

# 一种基于混合高斯模型的多目标进化算法\*

周爱民<sup>1</sup>, 张青富<sup>2</sup>, 张桂戌<sup>1</sup>

<sup>1</sup>(华东师范大学 计算机科学与技术系, 上海 200241)

<sup>2</sup>(School of Computer Science and Electronic Engineering, University of Essex, UK)

通讯作者: 张桂戌, E-mail: gxzhang@cs.ecnu.edu.cn

**摘要:** 目前,大多数多目标进化算法采用为单目标优化所设计的重组算子.通过证明或实验分析了几个典型的单目标优化重组算子并不适合某些多目标优化问题.提出了基于分解技术和混合高斯模型的多目标优化算法(multiobjective evolutionary algorithm based on decomposition and mixture Gaussian models,简称 MOEA/D-MG).该算法首先采用一个改进的混合高斯模型对群体建模并采样产生新个体,然后利用一个贪婪策略来更新群体.针对具有复杂 Pareto 前沿的多目标优化问题的测试结果表明,对给定的大多数测试题,该算法具有良好的效果.

**关键词:** 多目标优化;进化算法;MOEA/D;混合高斯概率模型

**中图法分类号:** TP181

中文引用格式: 周爱民,张青富,张桂戌.一种基于混合高斯模型的多目标进化算法.软件学报,2014,25(5):913-928. <http://www.jos.org.cn/1000-9825/4514.htm>

英文引用格式: Zhou AM, Zhang QF, Zhang GX. Multiobjective evolutionary algorithm based on mixture Gaussian models. Ruan Jian Xue Bao/Journal of Software, 2014,25(5):913-928 (in Chinese). <http://www.jos.org.cn/1000-9825/4514.htm>

## Multiobjective Evolutionary Algorithm Based on Mixture Gaussian Models

ZHOU Ai-Min<sup>1</sup>, ZHANG Qing-Fu<sup>2</sup>, ZHANG Gui-Xu<sup>1</sup>

<sup>1</sup>(Department of Computer Science and Technology, East China Normal University, Shanghai 200241, China)

<sup>2</sup>(School of Computer Science and Electronic Engineering, University of Essex, UK)

Corresponding author: ZHANG Gui-Xu, E-mail: gxzhang@cs.ecnu.edu.cn

**Abstract:** Recombination operators used in most current multiobjective evolutionary algorithms (MOEAs) were originally designed for single objective optimization. This paper demonstrates that some widely used recombination operators may not work well for multiobjective optimization problems (MOPs), and proposes a multiobjective evolutionary algorithm based on decomposition and mixture Gaussian models (MOEA/D-MG). In the algorithm, a reproduction operator based on mixture Gaussian models is used to model the population distribution and sample new trails solutions, and a greedy replacement scheme is then applied to update the population by the new trial solutions. MOEA/D-MG is applied to a variety of test instances with complicated Pareto fronts. The extensive experimental results indicate that MOEA/D-MG is promising for dealing with these continuous MOPs.

**Key words:** multiobjective optimization; evolutionary algorithm; MOEA/D; mixture Gaussian probability model

多目标优化问题(multiobjective optimization problem,简称 MOP)是在科学研究与生产应用领域广泛存在的一类挑战性优化问题.MOP 目标之间的冲突性会导致通常不存在一个最优解满足所有的优化目标.因此,人们往往需要对各个目标做出折衷得到 Pareto 最优解集<sup>[1]</sup>.对于一般 MOP 问题,目前还不存在解析方法求解.因此,我们常采用计算机算法来获得对 Pareto 解集的一个逼近.进化算法(evolutionary algorithm,简称 EA)是一类模仿自然界,特别是生物进化过程来求解复杂优化问题的计算机算法模型.EA 算法具有两个重要特征:(1) 对所求解

\* 基金项目: 国家重点基础研究发展计划(973)(2011CB707104); 国家自然科学基金(61273313, 61372147)

收稿时间: 2013-02-04; 修改时间: 2013-09-18; 定稿时间: 2013-10-31

问题的性质,如连续性、可微性等,不作特别假设;(2) 基于群体搜索,即,采用多点同时搜索问题最优解.因此,EA 算法特别适合求解复杂的非线性和黑盒 MOP 问题.

自从 1985 年第 1 种多目标进化算法(multiobjective evolutionary algorithm,简称 MOEA)提出以来<sup>[2]</sup>,MOEA 已发展成为求解 MOP 问题的主流方法之一;同时,MOEA 也已成为 EA 领域最热门的研究方向之一<sup>[3-5]</sup>.国际上,以 NSGA-II<sup>[6]</sup>,SPEA2<sup>[7]</sup>,PAES<sup>[8]</sup>,IBEA<sup>[9,10]</sup>,MOEA/D<sup>[11]</sup>等为代表的 MOEA 算法在众多应用领域获得了广泛的使用.同时,MOEA 研究在国内也获得了快速的发展<sup>[12-14]</sup>,研究主要集中在以下几个方向:(1) 算法设计,如基于实验设计的多目标算法<sup>[15-17]</sup>、基于差分进化的多目标算法<sup>[17,18]</sup>、基于粒子群的多目标算法<sup>[19,20]</sup>、基于免疫的多目标算法<sup>[21,22]</sup>、基于量子计算的多目标算法<sup>[23]</sup>、拟态物理学多目标算法<sup>[24]</sup>、多目标分布估计算法<sup>[25]</sup>及新 Pareto 支配关系设计<sup>[26]</sup>等;(2) 算法应用,如处理约束优化问题<sup>[27]</sup>、解决高维多目标(many-objective)优化问题<sup>[28]</sup>、路径规划<sup>[29]</sup>、分类<sup>[30]</sup>及其他实际应用<sup>[31-34]</sup>等;(3) 针对 MOEA 的特殊数据结构<sup>[35]</sup>、MOEA 算法评估<sup>[36]</sup>等.

根据 MOEA 算法所采用的基本思想,近 10 年左右出现的 MOEA 算法大致可以分为以下 3 类:

#### (1) 基于 Pareto 占优关系的 MOEA 算法

这类算法以 NSGA-II<sup>[6]</sup>和 SPEA2<sup>[7]</sup>等方法为代表,其基本思想是:采用 Pareto 占优关系对群体进行分类;然后,在每一类中计算个体的密度.通过占优关系和密度估计值,对群体进行全排序,并选择相对较优的个体进入下一代.人们常采用拥挤距离(crowding distance)、 $k$ -最近邻( $k$ -nearest neighbor)、 $\varepsilon$ -占优( $\varepsilon$ -domination)、网格划分(grading)等方法来估计个体的密度.因为基于 Pareto 占优关系的 MOEA 算法具有原理简单、容易理解、参数较少等优点,这类算法已引起了众多学者的深入研究和广泛的应用.

#### (2) 基于评估指标的 MOEA 算法

这类算法以 IBEA<sup>[9]</sup>和 HypE<sup>[10]</sup>等算法为代表,其基本思想是:用直接优化 Pareto 逼近集合的评估指标(performance indicator)的方式来间接优化原 MOP,这实际上是将多目标优化问题转换为一个单目标优化问题.在这类方法中,超体积(hypervolume)评估指标最为常用.这一类 MOEA 算法避免了 Pareto 占优引起的大量比较问题,但是由于超体积本身的计算量较大,需要引入蒙特卡洛估计等方法来提高计算速度<sup>[10]</sup>.

#### (3) 基于分解技术的 MOEA 算法

基于分解技术的 MOEA(MOEA based on decomposition,简称 MOEA/D)<sup>[11]</sup>是一类新的 MOEA 算法框架.和传统的基于分解的算法一样,MOEA/D 将 MOP 分解为一系列子问题来求解.然而传统方法需要分别求解每个子问题,因而计算量大.MOEA/D 将这些子问题有机组织起来同时求解而获得 Pareto 解集的一个逼近,由于算法只运行一次,所以能获得较高的计算效率.MOEA/D 中,每个子问题是一个单目标优化问题,因而 MOEA/D 算法框架能容纳各类单目标优化方法和局部搜索方法,并自然地建立了启发式算法与传统数学规划方法之间的联系.MOEA/D 算法框架目前已引起越来越多的研究者的关注.

重组算子(recombination operator)是 EA 算法的一个重要组成部分,其对父体进行某种变换以产生新个体.几乎所有的单目标优化重组算子,如粒子群算子、差分算子、人工免疫算子及各种杂交、变异算子,都已被应用于 MOEA 算法中.MOEA 在算法设计和应用上的成功使许多研究者忽略了 MOEA 重组算子的重要性.单目标优化问题与多目标优化问题最优解的根本区别在于:前者的解是单个(或可数个)孤立点,而后的解是一个 Pareto 解集.一些学者已经开始意识到单目标和多目标最优解在拓扑上的本质区别,并且证实:在旋转问题或具有复杂 Pareto 解集的 MOP 问题上,某些单目标重组算子在 MOEA 中效率低下<sup>[37,38]</sup>.为了提高 MOEA 算法在求解复杂 MOP 问题上的效率,利用 MOP 问题相关信息构造 MOEA 重组算子迫在眉睫<sup>[39]</sup>.相对于选择算子的研究工作,重组算子的工作还没有引起足够的重视:一方面,MOP 问题,特别是超多目标优化问题和离散 MOP 问题 Pareto 最优集解的特性研究不够充分;另一方面,利用问题特征设计针对 MOP 问题的 MOEA 重组算子也不是很多.我们首先将连续 MOP 问题的规则特性(regularity property)<sup>[40]</sup>引入到 MOEA 重组算子的设计中,并提出了基于规则特性的分布估计算法 RM-MEDA<sup>[38,41]</sup>.RM-MEDA 重组算子的核心思想是:利用概率分布模型去逼近 MOP Pareto 最优解集的流形(manifold),并用该模型采样产生新个体.RM-MEDA 能够解决一些变量之间有线性或非线性的复杂 MOP 问题.作为一种概念算法,RM-MEDA 主要在两个方面需要进一步改进:一是由于

使用了降维算法(dimension reduction algorithm),RM-MEDA 算法本身的时间复杂度远远大于一般的 MOEA;二是对 Pareto 解集具有复杂几何形状的 MOP 问题,RM-MEDA 效率不是很高。

基于上述研究基础,文献[42]提出:在 MOEA/D 算法框架中,采用基于高斯概率模型的重组算子来对种群建模和采样.初步实验结果表明:高斯概率模型能够有效逼近 Pareto 最优解集的流形,对具有复杂几何形状的 MOP 问题能够取得较为理想的实验效果.本文对文献[42]所提算法做了进一步的改进,提出了新的基于分解和混合高斯模型的多目标优化算法(multiobjective evolutionary algorithm based on decomposition and mixture Gaussian models,简称 MOEA/D-GM).本文工作与文献[42]工作的主要区别在于:(1) 针对几个经典的重组算子,证明或分析了它们在多目标优化中的不足,指出了利用 MOP 问题性质设计重组算子的必要性;(2) 提出了一个改进的基于混合高斯概率模型的重组算子,该算子充分利用 MOEA/D 中邻居个体的相似性来重用邻居模型,在保证建模质量的情况下降低建模的次数,从而提高建模效率;(3) 设计了一个贪婪更新策略,该策略总是寻找改进程度最大的子问题去更新,以有效推动种群进化。

## 1 问题定义

不失一般性,一个具有  $n$  个决策变量, $m$  个目标函数的 MOP 问题可以定义为

$$\begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T \\ \text{s.t. } x \in \Omega \end{cases} \quad (1)$$

其中,  $(x_1, x_2, \dots, x_n)^T \in \Omega$  是决策向量,  $\Omega$  是可行域空间,  $f_i: x \rightarrow R (i=1, \dots, m)$  是第  $i$  个目标函数.对于给定的 MOP,其目标之间往往相互冲突,一个解可能对于某个目标来说较好,而对于其他目标来讲可能较差.因此一般情况下, MOP 的最优解不是一个解而是一个被称做 Pareto 最优解集的集合。

**定义 1(Pareto 支配).** 假设  $x, y \in \Omega$  是上述 MOP 问题的可行解,称  $x$  Pareto 支配  $y$  (记作  $x \prec y$ ),当且仅当

$$\forall i=1, 2, \dots, m, f_i(x) \leq f_i(y) \wedge F(x) \neq F(y) \quad (2)$$

**定义 2(Pareto 最优解集).** Pareto 最优解集(Pareto set,简称 PS)是所有 Pareto 最优解的集合:

$$PS = \{x | \neg \exists y \in \Omega, y \prec x\} \quad (3)$$

**定义 3(Pareto 最优前沿).** Pareto 最优前沿(Pareto front,简称 PF)是 Pareto 最优解集在目标空间的投影:

$$PF = \{F(x) | x \in PS\} \quad (4)$$

连续 MOP 问题的 Pareto 解集不仅在拓扑结构上与单目标优化问题最优解不同,而且已经证明其具有如下优良的规则特性(regularity property)<sup>[40]</sup>:

**定理 1.** 在某些特定条件下,一个包含  $m$  个目标的连续 MOP 问题,其 Pareto 最优解集在解空间和目标空间中都是一个分段连续的  $(m-1)$  维流形。

该定理说明,连续 MOP 问题的 Pareto 最优解不是杂乱分布的,而是表现出一定的模式.因而,MOEA 可以通过对 Pareto 最优解集流形的逼近来间接地实现对 Pareto 最优解集的逼近。

## 2 传统重组算子的不足

当前,大部分进化重组算子,如杂交(crossover)算子和变异(mutation)算子等,是为单目标优化而设计.本节通过分析下述 3 种常见的重组算子来说明这些重组算子在多目标优化情况下的不足。

- 单点杂交算子:对父体  $x^A = (x_1^A, x_2^A, \dots, x_n^A)^T$  和  $x^B = (x_1^B, x_2^B, \dots, x_n^B)^T$ , 随机选择杂交位  $1 \leq k < n$ , 交换染色体元素,得到如下新个体:

$$x^a = (x_1^A, \dots, x_k^A, x_{k+1}^B, \dots, x_n^B) \text{ 和 } x^b = (x_1^B, \dots, x_k^B, x_{k+1}^A, \dots, x_n^A).$$

- 算术杂交算子:对父体  $x^A = (x_1^A, x_2^A, \dots, x_n^A)^T$  和  $x^B = (x_1^B, x_2^B, \dots, x_n^B)^T$ , 随机选择参数  $0.0 < \alpha < 1.0$ , 构造新解:

$$x^a = (\alpha x_1^A + (1-\alpha)x_1^B, \dots, \alpha x_n^A + (1-\alpha)x_n^B) \text{ 和 } x^b = ((1-\alpha)x_1^A + \alpha x_1^B, \dots, (1-\alpha)x_n^A + \alpha x_n^B).$$

- 高斯模型采样:从多元高斯分布模型  $x \sim N(\mu, \Sigma)$  采样得到新个体,其中,  $\mu$  和  $\Sigma$  分别是对群体建模得到的决

策向量的均值和协方差矩阵.

对于单目标优化问题,假设当前种群中父体靠近最优解,我们可以得出以下结论:

**引理 1.** 若  $\|x^A - x^*\| = \sqrt{\sum_{i=1}^n (x_i - x_i^*)^2} \leq \varepsilon, \|x^B - x^*\| = \sqrt{\sum_{i=1}^n (x_i - x_i^*)^2} \leq \varepsilon$ , 其中  $x^*$  为问题最优解,  $\varepsilon > 0$ , 则单点杂交后代  $x^a$  和  $x^b$  满足:  $\|x^a - x^*\| \leq \varepsilon$  或  $\|x^b - x^*\| \leq \varepsilon$ .

证明:假设结论不成立,即  $\|x^a - x^*\| > \varepsilon$  且  $\|x^b - x^*\| > \varepsilon$ , 显然,  $\|x^a - x^*\|^2 + \|x^b - x^*\|^2 > 2\varepsilon^2$ , 由前提条件可得:

$$0 \leq \|x^a - x^*\|^2 + \|x^b - x^*\|^2 = \|x^A - x^*\|^2 + \|x^B - x^*\|^2 \leq 2\varepsilon^2.$$

矛盾.所以假设不成立.得证. □

**引理 2.** 若  $\|x^A - x^*\| = \sqrt{\sum_{i=1}^n (x_i - x_i^*)^2} \leq \varepsilon, \|x^B - x^*\| = \sqrt{\sum_{i=1}^n (x_i - x_i^*)^2} \leq \varepsilon$ , 其中  $x^*$  为问题最优解,  $\varepsilon > 0$ , 则算术杂交两个后代均满足:  $\|x^a - x^*\| \leq \varepsilon$  且  $\|x^b - x^*\| \leq \varepsilon$ .

证明:由于  $f(x) = x^2$  是凸函数,可知:

$$\begin{aligned} \|x^a - x^*\|^2 &= \sum_{i=1}^n (\alpha x_i^A + (1-\alpha)x_i^B - x_i^*)^2 \\ &= \sum_{i=1}^n (\alpha(x_i^A - x_i^*) + (1-\alpha)(x_i^B - x_i^*))^2 \\ &\leq \sum_{i=1}^n (\alpha(x_i^A - x_i^*)^2 + (1-\alpha)(x_i^B - x_i^*)^2) \\ &= \alpha \sum_{i=1}^n (x_i^A - x_i^*)^2 + (1-\alpha) \sum_{i=1}^n (x_i^B - x_i^*)^2 \\ &= \alpha \|x^A - x^*\|^2 + (1-\alpha) \|x^B - x^*\|^2 \\ &\leq \alpha \varepsilon^2 + (1-\alpha) \varepsilon^2 \\ &= \varepsilon^2. \end{aligned}$$

即,  $\|x^a - x^*\| \leq \varepsilon$ . 同理可证  $\|x^b - x^*\| \leq \varepsilon$ . □

**引理 3.** 考虑一维优化问题,假设当前群体为  $\{x^1, x^2, \dots, x^m\}$ , 所有个体均满足  $\|x^k - x^*\| \leq \varepsilon, k=1, 2, \dots, m$ , 其中  $x^*$  为问题最优解,  $\varepsilon > 0$ , 群体样本均值和标准差为  $\mu$  和  $\sigma$ . 令  $y \sim N(\mu, \sigma^2)$  为从高斯分布采样得到的后代, 则

$$\text{Prob}(|y - x^*| \leq (1 + 2\sqrt{2})\varepsilon) > 0.95.$$

证明:样本均值  $\mu$  满足:

$$|\mu - x^*| = \left| \frac{1}{m} \sum_{k=1}^m x^k - x^* \right| = \frac{1}{m} \left| \sum_{k=1}^m (x^k - x^*) \right| \leq \frac{1}{m} \sum_{k=1}^m |x^k - x^*| \leq \varepsilon.$$

由  $\|x^k - x^*\| \leq \varepsilon$  及  $\|\mu - x^*\| \leq \varepsilon$  可知:  $-\varepsilon + x^* \leq x^k \leq \varepsilon + x^*$  及  $-\varepsilon - x^* \leq -\mu \leq \varepsilon - x^*$ , 则  $-2\varepsilon \leq x^k - \mu \leq 2\varepsilon$ , 即,  $|x^k - \mu| \leq 2\varepsilon$ . 样本均方差  $\sigma$  满足:

$$\sigma = \sqrt{\frac{1}{m-1} \sum_{k=1}^m (x^k - \mu)^2} \leq \sqrt{\frac{1}{m-1} \sum_{k=1}^m (2\varepsilon)^2} = 2\varepsilon \sqrt{\frac{m}{m-1}} < 2\sqrt{2}\varepsilon \quad (\text{假设个体数 } m > 2).$$

对采样个体  $y$ , 有  $|y - x^*| = |y - \mu + \mu - x^*| \leq |y - \mu| + |\mu - x^*| \leq |y - \mu| + \varepsilon$ .

由正态分布的特性可知,新个体  $y$  落入  $\mu$  左右均方差范围内的概率为 0.95, 即,  $\text{Prob}(|y - \mu| \leq \varepsilon) = 0.95$ .

显然,  $\text{Prob}(|y - x^*| - \varepsilon \leq \sigma) > 0.95$ , 即,  $\text{Prob}(|y - x^*| \leq \varepsilon + \sigma) > 0.95$ .

由  $\sigma < 2\sqrt{2}\varepsilon$  可得,  $\text{Prob}(|y - x^*| \leq \varepsilon + 2\sqrt{2}\varepsilon) > 0.95$ .

即,采样新个体以大于 0.95 的概率分布在最优解  $x^*$  周围.得证. □

上述 3 个引理表明:当种群逼近至最优解附近时,采用上述重组算子生成的子代个体仍然有较大的可能性分布在最优解附近.图 1 显示了在 2 维搜索空间中 3 个算子的示意图.

在多目标优化中,最优解不是一个孤立点  $x^*$ , 而是一个 Pareto 最优解集,并呈现出几何流形结构.在文献[38]中,我们已经指出:当种群靠近最优集合时,采用一般重组算子,后代个体可能偏离最优集合.图 2 显示了在 2 维搜

索空间中,上述 3 个重组算子的效率可能变得低下.

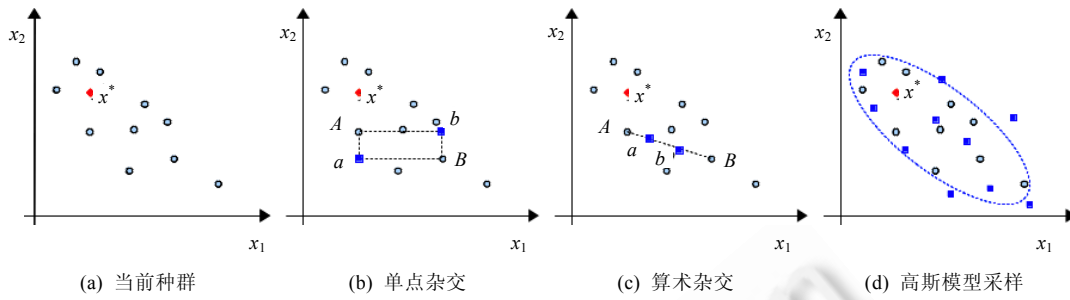


Fig.1 An illustration of reproduction operation in single objective optimization

图 1 单目标优化重组示意图

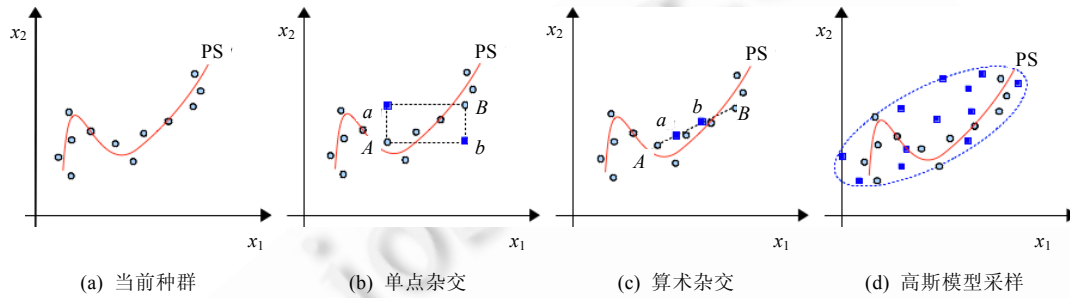


Fig.2 An illustration of reproduction operation in multiobjective optimization

图 2 多目标优化重组示意图

综合上述分析可知:当种群逼近最优解(集)时,一些常用的单目标优化重组算子,如单点杂交、算术杂交、高斯采样等,对单目标优化问题是稳定的,即,其后代个体以较大概率分布在最优解附近;这些算子对多目标优化问题是不稳定的,即,其后代以较大概率偏离最优解集.因此,需要针对多目标优化问题设计特别的重组算子.

### 3 基于混合高斯概率模型的多目标进化算法

#### 3.1 基于分解的多目标进化算法框架

基于分解技术的多目标进化算法 MOEA/D<sup>[11,43]</sup>是一类新的 MOEA 算法框架.MOEA/D 算法是对传统分解方法的有效扩展:和传统方法一样,MOEA/D 首先将一个 MOP 分解为一组单目标优化问题(子问题);与传统方法不同的是,MOEA/D 通过子问题之间的合作来同时优化这组子问题,从而获得对 Pareto 最优解集的一个逼近.相比传统方法串行解决子问题的方式,MOEA/D 这种并行求解子问题的策略能够有效提高算法搜索效率.同时,MOEA/D 框架可以自然地将各种单目标优化、局部搜索方法作为搜索算子使用.图 3 以 2 目标优化问题为例显示了 MOEA/D 算法的基本思想,其中,子问题 4 的邻居包含子问题 2~子问题 6.下面通过 MOEA/D 算法中的两个重要概念来进一步介绍该算法.

**定义 4(子问题(subproblem)).** 一个 MOP 问题  $\min F(x)$  可以转换为的一组子问题  $\min g^i(x), i=1,2,\dots,N$ , 每个子问题的最优解对应于原 MOP 问题的一个 Pareto 最优解.

如果子问题定义合理,那么这组子问题的最优解可以作为 Pareto 最优解集的一个逼近.通常,基于分解技术的算法通过给定一组权重向量  $w^i = (w_1^i, w_2^i, \dots, w_m^i)^T, i=1,2,\dots,N$  来定义子问题.本文使用切比雪夫(Chebyshev)方法来定义子问题:

$$g^i(x) = g(x|w^i, z^*) = \max_{j=1, \dots, m} \{w_j^i(f_j(x) - z_j^*)\} \tag{5}$$

其中,  $z^* = (z_1^*, z_2^*, \dots, z_m^*)$  是 MOP 问题的理想点(ideal point), 即  $z_j^* < \min_{x \in \Omega} f_j(x), j=1, 2, \dots, m$ . 可以证明: MOP 问题每一个 Pareto 最优解都对应一个切比雪夫子问题.

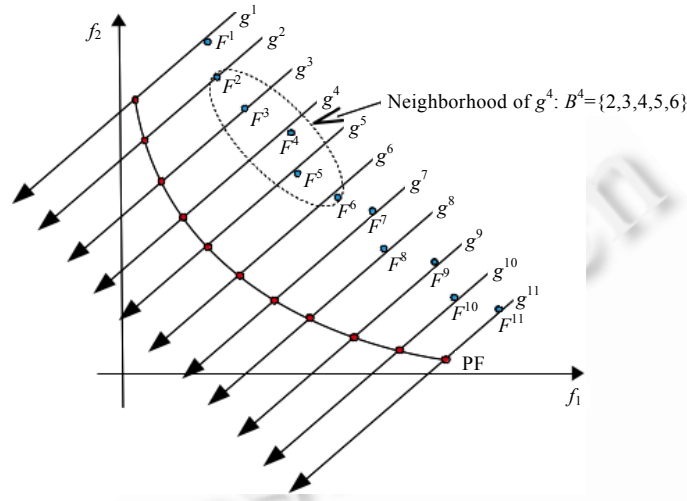


Fig.3 An illustration of MOEA/D  
图3 MOEA/D 算法示意图

权重向量  $w^i = (w_1^i, w_2^i, \dots, w_m^i)$  需满足条件  $0 \leq w_j^i \leq 1, j=1, 2, \dots, m \wedge \sum_{j=1}^m w_j^i = 1$ . 本文中,  $N$  个权重向量是事先在权重向量空间中均匀选择的, 具体算法可以参考文献[11]. 需要指出的是, 一组均匀给定的权重向量不一定能获得一组均匀分布的最优解. 权重向量的选择与 PF 的形状密切相关. 因此, 如何在运行过程中动态调整权重向量以获得均匀分布的最优解, 是目前 MOEA/D 算法设计中的一个研究热点.

**定义 5(邻域子问题(neighborhood)).** 一个子问题  $\min g^i$  的邻域子问题, 是一组与其最相似的  $K$  个子问题  $\{\min g^j | j \in B^i = \{i_1, i_2, \dots, i_K\}\}$ . MOEA/D 通过权重向量的距离来定义子问题的相似度, 权重向量距离越小, 则对应子问题越相似. 上标  $i_j$  表示  $w^{i_j}$  是与  $w^i$  第  $j$  近的权重向量. 显然,  $i_1 = i$ , 即, 子问题与其自身最相似.

邻域子问题在 MOEA/D 算法中的应用主要体现在两个方面: (1) 重组算子的父体主要来自邻域. 由于父体相距较近, 可以在一定程度上避免第 2.2 节中重组算子所出现的问题. (2) 新个体不仅更新其父亲而且更新其邻域个体, 使得好的新个体尽可能地保留到下一代. 正是因为子问题在重组和更新操作中的相互合作, MOEA/D 算法能够同时优化所有子问题.

MOEA/D 的早期版本采用静态邻域. 可以预见, 算法运行的不同阶段, 不同的 MOP 问题都需要不同的邻域. 如何在算法执行过程中动态地调整邻域以获得更好的算法性能, 也是 MOEA/D 算法设计中值得研究的.

### 3.2 基于混合高斯概率模型的重组算子

分布估计算法(estimation of distribution algorithm, 简称 EDA)<sup>[44]</sup> 是一类新兴的进化算法, 与常用的 EA 算法采用杂交变异等重组算子来产生后代的方式不同, EDA 利用一个概率模型来描述当前群体的分布, 并通过对该模型的采样来产生后代. 经典重组算子利用个体的局部信息来产生后代, 而 EDA 算法利用群体的全局分布信息来产生后代, 因而 EDA 算法更能从全局上指导搜索进程.

高斯模型是研究和应用最广泛的概率模型之一. 一个多元高斯分布的随机变量  $x = (x_1, x_2, \dots, x_n)^T$  可表示为

$$x \sim N(\mu, \Sigma) \tag{6}$$

其中,  $\mu$  为均值向量,  $\Sigma$  为协方差矩阵. 随机变量对应概率密度函数表示为

$$p(x) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\} \quad (7)$$

对于给定的一组数据  $x^1, x^2, \dots, x^K$ , 均值向量和协方差矩阵估计如下:

$$\mu = \frac{1}{K} \sum_{k=1}^K x^k \quad (8)$$

$$\Sigma = \frac{1}{K-1} \sum_{k=1}^K (x^k - \mu)(x^k - \mu)^T \quad (9)$$

多元高斯概率模型采样得到新点  $x=(x_1, x_2, \dots, x_n)^T$  的过程如图 4 所示.

**算法 1.** 多元高斯模型采样  $x=\text{GaussianSample}(\mu, \Sigma)$ .

Step 1. 采用 Cholesky 分解协方差矩阵得到一个下三角矩阵  $A$ , 满足  $\Sigma=AA^T$ .

Step 2. 产生单因素高斯分布向量  $y=(y_1, y_2, \dots, y_n)^T$ , 其中,  $y_j \sim N(0, 1)$ ,  $j=1, 2, \dots, n$  服从单位高斯分布.

Step 3.  $\ominus x=\mu+Ay$ .

Fig.4 Sampling process of multivariate Gaussian probability model

图 4 多元高斯概率模型采样过程

第 2 节的分析已经表明, 单个高斯概率模型不适合 MOP 问题, 因此我们需要考虑构造混合高斯模型. 一种直接的改进方法是: 首先采用聚类分析方法将群体划分成多个类别, 然后对每个类别分别构造高斯模型. 然而文献 [38] 已经验证, 这种方法没有充分考虑 MOP 问题的特性, 因此对一些问题求解效率低下. 文献 [44] 提出了一个混合高斯模型, 即, 对每个个体(子问题)构造一个高斯模型, 其概率密度函数表示为

$$p(x) = \sum_{i=1}^N \frac{1}{K} p^i(x) \quad (10)$$

其中,  $p^i(x)$  表示第  $i$  个子问题对应的概率密度函数;  $\mu^i$  和  $\Sigma^i$  为  $p^i(x)$  对应的均值向量和协方差矩阵, 其数值由第  $i$  个子问题对应的邻域子问题计算得到. 显然, 对每个子问题计算均值和协方差矩阵, 并通过图 4 算法采样新个体计算量较大. 本节将通过邻域子问题协方差矩阵的重用来降低混合高斯模型采样的复杂性.

对于第  $i(i=1, 2, \dots, N)$  个子问题, 构造其概率密度函数:

$$p^i(x) = (2\pi)^{-\frac{n}{2}} |\Sigma^i|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x-\mu^i)^T (\Sigma^i)^{-1}(x-\mu^i)\right\} \quad (11)$$

其中, 均值向量由第  $i$  个子问题当前的最好解表示, 即,  $\mu^i=x^i$ ; 协方差矩阵  $\Sigma^i$  或者由邻域子问题计算得出或者直接使用邻域子问题对应的协方差矩阵.

### 3.3 基于混合高斯概率模型的 MOEA/D 算法框架

图 5 显示了 MOEA/D-MG 算法的基本流程. 和经典 MOEA/D 一样, MOEA/D-MG 算法首先初始化子问题和理想点, 并构造每个子问题的邻域; 在主循环中, 对每个子问题构造新解并更新理想点和群体. MOEA/D-MG 算法与基本 MOEA/D 算法的不同主要体现在以下几点:

#### (1) 子问题采样顺序

为了强调分布稀疏的个体, 算法对群体进行排序. 主要基于下述几个因素: (a) 子问题采样中所用协方差矩阵, 或者采用该子问题邻域计算得到或者直接使用邻域子问题的协方差矩阵, 显然, 直接计算得到矩阵质量要高一些; (b) 为了得到分布均匀的最优解集, 需要强调当前解集中分布稀疏的个体. 若群体分布均匀, 则子问题的解应该靠近其邻域的中心. 本文采用二者的差来度量个体的稀疏程度. 群体排序算法如图 6 所示.

#### (2) 全局搜索与局部搜索

经典 MOEA/D 算法用杂交变异方式产生后代, 因此其全局搜索时父体范围为整个群体 (Step 2.3). 本文采用高斯采样, 若选择整个群体作为父群体, 会导致全局搜索时协方差矩阵一样而失去多样性. 基于此, 本文随机选择  $G$  个子问题作为父群体. 同时, 考虑到采样的多样性, 算法每次最多对  $T/2$  个子问题产生后代 (Step 2.5). 对采样得到的个体, 使用多项式变异算子变异.

理想情况下,新的采样策略每代仅需建立  $2N/T$  个概率模型,而文献[42]中需要建立  $N$  个概率模型.新的策略大大降低了建模的成本.

### (3) 群体更新

经典 MOEA/D 算法随机更新邻域内子问题.本文考虑所有子问题,并更新改进程度最大的子问题.算法流程如图 7 所示.

#### 算法 2. MOEA/D-MG 算法.

输入:  $N$  种群大小(子问题个数);  
 $T$  邻域大小,  $G$  全局搜索域大小,  $n_r$  最大更新子问题个数;  
 $\delta$  邻域搜索概率,  $P_m$  变异概率,  $\eta$  多项式变异算子参数;  
 $\{w^1, w^2, \dots, w^N\}$  权重向量集合.

Step 1. 初始化

- 1.1. 对每个子问题  $i=1, 2, \dots, N$ , 随机产生一个解  $x^i \in \Omega$ .
- 1.2. 对每个子问题  $i=1, 2, \dots, N$ , 构造邻域  $B^i = \{i_1, i_2, \dots, i_k\}$ , 使得  $w^{i_j}$  为与  $w^i$  第  $j$  近的权重向量.
- 1.3. 令理想点  $z^* = (z_1^*, z_2^*, \dots, z_m^*)$ , 其中  $z_j^* = \min_{x \in \{x^1, x^2, \dots, x^N\}} f_j(x)$ ,  $j=1, 2, \dots, m$ .

Step 2. 主循环

- 2.1. 对子问题进行排序  $\Pi = \text{SortPop}(\cdot)$ . //  $\Pi$  表示未产生后代子问题编号
- 2.2. 选择  $\Pi$  中第 1 个子问题, 令其编号为  $i$ .
- 2.3. 若  $\text{rand}(\cdot) < \delta$ , 则令  $A = B^i$ ; //  $\text{rand}(\cdot)$  表示随机产生  $[0.0, 1.0]$  内浮点数  
 否则, 随机选择  $G$  个子问题(包含子问题  $i$ ), 令这些子问题编号组成集合  $A$ .
- 2.4. 利用  $A$  所代表子问题对应决策变量计算协方差矩阵  $\Sigma$ .
- 2.5. 令  $C = A \cap \Pi$ , 若  $|C| > T/2$ , 从  $C$  中删除  $|C| - T/2$  个编号并保留编号  $i$ .
- 2.6. 对每个编号  $j \in C$ 
  - 1) 抽样产生后代  $y = \text{GaussianSample}(x^j, \Sigma)$ ;
  - 2) 变异后代  $x = \text{PolyMutation}(x, \eta, P_m)$ ;
  - 3) 更新理想点  $z_k^* = \min\{z_k^*, f_k(x)\}$ ,  $k=1, 2, \dots, m$ ;
  - 4) 更新群体  $\text{UpdatePop}(x, z^*, n_r)$ .
- 2.7. 令  $\Pi = \Pi \setminus C$ , 若  $\Pi \neq \emptyset$ , 则返回步骤 2.2.
- 2.8. 若停机条件未满足, 返回步骤 2.1.

Step 3. 输出 MOP 问题最优逼近  $\{x^1, x^2, \dots, x^N\}$ .

Fig.5 Procedure of MOEA/D-MG

图 5 MOEA/D-MG 算法流程

#### 算法 3. 群体排序 $\Pi = \text{SortPop}(\cdot)$ .

Step 1. 对每个个体  $x^i$ , 计算其邻域中心点  $\bar{x}^i = \frac{1}{T} \sum_{j=1}^T x^j$ ,  $i=1, 2, \dots, N$ .

Step 2. 对每个个体  $x^i$ , 计算其稀疏度  $s^i = \|x^i - \bar{x}^i\|$ ,  $i=1, 2, \dots, N$ .

Step 3. 对稀疏度  $s^1, s^2, \dots, s^N$  按由大到小的顺序排序, 所得序号为  $i^1, i^2, \dots, i^N$ . //  $i^j$  表示第  $j$  稀疏个体

Step 4. 输出  $\Pi = \{i^1, i^2, \dots, i^N\}$ .

Fig.6 Process of sorting population

图 6 群体排序过程

#### 算法 4. 群体更新 $\text{UpdatePop}(x, z^*, n_r)$ .

Step 1. 对每个个体  $x^i$ , 计算其子问题函数值  $g^i(x^i)$  和  $g^i(x)$ ,  $i=1, 2, \dots, N$ .

Step 2. 对每个个体  $x^i$ , 计算其改进度  $d^i = g^i(x^i) - g^i(x)$ ,  $i=1, 2, \dots, N$ .

Step 3. 对改进度按由大到小的顺序排序, 所得前  $n_r$  项对应子问题序号为  $i^1, i^2, \dots, i^{n_r}$ .

Step 4. 若  $d^j > 0$ , 令  $x^j = x$ ,  $j = i^1, i^2, \dots, i^{n_r}$ .

Fig.7 Process of updating population

图 7 群体更新过程

需要指出的是,新的更新方法采用了贪婪策略——从种群中选择改进程度最大的子问题来更新.相比经典



MOEA/D 的随机更新策略,贪婪策略能更快地推动群体进化.更新过程可以采用不同的贪婪策略,这也是值得将来进一步研究的方面.

#### 4 实验仿真

为检验提出的 MOEA/D-MG 算法的有效性,本文采用文献[43]中提出的 9 个具有复杂 PS 的测试函数:F1~F9.其中,F6 具有 3 个目标函数,其他问题具有 2 个目标函数;F6 和 F9 具有非凸 PF,其他问题具有凸 PF;F7 和 F8 是多峰问题,具有多个局部最优 PF.

由于采用的测试题可得到其理论最优解,本文采用 IGD<sup>[38]</sup>作为评估指标. $IGD(PF,PF^*)$ 计算所得解集 PF 与最优解集  $PF^*$ 之间的距离.

$$\begin{cases} IGD(PF,PF^*) = \frac{1}{|PF|} \sum_{x \in PF} d(x,PF^*) \\ d(x,PF^*) = \min_{y \in PF^*} \|x - y\| \end{cases} \quad (12)$$

其中, $| \cdot |$ 表示集合点的数目, $\| \cdot \|$ 表示向量的模.当 PF 靠近  $PF^*$ 时,IGD 可以同时度量 PF 的分布(diversity)和收敛(convergence)程度.在实验中,对 2 目标问题,我们从  $PF^*$ 中选择 500 个均匀分布的点代表  $PF^*$ ,对 3 目标问题选择 990 个代表点.

本文采用 NSGA-II-DE<sup>[43]</sup>,MOEA/D-DE<sup>[43]</sup>,MOEA/D-MG<sup>[42]</sup>与本文方法进行实验比较.为了加以区分,记文献[42]中方法为 MOEA/D-MG1,本文改进方法为 MOEA/D-MG2.这些方法参数设置如下:DE 算子控制参数  $CR=1.0, F=0.5$ ,多项式变异算子参数  $\eta=20, P_m=1/n$ .MOEA/D-DE 其他参数为:邻域大小  $T=20$ ,邻域搜索概率  $\delta=0.9$ ,子问题更新数目  $n_r=2$ .MOEA/D-MG1 其他参数为:对 2 目标问题,邻域大小  $T=15$ ;对 3 目标问题,邻域大小  $T=30$ ,邻域搜索概率  $\delta=0.8$ ,子问题更新数目  $n_r=2$ .MOEA/D-MG2 其他参数为:邻域大小  $T=10$ ,邻域搜索概率  $\delta=0.8$ ,全局搜索大小  $G=60$ ,子问题更新数目  $n_r=2$ .对于所有算法,2 目标问题种群大小  $N=299$ ,3 目标问题种群大小  $N=595$ ,算法执行 500 代停机.问题自变量维数为  $n=30$ (F1~F5,F9)或者  $n=10$ (F6~F8).对每个测试题,MOEA/D-MG 执行 30 次,MOEA/D-DE 和 NSGA-II-DE 执行 20 次.MOEA/D-MG1 和 MOEA/D-MG2 采用 Matlab 编写,MOEA/D-DE 和 NSGA-II-DE 采用 C++编写.

##### 4.1 MOEA/D-MG2实验结果与分析

表 1 是 MOEA/D-MG2 对 9 个测试函数运算结果的 IGD 指标的统计结果.从表中可见:随着计算代数的增加,IGD 度量值显著减小,特别是 100 代与 200 代之间变化明显.经过 500 代计算,除了 F6 和 F8 以外,其他问题均可得到较小的 IGD 均值.另外,除了 F8 以外,其他测试题均可得到较小的 IGD 均方差.这些统计结果表明:对于大多数测试题,MOEA/D-MG2 能够在较小的计算量内找到稳定而高质量的解集.由于 F8 是多峰问题,MOEA/D-MG2 可能陷入局部最优而导致 IGD 均值和均方差都较大.对于多峰问题 F7,MOEA/D-MG2 能够获取较好的结果,这说明 MOEA/D-MG2 具有解决多峰问题的能力.

**Table 1** Mean (std.) IGD values for MOEA/D-MG2 on generation 100, 200, 300, 400, and 500

**表 1** 100,200,300,400 和 500 代 MOEA/D-MG2 所得结果 IGD 度量均值与均方差

	100	200	300	400	500
F1	0.0024 (0.0001)	0.0017 (0.0000)	0.0015 (0.0000)	0.0014 (0.0000)	0.0014 (0.0000)
F2	0.0551 (0.0166)	0.0256 (0.0117)	0.0106 (0.0060)	0.0048 (0.0017)	0.0033 (0.0003)
F3	0.0252 (0.0045)	0.0083 (0.0022)	0.0043 (0.0011)	0.0031 (0.0006)	0.0025 (0.0002)
F4	0.0377 (0.0072)	0.0161 (0.0033)	0.0096 (0.0014)	0.0059 (0.0016)	0.0040 (0.0013)
F5	0.0234 (0.0029)	0.0119 (0.0014)	0.0090 (0.0011)	0.0076 (0.0009)	0.0070 (0.0011)
F6	0.0358 (0.0047)	0.0300 (0.0009)	0.0288 (0.0005)	0.0282 (0.0005)	0.0280 (0.0004)
F7	0.1819 (0.0472)	0.0039 (0.0017)	0.0017 (0.0001)	0.0015 (0.0000)	0.0014 (0.0000)
F8	0.2563 (0.0529)	0.1547 (0.0731)	0.0448 (0.0485)	0.0180 (0.0297)	0.0110 (0.0184)
F9	0.0532 (0.0096)	0.0334 (0.0102)	0.0193 (0.0087)	0.0108 (0.0065)	0.0067 (0.0033)

测试题具有复杂的 PS.图 8 显示了 MOEA/D-MG2 所获取的最终解集.对 F1~F4,F7 和 F8,获得的最优 PF 能

够很好地逼近真实 PF\*;对 F5 和 F9,获得的最优 PF 仍然存在少量间断部分;对 F6,端点附件的一些点距离端点仍有一些距离.从图 8(b)可知,这些问题的复杂 PS 给算法带来了困难,但是 MOEA/D-MG2 能够有效逼近大部分的 PS.图 8(c)和图 8(d)显示:在多次运行的情况下,MOEA/D-MG2 所获取的解集能够完全覆盖这些问题的 PS 和 PF.

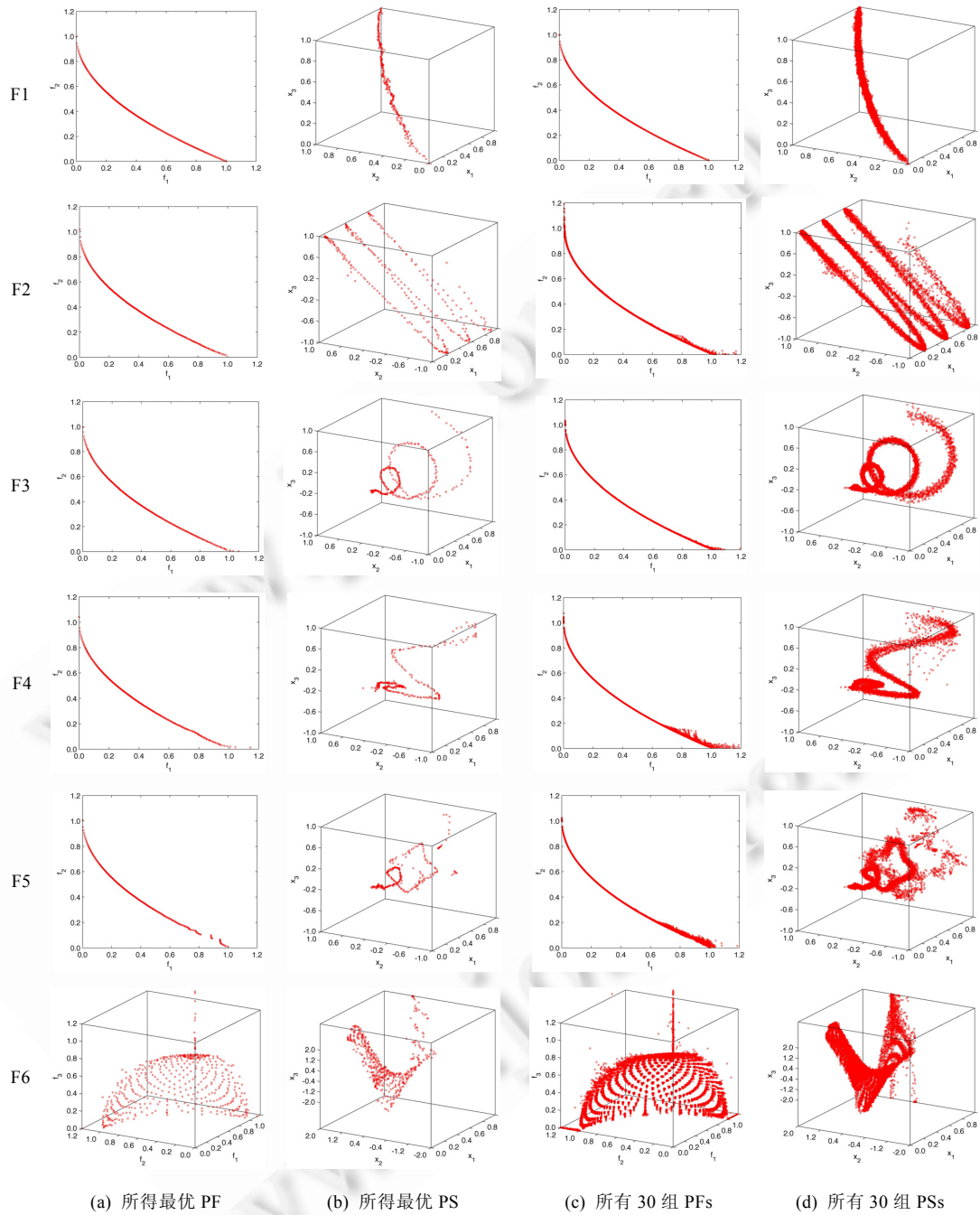


Fig.8 Final obtained approximations by MOEA/D-MG2

图 8 MOEA/D-MG2 算法所得最终解集

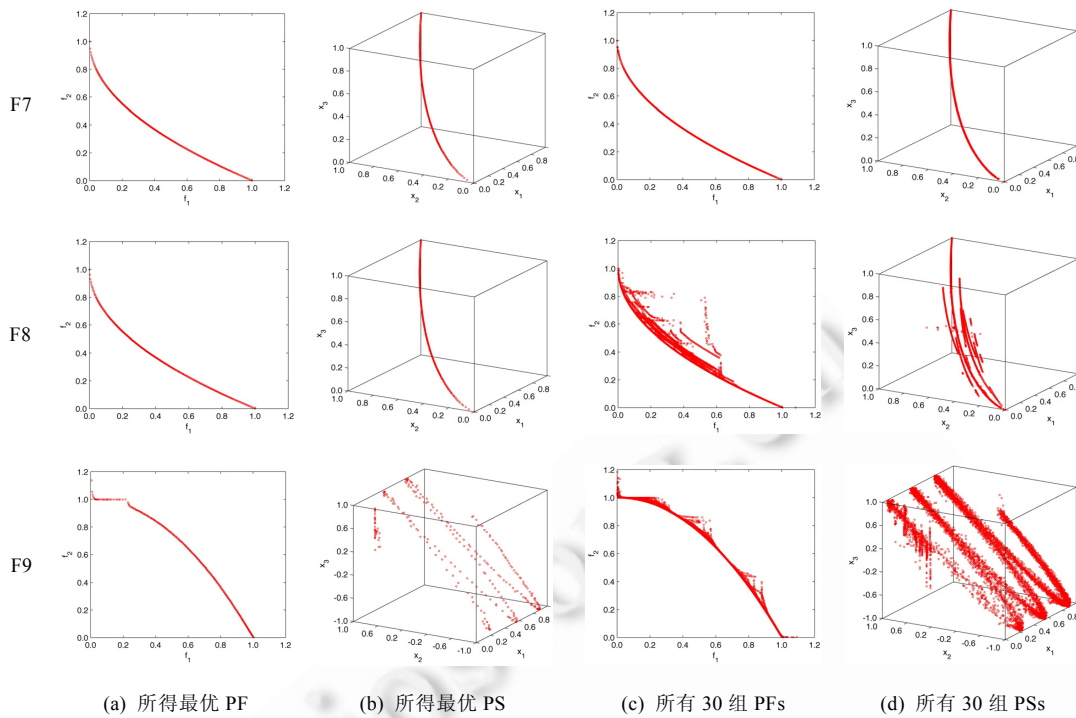


Fig.8 Final obtained approximations by MOEA/D-MG2 (Continued)

图 8 MOEA/D-MG2 算法所得最终解集(续)

4.2 MOEA/D-MG2参数分析

MOEA/D-MG2 算法的主要控制参数包括:邻域搜索大小  $T$ 、全局搜索大小  $G$  和邻域搜索概率  $\delta$ 。本节讨论这些参数对 MOEA/D-MG2 算法性能的影响。实验选取 F2, F3 和 F4 作为测试题, MOEA/D-MG2 的其他参数同第 4.1 节。

首先考虑邻域大小的影响,实验设置  $T=5, 10, 15, 20, 25; G=60, \delta=0.8$ 。

图 9 显示了在不同代数和  $T$  值情况下 MOEA/D-MG2 所得 PS 的 IGD 均值。图 9 表明:对 F2 和 F4,  $T=10$  最优;而对 F3,  $T=5$  最优。不同的  $T$  取值随着运行代数的增加,影响逐渐减小;而且在不同的  $T$  参数下, MOEA/D-MG2 算法均能取得较小的 IGD 统计均值。

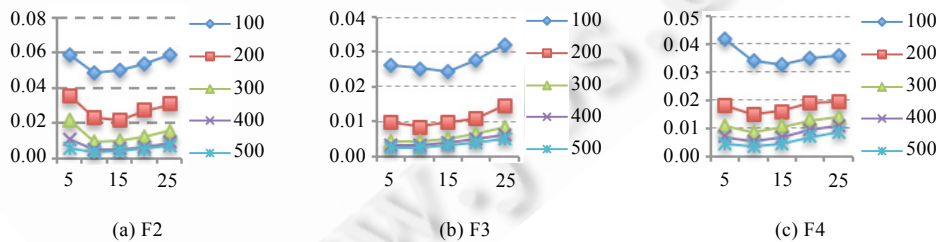
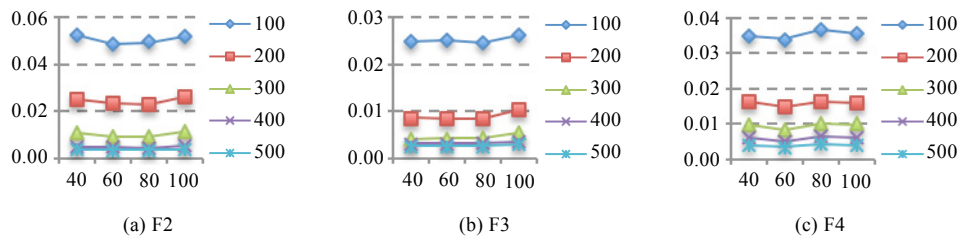


Fig.9 Influence of neighborhood size  $T$  to performance of MOEA/D-MG2

图 9 邻域大小  $T$  对 MOEA/D-MG2 算法性能的影响

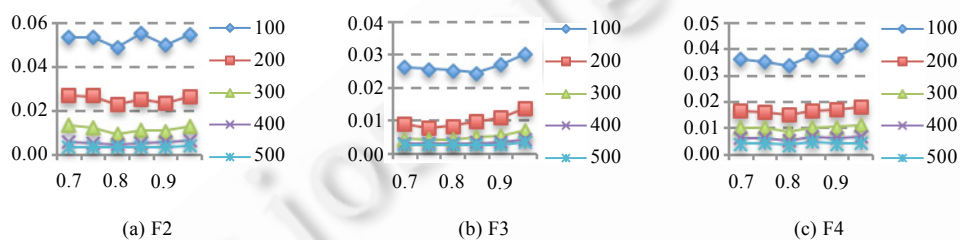
其次考虑全局搜索大小的影响,设置  $G=40, 60, 80, 100; T=10; \delta=0.8$ 。

图 10 显示了在不同代数和不同  $G$  值情况下, MOEA/D-MG2 所得 PS 的 IGD 均值。图 10 表明:不同  $G$  参数对 F2, F3 和 F4 的最后结果几乎没有影响。

Fig.10 Influence of global search size  $G$  to performance of MOEA/D-MG2图 10 全局搜索大小  $G$  对 MOEA/D-MG2 算法性能的影响

最后考虑邻域搜索概率的影响,设置 $\delta=0.7,0.75,0.8,0.85,0.9,0.95$ ;  $T=10$ ;  $G=60$ .

图 11 显示了在不同代数数和不同 $\delta$ 值情况下,MOEA/D-MG2 所得 PS 的 IGD 均值.图 11 表明:在前 200 代内,不同 $\delta$ 值会对结果有微小影响;但 300 代以后,不同 $\delta$ 值的影响变小;400 代以后,MOEA/D-MG 算法采用不同 $\delta$ 值均能取得较小的 IGD 统计均值.

Fig.11 Influence of probability of neighborhood search  $\delta$  to performance of MOEA/D-MG2图 11 邻域搜索概率 $\delta$ 对 MOEA/D-MG2 算法性能的影响

本节实验结果表明:

- (1) 不同的控制参数对算法性能的影响主要体现在算法搜索的早期,特别是 200 代以内;
- (2) 算法性能对控制参数不敏感,3 个主要参数均具有较大的取值范围.

### 4.3 组件对比实验

MOEA/D-MG2 算法主要包含了一个新的基于混合高斯模型的重组算子和一个贪婪更新策略.本节选择 F2,F3 和 F4 作为例子,讨论了这两个重要组件对算法性能的影响.

记文献[42]中的重组算子为 M1,本文所提改进算子为 M2;记文献[43]经典 MOEA/D 算法更新策略为 U1,本文所提策略为 U2.在 MOEA/D 框架下,这些算子可以产生 4 种组合:M1U1,M1U2,M2U1 和 M2U2.实验所用参数同第 4.1 节.

图 12 记录了 4 种策略分别对 F2,F3 和 F4 所得 IGD 均值(纵坐标)与运行代数(横坐标)的关系:

- 首先,固定更新策略来看重组算子的影响.从图 12 可见:若采用 U1 更新策略,M1 算子远优于 M2 算子;然而若采用 U2 更新策略,M1 算子则与 M2 算子在性能上不相上下.
- 然后,固定重组算子来看更新策略的影响.图 12 显示:无论采用 M1 算子还是 M2 算子,U2 更新策略总是远优于 U1 更新策略.

这些实验结果表明:(1) 由于采用了贪婪策略,本文所提更新策略相对经典 MOEA/D 算法的更新策略能够更快推动种群的进化;(2) 在采用贪婪更新策略的情况下,改进的基于混合高斯模型的重组算子与文献[42]所提出的算子性能相近.但是需要注意的是,本文中重组算子的建模次数要小于文献[42]所提出的算子.

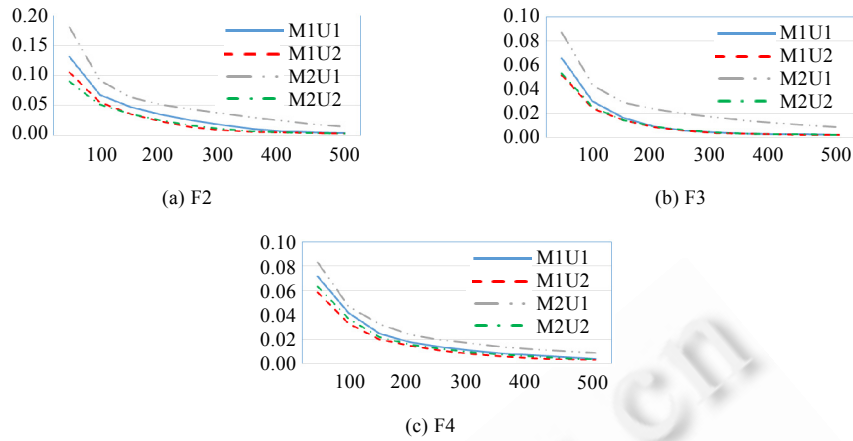


Fig.12 Influence of improved reproduction operator and the update strategy to algorithm performance

图 12 重组算子和更新策略对算法性能的影响

#### 4.4 算法对比实验

本节比较基于混合高斯模型的 MOEA/D 算法与文献[43]中所提出的 NSGA-II-DE 和 MOEA/D-DE 算法.

表 2 是这 4 种算法所得最后结果的统计 IGD 均值和均方差(NSGA-II-DE 和 MOEA/D-DE 测试结果源于文献[43],MOEA/D-MG 算法源代码可从 amzhou@cs.ecnu.edu.cn 获取).

**Table 2** Mean (std.) IGD values obtained by the four algorithms

**表 2** 4 种对比算法所得结果的统计 IGD 均值和均方差

	MOEA/D-DE <sup>[43]</sup>	NSGA-II-DE <sup>[43]</sup>	MOEA/D-MG1 <sup>[42]</sup>	MOEA/D-MG2
F1	0.0015(0.0000)	0.0044(0.0000)	<b>0.0009</b> (0.0000)	0.0014(0.0000)
F2	<b>0.0028</b> (0.0004)	0.0349(0.0066)	0.0040(0.0016)	0.0033(0.0003)
F3	0.0068(0.0099)	0.0296(0.0030)	<b>0.0024</b> (0.0005)	0.0025(0.0002)
F4	<b>0.0040</b> (0.0014)	0.0288(0.0021)	0.0046(0.0024)	<b>0.0040</b> (0.0013)
F5	0.0127(0.0069)	0.0288(0.0031)	<b>0.0055</b> (0.0007)	0.0070(0.0011)
F6	0.0289(0.0014)	0.0680(0.0072)	0.1192(0.0452)	<b>0.0280</b> (0.0004)
F7	0.0049(0.0063)	0.1171(0.0716)	0.0678(0.0485)	<b>0.0014</b> (0.0000)
F8	0.0998(0.0429)	0.1981(0.0494)	0.0137(0.0069)	<b>0.0110</b> (0.0184)
F9	<b>0.0035</b> (0.0008)	0.0395(0.0061)	0.0059(0.0020)	0.0067(0.0033)

统计结果表明:

- MOEA/D-DE 在 F2,F4 和 F9 上取得最好结果;
- MOEA/D-MG1 在 F1,F3 和 F5 上取得最好结果;
- MOEA/D-MG2 在 F4,F6~F8 上取得最好结果;
- 与 MOEA/D-DE,MOEA/D-MG1 和 MOEA/D-MG2 相比,NSGA-II-DE 不能取得最好结果.

需要指出的是:

- 对 F2 和 F9,MOEA/D-MG2 所得结果的 IGD 均值是 MOEA/D-DE 所得均值的 1.18 和 1.91 倍;
- 对 F3,F5,F7 和 F9,MOEA/D-DE 所得结果的 IGD 均值是 MOEA/D-MG2 所得均值的 2.72,1.81,3.50 和 9.07 倍;
- 对于 F1,F4 和 F6,两种算法所得结果类似.

对比 MOEA/D-MG1 和 MOEA/D-MG2 可见:

- 对 F6 和 F7,MOEA/D-MG2 所得结果远好于 MOEA/D-MG1 所得结果;
- 对于其他问题,两种算法所得结果比较类似.

## 5 结束语

基于分解技术的多目标算法是求解多目标优化问题的一类新型算法框架.目前大部分 MOEA/D 算法采用传统的重组算子.针对这种情况,本文证明或分析了几个典型重组算子在多目标优化下的不足.基于此,提出了基于混合高斯模型的 MOEA/D 算法 MOEA/D-MG. MOEA/D-MG 采用混合高斯模型来获取种群的分布,并对分布采样获得新个体;同时,采用一个贪婪策略来更新群体.实验分析结果表明:

- 混合高斯模型能够更有效地提取种群的结构信息,相比 DE 算子,它能够更有效地逼近复杂的 PS;
- 贪婪更新策略能够更有效地推动种群的进化.

需要指出的是,混合高斯模型的逼近能力也是有限的,对于过于复杂的问题如 F5,该模型只能部分逼近其 PS.结合 MOEA/D 算法的特性与多目标优化问题的特性,是进一步改进 MOEA/D-MG 算法的一个可行而有效的方法.

### References:

- [1] Miettinen K. Nonlinear Multiobjective Optimization. Boston: Kluwer Academic Publishers, 1999.
- [2] Schaffer JD. Multiple objective optimization with vector valued genetic algorithms. In: Proc. of the 1st Int'l Conf. on Genetic Algorithms. Hillsdale: L. Erlbaum Associates Inc., 1985. 93–100.
- [3] Deb K. Multi-Objective Optimization Using Evolutionary Algorithms. New York: John Wiley & Sons LTD, 2001.
- [4] Coello Coello CA. An updated survey of GA-based multiobjective optimization techniques. ACM Computing Surveys, 2000,32(2): 109–143. [doi: 10.1145/358923.358929]
- [5] Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q. Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation, 2011,1(1):32–49. [doi: 10.1016/j.swevo.2011.03.001]
- [6] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation, 2002,6(2):182–197. [doi: 10.1109/4235.996017]
- [7] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Proc. of the Evolutionary Methods for Design, Optimisation and Control. Athens: International Center for Numerical Methods in Engineering, 2002. 95–100.
- [8] Knowles JD, Corne DW. Approximating the nondominated front using the Pareto archived evolution strategy. Evolutionary Computation, 2000,8(2):149–172. [doi: 10.1162/106365600568167]
- [9] Zitzler E, Kunzli S. Indicator-Based selection in multiobjective search. In: Proc. of the Parallel Problem Solving from Nature (PPSN VIII). LNCS 3242, Berlin, Heidelberg: Springer-Verlag, 2004. 832–842. [doi: 10.1007/978-3-540-30217-9\_84]
- [10] Bader J, Zitzler E. HypE: An algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation, 2011, 19(1):45–76. [doi: 10.1162/EVCO\_a\_00009]
- [11] Zhang QF, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. on Evolutionary Computation, 2007,11(6):712–731. [doi: 10.1109/TEVC.2007.892759]
- [12] Zheng JH. Multiobjective Optimization Algorithms and Their Applications. Beijing: Science Press, 2007 (in Chinese).
- [13] Lei DM, Yan XP. Multiobjective intelligent optimization algorithms and their applications. Beijing: Science Press, 2009 (in Chinese).
- [14] Jiao LC, Shang RH, Ma WP. Multiobjective Immune Algorithms, Theory and Applications. Beijing: Science Press, 2010 (in Chinese).
- [15] Leung Y, Wang Y. Multi-Objective programming using uniform design and genetic algorithm. IEEE Trans. on Systems, Man, and Cybernetics, Part C, 2000,30(3):293–304. [doi: 10.1109/5326.885111]
- [16] Zeng SY, Kang LS, Ding LX. An orthogonal multi-objective evolutionary algorithm for multi-objective optimization problems with constraints. Evolutionary Computation, 2004,12(1):77–98. [doi: 10.1162/evco.2004.12.1.77]
- [17] Wang YN, Wu LH, Yuan XF. Multi-Objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. Soft Computing, 2010,14(3):193–209. [doi: 10.1007/s00500-008-0394-9]

- [18] Gong WY, Cai ZH. An improved multiobjective differential evolution based on Pareto-adaptive epsilon-dominance and orthogonal design. *European Journal of Operational Research*, 2009,198(2):576–601. [doi: 10.1016/j.ejor.2008.09.022]
- [19] Gong DW, Zhang Y, Zhang JH. Multi-Objective particle swarm optimization based on minimal particle angle. In: *Proc. of the Int'l Conf. on Intelligent Computing*. Berlin, Heidelberg: Springer-Verlag, 2005. 571–580. [doi : 10.1007/11538059\_60]
- [20] Zhan ZH, Zhang J. A parallel particle swarm optimization approach for multiobjective optimization problems. In: *Proc. of the Genetic and Evolutionary Computation Conf.* New York: ACM, 2010. 81–82. [doi : 10.1145/1830483.1830497]
- [21] Gong MG, Jiao LC, Du HF, Bo LF. Multiobjective immune algorithm with nondominated neighbor-based selection. *Evolutionary Computation*, 2008,16(2):225–255. [doi: 10.1162/evco.2008.16.2.225]
- [22] Chen JY, Lin QZ, Ji Z. A hybrid immune multiobjective optimization algorithm. *European Journal of Operational Research*, 2010, 204(2):294–302. [doi: 10.1016/j.ejor.2009.10.010]
- [23] Li BB, Wang L. A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 2007,37(3):576–591. [doi: 10.1109/TSMCB.2006.887946]
- [24] Wang Y, Zeng JC, Tan Y. An artificial physics optimization algorithm for multi-objective problems based on virtual force sorting. In: *Proc. of the 1st Int'l Conf. on Swarm, Evolutionary, and Memetic Computing*. Berlin, Heidelberg: Springer-Verlag, 2010. 615–622. [doi: 10.1007/978-3-642-17563-3\_72]
- [25] Zhang DM, Gong XS, Dai GM. Multi-Objective evolutionary algorithm for principal curve model based on multifractal. *Journal of Computer Research and Development*, 2011,48(9):1729–1739 (in Chinese with English abstract).
- [26] Yang DD, Ma JJ, Jiao LC, Gong MG, Si XY. Improved  $\epsilon$  dominance by ISomap. *Ruan Jian Xue Bao/Journal of Software*, 2011, 22 (10):2291–2304 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3933.htm> [doi: 10.3724/SP.J.1001.2011.03933]
- [27] Wang Y, Cai ZX, Guo GQ, Zhou YR. Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 2007,37(3):560–575. [doi: 10.1109/TSMCB.2006.886164]
- [28] Zou XF, Chen Y, Liu MZ, Kang LH. A new evolutionary algorithm for solving many-objective optimization problems. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 2008,38(5):1402–1412. [doi: 10.1109/TSMCB.2008.926329]
- [29] Wei W, Ouyang DT, Lü S, Feng YX. Multiobjective path planning under dynamic uncertain environment. *Chinese Journal of Computers*, 2011,34(5):836–846 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2011.00836]
- [30] Huo WG, Shao XL. A fuzzy associative classification method based on multi-objective evolutionary algorithm. *Journal of Computer Research and Development*, 2011,48(4):567–575 (in Chinese with English abstract).
- [31] Wang Z, Tang K, Yao X. Multi-Objective approaches to optimal testing resource allocation in modular software systems. *IEEE Trans. on Reliability*, 2010,59(3):563–575. [doi: 10.1109/TR.2010.2057310]
- [32] Wang Y, Li B, Chen YB. Digital IIR filter design using multi-objective optimization evolutionary algorithm. *Applied Soft Computing*, 2011,11(2):1851–1857. [doi: 10.1016/j.asoc.2010.05.034]
- [33] Li Q, Hao QF, Xiao LM, Li ZJ. Adaptive management and multi-objective optimization for virtual machine placement in cloud computing. *Chinese Journal of Computers*, 2011,34(12):2253–2264 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2011.02253]
- [34] Lu HC, Li YJ, Zhu YH. Simultaneous deployment and scheduling for real-time event capture in sensor networks. *Ruan Jian Xue Bao/Journal of Software*, 2011,22:182–190 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/11020.htm>
- [35] Shi C, Yan ZY, Shi ZZ, Zhang L. A fast multi-objective evolutionary algorithm based on a tree structure. *Applied Soft Computing*, 2010,10(2):468–480. [doi: 10.1016/j.asoc.2009.08.018]
- [36] Li MQ, Zheng JH. An indicator for assessing the spread of solutions in multi-objective evolutionary algorithms. *Chinese Journal of Computers*, 2011,34(4):647–664 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2011.00647]
- [37] Iorio AW, Li X. Rotated problems and rotationally invariant crossover in evolutionary multi-objective optimization. *Int'l Journal of Computational Intelligence and Applications*, 2008,7(2):149–186. [doi: 10.1142/S1469026808002223]
- [38] Zhang QF, Zhou AM, Jin YC. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans. on Evolutionary Computation*, 2008,12(1):41–63. [doi: 10.1109/TEVC.2007.894202]

- [39] Weinert K, Zabel A, Kersting P, Michelitsch T, Wagner T. On the use of problem-specific candidate generators for the hybrid optimization of multi-objective production engineering problems. *Evolutionary Computation*, 2009,17(4):527-544. [doi: 10.1162/evco.2009.17.4.17405]
- [40] Hillermeier C. *Nonlinear Multiobjective Optimization—A Generalized Homotopy Approach*. Basel: Birkhauser Verlag AG, 2001.
- [41] Zhou AM, Zhang QF, Jin YC. Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE Trans. on Evolutionary Computation*, 2009,13(5):1167-1189. [doi: 10.1109/TEVC.2009.2021467]
- [42] Zhou AM, Zhang QF, Zhang GX. A multiobjective evolutionary algorithm based on decomposition and probability model. In: *Proc. of the IEEE Congress of Evolutionary Computation (CEC 2012)*. Brisbane: IEEE Press, 2012. 1-8. [doi: 10.1109/CEC.2012.6252954]
- [43] Li H, Zhang QF. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. on Evolutionary Computation*, 2009,13(2):284-302. [doi: 10.1109/TEVC.2008.925798]
- [44] Larrañaga P, Lozano J. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston: Kluwer Academic Publishers, 2002.

#### 附中文参考文献:

- [12] 郑金华. 多目标进化算法及其应用. 北京: 科学出版社, 2007.
- [13] 雷德明, 严新平. 多目标智能优化算法及其应用. 北京: 科学出版社, 2009.
- [14] 焦李成, 尚荣华, 马文萍. 多目标优化免疫算法, 理论和应用. 北京: 科学出版社, 2010.
- [25] 张冬梅, 龚小胜, 戴光明. 基于多重分形主曲线模型多目标演化算法研究. *计算机研究与发展*, 2011, 48(9): 1729-1739.
- [26] 杨咚咚, 马晶晶, 焦李成, 公茂果, 司晓云. 一种改进 $\epsilon$ 支配的等度规映射方法. *软件学报*, 2011, 22(10): 2291-2304. <http://www.jos.org.cn/1000-9825/3933.htm> [doi: 10.3724/SP.J.1001.2011.03933]
- [29] 魏唯, 欧阳丹彤, 吕帅, 冯宇轩. 动态不确定环境下多目标路径规划方法. *计算机学报*, 2011, 34(5): 836-846. [doi: 10.3724/SP.J.1016.2011.00836]
- [30] 霍纬纲, 邵秀丽. 一种基于多目标进化算法的模糊关联分类方法. *计算机研究与发展*, 2011, 48(4): 567-575.
- [33] 李强, 郝沁汾, 肖利民, 李舟军. 云计算中虚拟机放置的自适应管理与多目标优化. *计算机学报*, 2011, 34(12): 2253-2264. [doi: 10.3724/SP.J.1016.2011.02253]
- [34] 陆汉城, 李燕君, 朱艺华. 传感器网络面向实时事件捕获的联合部署与调度策略. *软件学报*, 2011, 22: 182-190. <http://www.jos.org.cn/1000-9825/11020.htm>
- [36] 李密青, 郑金华. 一种多目标进化算法解集分布广度评价方法. *计算机学报*, 2011, 34(4): 647-664. [doi: 10.3724/SP.J.1016.2011.00647]



周爱民(1978—),男,湖北当阳人,博士,副教授,CCF 会员,主要研究领域为进化计算,机器学习,图像处理及应用.  
E-mail: amzhou@cs.ecu.edu.cn



张桂成(1970—),男,博士,教授,博士生导师,主要研究领域为图像分析与处理,计算智能,模式识别.  
E-mail: gxzhang@cs.ecnu.edu.cn



张青富(1965—),男,博士,教授,博士生导师,主要研究领域为多目标优化,分布式估计算法,启发式方法理论及应用.  
E-mail: qzhang@essex.ac.uk