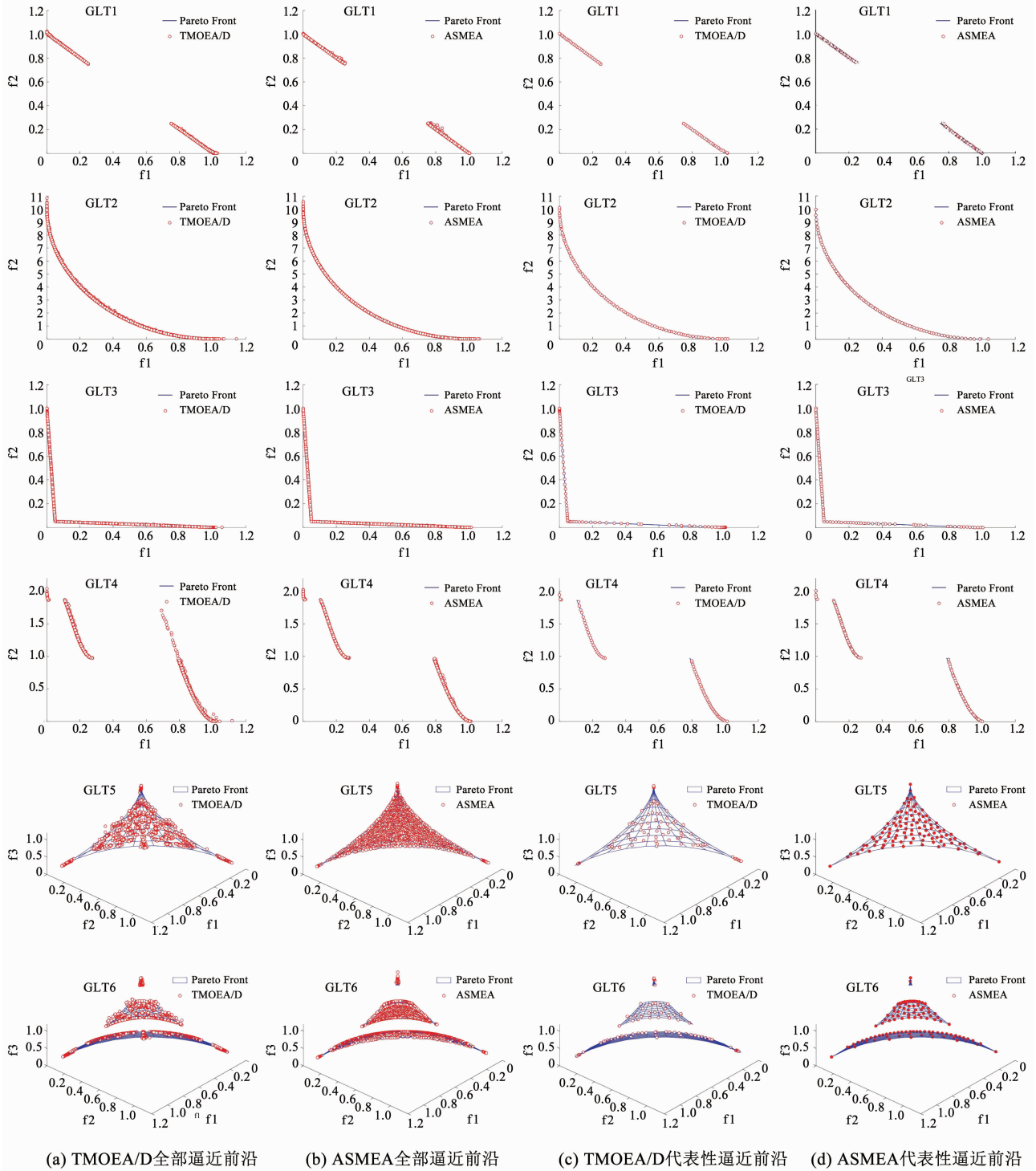


图 3 TMOEA/D 和 ASMEA 获得的全部逼近前沿和代表性逼近前沿

Fig. 3 All approximated fronts and representative approximated fronts obtained by TMOEA/D and ASMEA on GLT1-GLT6

### 3 进一步分析

#### 3.1 WFG 测试集检验求解性能

用具有复杂的 Pareto 前沿形状复杂 Pareto 问题特性的 WFG<sup>[24]</sup> 测试题来进一步验证 ASMEA 的性能. 将 ASMEA 及对比算法 NSGA-II、SMS-EMOA、RM-MEDA、TMOEA/D、MOEA/D-DE 的参数值见表 3. 对于 WFG 测试题, 计算 HV 指标值时的参考

点均选择  $r = (3, 5)^T$ . ASMEA 和对比算法独立计算 WFG 测试题 23 次获得的统计结果见表 4.

从表 4 的统计结果得出, 与 NSGA-II、RM-MEDA、TMOEA/D 及 MOEA/D-DE 各自的 23 次比较中, ASMEA 分别获得了 13、14、11 和 10 个明显较优结果. 虽然相对 SMS-EMOA 而言, ASMEA 获得了 5 个更优, 但 SMS-EMOA 与 ASMEA 在考虑最优解和次优解时性能相似. 最优平均秩的结果表明

ASMEA 略差于 SMS-EMOA. 但 ASMEA 在求解 GLT 整体最优性能.

测试集的性能优于 SMS-EMOA, 所以 ASMEA 拥有

表 3 求解 WFG 测试题的 NSGA-II、SMS-EMOA、RM-MEDA、TMOEA/D、MOEA/D-DE 及 ASMEA 的参数值

Tab.3 Parameter values for NSGA-II, SMS-EMOA, RM-MEDA, TMOEA/D, MOEA/D-DE, and ASMEA on WFG test instances

算法名称	参数设置
公共参数	$N = 100, T = 450, n = 30, m = 2$
NSGA-II	$F = 0.3, CR = 0.2$
SMS-EMOA	$F = 0.5, CR = 0.2$
RM-MEDA	$K = 5$
TMOEA/D	$F = 0.5, CR = 0.2$
MOEA/D-DE	$NS = 20, \beta = 0.9, F = 0.5, CR = 0.2$
ASMEA	$H = 5, HL = 15, \beta_0 = 0.5, F = 0.7, CR = 0.6, \tau_0 = 0.7$

表 4 NSGA-II、SMS-EMOA、RM-MEDA、TMOEA/D、MOEA/D-DE 以及 ASMEA 在 WFG 测试题上获得的统计结果

Tab.4 Statistical results obtained by NSGA-II, SMS-EMOA, RM-MEDA, TMOEA/D, MOEA/D-DE, and ASMEA on WFG test instances

测试题	IGD					
	NSGA-II	SMS-EMOA	RM-MEDA	TMOEA/D	MOEA/D-DE	ASMEA
WFG1	$1.417 \frac{\$}{7.08 \times 10^{-2}}[4]$	$1.616 \frac{\dagger}{5.18 \times 10^{-2}}[6]$	$1.172 \frac{\$}{8.30 \times 10^{-3}}[3]$	$9.376 \times 10^{-1} \frac{\$}{1.68 \times 10^{-2}}[1]$	$1.110 \frac{\$}{1.27 \times 10^{-2}}[2]$	$1.547 \frac{\dagger}{6.80 \times 10^{-2}}[5]$
WFG2	$1.450 \times 10^{-2} \frac{\dagger}{7.39 \times 10^{-4}}[3]$	$1.170 \times 10^{-2} \frac{\$}{7.28 \times 10^{-4}}[1]$	$2.970 \times 10^{-2} \frac{\dagger}{4.60 \times 10^{-3}}[5]$	$2.950 \times 10^{-2} \frac{\dagger}{4.40 \times 10^{-3}}[4]$	$3.810 \times 10^{-2} \frac{\dagger}{6.74 \times 10^{-4}}[6]$	$1.370 \times 10^{-2} \frac{\dagger}{7.57 \times 10^{-4}}[2]$
WFG3	$1.900 \times 10^{-1} \frac{\dagger}{8.46 \times 10^{-4}}[5]$	$1.140 \times 10^{-2} \frac{\$}{1.18 \times 10^{-4}}[1]$	$3.410 \times 10^{-2} \frac{\dagger}{5.10 \times 10^{-3}}[6]$	$1.240 \times 10^{-2} \frac{\dagger}{5.64 \times 10^{-4}}[3]$	$1.470 \times 10^{-2} \frac{\dagger}{2.72 \times 10^{-4}}[4]$	$1.180 \times 10^{-2} \frac{\dagger}{1.28 \times 10^{-4}}[2]$
WFG4	$5.710 \times 10^{-2} \frac{\$}{5.30 \times 10^{-3}}[3]$	$1.080 \times 10^{-2} \frac{\$}{2.92 \times 10^{-4}}[1]$	$9.410 \times 10^{-2} \frac{\dagger}{3.70 \times 10^{-3}}[6]$	$5.800 \times 10^{-2} \frac{\$}{2.24 \times 10^{-2}}[4]$	$3.290 \times 10^{-2} \frac{\$}{3.50 \times 10^{-3}}[2]$	$7.390 \times 10^{-2} \frac{\dagger}{4.80 \times 10^{-3}}[5]$
WFG5	$6.920 \times 10^{-2} \frac{\dagger}{6.26 \times 10^{-4}}[4]$	$6.650 \times 10^{-2} \frac{\$}{1.39 \times 10^{-4}}[1]$	$1.116 \times 10^{-1} \frac{\dagger}{1.44 \times 10^{-2}}[6]$	$8.560 \times 10^{-2} \frac{\dagger}{2.38 \times 10^{-2}}[5]$	$6.770 \times 10^{-2} \frac{\dagger}{5.30 \times 10^{-4}}[3]$	$6.660 \times 10^{-2} \frac{\dagger}{9.28 \times 10^{-5}}[2]$
WFG6	$3.335 \times 10^{-1} \frac{\dagger}{7.30 \times 10^{-3}}[3]$	$3.501 \times 10^{-1} \frac{\dagger}{3.44 \times 10^{-2}}[5]$	$3.261 \times 10^{-1} \frac{\dagger}{4.40 \times 10^{-3}}[2]$	$3.534 \times 10^{-1} \frac{\dagger}{2.72 \times 10^{-2}}[6]$	$3.350 \times 10^{-1} \frac{\dagger}{1.54 \times 10^{-2}}[4]$	$3.219 \times 10^{-1} \frac{\dagger}{1.81 \times 10^{-2}}[1]$
WFG7	$3.000 \times 10^{-2} \frac{\dagger}{1.90 \times 10^{-3}}[4]$	$1.110 \times 10^{-2} \frac{\$}{6.28 \times 10^{-4}}[1]$	$3.980 \times 10^{-2} \frac{\dagger}{7.80 \times 10^{-3}}[5]$	$2.641 \times 10^{-1} \frac{\dagger}{2.04 \times 10^{-1}}[6]$	$1.860 \times 10^{-2} \frac{\dagger}{5.46 \times 10^{-4}}[3]$	$1.140 \times 10^{-2} \frac{\dagger}{2.16 \times 10^{-4}}[2]$
WFG8	$6.130 \times 10^{-2} \frac{\dagger}{8.00 \times 10^{-3}}[4]$	$3.650 \times 10^{-2} \frac{\dagger}{7.40 \times 10^{-3}}[2]$	$1.609 \times 10^{-1} \frac{\dagger}{1.67 \times 10^{-2}}[6]$	$1.159 \times 10^{-1} \frac{\dagger}{9.16 \times 10^{-2}}[5]$	$5.000 \times 10^{-2} \frac{\dagger}{6.80 \times 10^{-3}}[3]$	$3.610 \times 10^{-2} \frac{\dagger}{9.30 \times 10^{-3}}[1]$
WFG9	$2.698 \times 10^{-1} \frac{\dagger}{1.49 \times 10^{-2}}[5]$	$2.677 \times 10^{-1} \frac{\dagger}{3.39 \times 10^{-2}}[4]$	$2.073 \times 10^{-1} \frac{\$}{2.50 \times 10^{-3}}[1]$	$2.715 \times 10^{-1} \frac{\dagger}{2.51 \times 10^{-2}}[6]$	$2.463 \times 10^{-1} \frac{\dagger}{3.15 \times 10^{-2}}[2]$	$2.631 \times 10^{-1} \frac{\dagger}{2.41 \times 10^{-2}}[3]$

测试题	HV					
	NSGA-II	SMS-EMOA	RM-MEDA	TMOEA/D	MOEA/D-DE	ASMEA
WFG1	$4.372 \frac{\$}{2.68 \times 10^{-1}}[4]$	$4.051 \frac{\dagger}{2.91 \times 10^{-1}}[5]$	$5.596 \frac{\$}{3.93 \times 10^{-2}}[3]$	$6.695 \frac{\$}{8.36 \times 10^{-2}}[1]$	$5.922 \frac{\$}{5.18 \times 10^{-2}}[2]$	$3.962 \frac{\dagger}{2.72 \times 10^{-1}}[6]$
WFG2	$1.141 \times 10^{1} \frac{\dagger}{4.20 \times 10^{-3}}[5]$	$1.146 \times 10^{1} \frac{\$}{3.58 \times 10^{-4}}[1]$	$1.127 \times 10^{1} \frac{\dagger}{3.45 \times 10^{-2}}[6]$	$1.145 \times 10^{1} \frac{\$}{6.30 \times 10^{-3}}[2]$	$1.144 \times 10^{1} \frac{\$}{8.51 \times 10^{-4}}[3]$	$1.142 \times 10^{1} \frac{\dagger}{5.20 \times 10^{-3}}[4]$
WFG3	$1.089 \times 10^{1} \frac{\dagger}{5.70 \times 10^{-3}}[5]$	$1.096 \times 10^{1} \frac{\$}{1.13 \times 10^{-4}}[1]$	$1.077 \times 10^{1} \frac{\dagger}{3.37 \times 10^{-2}}[6]$	$1.093 \times 10^{1} \frac{\dagger}{8.70 \times 10^{-3}}[3]$	$1.092 \times 10^{1} \frac{\dagger}{3.00 \times 10^{-3}}[4]$	$1.095 \times 10^{1} \frac{\dagger}{1.60 \times 10^{-3}}[2]$
WFG4	$8.348 \frac{\$}{5.66 \times 10^{-2}}[3]$	$8.671 \frac{\$}{3.40 \times 10^{-3}}[1]$	$8.110 \frac{\dagger}{2.13 \times 10^{-2}}[6]$	$8.258 \frac{\dagger}{2.01 \times 10^{-1}}[4]$	$8.536 \frac{\$}{2.42 \times 10^{-2}}[2]$	$8.235 \frac{\dagger}{3.04 \times 10^{-2}}[5]$
WFG5	$8.130 \frac{\dagger}{4.59 \times 10^{-2}}[4]$	$8.148 \frac{\dagger}{3.49 \times 10^{-2}}[2]$	$7.964 \frac{\dagger}{9.23 \times 10^{-2}}[5]$	$7.651 \frac{\dagger}{3.78 \times 10^{-1}}[6]$	$8.134 \frac{\dagger}{2.90 \times 10^{-2}}[3]$	$8.219 \frac{\dagger}{4.91 \times 10^{-2}}[1]$
WFG6	$6.331 \frac{\dagger}{4.01 \times 10^{-2}}[4]$	$6.263 \frac{\dagger}{1.69 \times 10^{-1}}[5]$	$6.381 \frac{\dagger}{2.31 \times 10^{-2}}[2]$	$6.249 \frac{\dagger}{1.10 \times 10^{-1}}[6]$	$6.338 \frac{\dagger}{7.85 \times 10^{-2}}[3]$	$6.408 \frac{\dagger}{9.21 \times 10^{-2}}[1]$
WFG7	$8.572 \frac{\dagger}{9.20 \times 10^{-3}}[4]$	$8.683 \frac{\$}{1.07 \times 10^{-2}}[1]$	$8.457 \frac{\dagger}{5.16 \times 10^{-2}}[5]$	$6.212 \frac{\dagger}{1.13}[6]$	$8.643 \frac{\dagger}{4.50 \times 10^{-3}}[3]$	$8.672 \frac{\dagger}{2.20 \times 10^{-3}}[2]$
WFG8	$8.309 \frac{\dagger}{5.71 \times 10^{-2}}[4]$	$8.505 \frac{\$}{4.37 \times 10^{-2}}[1]$	$7.682 \frac{\dagger}{9.72 \times 10^{-2}}[5]$	$7.314 \frac{\dagger}{9.68 \times 10^{-1}}[6]$	$8.445 \frac{\dagger}{4.82 \times 10^{-2}}[3]$	$8.470 \frac{\dagger}{5.82 \times 10^{-2}}[2]$
WFG9	$6.104 \frac{\dagger}{7.30 \times 10^{-2}}[5]$	$6.125 \frac{\dagger}{1.68 \times 10^{-1}}[4]$	$6.413 \frac{\$}{1.94 \times 10^{-2}}[1]$	$6.033 \frac{\dagger}{1.23 \times 10^{-1}}[6]$	$6.175 \frac{\dagger}{1.55 \times 10^{-1}}[2]$	$6.144 \frac{\dagger}{1.21 \times 10^{-1}}[3]$

注: 1) NSGA-II 的平均秩为 4.056, SMS-EMOA 的平均秩为 2.389, RM-MEDA 的平均秩为 4.389, TMOEA/D 的平均秩为 4.444, MOEA/D-DE 的平均秩为 3.000, ASMEA 的平均秩为 2.722; 2) NSGA-II 的 †/§/≈ 为 13/4/1, SMS-EMOA 的 †/§/≈ 为 5/10/3, RM-MEDA 的 †/§/≈ 为 14/4/0, TMOEA/D 的 †/§/≈ 为 11/4/3, MOEA/D-DE 的 †/§/≈ 为 10/5/3.

### 3.2 参数灵敏度分析

用带有不同参数的 ASMEA 算法对 GLT 的每道测试题独立运行 23 次,运用获得的 IGD 指标值的均值和标准差分析参数灵敏度.

#### 3.2.1 邻居交配池大小

分析 ASMEA 对  $H=5, 10, 15, 20, 25$  的灵敏度. ASMEA 算法采用不同  $H$  值时获得的 IGD 指标值的均值和标准差绘制于图 4(a) 中. 通过图 4(a) 可以看出,采用  $H=5$  的 ASMEA 对所有 GLT 测试题有最佳的求解效果.

#### 3.2.2 历史长度

设置  $HL=5, 10, 15, 20, 25$ , 在每个 GLT 测试题上独立运行 ASMEA 算法 23 次,利用不同  $HL$  的 ASMEA 算法获得的 IGD 值绘制于图 4(b) 中. 如图 4(b) 所示,当  $HL=15$  时,综合所有 GLT 测试题上的结果发现此时 ASMEA 表现最优.

#### 3.2.3 初始交配限制概率

设置  $\beta_0=0.1, 0.3, 0.5, 0.7, 0.9$ , 在 GLT 测试题上独立运行 ASMEA 算法 23 次,将  $\beta_0$  取不同值时 ASMEA 算法获得的 IGD 的均值和标准差如图 4(c) 所示. 从图 4(c) 中可看出,取  $\beta_0=0.9$  时,ASMEA 对于所有测试题均有较优异的性能,表明 ASMEA 性能对于  $\beta_0$  的取值不敏感.

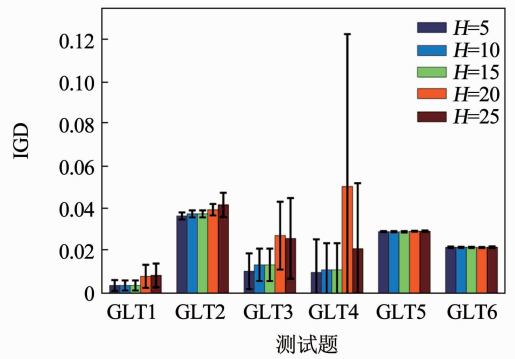
#### 3.2.4 初始学习率

设置学习率  $\tau_0=0.1, 0.3, 0.5, 0.7, 0.9$ , 将 ASMEA 在每道测试题上独立运行 23 次,绘制 IGD 指标值的均值和标准差如于图 4(d) 中. 从图 4(d) 中观察到,当  $\tau_0=0.7$  时,ASMEA 求解决所有的 GLT 测试题均有较好的性能. 图 4(d) 的结果表明 ASMEA 的性能对于初始学习率  $\tau_0$  的设置并不敏感.

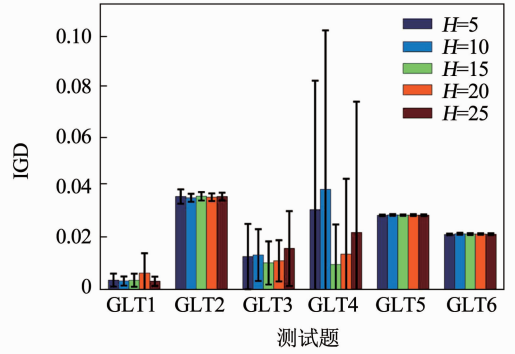
## 4 结 论

1) 本文设计了交配限制概率自适应更新策略以及聚类-演化融合策略,进而提出了一种自适应自组织多目标演化算法 (ASMEA). 在 ASMEA 的每一代,运用 SOM 聚类算法将种群划分为几个类. 根据交配限制概率,控制每个个体从邻居或差异较大的整个种群选择父代,进行交配产生新个体,从而加强开采或者勘探. 此外,在每一代,根据邻居和整个种群产生新个体的效用,进而对交配限制概率进行自适应更新.

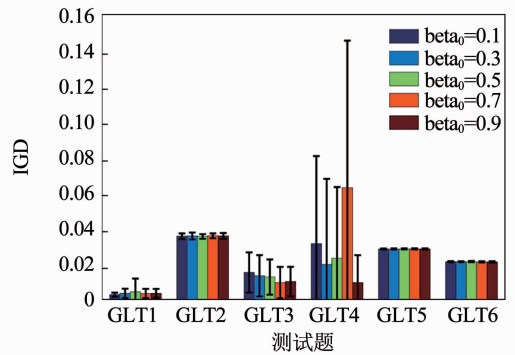
2) 将 ASMEA 与代表性算法 NSGA-II、SMS-EMOA、RM-MEDA、TMOEA/D、MOEA/D-DE 进行对比实验. 6 种算法分别求解了具有复杂 Pareto 前沿形状的 GLT 和 WFG 标准测试题. 求解结果表明



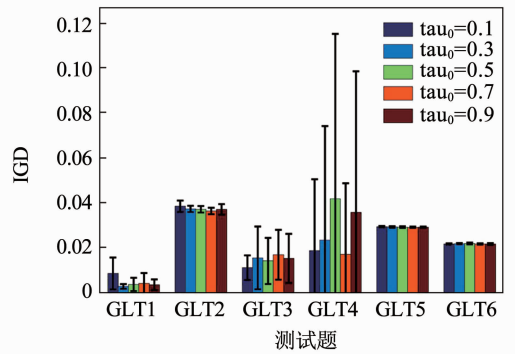
(a) 邻居交配池  $H$



(b) 历史长度  $HL$



(c) 初始交配限制概率  $\beta_0$



(d) 初始学习率  $\tau_0$

图 4 采用不同  $H, HL, \beta_0$  或  $\tau_0$  值的 ASMEA 在 GLT1-GLT6 上独立运行 23 次获得的 IGD 值的均值和标准差

Fig. 4 Mean and standard deviations of IGD values obtained by ASMEA with different  $H, HL, \beta_0$  or  $\tau_0$  values in 23 independent runs on GLT1-GLT6

ASMEA 具有整体更优的求解性能. 参数灵敏度实验结果表明 ASMEA 的性能未明显受到其使用参数变化的影响.

## 参考文献

- [1] DEB K. Multi-objective optimization using evolutionary algorithms [M]. New York, NY: John Wiley & Sons, Inc., 2001:13
- [2] DEB K, BEYER H G. Self-adaptive genetic algorithms with simulated binary crossover[J]. Evolutionary Computation, 2001, 9(2):199. DOI: 10.1162/106365601750190406
- [3] SCHAFFER J D. Multiple objective optimization with vector evaluated genetic algorithms [C]//Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications, Lawrence Erlbaum Associates. Pittsburgh, PA: DBLP, 1985: 94
- [4] STORN R, PRICE K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 344. DOI:10.1023/A:1008202821328
- [5] ZHANG Qingfu, ZHOU Aimin, JIN Yaochu. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm[J]. IEEE Transactions on Evolutionary Computation, 2008,12(1):43. DOI: 10.1109/TEVC.2007.894202
- [6] ZHOU Aimin, ZHANG Qingfu, JIN Yaochu. Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 1167. DOI: 10.1109/TEVC.2009.2021467
- [7] 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. DOI:10.1109/4235.996017
- [8] BEUME N, NAUJOKS B, EMMERICH M. SMS-EMOA: Multiobjective selection based on dominated hypervolume [J]. European Journal of Operational Research, 2007, 181(3): 1653. DOI: 10.1016/j.ejor.2006.08.008
- [9] LI Hui, ZHANG Qingfu. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 284. DOI: 10.1109/tevc.2008.925798
- [10] LI Hui, ZHANG Qingfu, DENG Jingda. Biased multiobjective optimization and decomposition algorithm[J]. IEEE Transactions on Cybernetics, 2016, 47(1): 1. DOI: 10.1109/TCYB.2015.2507366
- [11] SHANG Ronghua, DAI Kaiyun, JIAO Licheng, et al. Improved memetic algorithm based on route distance grouping for multiobjective large scale capacitated arc routing problems[J]. IEEE Transactions on Cybernetics, 2016, 46(4): 1001. DOI:10.1109/TCYB.2015.2419276
- [12] CHEN Renzhi, LI Ke, YAO Xin. Dynamic multiobjectives optimization with a changing number of objectives [J]. IEEE Transactions on Evolutionary Computation, 2018, 22(1): 157. DOI:10.1109/TEVC.2017.2669638
- [13] ZHANG Hu, ZHANG Xiujie, SONG Shenming, et al. An affinity propagation-based multiobjective evolutionary algorithm for selecting optimal aiming points of missiles [J]. Soft Computing, 2017, 21(11): 3028. DOI: 10.1007/s00500-015-1986-9
- [14] JIN Yaochu, SENDHOFF B. Connectedness, regularity and the success of local search in evolutionary multi-objective optimization [C]//Proceedings of the 2003 IEEE Congress on Evolutionary Computation. Canberra: IEEE, 2003:1910. DOI: 10.1109/CEC.2003.1299907
- [15] JIANG Shouyong, YANG Shengxiang. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts[J]. IEEE Transactions on Cybernetics, 2015, 46(2): 421. DOI: 10.1109/TCYB.2015.2403131
- [16] ISHIBUCHI H, NARUKAWA K, TSUKAMOTO N, et al. An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization [J]. European Journal of Operational Research, 2008, 188(1): 57. DOI: 10.1016/j.ejor.2007.04.007
- [17] SUN Jianyong, ZHANG Hu, ZHOU Aimin, et al. A new learning-based adaptive multi-objective evolutionary algorithm [J]. Swarm and Evolutionary Computation, 2019, 44: 304. DOI: 10.1016/j.swevo.2018.04.009
- [18] KOHONEN T. The self-organizing map[J]. Neurocomputing, 1998, 21(1/2/3): 1. DOI: 10.1016/S0925-2312(98)00030-7
- [19] BEUME N, NAUJOKS B, EMMERICH M. SMS-EMOA: Multiobjective selection based on dominated hypervolume [J]. European Journal of Operational Research, 2007, 181(3): 1653. DOI: 10.1016/j.ejor.2006.08.008
- [20] GU Fangqing, LIU Hailin, TAN K C. A multiobjective evolutionary algorithm using dynamic weight design method [J]. International Journal of Innovative Computing, Information and Control, 2012, 8(5B): 3677
- [21] ZITZLER E, THIELE L, LAUMANN M, et al. Performance assessment of multiobjective optimizers: An analysis and review[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(2): 117. DOI: 10.1109/TEVC.2003.810758
- [22] ZITZLER E, THIELE L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257. DOI: 10.1109/4235.797969
- [23] LIU Hailin, GU Fangqing, CHEUNG Y M. T-MOEA/D: MOEA/D with objective transform in multi-objective problems [C]//Proceedings of the 2010 International Conference of Information Science and Management Engineering. Xi'an: IEEE, 2010:282. DOI: 10.1109/ISME.2010.274
- [24] HUBAND S, BARONE L, WHILE L, et al. A scalable multi-objective test problem toolkit [C]// Proceedings of International Conference on Evolutionary Multi-Criterion Optimization. Berlin: Springer, 2005: 280. DOI: 10.1007/978-3-540-31880-4\_20