

# 基于目标分解的高维多目标并行进化优化方法

巩敦卫<sup>1,2</sup> 刘益萍<sup>1</sup> 孙晓燕<sup>1</sup> 韩玉艳<sup>1</sup>

**摘要** 高维多目标优化问题普遍存在且难以解决, 到目前为止, 尚缺乏有效解决该问题的进化优化方法. 本文提出一种基于目标分解的高维多目标并行进化优化方法, 首先, 将高维多目标优化问题分解为若干子优化问题, 每一子优化问题除了包含原优化问题的少数目标函数之外, 还具有由其他目标函数聚合成的一个目标函数, 以降低问题求解的难度; 其次, 采用多种群并行进化算法, 求解分解后的每一子优化问题, 并在求解过程中, 充分利用其他子种群的信息, 以提高 Pareto 非被占优解的选择压力; 最后, 基于各子种群的非被占优解形成外部保存集, 从而得到高维多目标优化问题的 Pareto 最优解集. 性能分析表明, 本文提出的方法具有较小的计算复杂度. 将所提方法应用于多个基准优化问题, 并与 NSGA-II、PPD-MOEA、 $\epsilon$ -MOEA、HypE 和 MSOPS 等方法比较, 实验结果表明, 所提方法能够产生收敛性、分布性, 以及延展性优越的 Pareto 最优解集.

**关键词** 进化优化, 高维多目标优化, 分解, 并行, Pareto 占优

**引用格式** 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

**DOI** 10.16383/j.aas.2015.c140832

## Parallel Many-objective Evolutionary Optimization Using Objectives Decomposition

GONG Dun-Wei<sup>1,2</sup> LIU Yi-Ping<sup>1</sup> SUN Xiao-Yan<sup>1</sup> HAN Yu-Yan<sup>1</sup>

**Abstract** Many-objective optimization problem is common in real-world applications, however, so far few evolutionary algorithms are suitable for them due to the difficulties of the problem. A parallel many-objective evolutionary optimization algorithm based on objectives decomposition is proposed. First, the many-objective optimization problem is decomposed into several sub-problems, which contain only some objectives of the original optimization problem together with a constructed objective by aggregating all the other objectives. Then, a multi-population parallel evolutionary algorithm is adopted to solve these sub-problems. The pressure on selecting non-dominated solutions for a sub-problem is improved by taking full advantage of the information obtained from other sub-populations. The final Pareto set of the optimized many-objective is achieved by archiving those sets of non-dominated solutions coming from the sub-populations. The performance of the proposed algorithm on reducing computation complexity is qualitatively analyzed. Furthermore, the algorithm is applied to several benchmark problems and compared with NSGA-II, PPD-MOEA,  $\epsilon$ -MOEA, HypE, and MSOPS. The results experimentally demonstrate that the algorithm is strengthened in obtaining solutions with better convergence, distribution and approximation.

**Key words** Evolutionary algorithm, many-objective optimization, decomposition, parallel, Pareto domination

**Citation** 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

在现实生活中, 存在很多包含多个目标的优化

问题, 如控制器设计<sup>[1]</sup>、电能调度<sup>[2]</sup>、半导体制造<sup>[3]</sup>等. 其共同点在于, 它们不仅包含多个目标, 而且这些目标之间是相互冲突的, 即不能找到一个解, 使其在所有目标上都达到最优, 这类优化问题通常称为多目标优化问题. 如果一个优化问题包含四个及以上目标, 那么, 该问题就称为高维多目标优化问题. 不失一般性, 本文考虑如下多目标优化问题:

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

式中,  $\mathbf{x}$  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

收稿日期 2014-12-01 录用日期 2015-04-08  
Manuscript received December 1, 2014; accepted April 8, 2015  
国家重点基础研究发展计划 (973 计划) (2014CB046306-2), 国家自然科学基金 (61375067), 江苏省自然科学基金 (BK2012566) 资助  
Supported by National Basic Research Program of China (973 Program) (2014CB046306-2), National Natural Science Foundation of China (61375067), and Natural Science Foundation of Jiangsu Province (BK2012566)  
本文责任编辑 沈轶  
Recommended by Associate Editor SHEN Yi  
1. 中国矿业大学信息与电气工程学院 徐州 221116 2. 兰州理工大学电气工程与信息工程学院 兰州 730050  
1. School of Information and Electronic Engineering, China University of Mining and Technology, Xuzhou 221116 2. College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050

数个数, 当  $m \geq 4$  时, 该问题是一个高维多目标优化问题. 作为典型的多目标进化优化方法 (Multi-objective evolutionary algorithm, MOEA), NSGA-II<sup>[4]</sup>、SPEA2<sup>[5]</sup> 等已成功地应用于大量多目标优化问题. 但是, 它们都基于 Pareto 占优关系进行个体比较, 因此, 仅适用于包含较少目标的优化问题. 实际的优化问题可能涉及高维多目标, 如果仍然基于已有的 Pareto 占优关系比较个体, 那么, 上述方法的应用效果将很差. 这是因为随着目标函数的增多, 种群中非被占优个体的数量将指数增加, 从而降低了 Pareto 最优解选择的压力.

为了将上述方法应用于高维多目标优化问题, 目前主要有三类途径: 1) 通过目标函数缩减<sup>[6-7]</sup>, 删除优化问题中冗余的目标函数, 将高维多目标优化问题转化为传统的多目标优化问题. 但是, 该方法仅适用于优化问题存在冗余目标函数的情况. 2) 改进密度估计策略<sup>[8-9]</sup>, 减少远离 Pareto 前沿的非被占优解数量, 提高算法的收敛性能. 但是, 该方法可能导致解集多样性降低. 3) 修改已有的 Pareto 占优关系<sup>[10-13]</sup>, 使得个体之间能够相互比较, 以提高 Pareto 最优解的选择压力. 但是, 该方法往往难以得到完整的 Pareto 前沿. 例如, 基于 Pareto 局部占优的 MOEA<sup>[10-11]</sup> 仅采用优化问题的部分目标函数, 计算进化个体的 Pareto 序值. 由于无法权衡所有的目标函数, 使得得到的优化解并非都是问题的全局最优解. 因此, 寻求更为有效的方法, 解决高维多目标优化问题, 是十分有意义的.

采用进化优化方法求解高维多目标优化问题时, 为了比较不同个体的性能, 需要计算该个体的所有目标函数值, 并对这些个体进行 Pareto 占优排序, 这往往需要消耗很长的时间. 多种群并行进化算法采用多个子种群协同进化, 以求解复杂的优化问题, 能够大大提高问题求解的速度. 如果采用多种群并行进化算法, 求解高维多目标优化问题时, 每一子种群优化不同的少量目标函数, 并基于 Pareto 局部占优, 比较子种群中进化个体的性能, 那么将无疑能够提高优化问题求解的效率.

鉴于此, 本文提出一种基于目标分解的高维多目标并行进化优化方法, 主要思想是: 基于相关关系将优化问题的目标函数分为若干组, 并在每组目标中加入一个聚合目标, 形成若干子优化问题; 在并行环境下, 采用多种群并行进化算法求解转化后的优化问题时, 每一子种群通过进化求解一个子优化问题, 并在求解过程中, 利用其他子种群的优势个体; 基于所有子种群产生的非被占优个体, 采用新的保存集形成策略, 构建优化问题的 Pareto 最优解集.

本文方法的创新之处在于: 1) 将高维多目标优化问题转化为若干个目标函数更少的多目标优化

(子) 问题, 从而易于采用传统的多目标进化优化方法求解; 2) 基于 Spearman 相关系数对目标函数分组, 保证了同一组的目标函数具有较高的相似度, 为子优化问题的高效求解提供了保障; 3) 在每一子问题中加入了聚合目标, 以融入全局优化信息, 有利于寻找优化问题的全局最优解; 4) 对于每组目标函数形成的子优化问题, 通过多种群并行进化算法的一个子种群, 在同构并行环境的一个进程上进化, 提高了问题求解的速度; 5) 采用改进的外部保存集形成策略, 基于各子种群的最优个体, 在一个新的进程上形成外部保存集, 保证了优化问题 Pareto 最优解集的逼近与分布性能.

本文的结构安排如下: 第 1 节综述相关工作; 提出的方法在第 2 节详细阐述; 第 3 节是所提方法在多个基准优化问题中的应用及对比实验; 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

### 1.1 高维多目标进化优化

作为一种特殊的多目标优化问题, 高维多目标优化问题的目标函数通常不少于 4 个. 若仍采用基于 Pareto 占优的 MOEA 求解该类问题, 其求解效率往往很低, 甚至难以得到问题的最优解. 引言中所述的三类改进方法, 使得基于 Pareto 占优的 MOEA 能够求解高维多目标优化问题. 除此之外, 还存在如下三类非 Pareto 占优方法: 1) 基于优化问题的所有目标, 计算新指标的值得, 评价个体的性能, 如 IBEA<sup>[14]</sup>、HypE<sup>[15]</sup> 等; 2) 采用目标聚合方法, 将高维多目标优化问题转化为若干单目标优化问题, 并行搜索 Pareto 最优解, 如 MSOPS<sup>[16]</sup>、MOEA/D<sup>[17]</sup> 等; 3) 采用集合进化优化方法, 求解高维多目标优化问题, 如文献 [18] 和文献 [19] 中提出的方法.

在上述方法中, Sato 等<sup>[10]</sup> 提出的基于 Pareto 局部占优的多目标进化算法 (PPD-MOEA), 仅利用部分目标函数比较个体的性能, 从而提高了 Pareto 最优解的选择压力. 在 PPD-MOEA 中, 算法以 NSGA-II 为框架, 采用 Pareto 局部占优关系, 代替传统的 Pareto 占优关系, 对个体进行占优等级排序; 在进化过程中, 通过不断变化这些部分目标, 使得所有目标都能够被优化; 此外, 基于占优区域控制, 更新外部保存集, 以提高 Pareto 最优解集的性能.

Jaimes 等<sup>[11]</sup> 也基于 Pareto 局部占优关系, 比较候选解的性能. 但是, 他们选择部分目标时, 基于目标函数之间的线性相关关系. 实验结果表明, 基于线性相关关系选择部分目标, 比随机选择部分目标取得了更好的优化结果.

但是, 基于 Pareto 局部占优关系, 得到的最优

解难以合理权衡所有目标函数. 这是因为, 采用上述方法得到的非被占优解往往不是全局占优的; 尽管全局占优解一定是局部占优解, 但是, 局部占优解却不一定是全局占优解.

为了克服上述缺陷, 本文通过目标聚合, 在已选择的部分目标中加入一个新的目标, 该目标融合了其余未选择目标的信息, 从而采用 Pareto 局部占优关系比较个体性能时, 能够权衡所有目标.

## 1.2 多种群并行进化优化

所谓多种群并行进化优化, 是指多个子种群并行进化, 以求解某一优化问题. 该方法将整个种群分为多个子种群, 且每个子种群独立进化; 在子种群进化过程中, 通过迁徙算子引入其他子种群的优势个体, 以提高子种群的性能. 可以采用多种方式迁徙子种群之间的个体, 常用的方式包括: 1) 完全迁徙, 每个子种群的优势个体迁徙到所有其他子种群; 2) 环状迁徙, 所有子种群形成一个环, 每个子种群的优势个体仅迁徙到环中相邻的两个子种群; 3) 星形迁徙, 以一个子种群作为中心, 连接其他所有子种群, 中心子种群的优势个体能够迁徙到所有其他子种群, 而其他子种群的优势个体仅能迁徙到中心子种群.

已有研究表明, 采用多种群并行进化优化方法求解多目标优化问题, 能够提高问题求解的效率<sup>[20-21]</sup>. 但是, 目前尚没有用于求解高维多目标优化问题的多种群并行进化优化方法. 本文针对分解

后每一子优化问题, 采用一个子种群进化求解, 使得不同子种群朝着不同的方向进化, 能够很好地求解高维多目标优化问题.

## 2 提出的方法

### 2.1 总体框架

本文提出一种基于目标分解的高维多目标并行进化优化方法, 其总体框架如图 1 所示.

首先, 将高维多目标优化问题分解为若干 ( $K$  个) 子问题, 每一子问题仅包含原问题的部分目标以及一个聚合目标, 且这些部分目标之间具有较大的相关性, 从而显著降低优化问题的求解难度; 其次, 在并行环境下, 采用基于传统 Pareto 占优的 MOEA 求解每一子问题, 并通过迁徙算子, 充分利用其他子种群提供的有价值的信息, 使得问题求解的效率大大提高; 最后, 基于各子种群产生的非被占优解形成外部保存集, 使得高维多目标优化问题的 Pareto 最优解集具有优越的性能.

容易看出, 高维多目标优化问题的分解、子优化问题的并行求解以及外部保存集的形成等, 是本文需要解决的关键技术. 在后续的 3 个小节中, 将详细阐述这些关键技术.

### 2.2 高维多目标优化问题的分解

本节通过目标分组与聚合方法, 将式 (1) 表示的

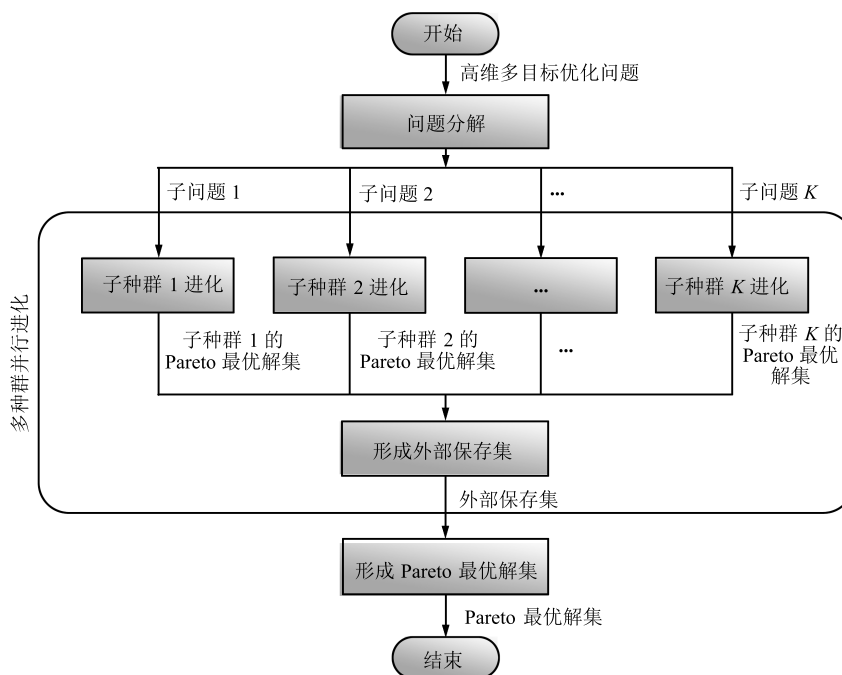


图 1 总体框架

Fig. 1 General framework

高维多目标优化问题分解为若干子问题, 其中, 每一子优化问题包含较少的目标函数, 在这些目标函数中, 有的是原优化问题的目标函数, 有的是原优化问题若干目标函数的聚合.

### 2.2.1 目标分组

为了求解式 (1) 描述的高维多目标优化问题, 本文基于目标函数之间的相关关系, 将该问题的目标函数分组. 在之前的研究中<sup>[6-7]</sup>, 目标函数之间的相关关系也被用来删除冗余的目标函数, 以缩减目标函数个数. 但是, 实际的优化问题很少具有冗余的目标函数. 如果删除相关程度较高的目标函数, 将难以得到完整的 Pareto 前沿. 因此, 本文保留所有的目标函数, 并将其分组. 进一步, 基于该分组, 将高维多目标优化问题分解为若干个传统的多目标优化问题, 从而降低问题求解的难度, 使得采用传统的多目标进化优化方法求解这些分解后的问题成为可能.

若两个目标函数之间的相关关系为负, 且一个目标函数减小, 那么, 另一个目标函数会增大; 反之, 若两个目标函数之间的相关关系为正, 那么, 它们会同时增加或减小. 这样一来, 如果一个候选解对某一目标函数是较优的, 那么, 该候选解对与之相似的目标函数也具有较优的性能. 因此, 求解目标函数之间相关程度较高的优化问题时, 算法将更容易找到同时对这些目标函数较优的折中解, 从而具有更佳的收敛性能. 鉴于此, 本文基于目标函数之间的相关关系, 将相关程度较高的目标函数分为一组, 对于提高问题求解的效率, 是非常有意义的.

为了按照相关关系对目标函数分组, 需要计算这些目标函数之间的相关程度. 本文通过计算目标函数之间的 Spearman 相关系数, 以反映这些目标函数之间的相关程度. Spearman 相关系数又称为秩相关系数, 它利用两个变量的秩次值, 进行相关关系分析, 对变量的形式及其满足的分布没有任何要求. 鉴于本文研究的优化问题, 包含的目标函数可能是非连续或者非线性的, 采集到的样本也可能不服从正态分布, 因此, 采用 Spearman 相关系数进行相关关系分析是完全可行的. 记目标函数  $f_i$  与  $f_j$  之间的 Spearman 相关系数为  $\rho_{ij}$ .  $\rho_{ij}$  的计算方法可参阅文献 [22]. 可知,  $\rho_{ij} \in [-1, 1]$ , 且  $\rho_{ij}$  越大, 目标函数  $f_i$  和  $f_j$  之间的相关程度就越大.

本文期望采用同构的并行环境, 求解分解后的若干多目标优化问题. 为了提高问题求解的效率, 希望并行环境的每一同构计算节点的负荷大体相同, 这样一来, 将目标函数均分是十分必要的. 本文将目标函数分为  $K$  组, 那么, 每组目标函数的个数不超过  $\lceil m/K \rceil$ . 组数  $K$  根据优化问题包含的目标函数个数以及可用的计算资源确定. 一般来讲,  $K$  的取

值应使得每组包含的目标函数个数小于或等于 3.

为了得到合理的目标函数分组, 需要采用一定的准则, 对分组结果进行评价. 本文借鉴文献 [23] 的方法, 首先, 计算组内目标函数相关系数的平均值  $p(G_k)$ ,  $k = 1, 2, \dots, K$ ; 然后, 对所有组求取上述指标的平均值, 记为  $\bar{p}$ , 用于反映分组结果的性能:

$$\bar{p} = \frac{1}{K} \sum_{k=1}^K p(G_k) \quad (2)$$

$$p(G_k) = \begin{cases} \frac{\sum_{f_i, f_j \in G_k} \rho_{ij}}{m_k - 1}, & m_k \geq 2 \\ 1, & m_k = 1 \end{cases} \quad (3)$$

式中,  $G_k$  表示由第  $k$  组目标函数形成的集合,  $m_k$  为组  $G_k$  包含的目标函数个数. 由式 (2) 和式 (3) 可知,  $\bar{p}$  越大, 组内目标函数之间的相似程度越高, 从而分组性能越好.

基于上述准则, 本文提出的目标函数分组方法的步骤如下:

**步骤 1.** 将目标函数随机分为  $K$  组, 且每组包含的目标函数个数不超过  $\lceil m/K \rceil$ , 令迭代次数  $q = 0$ , 计算  $\bar{p}(q)$ .

**步骤 2.** 判断分组终止准则 (本文设终止准则为  $q \geq m^3$ ) 是否满足, 若满足, 转步骤 5.

**步骤 3.** 随机选择两组, 如果这两组包含的目标函数均为  $\lceil m/K \rceil$  个, 那么, 从每组中随机选择一个进行交换; 如果只有一组包含  $\lceil m/K \rceil$  个目标函数, 那么, 从该组中随机选择一个, 放入另一组中; 否则, 从一组中随机选择一个目标, 放入另一组.

**步骤 4.** 令  $q = q + 1$ , 计算  $\bar{p}(q)$ , 如果  $\bar{p}(q) > \bar{p}(q - 1)$ , 那么, 更新分组结果, 转步骤 2.

**步骤 5.** 输出分组结果, 终止算法.

根据上述分组方法, 式 (1) 描述的高维多目标优化问题的目标函数可以分成如下  $K$  组:

$$\begin{aligned} G_1 &= \{f_{11}(\mathbf{x}), f_{12}(\mathbf{x}), \dots, f_{1m_1}(\mathbf{x})\} \\ G_2 &= \{f_{21}(\mathbf{x}), f_{22}(\mathbf{x}), \dots, f_{2m_2}(\mathbf{x})\} \\ &\vdots \\ G_K &= \{f_{K1}(\mathbf{x}), f_{K2}(\mathbf{x}), \dots, f_{Km_K}(\mathbf{x})\} \end{aligned} \quad (4)$$

式中,  $f_{kh}(\mathbf{x})$ ,  $k = 1, 2, \dots, K$ ,  $h = 1, 2, \dots, m_k$ , 为组  $G_k$  包含的第  $h$  个目标函数;  $m_k$  为组  $G_k$  包含的目标函数个数, 且满足  $\sum_{k=1}^K m_k = m$ ,  $m_k \leq \lceil m/K \rceil$ .

### 2.2.2 子优化问题的形成

基于式 (4) 中每组的目标函数, 可以形成一个传统的多目标优化问题, 其中, 每一多目标优化问题

均为原高维多目标优化问题的子问题. 但是, 如果通过求解这些子优化问题, 得到原优化问题的最优解, 那么, 这些最优解很可能是原优化问题的局部最优解, 而很难成为该优化问题的全局最优解. 这是因为, 每一子优化问题, 仅考虑了原优化问题的少部分目标函数. 为了得到原优化问题的全局最优解, 需要在每一子优化问题中, 增加反映其他目标函数信息的目标函数, 称之为这些目标函数的聚合目标.

对于组  $G_k$  包含的目标函数, 在形成的子优化问题中, 引入聚合目标, 记为  $f_{k0}(\mathbf{x})$ , 其表达式为

$$f_{k0}(\mathbf{x}) = F(G_1, \dots, G_{k-1}, G_{k+1}, \dots, G_K), \quad k = 1, 2, \dots, K \quad (5)$$

由式 (5) 可以看出,  $f_{k0}(\mathbf{x})$  是由除本组包含的目标函数之外, 其余目标函数聚合而成的新的目标函数, 因此, 它包含了这些目标函数的信息. 这样一来, 根据式 (4) 中组  $G_k$  包含的目标函数以及聚合目标  $f_{k0}(\mathbf{x})$ , 可以形成一个新的多目标优化问题, 从而式 (1) 描述的高维多目标优化问题可以分解为如下  $K$  个子问题:

$$\begin{cases} \min \mathbf{f}_{G_1}(\mathbf{x}) = \\ \quad (f_{10}(\mathbf{x}), f_{11}(\mathbf{x}), f_{12}(\mathbf{x}), \dots, f_{1m_1}(\mathbf{x})) \\ \text{s.t. } \mathbf{x} \in S \subset \mathbf{R}^n \end{cases} \\ \vdots \\ \begin{cases} \min \mathbf{f}_{G_K}(\mathbf{x}) = \\ \quad (f_{K0}(\mathbf{x}), f_{K1}(\mathbf{x}), f_{K2}(\mathbf{x}), \dots, f_{Km_K}(\mathbf{x})) \\ \text{s.t. } \mathbf{x} \in S \subset \mathbf{R}^n \end{cases} \quad (6)$$

求解式 (6) 中的每一子优化问题时, 既可以同时优化原优化问题的少数目标函数, 又因为聚合目标的存在, 可以对原优化问题的其他目标函数进行优化, 从而平衡了原优化问题的所有目标函数. 此外, 由于子优化问题目标函数的减少, 使得问题的求解难度显著降低, 从而 Pareto 最优解的选择压力大大提高, 这有利于问题的高效求解.

由式 (6) 可以看出,  $f_{k0}(\mathbf{x})$  对于每一子优化问题的形成与求解是非常关键的. 尽管式 (5) 给出了  $f_{k0}(\mathbf{x})$  的形式表达式, 但是, 采用不同的方法, 可以得到  $f_{k0}(\mathbf{x})$  不同的表达形式. 例如, 可以将其余目标函数进行算术运算, 从而聚合为一个目标函数; 也可以基于这些目标函数, 得到候选解在每组目标空

间上的 Pareto 序值, 并对这些序值进行算术运算, 作为该候选解的聚合目标值; 还可以计算候选解在其余目标空间上的超体积贡献度, 作为该候选解的聚合目标值, 等等. 容易知道, 虽然这些方法都能够反映候选解针对其余目标函数的优劣, 但是, 不同方法的计算复杂度是不同的. 鉴于后两种方法的计算复杂度较高, 因此, 本文采用第一种方法计算候选解的聚合目标值, 其表达式为

$$f_{k0}(\mathbf{x}) = \sqrt{\sum_{g=1, g \neq k}^K \sum_{h=1}^{m_g} (\omega_{gh} f_{gh}(\mathbf{x}))^2}, \quad k = 1, 2, \dots, K \quad (7)$$

其中,  $\omega_{gh}$  为目标函数  $f_{gh}$  的权重系数, 其取值可以根据该目标的重要程度确定. 本文假定每个目标函数都是同等重要的, 那么, 所有的  $\omega_{gh} = 1/(m - m_k)$ , 其中,  $g = 1, \dots, k-1, k+1, \dots, K; h = 1, 2, \dots, m_g$ . 可以看出, 式 (7) 反映了在其他子目标空间中, 候选解  $\mathbf{x}$  的目标函数值形成的向量到原点的欧氏距离. 采用式 (7) 计算候选解的聚合目标值, 不但需要的计算量小, 而且对候选解的区分能力也强.

### 2.3 子优化问题的并行求解

为了高效求解式 (1) 描述的高维多目标优化问题, 基于第 2.2 节建立的  $K$  个子优化问题, 本节通过多种群并行进化优化方法, 采用 NSGA-II<sup>[4]</sup> 求解每一子问题, 并在求解过程中充分利用其他子种群提供的有价值信息.

针对每一子优化问题, 采用一个子种群进化. 尽管在多种群并行进化算法中, 不同的子种群可以具有不同的规模, 但是, 由于本文分解后的不同子优化问题, 包含的目标函数个数大体相当, 使得优化不同子问题所需的计算量大体相同, 因此, 这里优化不同子问题的子种群规模均为  $n_p$ . 这样一来, 用于求解本文问题的子种群共有  $K$  个, 这些子种群的规模之和为  $N = n_p K$ . 鉴于每一子优化问题包含的目标函数比较少, 因此, 针对每一子种群中的不同进化个体, 可以采用传统的 Pareto 占优关系, 比较它们的性能. 此外, 由于不同子种群优化的目标函数不同, 因此, 不同子种群的进化方向不同, 这有利于整个种群保持良好的多样性.

在子种群进化求解过程中, 为了利用其他子种群提供的有价值信息, 需要在这些子种群之间迁徙部分进化个体. 进化个体迁徙时, 采用完全迁徙的方式, 迁徙间隔代数为  $T_m$ , 迁徙率为  $\alpha \in (0, 1)$ . 这样一来, 每隔进化代数  $T_m$ , 每一子种群需要选择  $[\alpha n_p]$  个优势个体作为迁徙个体, 并发送给其他子种群.

通常, 我们希望选择的迁徙个体具有较好的性能, 为了评价迁徙个体的性能, 与 NSGA-II 评价进化个体的方式一样, 也采用如下两个指标: 1) Pareto 序值, 通过 Pareto 占优关系, 计算某迁徙个体的 Pareto 序值, 用于评价该个体的逼近性能; 2) 拥挤距离, 基于 NSGA-II 提出的方法, 计算迁徙个体的拥挤距离, 用于衡量该个体的分布性能. 每一子种群选择迁徙个体的具体步骤本文不再赘述, 详细可参阅文献 [4]. 基于该方法选择的  $\lceil \alpha n_p \rceil$  个迁徙个体, 不但具有很好的逼近性能, 而且具有很好的分布性能, 这有利于其他子种群的进化.

鉴于不同子种群优化的目标函数不同, 因此, 一个子种群的优势个体, 对其他子种群而言, 未必也是优势个体. 这表明, 虽然某一子种群接收了其他子种群发送的优势个体, 该子种群也需要基于优化的子问题, 对这些发送的优势个体重新评价其性能. 进一步, 由于某一子种群接收了  $(K-1)\lceil \alpha n_p \rceil$  个迁徙个体, 而该子种群的规模  $n_p$  却是固定不变的, 因此, 需要对该子种群的当代个体以及接收的迁徙个体, 基于 Pareto 序值和拥挤距离重新排序, 以生成规模为  $n_p$  的子代种群.

## 2.4 外部保存集的形成

在多种群进化优化中, 采用外部保存集, 保留各子种群进化过程中的优势个体, 对于保证 Pareto 最优解集的完备性与多样性, 是非常必要的. 对于本文的基于目标分解的高维多目标并行进化优化方法而言, 也是如此. 一般地, 与通过合并各子种群的 Pareto 最优解集, 形成优化问题的 Pareto 最优解集相比, 外部保存集的 Pareto 最优解集, 具有更好的分布性能. 这是因为, 在高维空间上均匀分布的点集, 在低维子空间上也是均匀分布的, 但反之不然. 由于本文方法各子种群的 Pareto 最优解集, 都是在低维目标空间上得到的, 尽管这些解集中的解, 在低维目标空间上均匀分布, 但是, 它们在高维目标空间上却不一定是均匀分布的. 这样一来, 基于各子种群的 Pareto 最优解集, 在高维目标空间中, 按照一定的策略选择, 以形成外部保存集, 能够保证 Pareto 最优解集具有好的分布性能.

然而, 基于各子种群的 Pareto 最优解集形成保存集时, 难以基于这些解关于所有目标的 Pareto 序值, 得到逼近性能好的解. 这是因为, 需要优化的目标函数增多, 使得这些解的 Pareto 序值几乎相同. 因此, 本节提出一种能够兼顾逼近与分布性能的外部保存集形成策略.

尽管在高维目标空间中, 难以区分上述解的逼近性能, 但是, 对于来自同一子种群的 Pareto 最优解而言, 由于需要优化的目标函数较少, 因此, 基于

这些解在子种群中的 Pareto 序值, 仍然能够得到具有好的逼近性能的外部保存集. 记第  $t$  代的外部保存集为  $A^t = \cup_{k=1}^K A_k^t$ , 其中,  $A_k^t$ ,  $k = 1, 2, \dots, K$ , 表示  $A_k^t$  中来自第  $k$  个子种群的 Pareto 最优解集,  $|A^t| \leq N$ . 将  $A_k^t$  中的解与第  $k$  个子种群当前代的 Pareto 最优解集  $P_k^t$  合并, 得到  $C_k^t = A_k^t \cup P_k^t$ . 基于第  $k$  个子优化问题的目标函数, 计算  $C_k^t$  中解的 Pareto 序值, 选择 Pareto 序值为 1 的解, 形成一个新的集合, 记为  $C_k^{t'}$ . 考虑所有子优化问题, 将  $C_k^{t'}$ ,  $k = 1, 2, \dots, K$  合并, 可以得到  $C^t = \cup_{k=1}^K C_k^{t'}$ , 那么,  $C^t$  中的解对于所有的优化目标, 将具有好的逼近性能.

为了使得下一代外部保存集  $A^{t+1}$  中的解具有好的分布性能, 当  $|C^t| > N$  时, 在高维目标空间中, 基于拥挤距离, 从  $C^t$  中选择  $N$  个解, 得到下一代外部保存集  $A^{t+1}$ . 如果  $|C^t| \leq N$ , 那么,  $C^t$  即为下一代外部保存集.

通过上述方式形成的保存集, 不但具有好的逼近性能, 也具有好的分布性能, 这有利于寻找问题的 Pareto 最优解集.

## 2.5 算法步骤

本文提出的基于目标分解的高维多目标进化优化方法的步骤如下:

**步骤 1.** 采用第 2.2 节的方法, 将高维多目标优化问题分解为  $K$  个子问题;

**步骤 2.** 设定算法所需参数的值, 初始化  $K$  个子种群, 令  $t = 0$ ,  $A^t = \emptyset$ ;

**步骤 3.** 得到子种群的 Pareto 最优解集, 并发送至外部保存集;

**步骤 4.** 采用第 2.4 节的方法, 形成外部保存集;

**步骤 5.** 判断是否满足终止条件, 若是, 转步骤 9;

**步骤 6.** 判断是否满足迁徙条件, 若否, 转步骤 8;

**步骤 7.** 采用第 2.3 节的方法, 迁徙个体;

**步骤 8.** 子种群进化, 形成子代个体, 令  $t = t + 1$ , 转步骤 3;

**步骤 9.** 基于外部保存集, 得到 Pareto 最优解集并输出, 算法终止.

## 2.6 性能分析

本文方法主要分为三个部分: 问题分解、子问题的并行求解以及外部保存集的形成.

在问题分解时, 计算目标函数之间 Spearman 相关系数的计算复杂度为  $O(2mL^2)$ , 目标函数分组所需的计算复杂度为  $O(m^3)$ .

在并行求解子问题时, 每个子种群均采用 NSGA-II 求解一个子优化问题. NSGA-II 每进化

一代的计算复杂度为  $O(m(2N)^2)$ . 本文方法所有子种群每代迁徙个体和生成新种群所需的计算复杂度为  $O(m((\alpha + 1/K)^2 + (2/K)^2)N^2)$ . 由于  $\alpha \in (0, 1)$ ,  $K \geq 2$ , 因此,  $m((\alpha + 1/K)^2 + (2/K)^2)N^2 < m(2N)^2$ ,  $O(m((\alpha + 1/K)^2 + (2/K)^2)N^2) < O(m(2N)^2)$ . 这表明, 本文方法的子种群每进化一代, 需要的总的计算复杂度小于 NSGA-II; 此外, 迁徙率  $\alpha$  越小且分组个数  $K$  越大, 那么, 本文方法的计算复杂度越低. 需要说明的是, 每个进程仅实施一个子种群的进化, 因此, 每一进程完成每代进化, 需要的时间远远少于 NSGA-II.

形成外部保存集的最大计算复杂度为  $O(m/K(2N)^2)$ , 小于 NSGA-II 的相应操作. 由于采用一个新的进程来专门实施外部保存集的形成, 因此, 种群每进化一代, 形成外部保存集所需的时间都小于 NSGA-II.

对于本文方法处理的实际问题, 一般来说,  $m$ ,  $N$  和  $L$  均为同一数量级, 那么, 问题分解与种群进化一代所需的计算复杂度是相似的, 因此, 在整个问题求解过程中, 问题分解消耗的计算资源可以忽略不计. 此外, 通过上述分析可知, 所有子种群进化和外部保存集形成所需的计算复杂度, 均小于已有的典型方法. 后续的实验结果也验证了本节分析.

### 3 实验

本节通过实验评价本文所提方法的性能. 实验包括如下两部分: 1) 考察包含的参数和策略, 如目标函数分组个数、基于相关关系的目标函数分组, 以及外部保存集的形成对本文方法性能的影响; 2) 将本文方法与 NSGA-II、PPD-MOEA、 $\epsilon$ -MOEA、HypE 和 MSOPS 等比较, 验证本文方法的优越性.

#### 3.1 优化问题

采用 DTLZ 系列优化问题<sup>[24]</sup> 中的 DTLZ1、DTLZ2、DTLZ3、DTLZ5、DTLZ7 和多目标旅行商问题 (Multi-objective travelling salesman problem, MOTSP)<sup>[25]</sup>, 评价不同方法的性能. 其中, DTLZ1 是一个多模态优化问题, 该问题的 Pareto 前沿为一个平面; DTLZ2 是一个 Pareto 前沿为凹面的优化问题; DTLZ3 的 Pareto 前沿与 DTLZ2 相同, 但是, 该问题是一个多模态优化问题, 因此, 问题求解的难度增大; DTLZ5 的 Pareto 前沿为一条曲线, 选择该优化问题是为了考察不同方法在高维空间中收敛到低维 Pareto 前沿的能力; DTLZ7 具有一组不连续的 Pareto 前沿, 该问题能够检测不同方法在多个 Pareto 前沿保持分布度的能力; MOTSP 是一类著名的 NP 难问题. 在构建该问题时, 可以通过相关参数决定目标函数之间的相关程度. 当相关

参数的值分别为正、0 和负时, 该问题的相邻目标函数分别具有正、0 和负相关关系. 因此, 该问题非常适合用于考察本文方法的性能.

上述优化问题的数学模型可参阅相关文献. 需要说明的是, 对于 DTLZ 系列问题, 决策变量个数为  $n = m + v - 1$ , 其中,  $m$  为目标函数的个数,  $v$  为一个取值为整数的参数. 实验中, 对于优化问题 DTLZ1、DTLZ2、DTLZ3、DTLZ5 和 DTLZ7,  $v$  的取值分别为 5、10、5、10 和 20. 所有优化问题决策变量的取值范围均为  $[0, 1]$ . 对于 MOTSP, 节点数目为 30, 相关参数分别取 0.5、0 和 -0.5.

在实验的第一部分, 对于不同的参数或策略, 通过求解不同的优化问题, 评价这些参数或策略对本文方法性能的影响. 具体地, 1) 通过求解 24 目标优化问题 DTLZ2, 评价目标函数分组个数对本文方法性能的影响; 2) 基于 6 目标优化问题 DTLZ2, 构建并求解目标相关程度不同的多个优化问题, 以说明本文提出的基于相关关系的目标函数分组方法的有效性; 3) 通过求解 6 目标优化问题 DTLZ2, 评价本文所提外部保存集形成方法的有效性.

在实验的第二部分, 通过求解目标函数个数分别为 6、9、12 和 15 的优化问题 DTLZ1、DTLZ2、DTLZ3、DTLZ5 和 DTLZ7, 以及目标函数个数分别为 6 和 12 的优化问题 MOTSP, 并与其他方法比较, 验证本文所提方法的优越性.

#### 3.2 性能指标

采用世代距离 (Generation distance, GD)<sup>[26]</sup>、空间评价指标 (Spacing, SP)<sup>[26]</sup>、最大传播距离 (Maximum spread, MS)<sup>[27]</sup>、反世代距离 (Inverted generation distance, IGD)<sup>[28]</sup> 和超体积 (Hypervolume, HV)<sup>[15]</sup> 5 个性能指标, 以及并行程序耗时  $T_p$ , 评价不同方法的性能, 其中, GD 用于评价方法的逼近性能; SP 用于反映方法的分布性能; MS 用于衡量方法的延展性能; IGD 和 HV 可同时测量方法的逼近、分布以及延展性能;  $T_p$  用于考察并行算法的运行速度.

对于指标 GD、SP、MS、IGD 和 HV, 相应的计算方法可参阅上述文献. 容易知道, 某方法的 GD 值越小, 那么, 该方法的逼近性能越好, 特别地, 当  $GD = 0$  时, 该方法得到的所有优化解都在真实 Pareto 前沿上; 某方法的 SP 值越小, 那么, 该方法的分布性能越好, 所得 Pareto 前沿中优化解的分布越均匀; 某方法的 MS 值越大, 那么, 该方法的延展性能越好, 特别地, 当  $MS = 1$  时, 该方法所得 Pareto 前沿完全覆盖整个真实 Pareto 前沿; 某方法的 IGD 值越小或 HV 值越大, 那么, 该方法越能平衡逼近、分布以及延展性能, 所得 Pareto 前沿对真实 Pareto



前沿的拟合度越高.

对于指标  $T_p$ , 其计算方法为

$$T_p = p_n T_s \quad (8)$$

其中,  $p_n$  为程序包含的并行进程的个数,  $T_s$  为运行时间最长的进程所需的时间,  $T_p$  和  $T_s$  的单位均为秒.

在实验的第一部分, 采用 GD、SP、MS, 以及  $T_p$  等 4 个指标, 详细考察一些参数和策略对本文方法性能的影响; 在实验的第二部分, 采用 IGD 指标比较各方法求解 DTLZ 的性能, 并且还采用 HV 指标, 比较各方法求解 MOTSP 的性能. 需要说明的是, 由于 MOTSP 的真实 Pareto 前沿是未知的, 本文选取所有运行结果所得解的最大目标值, 作为计算超体积时的参考点.

### 3.3 对比方法

本文选取 NSGA-II<sup>[4]</sup>、PPD-MOEA<sup>[10]</sup>、 $\epsilon$ -MOEA<sup>[12]</sup>、HypE<sup>[15]</sup> 和 MSOPS<sup>[16]</sup> 作为对比方法. 由于本文方法的每一子种群都基于 NSGA-II 进化机制求解相应的子优化问题, 且选择迁徙个体和形成外部保存集时, 都用到了 NSGA-II 的进化个体选择机制, 因此, 将本文方法与 NSGA-II 对比, 是十分必要的; 鉴于本文方法基于 Pareto 局部占优关系比较进化个体的性能, 因此, 与 PPD-MOEA 对比也是很有意义的; 此外, 尽管  $\epsilon$ -MOEA、HypE 以及 MSOPS 与本文方法没有明显的共同之处, 但是, 它们均为求解高维多目标优化问题的有效方法. 因此, 与这些方法进行比较, 能够充分评价本文方法的性能.

上述对比方法中的 NSGA-II<sup>1</sup>、 $\epsilon$ -MOEA<sup>2</sup>、HypE<sup>3</sup> 以及 MSOPS<sup>4</sup> 的程序代码, 均可在论文作者的主页上下载, PPD-MOEA 的程序代码由我们根据原文算法自己编写. 本文的实验结果均通过运行这些程序得到.

### 3.4 参数设置

对于所有方法, 均采用相同的遗传算子. 对于 DTLZ, 采用模拟二元交叉和多项式变异, 且交叉和变异算子的分布指数分别为 20 和 10. 对于 MOTSP, 采用顺序交叉和逆转变异交叉, 变异概率分别为 0.8 和 0.01, 交配选择采用二元锦标赛方法. 算法的终止条件为个体评价次数达到预先设定的值, 对于优化问题 DTLZ1、DTLZ3 和 MOTSP, 该值为 100 000; 对于 DTLZ2、DTLZ5 和 DTLZ7, 该值为 30 000. 对于方法 NSGA-II、PPD-MOEA、HypE

和 MSOPS, 种群规模为 100. 对于本文方法, 当目标函数的分组个数为 3 或 6 时, 种群规模为 102; 当目标函数的分组个数为 8 时, 种群规模为 104, 使得不同子种群包含相同个数的进化个体. 对于  $\epsilon$ -MOEA, 由于该方法的种群规模与  $\epsilon$  的取值有关, 为了公平比较, 根据不同的优化问题, 采取不同的  $\epsilon$  取值, 使得在算法终止时, 种群规模保持在 100 左右.  $\epsilon$  的取值如表 1 和表 2 所列.

表 1 DTLZ 优化问题的  $\epsilon$  取值

Table 1 The settings of  $\epsilon$  for DTLZ problems

目标个数	DTLZ1	DTLZ2	DTLZ3	DTLZ5	DTLZ7
6	0.06	0.23	0.24	0.12	0.27
9	0.09	0.29	0.42	0.13	0.69
12	0.17	0.30	0.81	0.15	0.78
15	0.32	0.31	0.92	0.15	0.83

表 2 MOTSP 问题的  $\epsilon$  取值

Table 2 The settings of  $\epsilon$  for MOTSP

目标个数	相关参数	$\epsilon$
6	0.5, 0, -0.5	0.42, 1.3, 2.7
12	0.5, 0, -0.5	3.0, 4.1, 5.3

对于本文方法涉及的其他参数, 我们先前的研究表明, 合理地设置迁徙间隔与迁徙率, 不但对算法的运行速度影响较小, 而且能够提升算法性能. 因此, 在本文实验中,  $T_m = 1$ ,  $\alpha = 0.1$ . 此外, 如果没有特别说明, 当优化问题包含 6 或 9 个目标函数时, 分组个数为 3; 包含 12 个目标函数时, 分组个数为 4; 包含 15 个目标函数时, 分组个数为 5.

对于 PPD-MOEA, 每次选取的目标函数为 2 个, 进行目标变换的间隔代数为 10; 在 HypE 中, 估计超体积所需的采样点个数为 10 000; MSOPS 的权重向量为 100 个.

对于每一优化问题, 采用每种方法独立运行 30 次, 计算第 3.2 节各性能指标的值, 并统计它们的均值和方差. 此外, 采用 Mann-Whitney U 分布方法, 对不同方法得到的上述指标值进行假设检验, 以评价不同方法性能差异的显著性.

## 3.5 实验结果

### 3.5.1 不同参数或策略对本文方法性能的影响

#### 1) 目标函数分组个数

<sup>1</sup><http://www.iitk.ac.in/kangal/codes/nsga2/nsga2-v1.0.tar>

<sup>2</sup><http://www.iitk.ac.in/kangal/codes/eps-moea/epsmoea.tar>

<sup>3</sup><http://www.tik.ee.ethz.ch/pisa/selectors/hype/hype.c.source.tar.gz>

<sup>4</sup><http://homepage.ntlworld.com/evan.hughes1/msops.zip>



目标函数分组个数是本文方法新引入的参数, 它的取值与本文方法的收敛性能、计算资源的分配以及运行效率密切相关. 通过将 24 目标优化问题 DTLZ2 的目标函数分为 1、2、3、4、6、8、12 组, 考察目标函数分组个数对本文方法性能的影响. 容易知道, 当目标函数的分组个数仅为 1 时, 本文方法即为 NSGA-II. 图 2 给出了各性能指标随目标函数分组个数的变化趋势.

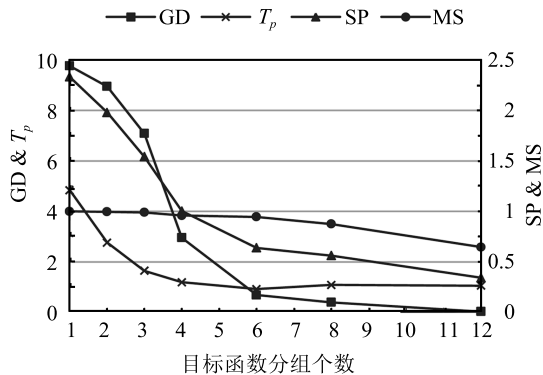


图 2 各项性能指标随目标函数分组个数的变化趋势

Fig. 2 Curves of metrics w.r.t. number of subproblems

由图 2 可以看出, 随着目标函数分组个数的增多, 1) GD 逐渐减小, 说明较多的目标函数分组个数, 能够改善本文方法所得 Pareto 前沿的逼近性能; 2) MS 逐渐减小, 说明目标函数分组个数越多, 本文方法所得 Pareto 前沿的延展性越差, 可能的原因是当目标函数分组个数很多时, 每个子种群包含的进化个体数很少, 使得子种群所得的 Pareto 最优解集难以完整覆盖真实的 Pareto 前沿; 3) SP 逐渐减小, 尽管如此, 这并不能说明本文方法所得 Pareto 前沿的分布更加均匀, 因为即使某两个近似 Pareto 前沿的分布性能相同, 对于逼近性能较好或延展性能较差的近似 Pareto 前沿, 它的 SP 指标值也会较小; 4) 本文方法的运行时间大体呈下降趋势, 并在分组个数为 6 时, 取得了最快的运行速度. 但是, 当分组个数为 8 时, 运行时间开始略微回升. 说明即使增加进程个数, 本文方法的运行效率也不一定提高. 但是, 只要目标函数分组个数大于 1, 本文方法的运行速度就会快于 NSGA-II.

基于上述实验结果, 对于 24 目标优化问题 DTLZ2, 当目标函数分为 6 组时, 可在多个性能指标上取得较为理想的结果.

## 2) 基于相关关系的目标函数分组方法

本部分参考文献 [29] 的方法, 基于 6 目标优化问题 DTLZ2, 构建目标相关程度不同的 5 个优化问题, 分别采用基于相关关系和随机的目标函数分组方法, 求解这些优化问题, 以评价本文所提目标函数

分组方法的性能. 这些问题的数学模型为

$$\begin{aligned} \min & \\ \begin{cases} g_i(\mathbf{x}) = f_i(\mathbf{x}), & i = 1, 2, 3 \\ g_i(\mathbf{x}) = \lambda f_{i-3}(\mathbf{x}) + (1 - \lambda) f_i(\mathbf{x}), & i = 4, 5, 6 \end{cases} \end{aligned} \quad (9)$$

其中,  $f_i(\mathbf{x})$ ,  $i = 1, 2, 3, 4, 5, 6$ , 分别为 6 目标优化问题 DTLZ2 的原目标函数.  $\lambda \in [0, 1]$  为相关强度系数, 且  $\lambda$  越大,  $g_i(\mathbf{x})$  与  $g_{(i-3)}(\mathbf{x})$  相似程度越高. 特别地, 当  $\lambda = 0$  时, 该优化问题即为 6 目标优化问题 DTLZ2; 当  $\lambda = 1$  时, 该优化问题具有 3 组完全相关的目标函数. 按照第 2.2 节给出的分组方法, 当  $\lambda$  取值较大时, 这些优化问题的目标函数更倾向于被分为如下 3 组: 1)  $G_1 = g_1(\mathbf{x}), g_4(\mathbf{x})$ ; 2)  $G_2 = g_2(\mathbf{x}), g_5(\mathbf{x})$ ; 3)  $G_3 = g_3(\mathbf{x}), g_6(\mathbf{x})$ . 本文设置  $\lambda = 0, 0.25, 0.5, 0.75, 1$ , 从而分别得到目标相关程度不同的多个优化问题.

对于基于相关关系和随机法得到的目标函数分组, 采用本文提出的方法求解上述优化问题, 实验结果如表 3 所列, 表中括号外的数据表示均值, 括号内的数据为标准差, 粗体显示的数据为两种方法得到的最优结果.

由表 3 可以看出, 1) 对于所有的优化问题, 基于相关关系的目标函数分组方法的 GD 指标均显著优于随机分组方法, 且  $\lambda$  越大, 即目标函数之间的相关程度越高, 基于相关关系的目标函数分组方法的收敛性越好; 2) 当  $\lambda = 0$  和  $\lambda = 0.25$  时, 两种方法的 SP 指标没有明显差异, 当  $\lambda > 0.25$  时, 基于相关关系的目标函数分组方法, 能够显著提升解集的分布性能; 3) 无论采用哪种分组方法, 得到的 Pareto 前沿的 MS 值均达到最大值; 4) 两种方法的  $T_p$  指标没有显著差异, 这说明, 计算目标分组花费的时间, 对算法总的运行时间几乎没有影响. 总体来讲, 基于相关关系的目标函数分组方法, 不仅所需的计算资源少, 而且能够明显提升算法的收敛与分布性能.

## 3) 外部保存集

将本文提出的外部保存集形成策略与传统的策略比较, 以评价本文的外部保存集形成策略对本文方法性能的影响. 这里传统的策略基于所有目标计算候选解的 Pareto 序值, 并基于序值和拥挤距离形成外部保存集. 分别基于这两种外部保存集形成策略, 并采用本文的方法, 求解 6 目标优化问题 DTLZ2, 实验结果如表 4 所列.

由表 4 可以看出, 1) 采用本文的外部保存集形成策略, 得到的 Pareto 前沿的 SP 值与传统的策略不相上下; 2) 两种策略得到的 Pareto 前沿的 MS 值均达到最大值, 但是, 传统策略的稳定性略优于本文

表 3 目标函数分组策略对本文方法性能的影响  
Table 3 The effect of the strategy of objectives decomposition

$\lambda$	GD		SP		MS		$T_p$	
	相关分组	随机分组	相关分组	随机分组	相关分组	随机分组	相关分组	随机分组
0	<b>1.029E-2</b> ( <b>3.9E-3</b> )	1.324E-2 (4.0E-3)†	2.218E-1 (1.2E-2)	<b>2.125E-1</b> ( <b>1.8E-2</b> )	1.000E+0 (1.1E-4)	1.000E+0 (1.5E-4)	3.931E-1 (3.8E-2)	<b>3.903E-1</b> ( <b>4.1E-2</b> )
0.25	<b>7.405E-3</b> ( <b>3.8E-3</b> )	1.291E-2 (4.2E-3)†	<b>1.992E-1</b> ( <b>1.3E-2</b> )	2.021E-1 (1.3E-2)	1.000E+0 (1.2E-4)	1.000E+0 (8.3E-5)	4.027E-1 (4.6E-2)	<b>3.998E-1</b> ( <b>3.3E-2</b> )
0.5	<b>2.037E-3</b> ( <b>7.1E-4</b> )	1.311E-2 (1.5E-3)†	<b>1.643E-1</b> ( <b>1.0E-2</b> )	1.936E-1 (2.7E-2)†	1.000E+0 (7.1E-5)	1.000E+0 (1.8E-4)	<b>3.891E-1</b> ( <b>7.1E-2</b> )	4.082E-1 (5.8E-2)
0.75	<b>8.214E-4</b> ( <b>4.2E-4</b> )	1.285E-2 (1.1E-3)†	<b>8.921E-2</b> ( <b>6.3E-3</b> )	1.284E-1 (1.9E-2)†	1.000E+0 (9.8E-5)	1.000E+0 (6.2E-5)	4.100E-1 (6.4E-2)	<b>4.081E-1</b> ( <b>4.5E-2</b> )
1	<b>3.436E-4</b> ( <b>2.0E-4</b> )	1.290E-2 (6.3E-4)†	<b>5.765E-2</b> ( <b>5.2E-3</b> )	6.228E-2 (4.7E-3)†	1.000E+0 (3.8E-6)	1.000E+0 (4.3E-6)	<b>3.915E-1</b> ( <b>5.6E-2</b> )	3.923E-1 (7.3E-2)

† 表示, 对于相关分组, 随机分组的该项指标具有显著差异 (Mann-Whitney U 分布检验, 置信水平为 0.05)

表 4 外部保存集形成策略对本文方法性能的影响  
Table 4 The effect of the strategy of external archive

	GD	SP	MS	$T_p$
本文策略	1.029E-2 (3.9E-3)	2.218E-1 (1.2E-2)	1.000E+0 (1.1E-4)	3.931E-1 (3.8E-2)
传统策略	4.364E-2 (3.0E-3)†	2.198E-1 (1.8E-2)	1.000E+0 (8.9E-5)	4.453E-1 (3.1E-2)†

† 表示, 对于本文策略, 传统策略的该项指标具有显著差异 (Mann-Whitney U 分布检验, 置信水平为 0.05)

的策略; 3) 虽然两种策略得到的 Pareto 前沿的 SP 和 MS 值没有显著差异, 但是, 对于其他两个指标 GD 和  $T_p$ , 本文的策略均显著优于传统的策略. 这说明, 采用本文提出的外部保存集形成策略, 得到的 Pareto 前沿在保证分布和延展性能的同时, 不但具有更好的逼近性能, 而且所需的计算时间更短. 因此, 采用本文提出的外部保存集形成策略, 能够提高求解高维多目标优化问题的性能.

通过第一部分的实验结果与分析, 可以得到如下结论: 本文提出的基于相关关系的目标函数分组方法和外部保存集形成策略是非常有效的. 此外, 合理的目标函数分组个数也有助于提高本文方法的性能.

### 3.5.2 与其他方法的对比

通过求解 6、9、12、15 目标优化问题 DTLZ1、DTLZ2、DTLZ3、DTLZ5、DTLZ7 及 6 和 12 目标优化问题 MOTSP, 并与 NSGA-II、PPD-MOEA、 $\varepsilon$ -MOEA、HypE 和 MSOPS 比较, 评价本文方法的性能. 表 5 列出了不同方法求解 DTLZ 所得

Pareto 前沿的 IGD 指标; 表 6 列出了这些方法求解 MOTSP 所得 Pareto 前沿的 HV 指标. 为了便于观察和对比, 以本文方法为基准, 对其他方法所得的 HV 指标进行标准化处理. 表中, 粗体显示的数据为这些方法得到的该问题的最优结果.

首先, 分析不同方法求解优化问题 DTLZ1 和 DTLZ3 的性能. 通过指标 IGD 的值可知, 1) 本文方法得到的 Pareto 前沿具有最好的性能, 其次是 MSOPS, 但是, 除了 6 目标优化问题 DTLZ1 和 15 目标优化问题 DTLZ3 之外, MSOPS 与本文方法在求解大多数优化问题时, 没有显著差异; 2) 虽然  $\varepsilon$ -MOEA 对于 6 目标优化问题 DTLZ1 和 DTLZ3 表现出很好的性能, 但是, 在更高维目标优化问题上, 该方法的性能却急剧下降, 这是因为, DTLZ1 和 DTLZ3 的目标函数值域很大, 其真实 Pareto 前沿的值域却很小, 导致  $\varepsilon$  的取值难以合理确定; 3) HypE、PPD-MOEA 和 NSGA-II 得到的 Pareto 前沿的性能明显劣于本文方法.

其次, 分析不同方法求解优化问题 DTLZ2 的性能. 根据不同方法得到的 Pareto 前沿的 IGD 指标, 这些方法性能优劣的排名依次是  $\varepsilon$ -MOEA、本文方法、MSOPS、PPD-MOEA、HypE 和 NSGA-II, 且这些方法的性能差异是显著的. 值得注意的是,  $\varepsilon$ -MOEA 对于不同目标函数的优化问题 DTLZ2, 显著优于本文方法. 这表明, 当  $\varepsilon$  取合适的值时,  $\varepsilon$ -MOEA 的性能也是能够优于本文方法的.

再次, 分析不同方法求解优化问题 DTLZ5 的性能. 根据 IGD 指标的值, 1) 本文方法和 MSOPS 都取得了很好的性能, 且对于 6 和 9 目标优化问题 DTLZ5, 本文方法显著优于 MSOPS; 虽然 MSOPS

表 5 不同方法求解 DTLZ 的 IGD 指标  
Table 5 Metric IGD of different methods for DTLZ problems

测试问题	目标个数	NSGA-II	PPD-MOEA	$\varepsilon$ -MOEA	HypE	MSOPS	本文方法
DTLZ1	6	4.784E+1 (2.5E+1)†	1.069E+0 (7.9E-1)†	1.157E-1 (1.1E-1)†	1.616E+0 (1.0E+0)†	1.127E-1 (4.5E-2)†	<b>7.853E-2</b> <b>(2.8E-3)</b>
	9	5.652E+1 (4.2E+1)†	2.326E+0 (1.5E+0)†	5.769E-1 (3.8E-1)†	1.777E+0 (8.4E-1)†	1.763E-1 (6.5E-2)	<b>1.628E-1</b> <b>(5.4E-2)</b>
	12	7.818E+1 (5.0E+1)†	3.388E+0 (2.5E+0)†	1.824E+0 (2.1E+0)†	2.522E+0 (9.7E-1)†	2.085E-1 (1.4E-1)	<b>1.828E-1</b> <b>(1.2E-1)</b>
	15	9.347E+1 (5.5E+1)†	4.924E+0 (4.3E+0)†	2.774E+0 (3.5E+0)†	3.184E+0 (2.7E+0)†	2.413E-1 (1.3E-1)	<b>1.961E-1</b> <b>(7.3E-2)</b>
DTLZ2	6	1.722E+0 (1.9E-1)†	4.709E-1 (4.8E-2)†	<b>2.659E-1</b> <b>(6.4E-3)†</b>	5.340E-1 (5.4E-2)†	3.228E-1 (2.1E-2)†	3.005E-1 (8.1E-3)
	9	2.071E+0 (1.4E-1)†	6.431E-1 (6.7E-2)†	<b>4.213E-1</b> <b>(1.1E-2)†</b>	6.967E-1 (4.2E-2)†	5.338E-1 (1.9E-2)†	4.909E-1 (1.3E-2)
	12	2.139E+0 (2.0E-1)†	7.527E-1 (6.7E-2)†	<b>5.010E-1</b> <b>(1.3E-2)†</b>	8.508E-1 (6.0E-2)†	6.174E-1 (1.7E-2)†	5.866E-1 (2.8E-2)
	15	2.194E+0 (1.9E-1)†	8.260E-1 (4.9E-2)†	<b>5.846E-1</b> <b>(4.1E-2)†</b>	9.393E-1 (2.9E-2)†	6.898E-1 (2.3E-2)†	6.474E-1 (1.9E-2)
DTLZ3	6	1.207E+2 (5.7E+1)†	1.071E+0 (5.6E-1)†	4.606E-1 (1.2E-1)†	3.259E+0 (3.0E+0)†	3.439E-1 (1.6E-2)	<b>3.248E-1</b> <b>(2.2E-2)</b>
	9	1.343E+2 (7.3E+1)†	2.510E+0 (1.5E+0)†	1.615E+0 (1.9E+0)†	3.439E+0 (2.4E+0)†	6.023E-1 (2.1E-1)	<b>5.616E-1</b> <b>(2.3E-2)</b>
	12	1.405E+2 (8.6E+1)†	3.789E+0 (1.9E+0)†	2.531E+1 (2.7E+1)†	3.874E+0 (1.8E+0)†	6.872E-1 (3.4E-1)	<b>6.460E-1</b> <b>(2.9E-2)</b>
	15	2.008E+2 (1.4E+2)†	4.621E+0 (3.9E+0)†	7.074E+1 (8.5E+1)†	5.514E+0 (3.4E+0)†	8.154E-1 (1.1E-1)†	<b>7.017E-1</b> <b>(6.4E-2)</b>
DTLZ5	6	1.461E-1 (3.2E-2)†	8.474E-2 (2.6E-2)†	8.187E-2 (1.4E-2)†	1.565E-1 (7.4E-2)†	3.144E-2 (5.8E-3)†	<b>2.140E-2</b> <b>(8.0E-3)</b>
	9	4.711E-1 (3.0E-1)†	1.106E-1 (2.1E-2)†	1.014E-1 (2.3E-2)†	1.980E-1 (7.7E-2)†	3.240E-2 (2.9E-3)†	<b>2.856E-2</b> <b>(5.4E-3)</b>
	12	9.118E-1 (2.6E-1)†	1.660E-1 (6.0E-2)†	1.019E-1 (1.8E-2)†	2.096E-1 (9.9E-2)†	<b>3.252E-2</b> <b>(5.8E-3)</b>	3.322E-2 (1.0E-2)
	15	1.754E+0 (6.3E-1)†	1.946E-1 (7.2E-2)†	1.259E-1 (2.4E-2)†	2.313E-1 (6.2E-2)†	<b>3.375E-2</b> <b>(8.7E-3)</b>	3.623E-2 (7.6E-3)
DTLZ7	6	1.401E+0 (3.1E-1)†	1.104E+0 (2.0E-1)†	<b>5.976E-1</b> <b>(1.1E-2)†</b>	1.186E+0 (3.6E-1)†	3.778E+0 (1.7E-1)†	8.653E-1 (2.2E-1)
	9	1.877E+0 (7.5E-1)†	1.873E+0 (6.2E-1)†	1.467E+0 (3.2E-1)†	1.637E+0 (5.2E-1)†	7.157E+0 (1.7E-1)†	<b>1.185E+0</b> <b>(3.3E-1)</b>
	12	3.482E+0 (1.6E+0)†	4.113E+0 (1.6E+0)†	6.477E+0 (2.1E+0)†	2.216E+0 (9.2E-1)†	1.152E+1 (2.2E-1)†	<b>1.664E+0</b> <b>(2.6E-1)</b>
	15	3.608E+0 (1.2E+0)†	4.908E+0 (1.5E+0)†	1.007E+1 (1.7E+0)†	2.951E+0 (8.0E-1)†	1.481E+1 (6.9E-2)†	<b>2.007E+0</b> <b>(5.1E-1)</b>

† 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 6 不同方法求解 MOTSP 的 HV 指标  
Table 6 Metric HV of different methods for MOTSP

目标个数	相关参数	NSGA-II	PPD-MOEA	$\varepsilon$ -MOEA	HypE	MSOPS	本文方法
6	0.5	37.3†	52.1†	94.6†	39.6†	89.2†	<b>100</b>
	0	32.6†	41.1†	<b>101.5</b>	45.0†	97.2†	100
	-0.5	31.2†	39.9†	98.7	41.7†	87.8†	<b>100</b>
12	0.5	29.4†	49.7†	79.6†	35.1†	96.1†	<b>100</b>
	0	24.6†	46.9†	81.3†	36.8†	93.2†	<b>100</b>
	-0.5	21.5†	45.7†	72.7†	32.7†	90.3†	<b>100</b>

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

求解 12 和 15 目标优化问题 DTLZ5 时, 得到了更好的 IGD 指标均值, 但是, 与本文方法得到的 IGD 指标均值不具有显著差异. 本文方法取得很好的性能的原因是, 在优化问题 DTLZ5 中, 前  $m-1$  个目标函数之间的相关程度非常高, 通过目标函数的分组与聚合, 求解任一子优化问题, 即可得到原优化问题的近似 Pareto 前沿, 还大幅度减少了目标函数; 2) 虽然 PPD-MOEA、HypE 和 NSGA-II 得到的 Pareto 前沿的 IGD 值也较好, 但是劣于  $\varepsilon$ -MOEA.

然后, 分析不同方法求解优化问题 DTLZ7 的性能. 从 IGD 指标值可以知道, 1) 对于 6 目标优化问题 DTLZ7,  $\varepsilon$ -MOEA 取得了最佳性能, 本文方法次之, 但是, 在更高维目标优化问题上, 本文方法保持了优越的性能, 而  $\varepsilon$ -MOEA 的性能却明显下降, 这仍归因于  $\varepsilon$  的取值; 2) 虽然求解 6 和 9 目标优化问题 DTLZ7, PPD-MOEA 与 HypE 的性能略优于 NSGA-II, 但是, 它们在求解 12 和 15 目标优化问题时, 性能却劣于 NSGA-II; 3) 在这些方法中, MSOPS 取得了最差的性能.

最后, 分析不同方法求解优化问题 MOTSP 的性能. 根据表 6 中的 HV 指标, 1) 除了相关参数为 0 的 6 目标优化问题 MOTSP 之外, 本文方法均取得最好的性能; 2) 虽然求解相关参数为 0 和 -0.5 的 6 目标优化问题 MOTSP 时, 本文方法与  $\varepsilon$ -MOEA 没有显著差异, 但是, 当该问题的相关参数为 0.5 时, 即目标函数具有较大的相关关系时, 本文方法明显优于  $\varepsilon$ -MOEA. 这说明, 本文方法对于目标相关程度较高的优化问题更有效; 3)  $\varepsilon$ -MOEA 求解 6 目标优化问题 MOTSP 时取得的很好的性能, 优于大部分其他方法, 但是, 在求解 12 目标优化问题 MOTSP 时, MSOPS 取得了仅次于本文方法的性能.

通过第二部分的实验结果与分析, 可以得到如下结论: 1) 由于 Pareto 占优的局限性, NSGA-II 在处理大多数优化问题时, 均取得了最差的性能, 但是, 在求解 Pareto 前沿不连续的优化问题时, 其他方法并不一定比其更优; 2)  $\varepsilon$ -MOEA、MSOPS、HypE

和 PPD-MOEA 均能够有效求解大部分高维多目标优化问题, 但是, 对于  $\varepsilon$ -MOEA 和 MSOPS, 需要设置合适的参数, 以获得优异的性能; 3) 除了求解优化问题 DTLZ2、6 目标优化问题 DTLZ7 以及相关参数为 0 的 6 目标优化问题 MOTSP, 本文方法得到的 Pareto 前沿劣于  $\varepsilon$ -MOEA 之外, 对于其他优化问题, 本文方法均具有最佳的性能.

#### 4 总结及下一步的工作

高维多目标优化问题包含的目标函数增多, 使得该问题的求解非常困难. 如果采用合适的方法, 将目标函数分解, 那么, 有可能降低问题求解的难度. 本文提出一种基于目标分解的高维多目标优化问题并行进化求解方法, 通过目标函数分组与聚合, 将一个高维多目标优化问题分解为若干子问题, 从而降低了问题求解的难度. 与其他分解方法不同的是, 本文基于 Spearman 相关系数对目标函数分组, 保证了同一组的目标函数具有较高的相似度, 这为子优化问题的高效求解提供了保障; 对于每组目标函数形成的子优化问题, 通过多种群并行进化算法的一个子种群, 在同构并行环境的一个子进程上进化, 提高了问题求解的速度; 采用改进的外部保存集形成策略, 基于各子种群的最优个体, 在一个新的进程上, 形成外部保存集, 保证了优化问题 Pareto 最优解集的逼近与分布性能.

将所提方法应用于求解 DTLZ 和 MOTSP 优化问题, 并与多种典型的进化优化方法比较. 实验结果表明, 本文提出的基于 Spearman 相关系数的目标函数分组方法能够有效提高所提方法的性能; 此外, 目标函数分组个数也会影响本文方法的性能, 并通过实验给出了分组个数的合理取值, 以得到各项指标均衡的 Pareto 最优解集; 最后, 与其他 5 种方法相比, 本文方法求解多数优化问题时, 取得了最佳的性能, 表明了本文方法是一种非常优越的求解高维多目标优化问题的方法.

在本文的实验中, 仅通过几个典型的基准优化问题来验证本文方法的有效性, 还没有将其应用于实际的优化问题. 因此, 下一步将考虑本文方法在更多实际优化问题中的应用, 以全面评价所提方法的性能; 除了本文给出的目标函数聚合方法之外, 还可以采用其他的聚合方法, 如 MOEA/D 中的切比雪夫方法和 PBI (Penalty-based boundary intersection) 方法等. 这些方法基于理想点聚合目标函数, 具有不同的几何意义, 可能更加适用于具有某种类型 Pareto 前沿的优化问题. 因此, 寻找合适的目标函数聚合方法, 也是下一步需要研究的问题; 最后, 对于不同的子优化问题, 包含的目标函数的相关程度是不同的, 一般来讲, 相关度较高的目标函数形成的子优化问题, 对问题 Pareto 最优解集的贡献度较小, 但是, 本文用于求解任一子优化问题的子种群规模都是相同的, 这使得消耗了很大的计算量来用于求解对 Pareto 最优解集贡献较小的子问题. 如果根据对 Pareto 最优解集的贡献度, 合理确定用于求解每一子优化问题的子种群规模, 那么, 将会进一步提高本文方法的求解高维多目标优化问题的效率. 因此, 多种群并行遗传算法每一子种群规模的确定, 是我们需要进一步研究的问题.

## References

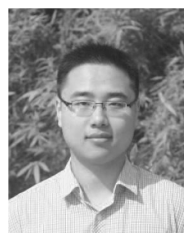
- Zhang Xiao-Dong, Yao Xiao-Lan, Wu Qing-He, Li Dao-Ping. Design and application of a class of multiplexed receding horizon controllers for looper control system. *Acta Automatica Sinica*, 2011, **37**(3): 380–384  
(张晓东, 姚小兰, 伍清河, 李道平. 一类基于滚动时域优化原理的多路控制器设计及其在活套控制中的应用. *自动化学报*, 2011, **37**(3): 380–384)
- Kong Wei-Jian, Chai Tian-You, Ding Jin-Liang, Wu Zhi-Wei. A real-time multiobjective electric energy allocation optimization approach for the smelting process of magnesia. *Acta Automatica Sinica*, 2014, **40**(1): 51–61  
(孔维健, 柴天佑, 丁进良, 吴志伟. 镁砂熔炼过程全厂电能分配实时多目标优化方法研究. *自动化学报*, 2014, **40**(1): 51–61)
- 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)
- 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
- Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Proceedings of the 5th Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Athens, Greece: CIMNE, 2001. 95–100
- Deb K, Saxena D K. Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: Proceedings of the 2006 IEEE Congress on Evolutionary Computation. Vancouver, Canada: IEEE, 2006. 3353–3360
- Jaimes A L, Coello C A C, Chakraborty D. Objective reduction using a feature selection technique. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation. New York, USA: ACM, 2008. 673–680
- Adra S F, Fleming P J. Diversity management in evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2011, **15**(2): 183–195
- Li M Q, Yang S X, Liu X H. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2014, **18**(3): 348–365
- Sato H, Aguirre H E, Tanaka K. Pareto partial dominance MOEA and hybrid archiving strategy included CDAS in many-objective optimization. In: Proceedings of the 2010 IEEE Congress on Evolutionary Computation. Barcelona, Spain: IEEE, 2010. 1–8
- Jaimes A L, Coello C A C, Aguirre H, Tanaka K. Objective space partitioning using conflict information for solving many-objective problems. *Information Sciences*, 2014, **268**: 305–327
- Deb K, Mohan M, Mishra S. Evaluating the  $\epsilon$ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolutionary Computation*, 2005, **13**(4): 501–525
- Kukkonen S, Lampinen J. Ranking-dominance and many-objective optimization. In: Proceedings of the 2007 IEEE Congress on Evolutionary Computation. Singapore: IEEE, 2007. 3983–3990
- Zitzler E, Künzli S. Indicator-based selection in multiobjective search. In: Proceedings of the 8th International Conference on Parallel Problem Solving from Nature. Berlin: Springer-Verlag, 2004. 832–842
- Bader J, Zitzler E. HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 2011, **19**(1): 45–76
- Hughes E J. Multiple single objective Pareto sampling. In: Proceedings of the 2003 IEEE Congress on Evolutionary Computation. Canberra, Australia: IEEE, 2003. 2678–2684
- Zhang Q F, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, **11**(6): 712–730
- Zitzler E, Thiele L, Bader J. On set-based multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2010, **14**(1): 58–79
- Gong Dun-Wei, Ji Xin-Fang, Sun Xiao-Yan. Solving many-objective optimization problems using set-based evolutionary algorithms. *Acta Electronica Sinica*, 2014, **42**(1): 77–83  
(巩敦卫, 季新芳, 孙晓燕. 基于集合的高维多目标优化问题的进化算法. *电子学报*, 2014, **42**(1): 77–83)
- Toro F, Ortega J, Fernandez J, Diaz A. PSFGA: a parallel genetic algorithm for multiobjective optimization. In: Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing. Canary Islands: IEEE Computer Society, 2002. 384–391
- Qi R B, Ma T Y, Sun F, Qian F. Evolutionary algorithm based on the evolution of Pareto archive and individual migration. In: Proceedings of the 2011 International Symposium on Advanced Control of Industrial Processes. Hangzhou, China: IEEE, 2011. 90–95

- 22 Wang Xing. *Non-Parametric Statistics*. Beijing: Tsinghua University Press, 2009. 180–184  
(王星. 非参数统计. 北京: 清华大学出版社, 2009. 180–184)
- 23 Murata T, Taki A. Examination of the performance of objective reduction using correlation-based weighted-sum for many objective knapsack problems. In: Proceedings of the 10th International Conference on Hybrid Intelligent Systems. Atlanta, GA: IEEE, 2010. 175–180
- 24 Deb K, Thiele L, Laumanns M, Zitzler E. Scalable test problems for evolutionary multiobjective optimization. In: Proceedings of the 2005 Evolutionary Multiobjective Optimization. London: Springer-Verlag, 2005. 105–145
- 25 Corne D W, Knowles J D. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In: Proceedings of the 9th Annual Conference Genetic Evolutionary Computation. London, England: ACM, 2007. 773–780
- 26 van Veldhuizen D A, Lamont G B. On measuring multiobjective evolutionary algorithm performance. In: Proceedings of the 2000 IEEE Congress on Evolutionary Computation. San Diego, CA, US: IEEE, 2000. 204–211
- 27 Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, 2000, 8(2): 173–195
- 28 Zhang Q F, Zhou A M, Zhao S Z, Suganthan P N, Liu W D, Tiwari S. Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition, Technical Report CES-487, University of Essex, Colchester, UK, and Nanyang Technological University, Singapore, 2008.
- 29 Ishibuchi H, Akedo N, Nojima Y. Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. *IEEE Transactions on Evolutionary Computation*, 2015, 19(2): 264–283



**巩敦卫** 中国矿业大学信息与电气工程学院教授. 1999 年在中国矿业大学获博士学位. 主要研究方向为进化计算与应用. E-mail: dwgong@vip.163.com  
(**GONG Dun-Wei** Professor at the School of Information and Electronic 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.)



**刘益萍** 中国矿业大学信息与电气工程学院博士研究生. 2012 年在中国矿业大学获学士学位. 主要研究方向为多目标进化优化. 本文通信作者.

E-mail: lyp.cumt@gmail.com

(**LIU Yi-Ping** Ph. D. candidate at the School of Information and Electronic Engineering, China University of

Mining and Technology. He received his bachelor degree from China University of Mining and Technology in 2012. His research interest covers evolutionary computation and multi-objective optimization. Corresponding author of this paper.)



**孙晓燕** 中国矿业大学信息与电气工程学院教授. 2009 年在中国矿业大学获博士学位. 主要研究方向为交互式进化计算. E-mail: xysun78@126.com

(**SUN Xiao-Yan** Professor at the School of Information and Electronic Engineering, China University of Mining and Technology. She received her

Ph. D. degree from China University of Mining and Technology in 2009. Her research interest covers interactive evolutionary computation.)



**韩玉艳** 中国矿业大学信息与电气工程学院博士研究生. 2012 年在聊城大学获硕士学位. 主要研究方向为调度优化.

E-mail: yuyanhan1023@163.com

(**HAN Yu-Yan** Ph. D. candidate at the School of Information and Electronic Engineering, China University of Mining and Technology. She received

her master degree from Liaocheng University in 2012. Her research interest covers scheduling optimization.)