

# 基于在线感知 Pareto 前沿划分目标空间的多目标进化优化

封文清<sup>1</sup> 巩敦卫<sup>1</sup>

**摘要** 多目标进化优化是求解多目标优化问题的可行方法。但是, 由于没有准确感知并充分利用问题的 Pareto 前沿, 已有方法难以高效求解复杂的多目标优化问题。本文提出一种基于在线感知 Pareto 前沿划分目标空间的多目标进化优化方法, 以利用感知的结果, 采用有针对性的进化优化方法求解多目标优化问题。首先, 根据个体之间的拥挤距离与给定阈值的关系感知优化问题的 Pareto 前沿上的间断点, 并基于此将目标空间划分为若干子空间; 然后, 在每一子空间中采用 MOEA/D (Multi-objective evolutionary algorithm based on decomposition) 得到一个外部保存集; 最后, 基于所有外部保存集生成问题的 Pareto 解集。将提出的方法应用于 15 个基准数值函数优化问题, 并与 NSGA-II、RPEA、MOEA/D、MOEA/D-PBI、MOEA/D-STM 和 MOEA/D-ACD 等比较。结果表明, 提出的方法能够产生收敛和分布性更优的 Pareto 解集, 是一种非常有竞争力的方法。

**关键词** 多目标进化优化, Pareto 前沿, 间断点, 目标空间划分, MOEA/D

**引用格式** 封文清, 巩敦卫. 基于在线感知 Pareto 前沿划分目标空间的多目标进化优化. 自动化学报, 2020, 46(8): 1628–1643

**DOI** 10.16383/j.aas.2018.c180323



开放科学 (资源服务) 标识码 (OSID):

## Multi-objective Evolutionary Optimization With Objective Space Partition Based on Online Perception of Pareto Front

FENG Wen-Qing<sup>1</sup> GONG Dun-Wei<sup>1</sup>

**Abstract** Multi-objective evolutionary optimization is a feasible way to solve multi-objective optimization problems. However, previous methods have difficulties in efficiently tackling a complex multi-objective optimization problem, since they cannot accurately perceive the shape of the Pareto front and take full advantages of it. A multi-objective evolutionary algorithm with objective space partition based on online perceiving the Pareto front of an optimization problem is proposed in this study. In the proposed method, a series of discontinuous points on a Pareto front which is getting from the relationship between the crowded distance of individuals and a given threshold are first detected. Then, the objective space is divided into a number of sub-spaces based on these points. In each sub-space, the multi-objective evolutionary algorithm based on decomposition (MOEA/D) is employed to obtain an external archive set. Finally, the Pareto optimal set of the problem is formed based on all the external sets. The proposed method is compared with NSGA-II, RPEA, MOEA/D, MOEA/D-PBI, MOEA/D-STM, and MOEA/D-ACD on 15 benchmark optimization problems. The results empirically demonstrate that the proposed method has capabilities in obtaining a Pareto optimal set with good performance in convergence and diversity, and is a competitive alternative for multi-objective optimization.

**Key words** Multi-objective evolutionary optimization, Pareto front, discontinuous point, objective space division, MOEA/D

**Citation** Feng Wen-Qing, Gong Dun-Wei. Multi-objective evolutionary optimization with objective space partition based on online perception of Pareto front. *Acta Automatica Sinica*, 2020, 46(8): 1628–1643

收稿日期 2018-05-18 录用日期 2018-08-28  
Manuscript received May 18, 2018; accepted August 28, 2018  
国家重点研发计划项目 (2018YFB1003802-01), 国家自然科学基金 (61773384, 61763026, 61673404, 61573361, 61503220), 国家 973 科技计划项目 (2014CB046306-2) 资助  
Supported by The National Key Research and Development Program of China (2018YFB1003802-01), National Natural Science Foundation of China (61773384, 61763026, 61673404, 61573361, 61503220), and National Basic Research Program of China (973 Program) (2014CB046306-2)

本文责任编辑 魏庆来  
Recommended by Associate Editor WEI Qing-Lai  
1. 中国矿业大学信息与控制工程学院 徐州 221116  
1. School of Information and Control Engineering, China Uni-

versity of Mining and Technology, Xuzhou 221116

优化是工程实践和科学研究中主要的问题形式之一, 其中, 仅有 1 个目标函数的优化问题, 称为单目标优化问题; 目标函数超过 1 个且存在冲突的最优化问题, 称为多目标优化问题 (Multi-objective optimization problems, MOPs). 与单目标优化问题相比, 多目标优化问题更加普遍地存在于科学研究和工程应用领域, 如半导体制造、工业调度以及控制系统<sup>[1-4]</sup>. 其特点在于, 优化问题不仅包含多个目

versity of Mining and Technology, Xuzhou 221116

标, 而且目标之间相互冲突, 从而难以得到使各目标都达到最优的解, 而只存在一个折衷解的集合, 称为 Pareto 最优解集 (Pareto-optimal set)<sup>[5]</sup>. 针对多目标优化问题, 出现众多优异的多目标进化算法<sup>[6-9]</sup>, 其中文献 [10] 在 2009 年提出基于分解的多目标进化算法 (Multi-objective evolutionary algorithm based on decomposition, MOEA/D), 该算法首先将多目标优化问题分解成多个单目标子问题, 然后并行优化各子问题, 并基于聚合函数选择最优解, 提高了 Pareto 解集的选择压力, 并通过实验验证了所提方法对解决具有复杂 Pareto 前沿形状的问题有较好的效果.

针对多目标优化问题, 最初采用数学规划方法, 将多目标优化问题转化为单目标问题, 典型的方法包括: 加权和法、切比雪夫法 (Tchebycheff approach), 以及边界交集法等<sup>[11-12]</sup>. 多目标优化问题的目标函数和约束函数可能是非线性、不可微或不连续的, 使得传统的数学规划方法求解效率偏低, 且对权值和目标的次序敏感. 20 世纪 80 年代中期, 进化优化应用于求解多目标优化问题, 即多目标进化. 90 年代中期, 多目标优化开始迅速发展, 大体分为两个阶段:

第一阶段主要采用基于 Pareto 占优的个体选择和适应值共享的多样性保持, 典型的方法有多目标遗传算法 (Multi-objective genetic algorithms, MOGA)<sup>[13]</sup>、非支配排序遗传算法 (Non-dominated sorting genetic algorithm, NSGA)<sup>[14]</sup> 和小生境遗传算法 (Niche Pareto genetic algorithm, NPGA)<sup>[15]</sup>. MOGA 首先基于 Pareto 占优, 将种群中的个体划分不同的等级, 并基于此对个体排序; 然后通过适应度共享调整个体的适应值, 并基于调整后的适应值选择优势个体. MOGA 大的选择压力, 导致种群往往早熟收敛. NSGA 选择非被占优解之后共享一个虚拟适应值, 并选择后续的非被占优解. NSGA 不但计算复杂度高, 而且需要预先设定共享参数的值. 在 NPGA 中, 采用基于 Pareto 占优的锦标赛选择并辅以小生境技术, 生成子代种群. 但是, 该算法中小生境半径的选取和调整困难, 且需要确定比较集的规模.

第二阶段主要以外保存集用来存放进化过程中的非被占优个体, 保证了解集的分布性, 代表性的方法有强度 Pareto 进化算法 (Strength Pareto evolutionary algorithm II, SPEA-II)<sup>[16]</sup>、非支配排序遗传算法 II (Non-dominated sorting genetic algorithm II, NSGA-II)<sup>[17]</sup> 和 Pareto 包络选择算法 II (Pareto envelop-based selection algorithm-II, PESA-II)<sup>[18]</sup>. NSGA-II 基于序值的快速非被占优解排序, 降低了计算复杂度. PESA-II 基于网格选择

个体, 使得解集具有好的收敛性, 特别适合于求解高维多目标优化问题. 但是, 该算法执行一次选择操作只能选择一个个体, 导致计算消耗增多, 且解集的多样性不佳. SPEA 约定适应度低的个体对应高的选择概率, 并设置除进化种群外的一个外部种群, 保存当前非被占优个体. 当外部种群的个体数目超过设定值时, 采用聚类方法删减个体, 从进化种群和外部种群中选择个体进行交叉和变异操作. 但是, 该算法的计算复杂度较高.

实际的多目标优化问题的 Pareto 前沿往往具有复杂的特征, 可能是长尾、多峰或不连续的. 如果问题的 Pareto 前沿具有长尾, 那么, 获得好的分布性能的解集将非常困难; 多峰优化问题往往需要寻找多个极值点, 而已有方法主要求解优化问题的全局极值, 因此, 非常具有挑战性; 当优化问题的 Pareto 前沿不连续时, 间断边界搜索的不准确, 往往导致解集的分布范围非常有限. 对于这些优化问题, 已有方法通常难以有效求解.

为了有效求解具有复杂 Pareto 前沿的优化问题, 文献 [19-20] 提出了新的非被占优解保留方法, 并且通过优化 Pareto 解集的评价指标, 代替优化原来的问题; 虽然在多种评价指标中, 超体积 (Hypervolume) 最为常用, 但是, 鉴于其计算耗时长, 因此, 利用蒙特卡洛方法估计超体积. 文献 [21-22] 是 MOEA/D 的改进, 通过分析 Tchebycheff 聚合函数, 确定初始的方向向量, 且引入方向向量的自适应调整, 获得具有好的分布性的解集. 文献 [23] 提出了一个参考向量引导的多目标优化算法, 采用角度惩罚距离的标量化方法, 平衡目标空间中解的收敛和多样性, 根据目标函数的尺度动态调整参考向量的分布. 文献 [24] 针对规则和不规则 POF 的问题, 提出基于分解的参考向量自适应的 g-DBEA 算法, 采用学习期间积累的关于特定的参考矢量的非支配解的数量的信息, 引导参考矢量的适应.

需要指出的是, 这些工作假定优化问题的 Pareto 前沿形状可粗略估计, 并基于 Pareto 前沿形状, 设计有针对性的求解方法. 对于实际的优化问题, 与该问题相关的数据往往难以获取, 从而难以估计问题的 Pareto 前沿. 如果基于进化求解过程获取的数据, 在线感知问题的 Pareto 前沿, 那么, 在后续的求解过程中, 将能够基于感知的 Pareto 前沿, 设计有针对性的优化问题求解方法, 对于优化问题的解决, 是非常有帮助的.

鉴于此, 本文研究多目标优化问题 Pareto 前沿的在线感知问题, 并提出基于在线感知 Pareto 前沿划分目标空间的多目标进化优化方法. 首先, 基于拥挤距离, 感知优化问题的 Pareto 前沿是否存在间断点; 然后, 如果存在间断点, 那么, 将目标空间划分为

若干子空间,并在每一子空间上采用 MOEA/D,得到一系列外部保存集,基于此,生成问题的 Pareto 解集;否则,采用传统的 MOEA/D,生成问题的 Pareto 解集.

本文的贡献主要体现在如下 3 个方面: 1) 提出了基于拥挤距离的优化问题 Pareto 前沿间断点感知方法; 2) 提出了基于在线感知 Pareto 前沿划分目标空间的多目标进化优化方法; 3) 将提出的方法应用于 15 基准数值函数优化问题,并与 NSGA-II、基于参考点的进化算法 (Reference points-based evolutionary algorithm, RPEA)<sup>[25]</sup>、MOEA/D、MOEA/D-PBI<sup>[26]</sup>、MOEA/D-STM<sup>[27]</sup> 和 MOEA/D-ACD<sup>[28]</sup> 等比较,验证了所提方法的有效性.

本文结构安排如下: 第 1 节综述相关工作; 本文的方法在第 2 节提出; 第 3 节将提出的方法应用于基准数值函数优化问题,并与已有方法对比; 最后,第 4 节总结全文,并指出需要进一步研究的问题.

## 1 相关工作

不失一般性,本文考虑如下多目标优化问题<sup>[5]</sup>:

$$\begin{aligned} \min \mathbf{f}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{s. t. } \mathbf{x} &\in S \subset \mathbf{R}^n \end{aligned} \quad (1)$$

其中:  $\mathbf{x}$  为  $n$  维决策向量,  $S$  为  $\mathbf{x}$  的可行域,  $f_i(\mathbf{x}), i = 1, 2, 3, \dots, m$ , 为第  $i$  个目标函数,  $m$  为目标函数个数.

在多目标优化中,常用到如下重要定义.

**定义 1.** Pareto 占优: 考虑问题 (1) 的任意 2 个解  $\mathbf{x}_1, \mathbf{x}_2 \in S$ , 且  $\mathbf{x}_1 \neq \mathbf{x}_2$ , 如果  $\forall i = 1, 2, 3, \dots, m$ , 使得  $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ , 且  $\exists j = 1, 2, 3, \dots, M$  使得  $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$ , 那么, 称  $\mathbf{x}_1$  占优  $\mathbf{x}_2$ , 记为  $\mathbf{x}_1 \succ \mathbf{x}_2$ .

**定义 2.** Pareto 最优解集: 对问题 (1) 的一个解  $\mathbf{x}^* \in S$ , 如果不存在  $\mathbf{x}' \in S$ , 使得  $\mathbf{x}' \succ \mathbf{x}_1^*$ , 那么, 称  $\mathbf{x}_1^*$  为该问题的 Pareto 最优解. 所有 Pareto 最优解构成的集合, 称为 Pareto 最优解集.

**定义 3.** Pareto 前沿: Pareto 最优解集在目标空间中形成的 (超) 曲面, 称为 Pareto 前沿.

很多实际优化问题的 Pareto 前沿往往具有复杂的特性. 为了有效求解该问题, 感知 Pareto 前沿的特性, 并基于此设计有针对性的进化优化方法, 是非常必要的. 在诸多复杂的 Pareto 前沿特性中, 不连续即 Pareto 前沿存在间断区域, 是非常有代表性的. 对于该类问题, 如果仅通过增加种群规模, 受到非被占优个体的选择方法的影响, 对间断区域的描述往往不理想, 且该方法所需的计算量大. 鉴于此, 已有方法主要通过改变个体的分布, 求解 Pareto 前沿含有间断区域的优化问题, 主要分为如下 3 类.

1) 优先保留稀疏个体<sup>[29-33]</sup>. 在环境选择中, 当个体的序值相同时, NSGA-II 选择拥挤距离大的个体进入下一代. 但是, 该方法难以寻找孤立点, 也难以获得分布均匀的 Pareto 解集. 公茂果等<sup>[34]</sup> 根据个体在目标空间的相似度, 对 Pareto 前沿自适应划分, 并基于划分的区域, 选择代表性个体, 提高了 Pareto 解集的分布性能. 但是, 该方法得到解集的收敛性能尚需进一步提高, 此外, 非被占优个体的选择也受 Pareto 前沿形状的限制.

2) 引导个体向间断区域进化<sup>[35]</sup>. 差分进化采用特殊的变异算子, 并通过将差分向量加入基向量生成新的个体, 引导个体向间断区域进化, 具有很好的局部开发能力, 但是, 差分进化参数和优化性能的关系分析比较复杂. Deng 等<sup>[35]</sup> 提出了用于求解约束多目标优化问题的蚁群算法, 该算法保存发现的所有非被占优解, 用于指导蚂蚁向间断区域进化, 以提高解的分布性能, 但是, 对间断区域的感知与搜索时间长, 且易陷入局部最优.

3) 个体均匀选择. 依据间断区域方向向量对应的解, 描述间断区域; 根据个体的均匀选择, 获得间断区域的 Pareto 分布; 基于分解机制的多目标进化算法依靠方向向量和聚合函数的选择, 具有较优的收敛速度和分布性能, 但是, 对间断区域的搜索受方向向量的制约, 存在较大不确定性. 随后, 提出有针对性的改变解的分布, 提高复杂 Pareto 前沿上个体的分布性能, 以获得间断区域的 Pareto 前沿. 其中, MOEA/D-PBI<sup>[36-37]</sup> 是基于惩罚的边界交叉函数的分解多目标进化算法, 该算法在一定程度上隐式地考虑了解与方向向量的接近程度, 但惩罚策略的性能受惩罚系数  $\theta$  的限制. 自适应方向向量调整方法 (MOEA/D-AWA), 基于新的方向向量初始化技术, 自适应地调整复杂 Pareto 前沿的方向向量, 以获取均匀性的解, 增强对间断区域的搜索能力, 但是, 该方法没有采用特定的种群替换策略. MOEA/D-STM<sup>[27]</sup> 利用稳态匹配模型协调解的选择过程, 把子问题和解作为两个不同的代理, 子问题倾向于选择聚合函数值更小的解, 解倾向于选择与其距离更近的子问题, 以平衡多样性和收敛性. MOEA/D-TPN<sup>[38]</sup> 针对复杂 Pareto 前沿优化问题, 将方向向量分为中间和外部区域两部分, 通过计算这两个区域个体间的拥挤距离, 判断 Pareto 前沿的凹凸性, 并采用坐标变化方法, 将凹 Pareto 前沿转化为凸 Pareto 前沿, 以获得更好的分布性能, 但是, 该方法需要设置大量参数. 此外, 改进的 MOEA/D-ACD 自适应调整约束分解策略<sup>[28]</sup>, 提出对子问题施加约束, 利用从搜索中收集的信息自适应调整约束的策略, 虽然能获得良好的分布, 但是约束条件的调整降低了解的收敛速度.

本文根据个体之间的拥挤距离与给定阈值的关系, 搜索间断点; 基于相邻的间断点, 将目标空间划分为多个子空间, 并在每个子空间上采用 MOEA/D, 生成该子空间上的 Pareto 前沿, 进一步形成含有间断区域 Pareto 前沿. 第 2 节, 将详细阐述本文提出的基于在线感知 Pareto 前沿划分目标空间的多目标进化优化方法.

## 2 提出的方法

### 2.1 总体框架

本节提出一种基于在线感知 Pareto 前沿划分目标空间的多目标进化优化方法, 思想是: 基于个体之间的拥挤距离, 判断 Pareto 前沿是否存在间断点. 如果存在, 首先确定间断点的位置; 然后, 将目标空间划分为多个子空间, 并在每一子空间上采用局部 MOEA/D 求解, 通过合并不同子空间上的 Pareto 前沿, 形成优化问题的 Pareto 前沿; 否则, 在整个空间上采用传统的 MOEA/D 求解.

容易看出, 间断点判断、目标空间划分、局部 MOEA/D 设计, 以及 Pareto 前沿形成, 是需要解决的关键技术. 在后续的几小节, 将详细阐述这些关键技术的实现方法.

### 2.2 间断点的判定

判定优化问题的 Pareto 前沿是否存在间断点, 一种最直观的想法是: 考察种群进化到某代为止得到的外部保存集, 如果该集合中某元素的拥挤距离

大于一个设定的阈值, 那么, 在该元素及其相邻的元素之间可能存在 Pareto 前沿间断点. 基于此, 本小节提出一种基于拥挤距离的 Pareto 前沿间断点判定方法, 该方法首先基于种群进化, 得到一个外部保存集; 然后, 采用已有方法, 计算该集合中每一元素的拥挤距离; 最后, 设定一个拥挤距离阈值, 并将外部保存集中每一元素的拥挤距离与该阈值比较, 如果大于该阈值, 那么, 该元素及其相邻的元素可能存在间断点; 否则, 不存在间断点. 需要注意的是, 这里, 基于拥挤距离判定是否存在间断点, 判定结果的准确性与外部保存集中元素的分布、阈值的设定等有关. 不同的分布和设定值, 将导致不同的判定结果. 因此, 某一分布和阈值下的判定结果不一定是准确的.

现在给出判定优化问题 Pareto 前沿间断点的具体方法. 首先, 将每代种群进化得到的非被占优个体放入外部保存集, 并基于此, 更新外部保存集. 记种群进化到第  $t$  代, 得到的外部保存集为  $A(t)$ , 该集合包含的元素为  $a_1, a_1, \dots, a_{|A(t)|}$ , 其中,  $|A(t)|$  为  $A(t)$  包含的元素个数. 对  $A(t)$  的元素按照第 1 个目标值从小到大排序, 如果有多个元素的第 1 个目标值相同, 那么, 再对这些元素按照第 2 个目标值从小到大排序, 以此类推, 得到一个有序的外部保存集. 在不引起混淆的情况下, 仍记排序后的外部保存集为  $A(t)$ .

然后, 计算第  $i$  个元素  $a_i$  的拥挤距离. 采用式 (2) 的方法,  $a_i$  的拥挤距离记为  $C(a_i)$ , 可以表示为:

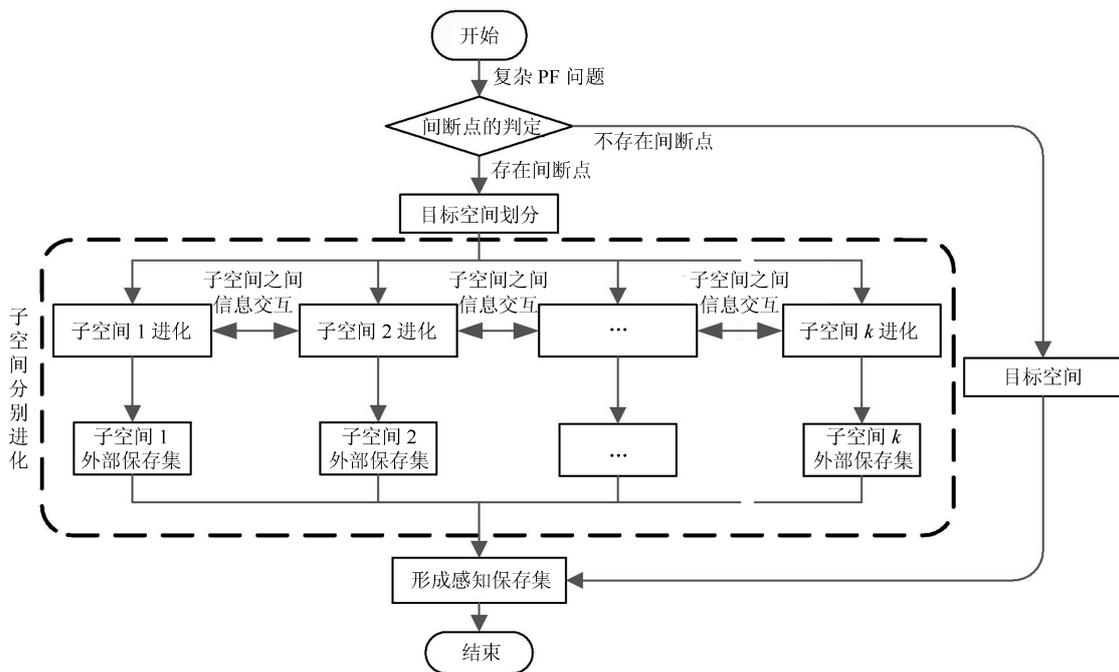


图 1 总体框架

Fig. 1 General framework

$$C(a_i) = \sum_{j=1}^m (f_j(a_{i+1}) - f_j(a_{i-1})) \quad (2)$$

最后, 判定是否存在间断点, 并确定间断点的位置.

**定义 4.** 间断区域和间断点:

记拥挤距离阈值为  $\delta$ , 当  $C(a_i) \geq \delta$  时,  $a_{i-1}$  和  $a_i$ 、 $a_i$  和  $a_{i+1}$  中至少有一对在目标空间的距离比较大. 不妨设  $a_{i-1}$  和  $a_i$  在目标空间的距离比较大, 那么, 在近似 Pareto 前沿上,  $a_{i-1}$  和  $a_i$  之间的区域很可能是一个间断区域, 此时,  $a_{i-1}$  和  $a_i$  是间断点.

容易看出, 间断点判定与  $\delta$  密切相关, 不同的  $\delta$  取值, 得到的间断点可能有很大差别. 现在给出一种  $\delta$  的取值方法. 对于  $A(t)$  中元素  $i$  的拥挤距离为  $C(a_i)$ . 对于所有的目标函数值, 能够得到这些拥挤距离的平均值. 那么,  $\delta$  可以表示为:

$$\delta = \frac{\alpha}{|A(t)|} \sum_{i=1}^{|A(t)|} C(a_i) \quad (3)$$

式中,  $\alpha$  为调整系数, 取值通过实验设定. 由式 (3) 可知,  $\alpha$ 、 $|A(t)|$  和  $C(a_i)$  是影响  $\delta$  的因素. 其中,  $C(a_i)$  由 Pareto 前沿决定,  $|A(t)|$  为  $A(t)$  包含的元素个数, 是固定值, 那么,  $\alpha$  是决定  $\delta$  取值的关键.

由式 (3) 可知, 拥挤距离阈值  $\delta$  为种群个体拥挤距离平均值的  $\alpha$  倍. 如果点  $a_i$  和  $a_{i-1}$  满足  $C(a_i) \geq \delta$ 、 $C(a_{i-1}) \geq \delta$ , 那么, 将点  $a_i$  和  $a_{i-1}$  加入间断点集合, 并输出间断点集合.

如图 2 所示, 其中, 图 2(a) 表示选取  $\alpha$  计算拥挤距离阈值, 分别计算每个个体的拥挤距离, 其中,  $C$  点和  $D$  点的拥挤距离最大, 其次是  $A$  点和  $B$  点, 然后是  $E$  点和  $F$  点; 从图 2(a) 可看出,  $C$  点和  $D$  点的拥挤距离明显大于  $\delta$ ,  $A$  点和  $B$  点的拥挤距离略大于  $\delta$ , 得到间断点  $C$  点、 $D$  点和  $A$  点、 $B$  点; 图 2(b) 中可以看出,  $E$  点和  $F$  点的拥挤距离小于  $\delta$ ,  $E$  点和  $F$  点之间不存在间断区域,  $E$  点和  $F$  点不是间断点. 此时, 间断点集合为  $\{A, B, C, D\}$ .

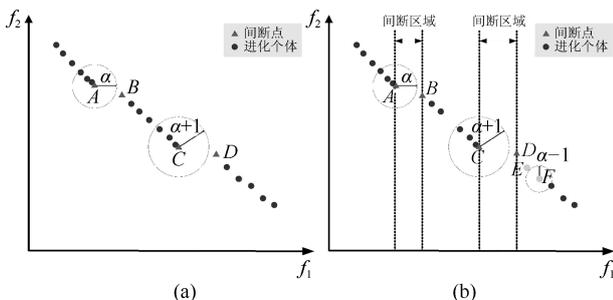


图 2 搜索间断点

Fig. 2 Searching for discontinuous points

**算法 1. 搜索间断点**

输入: 调整系数  $\alpha$ , 种群  $A(t)$   
 输出: 间断点保存集  $BP = \{BP_1, \dots, BP_l\}$   
 1) 根据式 (3) 计算拥挤距离阈值  $\delta$   
 2) for  $i = 1$  to  $|A(t)|$   
 3) if  $C(a_i) > \delta$   
 4)  $BP \leftarrow BP \cup a_i$   
 5) end  
 6) end  
 7) return  $BP = \{BP_1, \dots, BP_l\}$

在算法 1 中, 首先, 计算  $\delta$  (第 1 步); 接着, 依次计算每个个体的拥挤距离, 判断是否大于  $\delta$  (第 3 步), 若大于, 则将该点加入间断点  $BP$  中 (第 4 步); 最后, 算法结束 (第 7 步), 并输出间断点保存集.

**2.3 目标空间划分**

基于第 2.2 节得到的间断点, 本小节将目标空间划分, 思想是: 首先, 基于间断点, 得到若干网格<sup>[21]</sup>; 然后, 合并满足条件的网格, 得到划分后的目标空间.

首先, 以间断点作为分界点, 得到目标空间的若干网格. 以 2 目标优化问题为例, 说明网格的形成过程. 设间断点分别为  $a_{k-1}$  和  $a_k$ , 那么, 分界点在目标空间的坐标分别为  $(f_1(a_{k-1}), f_2(a_{k-1}))$  和  $(f_1(a_k), f_2(a_k))$ . 这样一来,  $f_1(a_{k-1})$  和  $f_1(a_k)$  将  $f_1$  轴分为 3 部分,  $f_2(a_{k-1})$  和  $f_2(a_k)$  将  $f_2$  轴也分为 3 部分, 如式 (4) 所示.

$$\begin{cases} f_{j,1} : f_j \leq \min\{f_j(a_{k-1}), f_j(a_k)\} \\ f_{j,2} : \min\{f_j(a_{k-1}), f_j(a_k)\} < f_j < \max\{f_j(a_{k-1}), f_j(a_k)\} \\ f_{j,3} : f_j \geq \max\{f_j(a_{k-1}), f_j(a_k)\} \end{cases} \quad (4)$$

式中,  $f_{j,1}$ 、 $f_{j,2}$  和  $f_{j,3}$  分别表示基于间断点, 得到的  $f_1$  轴的第 1、第 2 和第 3 部分,  $j = 1, 2$ . 选取  $f_1$  和  $f_2$  轴的某一部分, 得到的包围区域形成一个网格. 进一步, 将网格在各目标方向的最大值, 作为该网格的坐标. 如图 3 所示, 基于 2 个间断点, 形成 9 个网格.

类似的, 可以得到 3 及以上目标优化问题网格的形成过程. 记间断点的个数为  $l$ , 那么, 优化问题的 Pareto 前沿由  $l/2 + 1$  个分片的 Pareto 前沿构成. 此时, 网格形成的伪代码如算法 2 所示.

**算法 2. 网格形成**

输入: 间断点个数  $l$ , 间断点集合  $BP = \{BP_1, \dots, BP_l\}$   
 输出: 网格集合  $G$   
 1) for  $n=1$  to  $l/2$   
 2)  $f_{1,i_1}, f_{2,i_2}, \dots, f_{r,i_3} \leftarrow F(BP_{(2n-1)})$  and  $F(BP_{(2n)})$   
 3)  $(f_{1,i_1}, f_{2,i_2}, \dots, f_{r,i_3}) \leftarrow f_{1,i_1}, f_{2,i_2}, \dots, f_{r,i_3}$ ,

- $i_1, i_2, i_3 = 1, 2, 3$
- 4)  $G \leftarrow G \cup \{(f_{1,i_1}, f_{2,i_2}, \dots, f_{r,i_r})\}$
- 5) end
- 6) return  $G$

算法 2 描述了基于间断点分割目标空间的方法. 为了分割含有多个间断点的多维目标空间, 首先, 选取间断点  $F(BP_{(1)})$  和  $F(BP_{(2)})$  (第 2 步), 依式 (4) 计算坐标轴被间断点分割后的坐标范围,  $f_{1,i_1}, f_{2,i_2}, \dots, f_{r,i_r}$  在目标空间形成的区域, 称为一个网格 (第 3 步), 并保存到网格集合  $G$  中 (第 4 步). 重复以上操作, 直到间断点  $F(BP_{(l-1)})$  和  $F(BP_{(l)})$  为止, 得到网格集合  $G$ , 算法结束 (第 6 步).

将目标空间划分成一系列网格之后, 进化种群的任一个体, 均落入某一网格之中. 在这些网格中, 有的不包含进化个体, 称为空网格; 包含个体的网格, 称为正常网格. 将空网格与正常网格合并, 形成更大的子空间, 有利于在该子空间上采用合适的进化方法, 生成问题的 Pareto 解集. 网格合并时, 主要利用间断点与网格的占优关系, 将间断区域和非

支配间断区域的网格合并, 接下来以二维空间为例, 策略如下.

首先, 寻找间断区域所在的网格. 在所有的空网格中, 满足  $(f_{1,i}, f_{2,j}) > F(a_{k-1}), (f_{1,i}, f_{2,j}) > F(a_k), i, j = 1, 2, \dots, m$  的空网格, 占优间断点  $F(a_{k-1})$  和  $F(a_k)$ , 那么, 剩余的网格即为间断区域所在的网格, 如图 4 中的网格 ⑤、⑥、⑧ 和 ⑨.

然后, 依据间断点与网格的占优关系合并网格. 考虑每一对间断点, 选取距理想点最近的作为分割点, 合并分割点所在的网格和占优分割点的网格. 如图 4 所示, 在  $a_{k-1}$  和  $a_k$  中,  $a_{k-1}$  距理想点 (坐标原点) 近, 那么, 以  $a_{k-1}$  作为分割点, 合并网格 ①、④ 和 ⑦.

最后, 形成若干子空间. 基于分割点得到分割线 (面), 处于分割面两侧的区域, 为分割之后的子空间. 如图 4 所示, 基于  $a_{k-1}$ , 得到一条分割线, 位于分割线左侧的网格与占优区域一起, 形成子空间  $PF_1$ ; 位于分割线右侧的网格与间断区域一起, 形成子空间  $PF_2$ .

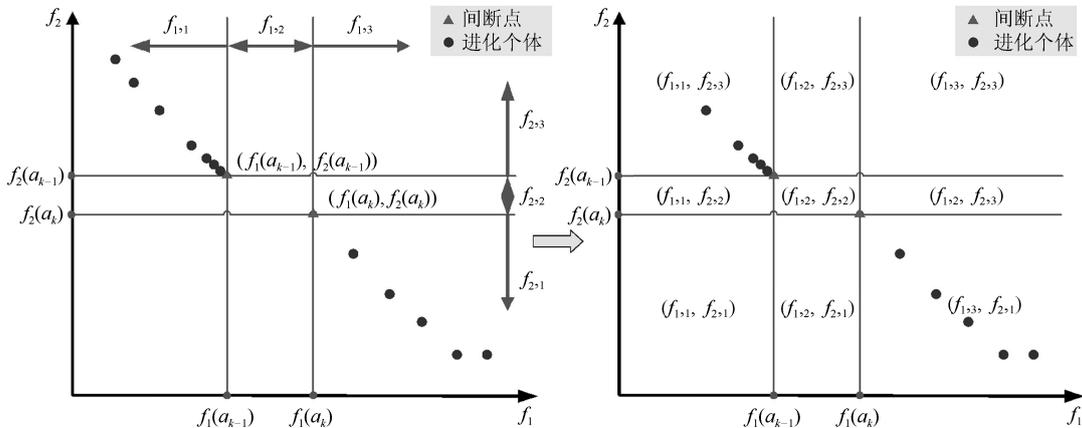


图 3 网格的形成

Fig. 3 Forming grids

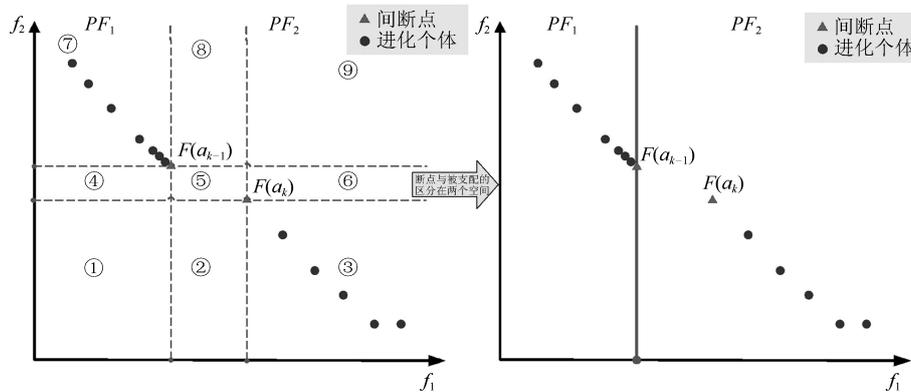


图 4 目标空间划分

Fig. 4 Dividing the objective space

### 2.4 局部 MOEA/D 设计

第 2.3 节中, 将存在间断区域的 Pareto 前沿的目标空间划分为多个子空间, 接下来给出局部 MOEA/D 的设计方法.

在不增加计算量的前提下, 为了获得 Pareto 前沿良好的分布和收敛性能, 本文采用如下方法: 针对收敛性, 采用子空间信息交互的方法, 即利用其他子空间种群进化所提供的有价值信息, 促进本空间个体的进化; 针对分布性, 在子空间中采用 MOEA/D 的同时增加对解的选择条件, 保证解的分布性能.

局部 MOEA/D 的步骤如下:

步骤 1. 初始化算法总代数  $max\_gen$ , 令  $t_i = 0$ .

步骤 2. 生成个体  $y$ . 选取子空间  $i$  中的个体, 经交叉变异获得个体  $y$ .

步骤 3. 计算  $y$  所在的子空间  $j$ . 如果  $i \neq j$ , 那么, 子空间  $i$  和  $j$  进行信息交互; 否则,  $y$  参与子空间  $i$  非被占优解的选择.

步骤 4. 子空间  $i$  的种群进化, 形成子代个体, 更新外部保存集  $A_i(t)$ .

步骤 5. 判断子空间  $i$  是否进化了  $max\_gen$  代, 如果满足, 转步骤 6; 否则, 令  $t_i = t_i + 1$ , 转步骤 2.

步骤 6. 输出子空间  $i$  的进化结果和外部保存集.

接下来, 给出局部 MOEA/D 中涉及到的主要策略: 子空间的信息交互和子空间内非被占优解的选择.

子空间之间通过信息交互协同进化, 加快了子空间内个体的收敛速度. 信息交互方法为: 首先, 选取子空间内个体  $x$ , 进行交叉和变异操作, 生成新个体  $y$ ; 然后, 计算  $y$  在目标空间的位置  $F(y) = \{f_1(y), \dots, f_m(y)\}$ , 并与每个子空间坐标依次比较, 判断  $y$  所在的子空间; 最后,  $y$  参与所在子空间非被占优解的选择.

$$\begin{cases} PF_1 : f_r(y) \leq f_r(X_{BP(1)}) \\ PF_{i+1} : f_r(X_{BP(i)}) < f_r(y) < f_r(X_{BP(i+1)}) \\ PF_{k+1} : f_r(y) \geq f_r(X_{BP(k)}) \end{cases} \quad (5)$$

式中  $i = 1, \dots, k - 1$ .

如图 5 所示,  $C$  为分割点,  $f_1(C) = f_1(X_c)$  为分割线, 子空间  $PF_1$  和  $PF_2$  的范围为  $(-\infty < f_1(x) \leq f_1(X_c), -\infty < f_2(x) < +\infty)$  和  $(f_1(X_c) < f_1(x), -\infty < f_2(x) < +\infty)$ . 以点  $C$  为例, 首先, 将  $C$  对应的元素  $X_C$  进行变异, 得到新个体  $y$ ; 然后, 判断  $y$  与子空间  $PF_1$  或  $PF_2$  的关系, 假设存在两种情况, 生成的新个体分别用  $C'$  和  $C''$  表示: 如果  $C'$  位于  $PF_1$  子空间, 即  $f_1(C') < f_1(X_c)$ , 那么,  $C'$  参与  $PF_1$  子空间非被占优解的选择; 如果  $C''$  位于  $PF_2$  子空间, 即  $f_1(C'') \leq f_1(X_c)$ , 那么,  $C''$  参与  $PF_2$

子空间非被占优解的选择.

局部 MOEA/D 选择非被占优解时, 要求解、理想点与方向向量的夹角小于  $\langle \lambda^i, \lambda^{i+1} \rangle / 2$ , 以获得更好的分布性能. 对于经交叉和变异产生的新解  $y$ , 满足如下条件时,  $y$  占优  $x$ , 为方向向量  $\lambda^i$  对应的解, 将  $y$  加入子空间的外部保存集中, 同时将  $x$  从子空间的外部保存集中删除.

1) 当  $x, y$  满足  $\langle \lambda^i, F(y) - z^* \rangle < \langle \lambda^i, \lambda^{i+1} \rangle / 2$  和  $\langle \lambda^i, F(x) - z^* \rangle > \langle \lambda^i, \lambda^{i+1} \rangle / 2$  时, 如图 6 (a) 所示;

2) 当  $x, y$  满足  $\langle \lambda^i, F(y) - z^* \rangle < \langle \lambda^i, \lambda^{i+1} \rangle / 2$  和  $\langle \lambda^i, F(x) - z^* \rangle < \langle \lambda^i, \lambda^{i+1} \rangle / 2$ , 且满足  $g(y|\lambda^i, z^*) < g(x|\lambda^i, z^*)$  时, 如图 6 (b) 所示;

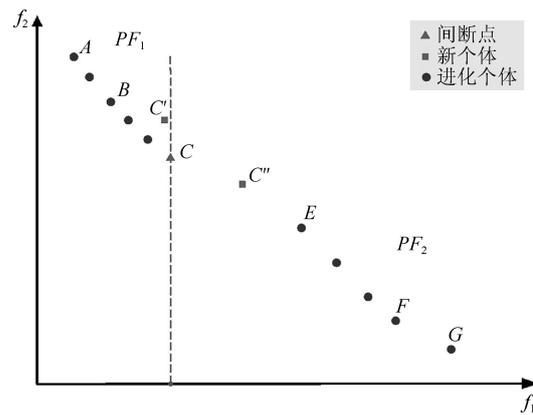


图 5 子空间信息交互

Fig. 5 Information interaction among subspaces

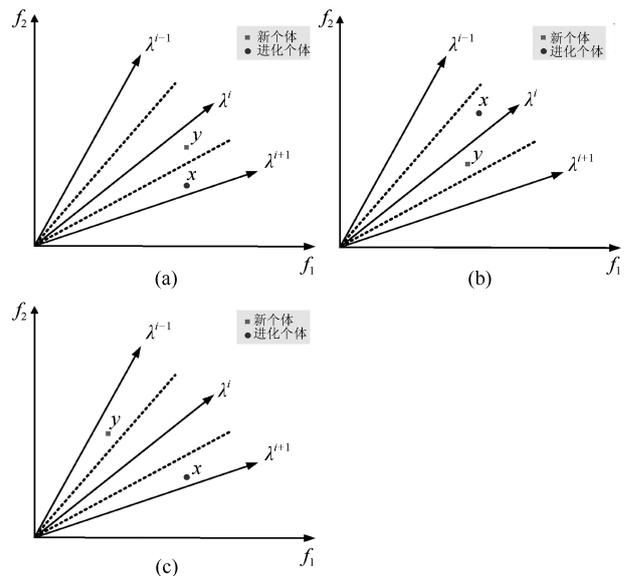


图 6 个体分布的调整

Fig. 6 Adjusting the distribution of individuals

3) 当  $x, y$  满足  $\langle \lambda^i, F(y) - z^* \rangle > \langle \lambda^i, \lambda^{i+1} \rangle / 2$  和  $\langle \lambda^i, F(x) - z^* \rangle > \langle \lambda^i, \lambda^{i+1} \rangle / 2$ , 且满足  $g(y|\lambda^i, z^*) < g(x|\lambda^i, z^*)$  时, 如图 6 (c) 所示.

其中,  $g(\cdot)$  为加权 Tchebycheff 函数, 对于第  $i$  个子问题, 该函数可以表示为:

$$\min g(x|\lambda^i, z^*) = \max_{1 \leq j \leq m} \{\lambda_j^i |f_j(x) - z_j^*|\},$$

$$x \in A(t)$$

式中,  $\langle \lambda^i, F(y) - z^* \rangle$  为方向向量  $\lambda^i$  和  $F(y) - z^*$  的夹角,  $\langle \lambda^i, \lambda^{i+1} \rangle$  为相邻方向向量的夹角, 如图 6(c) 所示.

### 2.5 Pareto 前沿形成

在种群进化中, 通常采用外部保存集保留优势个体, 但在 Pareto 前沿不连续这一类特殊问题中, 不连续部分的感知结果往往包含被占优解, 此时, 将感知结果保存在一个外部保存集中是不可行的. 因此, 本文提出每个子空间采用一个独立的外部保存集, 用于存放该子空间中非被占优的解. 子空间的保存集中, 不仅包含 Pareto 前沿的非被占优解, 也包含 Pareto 前沿间断部分的感知结果.

第 2.3 节将目标空间分割为若干个子空间, 每个子空间包含的个体数量是不规则的. 为了获得更好的分布性能, 本小节根据子空间内 Pareto 前沿的跨度, 设定子空间外部保存集的规模, 即 Pareto 前沿跨度较大的子空间, 需包含较多的个体. 子空间内个体的数量, 通过子空间外部保存集规模的调整实现.

子空间  $h$  外部保存集大小为  $|A_h(t)|$ . 对于子空间  $h$ , 包含  $|P_h(t)|$  个个体, 对于  $|P_h(t)|$  中的任意元素  $a_i$  和  $a_j$ , 计算子空间  $h$  内个体在目标函数  $r$  上的差  $|f_r(a_i) - f_r(a_j)|$ , 那么, 子空间 Pareto 前沿的跨度为  $L_h$ .

$$L_h = \sum_{r=1}^m \max_{a_i, a_j \in P_h(t)} |f_r(a_i) - f_r(a_j)|,$$

$$i, j = 1, \dots, |P_h(t)|, i \neq j \quad (6)$$

记优化问题 Pareto 前沿的跨度为  $L$ ,  $l$  为间断点个数, 那么,

$$L = \sum_{h=1}^{l+1} L_h \quad (7)$$

子空间  $h$  的外部保存集规模为

$$|A_h(t)| = \frac{L_h |A(t)|}{L} \quad (8)$$

确定子空间  $h$  的外部保存集规模后, 需要从  $P_h(t)$  选择个体保存到外部集合  $A_h(t)$  中. 子空间  $h$  包含  $|P_h(t)|$  个个体, 如果  $|P_h(t)|$  大于子空间的外部保存集规模  $|A_h(t)|$ , 那么, 删除拥挤距离最小的个体, 并从外部保存集中删除对应的元素; 如果存在多个最小拥挤距离相同的个体, 那么, 随机删除其中一个;

否则, 选择拥挤距离最大的个体, 进行交叉和变异运算, 生成新的个体, 并保存到该子空间的外部保存集中; 如果存在多个最大拥挤距离相同的个体, 那么, 随机选择其中任意一个, 进行交叉和变异运算, 直到子空间包含的个体数量等于子空间外部保存集的规模.

如图 7 所示,  $PF_1$  和  $PF_2$  分别为两个子空间, 子空间上 Pareto 前沿的跨度为  $L_1$  和  $L_2$ , 且  $L_1 < L_2$ . 比较  $PF_1$  和  $PF_2$  中个体数量  $|P_1(t)|$  和  $|P_2(t)|$  与外部保存集规模  $|A_1(t)|$  和  $|A_2(t)|$  大小. 如果  $|P_1(t)| > |A_1(t)|$ , 且  $|P_2(t)| < |A_2(t)|$ , 那么, 计算  $PF_1$  和  $PF_2$  中个体的拥挤距离, 删除  $PF_1$  中拥挤距离最小的个体  $C$ , 并更新  $PF_1$  的外部保存集; 选择  $PF_2$  中拥挤距离大的个体  $G$  进行交叉和变异操作, 得到  $G'$ , 并将  $G$  和  $G'$  保存到  $PF_2$  的外部保存集中, 直至  $|P_1(t)| = |A_1(t)|$  和  $|P_2(t)| = |A_2(t)|$ .

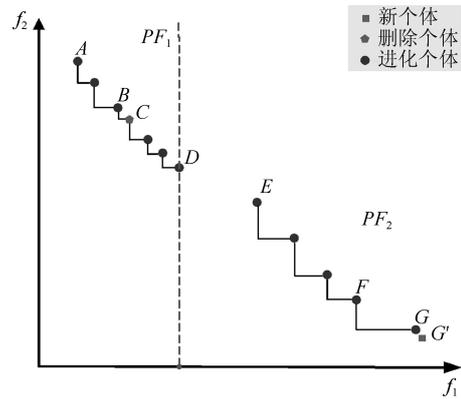


图 7 个体数量调整

Fig. 7 Adjusting the number of individuals

算法 3 提出调整子空间内个体数量的方法. 首先, 计算每个子空间 Pareto 前沿的跨度 (第 1 步至第 3 步); 然后, 采用式 (7), 计算目标空间中优化问题 Pareto 前沿的跨度  $L$  (第 4 步); 接着, 采用式 (8), 计算子空间外部保存集的规模 (第 6 步) 并根据子空间包含个体数量与外部保存集规模的关系, 增加或删除个体 (第 7 步至第 13 步); 最后, 输出子空间的外部保存集 (第 15 步).

#### 算法 3. 子空间外部保存集的生成

输入: 间断点个数  $l$ , 间断点集合  $BP = \{BP_1, \dots, BP_l\}$

输出: 子空间外部保存集  $A_h(t)$

- 1) for  $h = 1$  to  $l/2 + 1$
- 2) 根据式 (6), 计算各个子空间的跨度  $L_h$
- 3) end
- 4) 根据式 (7), 计算优化问题 Pareto 前沿的跨度  $L$
- 5) for  $h = 1$  to  $l/2 + 1$
- 6) 根据式 (8), 计算子空间  $h$  的外部保存集大小

```

|Ah(t)|
7) while |Ah(t)| ≠ |Ph(t)|
8)   if |Ah(t)| > |Ph(t)|
9)     find  $x = \min(C(x_i))$ , 产生新解  $y$ ,  $P_h(t) =$ 
 $P_h(t) \cup y$ 
10)   else
11)     find  $x = \max(C(x_i))$ ,  $P_h(t) = P_h(t) - x$ 
12)   end
13) end
14) end
15) return  $\{P_1(t), \dots, P_i(t), \dots, P_{l/2+1}(t)\}$ 

```

合并各子空间的外部保存集  $A_h(t)$ , 得到整个种群的外部保存集, 记为  $A(t) = \cup_{h=1}^{l/2+1} A_h(t)$ , 其中, 所有子空间保存集中个体的数量之和等于种群规模, 即  $N = \sum_{h=1}^{l/2+1} |A_h(t)|$ . 集合  $A(t)$  不仅包含全局最优解, 也包含描述间断区域的局部最优解, 以准确详细地感知 Pareto 前沿在间断区域的变化.

基于上述工作, 本文提出的基于在线感知 Pareto 前沿划分目标空间的多目标进化优化方法, (Multi-objective evolutionary optimization with objective space partition based on online perception of Pareto front) 记为 MOEA-PPF, 如算法 4 所示.

#### 算法 4. MOEA-PPF 算法

输入:  $N$ : 种群大小,  $T$ : 邻域大小,  $B(i) = \{i_1, \dots, i_T\} : \lambda^{i_1}, \dots, \lambda^{i_T}$  是方向向量  $\lambda^{i_1}$  最近的  $T$  个方向向量,  $\bar{z}$ : 理想点,  $\mathbf{z} = (z_1, \dots, z_r)^T$ ,  $z_i = \min\{f_i(\mathbf{x}^1), f_i(\mathbf{x}^2), \dots, f_i(\mathbf{x}^N)\}$ ,  $\alpha$ : 调整系数  
 输出: 外部保存集  $EP$

```

1) Initialize:
   初始种群  $\mathbf{x}^1, \dots, \mathbf{x}^N$ , 初始化  $\mathbf{z} = (z_1, \dots, z_r)^T$ ,  $z_i =$ 
 $\min\{f_i(\mathbf{x}^1), f_i(\mathbf{x}^2), \dots, f_i(\mathbf{x}^N)\}$ , 令  $gen = 1$ 
2) while  $gen < max\_gen$ 
3)   if  $gen == max\_gen/2$ 
4)     利用第 3.3 节, 分割空间;
5)     初始化子空间的外部保存集, 子空间邻域  $B'(i)$ ;
       理想点  $\mathbf{z}(1), \mathbf{z}(2), \dots, \mathbf{z}(l/2 + 1)$ 
6)   end
7)   if  $gen < max\_gen/2$ 
8)     采用 MOEA/D 方法, 更新外部保存集  $A(t)$ 
9)   else
10)    采用局部 MOEA/D 方法, 更新外部保存集
        $A_h(t)$ , 其中,  $h = 1, \dots, l/2 + 1$ 
11)   end
12)    $gen = gen + 1$ 
13) end
14) return  $EP = A_1(t) \cup \dots \cup A_{k+1}(t)$ 

```

在算法 4 中, 首先, 在种群进化前期, 采用传统 MOEA/D, 并将最优解保存在外部保存集  $A(t)$

中; 然后, 当种群进化到最大运行代数/2 时 (第 4 步), 根据算法 1 获得间断点; 采用算法 2 划分子空间; 依据算法 3 初始化子空间的外部保存集; 此外, 初始化子空间邻域  $B'(i)$  以及子空间的理想点  $\mathbf{z}(1), \mathbf{z}(2), \dots, \mathbf{z}(l/2 + 1)$  (第 5 步); 接着, 在种群进化后期, 采用局部 MOEA/D, 更新与子空间  $h$  对应的理想点  $\mathbf{z}(h)$  和外部保存集  $A_h(t)$  (第 10 步); 最后, 将子空间的外部保存集合并, 得到 Pareto 解集并输出  $EP$  (第 14 步).

### 3 实验与结果

本文基于拥挤距离搜索间断点, 在个体进化过程中, 主要基于 MOEA/D 方法, 因此与 MOEA/D 和其改进算法相比较是十分必要的. 本节通过实验, 评价所提方法的性能. 实验包括如下 2 部分: 1) 考察算法中参数  $\alpha$  的取值, 对优化问题 Pareto 前沿感知的影响; 2) 将提出的方法与 NSGA-II、RPEA、MOEA/D、MOEA/D-PBI、MOEA/D-STM 和 MOEA/D-ACD 等流行的多目标进化优化方法比较, 评价所提方法的性能.

#### 3.1 优化问题和性能指标

ZDT 和 WFG 是目前应用最广泛的基准优化问题. 这些基准优化问题的目标函数个数和决策变量个数都可以由用户给定. WFG 系列问题具有很多特性, 求解难度高. 本文选择 ZDT1、ZDT2、ZDT3、ZDT4、ZDT6、KUR 和 WFG1-WFG9<sup>[39-41]</sup> 作为测试优化问题. 在这些子问题中, ZDT1、ZDT2、ZDT4、ZDT6、WFG3、WFG4、WFG5、WFG6、WFG7 和 WFG9 具有连续的 Pareto 前沿; ZDT1 函数的 Pareto 前沿分布均匀、凸型、没有局部极值, 用来测试算法求解凸 Pareto 前沿优化问题的能力; ZDT2 函数的 Pareto 前沿是凹型、没有局部极值; ZDT3 函数具有 5 段不连续的 Pareto 前沿, 用来检测算法求解不连续 Pareto 前沿优化问题的能力; ZDT4 是比较复杂的函数, 第一个变量的取值范围与其他变量不同, 且包含局部极值, 算法极易陷于局部极值 (伪 Pareto 前沿), 它的 Pareto 前沿形状与 ZDT1 函数相同, 该函数检测算法跳出局部最优解的能力; ZDT6 函数的 Pareto 前沿与 ZDT1 函数相似, 但它的 Pareto 前沿分布不均匀. KUR 的 Pareto 前沿由 2 个不连续的曲线组成, WFG4-WFG9 的 Pareto 前沿面为凹函数, WFG1 具有混合的 Pareto 前沿, 且由 5 段不连续的曲线组成, WFG2 具有凸且分离的 Pareto 前沿, WFG3 具有线性且退化的 Pareto 前沿, WFG8 则具有凹且不连续的 Pareto 前沿. 这些函数用来测试算法求解 Pareto 前沿是凹、凸、连续、不连续和包含多个局

部极值优化问题的能力. 由于这些测试函数的前沿不同, 该系列问题非常适合于考察本文算法的性能.

实验中, 采用广泛使用的世代距离 (Generation distance, GD) 测度, 反映某方法所得 Pareto 前沿的收敛性能, 采用反世代距离 (Inverted generation distance, IGD) 测度<sup>[42]</sup>, 反映某方法所得 Pareto 前沿的综合性能, 此外, 提出完备率 (Completeness rate, CR) 测度, 评价某方法所得 Pareto 前沿的分布性能.

记  $P^*$  为优化问题真实的 Pareto 解集,  $P$  为采用任一方法所得的 Pareto 解集, 那么,  $P$  的 GD 值和 IGD 值可以表示为:

$$GD(P^*, P) = \frac{\sum_{v \in P} d(v, P^*)}{|P|} \quad (9)$$

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (10)$$

式 (9) 中,  $d(v, P^*)$  为  $P$  中的元素  $v$  到  $P^*$  的距离,  $|P|$  为  $P$  中元素的个数; 式 (10) 中,  $d(v, P)$  为  $P^*$  中的元素  $v$  到  $P$  的距离,  $|P^*|$  为  $P^*$  中元素的个数. Pareto 解集的 GD 值越小, 那么, 该方法所得 Pareto 前沿的收敛性能越好; Pareto 解集的 IGD 值越小, 那么, 该方法所得 Pareto 前沿的收敛性和分布性越好, 从而综合性能越好.

计算 Pareto 解集的 CR 测度时, 首先, 将目标函数均匀划分为若干区间; 然后, 将 Pareto 前沿上的个体向目标函数投影; 最后, 统计包含投影的区间占整个区间的比.  $P$  的 CR 值可以表示为:

$$CR(P^*, P) = \sum_{r=1}^m \frac{flag(r, k)}{|P|}, \quad k = 1, \dots, \frac{|P^*|}{m} \quad (11)$$

式中,  $flag(r, k)$  是一个标志函数, 如果第  $r$  个目标函数的第  $k$  个区间内存在 Pareto 前沿上个体的投影, 那么, 该函数的值为 1; 否则为 0, 即  $CR \in [0, 1]$ . Pareto 解集的 CR 值越大, 那么, Pareto 前沿越接近优化问题真实的 Pareto 前沿, 从而该方法对 Pareto 前沿的感知能力越强.

#### 算法 5. 计算 CR 测度

输入:  $P^*, P, m$

输出: CR

- 1) for  $r = 1$  to  $m$
- 2) for  $k = 1$  to  $(|P^*|/m)$
- 3)  $flag(r, k) = 0$ ;
- 4) end
- 5) end
- 6) for  $r = 1$  to  $m$
- 7)  $Scale(r) = (\max(P^*(r)) - \min(P^*(r))) / (|P^*|/m)$
- 8) for  $j = 1$  to  $|P|$

- 9) for  $k = 1$  to  $(|P^*|/m) - 1$
- 11) if  $(\min(P^*(r)) + k \times Scale(r) < P(j, r))$   
 $\leq \min(P^*(r)) + (k + 1) \times Scale(r))$
- 11)  $flag(r, k) = 1$ ;
- 12) end
- 13) end
- 14) end
- 15) end
- 16) return  $CR = (\sum_{r=1}^m \sum_{k=1}^{(|P^*|/m)} flag(r, k)) / |P|$

### 3.2 对比方法与参数设置

选择 NSGA-II、RPEA、MOEA/D、MOEA/D-PBI、MOEA/D-STM 和 MOEA/D-ACD, 作为对比方法, 评价所提方法的性能. 上述方法主要依据均匀分布的方向向量和个体之间的拥挤距离等选择策略, 没有感知和充分利用优化问题的 Pareto 前沿的信息, 对 Pareto 前沿的感知, 是发现问题特性的一种途径, 根据所研究问题的特性, 设计有针对性的方法, 而优化方法的针对性, 则决定了求解问题的效率. 本文基于进化求解过程获取的数据, 在线感知问题的 Pareto 前沿, 基于感知的 Pareto 前沿, 设计 MOEA-PPF. 由于本文采用局部 MOEA/D 求解优化问题, 因此, 选择 MOEA/D 及其改进算法进行对比, 是十分必要的. 此外, NSGA-II 的个体选择和外部保存集更新策略比较有代表性, 因此, 与该方法对比, 能够评价所提方法的性能. RPEA 是基于参考点引导的进化算法, 与本文基于参考点 (向量) 的方法相似, 因此本文方法非常有必要与 RPEA 对比.

所有的方法均采用相同的种群规模和进化终止代数, MOEA-PPF 设置  $\alpha$  为 13, 比较所得的 Pareto 前沿完备性和 IGD 性能.

种群大小: 选择的对比方法和测试用例统一使用  $N = 500$

运行次数和终止条件: 每个算法独立运行 30 次. 在进化 800 代后, 算法终止. MOEA/D、MOEA/D-PBI、MOEA/D-STM、MOEA/D-ACD 和 MOEA-PPF 邻域大小设置为  $T = 20$ .

RPEA 的参数设置:  $a_{ref} = 0.4$ ,  $\delta = 0.05$ . MOEA-PPF 调整系数设置为:  $\alpha = 13$ .

### 3.3 实验结果与分析

#### 1) $\alpha$ 取值对优化问题 Pareto 前沿感知的影响

参数  $\alpha$  的设置对 MOEA-PPF 的性能起关键的作用. 本部分通过采用 MOEA-PPF 求解 ZDT3, 考察参数  $\alpha$  的影响. 图 8 分别展示了种群进化 800 代后, 不同  $\alpha$  的 IGD 指标、GD 指标和运算时间; 图 9 展示了  $\alpha$  为 3、4、13、22 以及 23 时, IGD 随进化代数变化的曲线, 其中, 图 9 (b) 为图 9 (a) 的局部放大图.

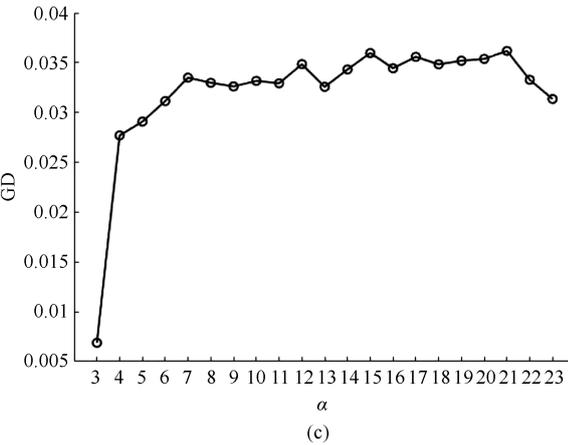
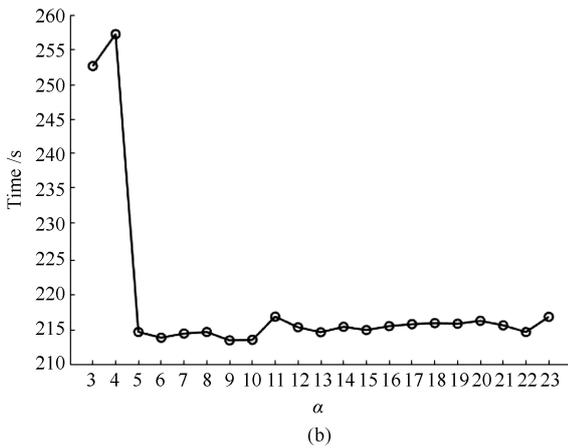
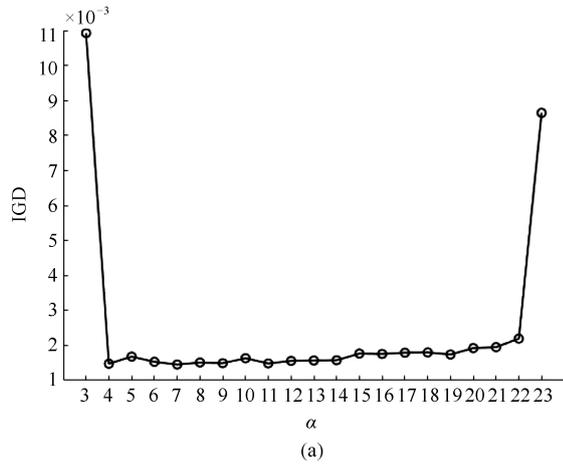


图 8 不同时  $\alpha$  的 IGD、GD 和运算时间

Fig.8 Curves of IGD, GD and time for different values of  $\alpha$

从图 8(a) 可以看出, 参数  $\alpha$  的值增大到 5 时, IGD 指标下降的较快, 这说明, 较小的  $\alpha$  值阻碍进化, 获得的 IGD 性能较差; 参数  $\alpha$  的值增大到 14 时, IGD 指标无变化, 此时, MOEA-PPF 感知的 Pareto 前沿相似; 参数  $\alpha$  的值增大到 21 时, IGD 指标缓慢增长, 这意味着, 随着  $\alpha$  增大, 算法对 Pareto 前沿的感知能力降低; 参数  $\alpha$  的值增大到 23 时,

IGD 指标快速上升, 这意味着, 过大的  $\alpha$  将明显阻碍算法获得的 Pareto 前沿的性能.

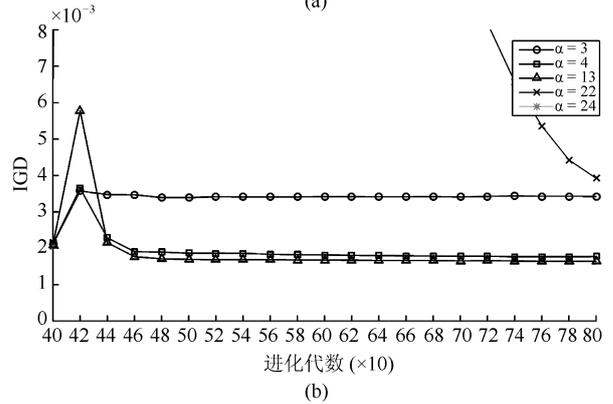
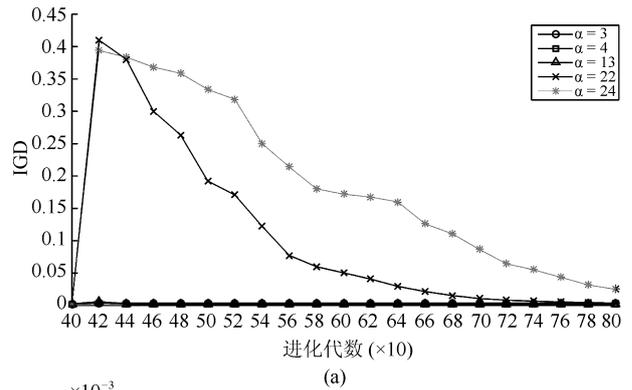


图 9 IGD 指标随  $\alpha$  的变化曲线

Fig.9 Curves of IGD with respect to the number of generations for different values of  $\alpha$

从图 8(b) 可以看出, 参数  $\alpha$  的值增大到 7 时, GD 指标上升的快, 这说明, 较小的  $\alpha$  值促进种群的收敛, 获得的 GD 性能较好; 参数  $\alpha$  的值增大到 21 时, GD 指标缓慢增长, 这意味着, 随着  $\alpha$  增大, 种群的收敛性能略有降低; 参数  $\alpha$  的值增大到 23 时, GD 指标略有下降, 这意味着, 种群的收敛性能提升.

从图 8(c) 可以看出, 参数  $\alpha$  的值增大到 5 时, 运算时间快速下降, 这说明, 较小的  $\alpha$  值阻碍进化, 消耗运算时间较长; 参数  $\alpha$  的值增大到 23 时, 运算时间无明显变化,  $\alpha$  的值继续增大, 不会明显增加算法的运算时间.

根据图 8 可知, 参数  $\alpha$  的取值, 需要综合分析算法的 IGD、GD 和运算时间. 较小的  $\alpha$  值, 可以获得较好的 GD 指标, 但 IGD 指标较差, 运算时间较长; 较大的  $\alpha$  值, 可以获得较好的 GD 指标, 但 IGD 指标较差; 对比 GD 和 IGD 的定义可知, GD 反应种群的收敛性能, IGD 综合反应种群的收敛性能和分布性能, 在  $\alpha$  取值时, 优先考虑 IGD 值和运算时间, 再考虑 GD 值.

图 9 为参数  $\alpha$  选择 3、4、13、22 以及 23 时, IGD 随进化代数变化的曲线. 从图 9 中可以看出, 1)

参数  $\alpha = 13$  时, IGD 指标最优. 这说明, 算法感知的 Pareto 前沿具有良好的性能; 2) 算法运行初始阶段, IGD 指标上升, 随后,  $\alpha$  为 3、4、13 的 IGD 指标在大约 60 代后, IGD 指标的变化趋于平缓,  $\alpha$  选择 22 和 23 时, IGD 指标下降较慢. 这说明, 算法运行初始化阶段, 子空间重新分配, 导致 IGD 指标变差, 但在  $\alpha$  选取合适的情况下, IGD 指标可快速下降. 综上所述,  $\alpha$  应设置为一个合理的值, 以取得最佳的综合性能.

图 10 为  $\alpha$  选择 3、13、22 以及 23 时, 所获得的 Pareto 前沿. 从中可以看出, 在  $\alpha = 13$  时, 可准确感知 Pareto 前沿在间断区域的变化.

2) 与其他方法的对比

表 1 和 2 列出不同方法求解 ZDT1、ZDT2、ZDT3、ZDT4、ZDT6、KUR 和 WFG1-WFG9 所得 Pareto 解集的 IGD 和 CR 值.

首先, 针对 Pareto 前沿连续的测试问题, 比较各方法所得 Pareto 解集的 IGD 值. 当求解测试问题 ZDT1 和 ZDT2 时, 对于 IGD 测度, 方法优劣的排

序为: MOEA/D-ACD、MOEA/D-STM、MOEA-PPF、MOEA/D、MOEA/D-PBI、RPEA 和 NSGA-II. ZDT1 和 ZDT2 的 Pareto 前沿均为连续, MOEA/D-ACD 选择优势个体时, 依据个体与方向向量的夹角, 因此, 具有更好的分布性; MOEA/D-STM 依据个体与方向向量的距离选择优势个体, 性能劣于 MOEA/D-ACD; MOEA-PPF 没有搜索到间断点, 无法划分目标空间, 此外, 采用 MOEA-PPF 对非被占优解的选择与 MOEA/D 相同, 因此, 与 MOEA/D 具有类似的性能, 两种方法的 IGD 指标不具有显著差异; MOEA/D-PBI 采用 PBI 聚合方法, 收敛性能劣于 MOEA/D; RPEA 采用参考点引导进化, 收敛性能优于 NSGA-II.

考虑测试问题 ZDT4 的 IGD 测度, 这些方法的优劣排序为: MOEA/D-STM、MOEA-PPF、MOEA/D-ACD、MOEA/D、MOEA/D-PBI、RPEA 和 NSGA-II. ZDT4 包括周期分量, MOEA/D-STM、MOEA-PPF、MOEA/D、MOEA/D-PBI、RPEA 和 NSGA-II 保持与求解

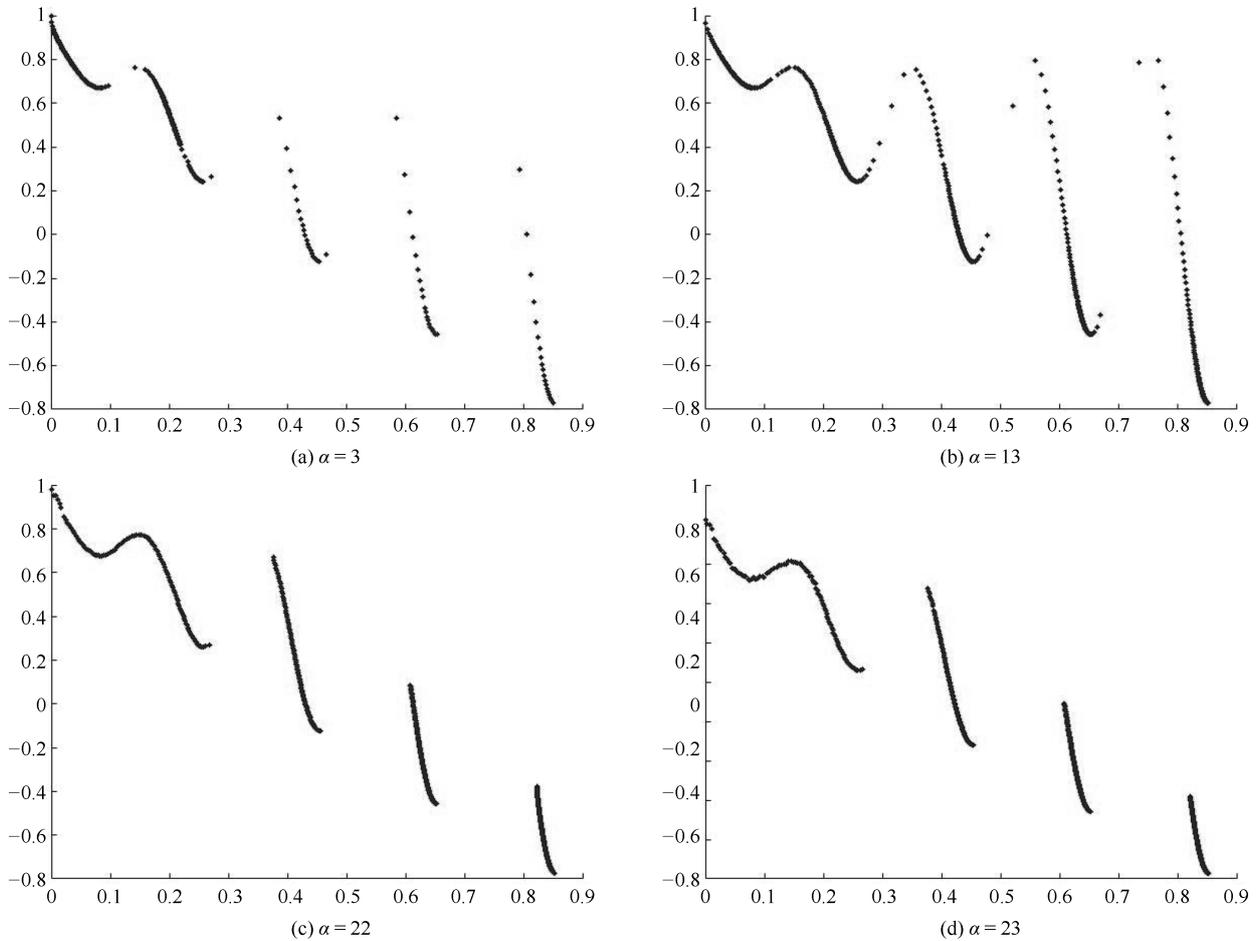


图 10  $\alpha$  取不同值时, ZDT3 的 Pareto 前沿  
Fig. 10 Pareto front of different values of  $\alpha$  for ZDT3

表 1 不同方法求解测试函数的 IGD 值  
Table 1 The values of IGD obtained by different algorithms

优化问题	NSGA-II	RPEA	MOEA/D	MOEA/D-PBI	MOEA/D-STM	MOEA/D-ACD	MOEA-PPF
ZDT1	$6.546 \times 10^{-3}\dagger$	$2.260 \times 10^{-3}\dagger$	$8.708 \times 10^{-4}$	$1.608 \times 10^{-3}\dagger$	$7.637 \times 10^{-4}\dagger$	<b><math>7.629 \times 10^{-4}\dagger</math></b>	$8.668 \times 10^{-4}$
ZDT2	$1.685 \times 10^{-2}\dagger$	$4.700 \times 10^{-3}\dagger$	$8.392 \times 10^{-4}$	$1.110 \times 10^{-3}\dagger$	$7.507 \times 10^{-4}\dagger$	<b><math>7.502 \times 10^{-4}\dagger</math></b>	$8.368 \times 10^{-4}$
ZDT3	$2.647 \times 10^{-3}\dagger$	$3.534 \times 10^{-3}\dagger$	$2.038 \times 10^{-3}\dagger$	$3.978 \times 10^{-3}\dagger$	$2.060 \times 10^{-3}\dagger$	$2.050 \times 10^{-3}\dagger$	<b><math>1.620 \times 10^{-3}</math></b>
ZDT4	$2.463 \times 10^{-2}\dagger$	$2.198 \times 10^{-3}\dagger$	$7.791 \times 10^{-4}$	$1.277 \times 10^{-3}\dagger$	<b><math>7.637 \times 10^{-4}\dagger</math></b>	$7.819 \times 10^{-4}$	$7.719 \times 10^{-4}$
ZDT6	$8.319 \times 10^{-4}\dagger$	$1.177 \times 10^{-1}\dagger$	$3.994 \times 10^{-4}$	$4.006 \times 10^{-4}$	<b><math>3.811 \times 10^{-4}\dagger</math></b>	$5.225 \times 10^{-4}\dagger$	$3.988 \times 10^{-4}$
KUR	$3.962 \times 10^{-1}$	$3.284 \times 10^{-2}\dagger$	$1.012 \times 10^{-2}$	$1.029 \times 10^{-2}$	$9.779 \times 10^{-3}\dagger$	$9.853 \times 10^{-3}$	<b><math>9.609 \times 10^{-3}</math></b>
WFG1	$6.949 \times 10^{-1}\dagger$	$4.685 \times 10^{-1}\dagger$	$4.553 \times 10^{-3}$	$3.248 \times 10^{-2}\dagger$	$8.218 \times 10^{-3}$	$4.680 \times 10^{-1}\dagger$	<b><math>3.940 \times 10^{-3}</math></b>
WFG2	$3.968 \times 10^{-3}$	$1.049 \times 10^{-3}\dagger$	$5.063 \times 10^{-3}$	$4.850 \times 10^{-2}\dagger$	$4.805 \times 10^{-3}$	$5.241 \times 10^{-3}$	<b><math>3.748 \times 10^{-3}</math></b>
WFG3	$3.198 \times 10^{-3}$	$5.316 \times 10^{-3}\dagger$	$2.953 \times 10^{-3}$	$4.498 \times 10^{-3}\dagger$	<b><math>2.597 \times 10^{-3}\dagger</math></b>	$2.747 \times 10^{-3}\dagger$	$2.951 \times 10^{-3}$
WFG4	$2.665 \times 10^{-2}\dagger$	$1.816 \times 10^{-2}\dagger$	$5.724 \times 10^{-3}$	$1.886 \times 10^{-2}\dagger$	$8.427 \times 10^{-3}\dagger$	$1.215 \times 10^{-2}\dagger$	<b><math>5.122 \times 10^{-3}</math></b>
WFG5	$6.206 \times 10^{-3}\dagger$	$6.777 \times 10^{-2}\dagger$	$6.326 \times 10^{-2}$	$6.852 \times 10^{-2}$	<b><math>6.179 \times 10^{-2}\dagger</math></b>	$6.287 \times 10^{-2}\dagger$	$6.510 \times 10^{-2}$
WFG6	$5.278 \times 10^{-2}\dagger$	$7.205 \times 10^{-2}\dagger$	$5.560 \times 10^{-2}$	$5.597 \times 10^{-2}$	$2.862 \times 10^{-3}\dagger$	<b><math>2.672 \times 10^{-3}</math></b>	$5.287 \times 10^{-2}$
WFG7	$3.186 \times 10^{-3}\dagger$	$3.014 \times 10^{-2}\dagger$	$2.663 \times 10^{-3}$	$3.850 \times 10^{-3}\dagger$	<b><math>2.585 \times 10^{-3}\dagger</math></b>	$2.666 \times 10^{-3}$	$2.689 \times 10^{-3}$
WFG8	$1.129 \times 10^{-1}$	$1.017 \times 10^{-1}\dagger$	$1.033 \times 10^{-1}\dagger$	$1.074 \times 10^{-1}$	$9.793 \times 10^{-2}\dagger$	<b><math>8.302 \times 10^{-2}\dagger</math></b>	$1.029 \times 10^{-1}$
WFG9	$1.915 \times 10^{-2}$	$1.272 \times 10^{-2}\dagger$	$1.533 \times 10^{-2}$	$2.155 \times 10^{-2}$	$1.229 \times 10^{-2}$	<b><math>8.062 \times 10^{-3}</math></b>	$1.522 \times 10^{-2}$

† 表示对比方法与本文方法的 IGD 指标具有显著差异 (Mann-Whitney U 分布检验, 置信水平为 0.05)

表 2 不同方法获得的 CR 值  
Table 2 Metric CR obtained by different algorithms

优化问题	NSGA-II	RPEA	MOEA/D	MOEA/D-PBI	MOEA/D-STM	MOEA/D-ACD	MOEA-PPF
ZDT1	0.9400	0.6860	<b>0.9440</b>	0.9360	<b>0.9440</b>	<b>0.9440</b>	<b>0.9440</b>
ZDT2	0.9260	0.6380	<b>0.9960</b>	0.9940	<b>0.9960</b>	<b>0.9960</b>	<b>0.9960</b>
ZDT3	0.6440	0.4340	0.5760	0.6460	0.5760	0.6720	<b>0.7480</b>
ZDT4	0.9340	0.6320	<b>0.9440</b>	0.9340	<b>0.9440</b>	0.9420	<b>0.9440</b>
ZDT6	0.9460	0.6640	<b>0.9960</b>	<b>0.9960</b>	<b>0.9960</b>	<b>0.9960</b>	<b>0.9960</b>
KUR	0.8020	0.6060	0.7800	<b>0.9160</b>	0.7780	0.8940	0.9140
WFG1	0.7320	0.3620	0.8700	0.8240	0.8480	0.4520	<b>0.8720</b>
WFG2	0.6100	0.3680	0.4880	0.6060	0.4860	0.6220	<b>0.7360</b>
WFG3	0.9460	0.8260	0.9920	<b>0.9960</b>	<b>0.9960</b>	<b>0.9960</b>	<b>0.9960</b>
WFG4	0.9280	0.5820	0.9500	0.9520	0.8440	0.8340	<b>0.9560</b>
WFG5	0.9020	0.5760	0.9180	0.9180	<b>0.9540</b>	0.9140	0.9200
WFG6	0.9000	0.5500	0.9460	0.9580	<b>0.9620</b>	<b>0.9620</b>	0.9580
WFG7	0.9240	0.5780	0.9660	<b>0.9700</b>	0.9680	0.9620	0.9680
WFG8	0.5400	0.3500	0.9060	<b>0.9220</b>	0.9140	0.9180	0.9000
WFG9	0.9220	0.6340	0.9260	<b>0.9280</b>	0.8920	0.9220	0.8980

ZDT1 和 ZDT2 相同的性能排序; MOEA/D-ACD 保留了与某些与方向向量距离较远的个体, 使得性能劣于 MOEA/D-STM 和 MOEA-PPF.

对于 IGD 测度, 不同方法求解 ZDT6 的性能优劣排序为: MOEA/D-STM、MOEA-PPF、MOEA/D、MOEA/D-PBI、MOEA/D-ACD、NSGA-II 和 RPEA. 此时, MOEA/D-ACD 产生的部分解用于描述 Pareto 前沿在  $f_2 \in [1, 10]$

的变化趋势, 导致 IGD 性能劣于 MOEA/D.

当求解测试问题 WFG1 时, MOEA-PPF 显著优于 MOEA/D-PBI、MOEA/D-ACD、NSGA-II 和 RPEA, 与 MOEA/D-STM 和 MOEA/D 方法相比, MOEA-PPF 虽然得到了更好的 IGD 指标, 但不具备显著性差异, 这是由于 WFG1 测试问题的 Pareto 前沿为连续的, 本文方法采用非被占优解的选择与 MOEA/D 相同; 当求解测试问

题 WFG4 时, MOEA-PPF 显著优于 MOEA/D-STM、MOEA/D-PBI、MOEA/D-ACD、NSGA-II 和 RPEA, 本文方法与 MOEA/D 具有相同的选择策略, 两种算法不具备显著性差异。

当求解测试问题 WFG3、WFG5、WFG6、WFG7 和 WFG9 时, MOEA-PPF 和 MOEA/D 获得的 IGD 指标不具备显著性差异, 本文方法的收敛性和分布性与 MOEA/D 方法相同。

然后, 针对 Pareto 前沿不连续的优化问题, 比较各方法所得 Pareto 解集的 IGD 值。从表 1 可以看出, 求解 ZDT3、KUR 和 WFG2 时, MOEA-PPF 的性能最优。对于 ZDT3 的 IGD 测度, 方法的性能优劣排序为: MOEA-PPF、MOEA/D、MOEA/D-ACD、MOEA/D-STM、NSGA-II、RPEA 和 MOEA/D-PBI。这是因为 ZDT3 的 Pareto 前沿由 5 个分段曲线组成, MOEA-PPF 划分目标空间, 并根据各段曲线的分布调整每一子空间上进化种群的规模, 且通过目标空间的划分, 避免了 Pareto 前沿间断点附近个体的聚集, 具有更好的分布性能; MOEA/D-ACD 虽然利用部分个体描述 Pareto 前沿在间断区域的分布, 但是, 没有避免个体在间断点附近的聚集, 因此, 性能劣于 MOEA/D, 仅略优于 MOEA/D-STM。对分段 Pareto 前沿的描述, 最优的是 MOEA-PPF, 其次是 MOEA/D-ACD。

对于 KUR 和 WFG2, 它们的 Pareto 前沿分别由 3 和 5 个分段曲线组成。MOEA/D-ACD 利用部分个体描述间断区域的 Pareto 前沿, 导致 IGD 性能变差。但是, 求解 WFG2 时, NSGA-II 优于 MOEA/D、MOEA/D-PBI、MOEA/D-ACD、MOEA/D-STM 和 RPEA, 主要由于 NSGA-II 依据占优关系选择优势个体, 避免了个体聚集, 从而获得更多的个体描述 Pareto 前沿, 得到更优的 IGD 值。

当求解测试问题 WFG8 时, 对比方法的性能优劣排序为: MOEA/D-ACD、MOEA/D-STM、MOEA-PPF、RPEA、MOEA/D、MOEA/D-PBI 和 NSGA-II。此时, MOEA-PPF 的收敛性能劣于 MOEA/D-ACD 和 MOEA/D-STM, 但在对比方法中, 只有 MOEA-PPF 和 MOEA/D-PBI 利用部分个体描述 Pareto 前沿在间断区域的分布, 两种算法的 CR 测度最高。

现在, 考察不同方法 CR 测度的对比。由表 2 可知, 对于 Pareto 前沿连续的测试问题 ZDT1、ZDT2、ZDT4、ZDT6 和 WFG3, MOEA-PPF 与 MOEA/D、MOEA/D-STM 和 MOEA/D-ACD 的性能相同; 对于 ZDT3、KUR 和 WFG2 问题, MOEA-PPF 明显优于其他方法。这说明, 在对

优化问题 Pareto 前沿的描述上, MOEA-PPF 具有显著优越性。

通过上述实验结果, 可以得到如下结论: 1) 对于求解 Pareto 前沿连续的优化问题, MOEA-PPF 的 IGD 性能劣于 MOEA/D-ACD 和 MOEA/D-STM, 但略优于 MOEA/D, 此外, 这些方法具有相近的 CR 性能; 2) 当求解 Pareto 前沿不连续的优化问题时, MOEA-PPF 的 IGD 和 CR 性能均明显优于对比方法。这说明, 本文方法是一种非常有竞争力的多目标优化方法。

## 4 结束语

具有复杂 Pareto 前沿的多目标优化问题是非常普遍的, 有效求解该问题的前提是及时准确地感知 Pareto 前沿。鉴于此, 本文提出一种基于在线感知 Pareto 前沿划分目标空间的多目标进化优化方法。基于种群进化的信息, 在线感知 Pareto 前沿的特性, 如果 Pareto 前沿是不连续的, 那么, 采用 MOEA-PPF 求解; 否则, 采用传统 MOEA/D 求解。将提出的方法分别应用于 15 个连续和不连续基准优化问题, 并与 6 个流行的多目标进化优化方法对比, 通过反世代距离和覆盖率等测度的实验结果表明, 本文方法的确是求解多目标优化问题的有效方法。

需要指出的是, 本文判定优化问题 Pareto 前沿的间断点时, 需要利用种群进化信息, 它们的质量直接影响间断点判定的准确性, 从而影响后续种群的进化方向。因此, 如何提高用于判定间断点的进化种群的质量, 是今后需要研究的问题。此外, 本文仅针对 Pareto 前沿不连续的优化问题, 给出了 MOEA-PPF 求解方法。实际的复杂优化问题, Pareto 前沿会有更多的表现特性, 如在某些区域分布密集, 而另一些区域分布稀疏、Pareto 前沿分段个数多等。对这些问题, 本文方法的求解性能往往难以令人满意。需要在 Pareto 前沿感知和进化求解等方面, 研究有针对性方法, 这也是今后需要进一步研究的问题。最后, 本文提出的方法能否成功应用于实际的优化问题, 需要进一步验证, 不但有利于方法的改进与完善, 也有助于实际问题的解决。

## References

- 1 Zuo Xing-Quan, Wang Chun-Lu, Zhao Xin-Chao. Combining multi-objective immune algorithm and linear programming for double row layout problem. *Acta Automatica Sinica*, 2015, **41**(3): 528–540  
(左兴权, 王春露, 赵新超. 一种结合多目标免疫算法和线性规划的双行设备布局方法. *自动化学报*, 2015, **41**(3): 528–540)
- 2 Han Y Y, Gong D W, Jin Y C, Pan Q K. Evolutionary multi-objective blocking lot-streaming flow shop scheduling with

- interval processing time. *Applied Soft Computing*, 2016, **42**: 229–245
- 3 Li Y M, Tong S C. Adaptive fuzzy output-feedback stabilization control for a class of switched nonstrict-feedback nonlinear systems. *IEEE Transactions on Cybernetics*, 2017, **47**(4): 1007–1016
- 4 Duan Kai-Rong, Zhang Gong-Xuan. Multi-objective immune system algorithm for task scheduling in cloud computing. *Journal of Computer Applications*, 2016, **36**(2): 324–329  
(段凯蓉, 张功萱. 基于多目标免疫系统算法的云任务调度策略. 计算机应用, 2016, **36**(2): 324–329)
- 5 Deb K. *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester: John Wiley & Sons, 2001. 277–293
- 6 Liu Yuan, Zheng Jin-Hua, Zou Juan, Yu Guo. An evolutionary algorithm through neighborhood competition for multi-objective optimization. *Acta Automatica Sinica*, 2018, **44**(7): 1304–1320  
(刘元, 郑金华, 邹娟, 喻果. 基于邻域竞赛的多目标优化算法. 自动化学报, 2018, **44**(7): 1304–1320)
- 7 Qiao Jun-Fei, Han Gai-Tang, Zhou Hong-Biao. Knowledge-based intelligent optimal control for wastewater biochemical treatment process. *Acta Automatica Sinica*, 2017, **43**(6): 1038–1046  
(乔俊飞, 韩改堂, 周红标. 基于知识的污水生化处理过程智能优化方法. 自动化学报, 2017, **43**(6): 1038–1046)
- 8 Ding Jin-Liang, Yang Cui-E, Chen Li-Peng, Chai Tian-You. Dynamic multi-objective optimization algorithm based on reference point prediction. *Acta Automatica Sinica*, 2017, **43**(2): 313–320  
(丁进良, 杨翠娥, 陈立鹏, 柴天佑. 基于参考点预测的动态多目标优化算法. 自动化学报, 2017, **43**(2): 313–320)
- 9 Gong Dun-Wei, Liu Yi-Ping, Sun Xiao-Yan, Han Yu-Yan. Parallel many-objective evolutionary optimization using objectives decomposition. *Acta Automatica Sinica*, 2015, **41**(8): 1438–1451  
(巩敦卫, 刘益萍, 孙晓燕, 韩玉艳. 基于目标分解的高维多目标并行进化优化方法. 自动化学报, 2015, **41**(8): 1438–1451)
- 10 Li H, Zhang Q F. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2009, **13**(2): 284–302
- 11 Geoffrion A M. Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 1968, **22**(3): 618–630
- 12 Zadeh L. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 1963, **8**(1): 59–60
- 13 Fonseca C M, Fleming P J. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: Proceedings of the 5th International Conference on Genetic Algorithms. San Mateo, USA: ACM, 1993. 416–423
- 14 Srinivas N, Deb K. Multiobjective optimization using non-dominated sorting in genetic algorithms. *Evolutionary Computation*, 1994, **2**(3): 221–248
- 15 Horn J, Nafpliotis N, Goldberg D E. A niched Pareto genetic algorithm for multiobjective optimization. In: Proceedings of the 1st IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence. Orlando, USA: IEEE, 1994. 82–87
- 16 Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. In: Proceedings of 2002 Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems. Berlin, Germany: Springer-Verlag, 2002. 95–100
- 17 Deb K, Pratap A, Agarwal A, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(2): 182–197
- 18 Corne D W, Jerram N R, Knowles J D, Oates M J. PESA-II: region-based selection in evolutionary multiobjective optimization. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. San Francisco, California, USA: Morgan Kaufmann Publishers, 2001. 283–290
- 19 Bader J, Zitzler E. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 2011, **19**(1): 45–76
- 20 Zitzler E, Künzli S. Indicator-based selection in multiobjective search. In: Proceedings of the 8th International Conference on Parallel Problem Solving from Nature. Birmingham, UK: Springer-Verlag, 2004. 832–842
- 21 Gu F Q, Liu H L. A novel weight design in multi-objective evolutionary algorithm. In: Proceedings of the 2010 International Conference on Computational Intelligence and Security. Nanning, China: IEEE, 2010. 137–141
- 22 Qi Y T, Ma X L, Liu F, Jiao L C, Sun J Y, Wu J S. MOEA/D with adaptive weight adjustment. *Evolutionary Computation*, 2014, **22**(2): 231–264
- 23 Cheng R, Jin Y C, Olhofer M, Sendhoff B. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2016, **20**(5): 773–791
- 24 Asafuddoula M, Singh H K, Ray T. An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors. *IEEE Transactions on Cybernetics*, 2018, **48**(8): 2321–2334
- 25 Liu Y P, Gong D W, Sun X Y, Zhang Y. Many-objective evolutionary optimization based on reference points. *Applied Soft Computing*, 2017, **50**: 344–355
- 26 Zhang Q F, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, **11**(6): 712–731
- 27 Li K, Zhang Q F, Kwong S, Li M Q, Wang R. Stable matching-based selection in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2014, **18**(6): 909–923

- 28 Wang L P, Zhang Q F, Zhou A M, Gong M G, Jiao L C. Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 2016, **20**(3): 475–480
- 29 Storn R, Price K. Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, **11**(4): 341–359
- 30 Dorigo M, Maniezzo V, Colomi A. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 1996, **26**(1): 29–41
- 31 Branke J, Mostaghim S. About selecting the personal best in multi-objective particle swarm optimization. *Parallel Problem Solving from Nature-PPSN IX*. Berlin, Island: Springer, 2006. 52–532
- 32 Lei De-Ming, Wu Zhi-Ming. Pareto archive multi-objective particle swarm optimization. *Pattern Recognition and Artificial Intelligence*, 2006, **19**(4): 475–480  
(雷德明, 吴智铭. Pareto 档案多目标粒子群优化. 模式识别与人工智能, 2006, **19**(4): 475–480)
- 33 Huang V L, Suganthan P N, Liang J J. Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems. *International Journal of Intelligent Systems*, 2006, **21**(2): 209–226
- 34 Gong Mao-Guo, Cheng Gang, Jiao Li-Cheng, Liu Chao. Nondominated individual selection strategy based on adaptive partition for evolutionary multi-objective optimization. *Journal of Computer Research and Development*, 2011, **48**(4): 545–557  
(公茂果, 程刚, 焦李成, 刘超. 基于自适应划分的进化多目标优化非支配个体选择策略. 计算机研究与发展, 2011, **48**(4): 545–557)
- 35 Deng L Y, Lin V, Chen M. Hybrid ant colony optimization for the resource-constrained project scheduling problem. *Journal of Systems Engineering and Electronics*, 2010, **21**(1): 67–71
- 36 Zheng W, Wu X X, Yang X B, Cao S C, Liu W X, Lin J. Test suite minimization with mutation testing-based many-objective evolutionary optimization. In: Proceedings of the 2017 International Conference on Software Analysis, Testing and Evolution. Harbin, China: IEEE, 2017. 32–36
- 37 Wang R, Ishibuchi H, Zhang Y, Zheng X K, Zhang T. On the effect of localized PBI method in MOEA/D for multi-objective optimization. In: Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI). Athens, Greece: IEEE, 2016. 1–8
- 38 Jiang S Y, Yang S X. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts. *IEEE Transactions on Cybernetics*, 2016, **46**(2): 421–437
- 39 Deb K, Thiele L, Laumanns M, Zitzler E. Scalable test problems for evolutionary multiobjective optimization. *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. London: Springer, 2005. 187–196
- 40 Kursawe F. A variant of evolution strategies for vector optimization. In: Proceedings of the 1st Workshop Parallel Problem Solving from Nature. Berlin, Heidelberg, Germany: Springer, 1991. 193–197
- 41 Huband S, Barone L, While L, Hingston P. A scalable multi-objective test problem toolkit. *Evolutionary Multi-Criterion Optimization*. Berlin, Heidelberg: Springer, 2005. 280–295
- 42 Zhang Q, Zhou A, Zhao S, Suganthan P N, Liu W, Tiwari S. Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition, Technical Report CES-487, University of Essex and Nanyang Technological University, Singapore, 2008.



**封文清** 中国矿业大学信息与控制工程学院博士研究生。2013 年获得黑龙江大学硕士学位。主要研究方向为多目标进化优化与应用。

E-mail: fwq\_cumt@163.com

(**FENG Wen-Qing** Ph.D. candidate at the School of Information and Control Engineering, China University

of Mining and Technology. She received her master degree from Heilongjiang University in 2013. Her research interest covers multi-objective evolutionary optimization and its applications.)



**巩敦卫** 中国矿业大学信息与控制工程学院教授。1999 年获得中国矿业大学博士学位。主要研究方向为进化计算与应用。本文通信作者。

E-mail: dwgong@vip.163.com

(**GONG Dun-Wei** Professor at the School of Information and Control Engineering, China University of Mining and

Technology. He received his Ph. D. degree from China University of Mining and Technology in 1999. His research interest covers evolutionary computation and its applications. Corresponding author of this paper.)