

# 一种基于混合个体选择机制的多目标进化算法\*

陈晓纪<sup>1</sup>, 石川<sup>1</sup>, 周爱民<sup>2</sup>, 吴斌<sup>1</sup>

<sup>1</sup>(北京邮电大学 计算机学院,北京 100876)

<sup>2</sup>(华东师范大学 计算机科学与软件工程学院,上海 200062)

通讯作者: 石川, E-mail: shichuan@bupt.edu.cn

**摘要:** 在多目标进化算法中,如何从后代候选集中选择最优解显著地影响优化过程.当前,最优解的选择方式主要是基于实际目标值或者代理模型估计目标值.然而,这些选择方式往往是非常耗时或者存在精度差等问题,特别是对于一些实际的复杂优化问题.最近,一些研究人员开始利用有监督分类辅助后代选择,但是这些工作难以准备准确的正例和负例样本或者存在耗时的参数调整等问题.为了解决这些不足,本文提出了一种新颖的融合分类与代理的混合个体选择机制用于从后代候选集中选择最优解.在每一代优化中,首先,利用分类器选择优良解;然后,设计了一个轻量级的代理模型用于估计优良解的目标值;最后,利用这些目标值对优良解进行排序并选择最优解作为后代解.基于典型的多目标进化算法 MOEA/D,本文利用混合个体选择机制设计了新的算法框架 MOEA/D-CS,与当前流行的基于分解多目标进化算法比较,实验结果表明本文提出的算法取得了最好的性能.

**关键词:** 多目标优化;进化算法;后代选择;相似性;MOEA/D

**中图法分类号:** TP311

中文引用格式: 陈晓纪,石川,周爱民,吴斌.一种基于混合个体选择机制的多目标进化算法.软件学报.

英文引用格式: Chen XJ, Shi C, Zhou AM, Wu B. A multiobjective evolutionary algorithm based on hybrid individual selection mechanism. Ruan Jian Xue Bao/Journal of Software(in Chinese).

## A Multiobjective Evolutionary Algorithm Based on Hybrid Individual Selection Mechanism

CHEN Xiao-Ji<sup>1</sup>, SHI Chuan<sup>1</sup>, ZHOU Ai-Min<sup>2</sup>, WU Bin<sup>1</sup>

<sup>1</sup>(College of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China)

<sup>2</sup>(College of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China)

**Abstract:** In multiobjective evolutionary algorithms, how to select the optimal solutions from the offspring candidate set significantly affects the optimization process. At present, the selection of the optimal solutions is largely based on the real objective values or surrogate model to estimate objective values. However, these selections are usually very time-consuming or exist poor accuracy problems, especially for some real complex optimization problems. Recently, some researchers began to employ supervised classification to assist offspring selection, but these works are difficult to prepare the exact positive and negative samples or exist time-consuming parameter adjustment problems. In order to solve these disadvantages, we propose a novel hybrid individual selection mechanism through integrating classification and surrogate to select the optimal solutions from the offspring candidate set. Concretely, in each generation, the selection mechanism employs a classifier to select good solutions firstly; then, it designs a cheap surrogate model to estimate objective values of each good solution; finally, it sorts these good solutions according to objective values and selects the optimal solution as the offspring solution. Based on the typical multiobjective evolutionary algorithm MOEA/D, we employ the hybrid individual selection mechanism to

\* 基金项目: 国家重点基础研究发展计划(2017YFB0803304); 国家自然科学基金(61772082,61375058)

Foundation item: National Key Research and Development Program of China (2017YFB0803304); National Natural Science Foundation of China (61772082, 61375058).

收稿时间: 0000-00-00; 修改时间: 0000-00-00; 采用时间: 0000-00-00; jos 在线出版时间: 0000-00-00

CNKI 在线出版时间: 0000-00-00

design a new algorithm framework MOEA/D-CS. Compared with the current popular multiobjective evolutionary algorithms based on decomposition, experimental results show that the proposed algorithm obtains the best performance.

**Key words:** multiobjective optimization; evolutionary algorithm; offspring selection; similarity; MOEA/D

多目标优化问题(Multiobjective Optimization Problem 简称 MOP)是指那些同时优化多个目标的问题.一般来说这些单个目标是相互冲突的,而且总体目标没有单个最优解<sup>[1]</sup>.求解多目标优化问题常常是困难的,它不像单目标优化问题(Single Objective Optimization Problem,简称 SOP)那样只有一个最优解;多目标优化问题得到的是一组最优解.因为进化算法(EAs)是基于种群的优化算法,能够多次迭代优化逼近 MOP 的 Pareto 前沿,所以进化算法(EAs)广泛应用于解决 MOP 问题<sup>[2]</sup>.这些 EAs 被称为多目标进化算法(MOEAs).近年来,大量 MOEAs 被提出,这些算法可以分为三类<sup>[1]</sup>:基于 Pareto 支配的算法<sup>[3-6]</sup>,基于指标的算法<sup>[7-9]</sup>和基于分解的算法<sup>[10,11]</sup>.MOEA/D<sup>[10]</sup>是最近最具代表性的基于分解的算法.MOEA/D 的基本思想是将 MOP 分解为一组标量目标子问题.相邻子问题相互协作产生新的后代解集,而新产生的解不仅会更新相应子问题,也会更新相邻子问题.通过这种方式,所有子问题同时被优化,最终所有子问题的解集构成了原始 MOP 的 Pareto 解集(PS)与 Pareto 前沿(PF).

后代选择是 MOEA 的核心问题之一<sup>[12]</sup>.后代选择显著地影响迭代优化的收敛速度和解集的多样性.目前主要有三种后代选择方式:(1)非支配排序是一种非常流行的方法.对于多目标优化问题,通常存在一个解集,这些解之间对于所有目标函数而言是无法比较优劣的,其特点是无法在改进任何目标函数的同时不削弱至少一个目标函数.它根据解之间的支配关系对所有解进行排序,称之为非支配排序.典型的非支配排序算法包括 NSGA-II<sup>[4]</sup>和 SPEA2<sup>[5]</sup>.非支配排序方法往往不适用于代价昂贵的多目标优化问题.(2)代理也是选择后代最优解的常用方法.代理模型是指计算量小,但是其计算结果和高精度模型的计算分析结果相近的分析模型.在优化设计时,可以用代理模型替代高精度分析模型.典型的利用代理模型解决多目标优化问题的算法包括 MOEA/D-EGO<sup>[13]</sup>和 MOEA/D-RBF<sup>[14]</sup>.通常,利用某些方法获取的样本点集构造代理模型往往需要拟合复杂的数学模型,这个过程就是非常耗时的.同时,为了保证代理模型具有较高的精度,代理模型需要优化许多参数<sup>[15]</sup>.(3)最近,分类也成为选择后代最优解的常用方法.分类就是按照某种标准给对象设置标签,再根据标签来区分归类.分类算法利用有监督机器学习方法构造分类器用于后代选择.典型的分类算法包括 MOEA/D-CPS<sup>[16]</sup>,CPS-MOEA<sup>[12]</sup>和 MOEA/D-SVM<sup>[17]</sup>.分类算法将后代选择看成是一个分类问题.然而,分类后优良解的数量往往不止一个,利用随机选择方式很难从优良解集中获取最优解.此外,构建复杂分类器往往存在优化参数耗时等问题.

考虑到现有后代选择方式的缺陷,本文提出了一种融合分类与代理的混合个体选择机制;进一步基于经典的 MOEA/D 框架,利用混合个体选择机制设计了新的多目标进化算法,称为 MOEA/D-CS(MOEA/D based on classification and surrogate).由于分类能够以较小的时间代价提升算法性能以及代理能够较为准确的估计目标值,有许多工作利用分类或者代理实现后代选择,但是单独利用分类或者代理存在上述缺陷,本文融合二者的优点实现了一种混合个体选择机制.在混合个体选择机制实现过程中,首先利用分类器过滤候选解并且只有优良解被保留下来;然后利用轻量级代理模型估计优良解的目标值;最后利用目标值排序策略获得最优解.在 19 个具有 2 目标或 3 目标的多目标优化测试函数上,本文比较了 MOEA/D-CS 算法与其他多个当前流行的基于分解多目标进化算法.实验结果表明,在没有显著增加时间代价的情况下,MOEA/D-CS 算法在绝大部分测试函数上取得了最好的性能.

本文的其余部分安排如下.第 1 节介绍了相关工作.第 2 节介绍了算法实现的细节.第 3 节进行了大规模实验,并对实验结果和敏感性参数进行了深入分析.第 4 节总结了全文,并对未来的研究工作进行了阐述.

## 1 相关工作

在过去的二十几年时间里,各种多目标进化算法相继被提出.总体来说,多目标进化算法主要包含三类.第

一类算法的特点是基于个体支配关系与精英保留策略.David 等人<sup>[18]</sup>设计的多目标进化算法 VEGA 被认为是利用进化算法解决多目标优化问题的先驱工作.Srinivas 和 Deb<sup>[19]</sup>提出的 NSGA 奠定了多目标进化算法的基本框架.NSGA-II<sup>[4]</sup>是 NSGA 的改进版本.NSGA-II 使用精英保留策略,保留最好的父代个体和子代个体.Zitzler 和 Thiele<sup>[20]</sup>提出的 SPEA 利用外部种群实现精英保留策略.SPEA 的改进版本 SPEA2<sup>[5]</sup>也相继被提出.SPEA2<sup>[5]</sup>使用最近邻密度估计来实现更高搜索效率,新的归档截断方法保证边界解被保留下来.第二类算法的特点是使用指示器指导搜索过程.Zitzler 和 Künzli<sup>[7]</sup>首先提出了一个基于指标的进化算法(IBEa).在 IBEa 中,不需要任何额外的多样性保留机制.与其他 MOEAs 相比,IBEa 只比较成对的个体而不是整个近似集.Basseur 和 Zitzler<sup>[8]</sup>提出了一个以指标为基础的处理不确定性问题的模型,其中每个解在目标空间中被分配一个概率.Bader 和 Zitzler<sup>[9]</sup>提出了一种基于快速超体积的 MOEA 算法,该算法用于处理多个目标的优化问题.为了减少超体积计算的开销,提出了一种基于蒙特卡罗模拟的快速估计超体积近似集的方法.第三类算法使用分解和降维<sup>[21]</sup>来解决多目标优化问题.这个类型的典型算法是 MOEA/D<sup>[10,11]</sup>.这种方法将 MOP 分解为一组子问题并同时优化这些子问题.后代繁殖和环境选择都是基于子问题的.伴随着多目标进化算法的发展历程,多目标进化算法在理论研究方面也取得了一定的进展.在一个基于种群的多目标问题上,Laumanns<sup>[22]</sup>等人首次分析了算法的运行时间.在某些多目标优化问题上,Laumanns<sup>[23]</sup>等人证明了基于特定种群的多目标优化算法具有较低的运行时间和算法运行时间与 Pareto 前沿是密切相关的,而且算法总的运行时间是有界的.Qian<sup>[24]</sup>等人从理论上研究了重组算子对多目标进化算法的影响.在一系列多目标优化基准问题上,结合重组算子的多目标进化算法不仅可以减少算法的运行时间,而且能够使最优解集更靠近 Pareto 前沿.

利用多目标进化算法解决多目标优化问题,如何选择后代个体是一个非常重要的问题.在多目标进化算法中,主要有 3 种后代选择方式:(1)传统的算法使用非支配排序<sup>[4,5]</sup>选择后代解集.NSGA-II<sup>[4]</sup>算法首先利用非支配排序找出种群中的非支配解;然后,一部分非支配解通过交叉与变异生成子代解,并且利用实际目标值评价每一个子代解;最后,整合父代种群与子代种群,利用非支配排序保存下一次迭代优化需要的种群.SPEA2<sup>[5]</sup>算法除了采用传统的交叉与变异产生子代解,还利用非支配排序与密度估计方法计算个体的适应值,算法精度要高于 NSGA-II<sup>[4]</sup>.(2)近年来,许多文献利用代理选择后代最优解.文献<sup>[25,26]</sup>利用回归或排序技术用来构建代理模型,这样可以减少目标值评价的次数.文献<sup>[27]</sup>使用代理模型进行局部搜索和预选择.MOEA/D-EGO<sup>[13]</sup>利用高斯随机过程解决代价昂贵的多目标优化问题.MOEA/D-EGO<sup>[13]</sup>利用高斯过程为每个目标分别建模,然后获得每个分解子问题的模型.但是 MOEA/D-EGO<sup>[13]</sup>利用种群的聚类信息建立代理模型,没有考虑种群的整体信息对代理模型精度的影响,这可能导致算法的精度较低,而且需要调整较多参数.同时,利用局部信息更新代理模型可能导致错误的搜索.文献<sup>[14]</sup>利用集成的径向基函数网络构建代理模型 MOEA/D-RBF<sup>[14]</sup>,它保持了固定大小的训练集,但是只针对较小的种群规模和迭代次数.(3)最近,随着机器学习算法的广泛应用,有监督分类算法已经成为选择后代最优解的常用方法.Yu<sup>[28]</sup>等人提出了 RACOS 算法,该算法利用分类模型对优良解与不良解进行分类.RACOS 算法在处理高维优化问题时具有明显的优势.Zhang 等人<sup>[16]</sup>提出的 MOEA/D-CPS 算法利用非支配排序和 KNN<sup>[29]</sup>构造分类器模型,在具有复杂 Pareto 前沿的多目标优化问题上取得了较好的性能.Zhang 等人<sup>[12]</sup>利用非支配排序和 CART<sup>[30]</sup>构造分类器模型,与 NSGA-II<sup>[4]</sup>算法结合,提出了 CPS-MOEA 算法,在大量多目标优化测试函数上取得了较好的性能.Lin 等人<sup>[17]</sup>利用 SVM<sup>[31]</sup>训练分类器,提出了 MOEA/D-SVM 算法,提升了 MOEA/D 的性能.然而,上述后代选择方式普遍存在着时间代价较高,难以准备准确的正例和负例样本或者很难保证代理模型的精度.因此,往往不能适用于代价昂贵的多目标优化问题.

## 2 MOEA/D-CS 算法

后代选择是多目标进化算法中一个非常重要的问题.不同于以往后代选择方式,本文融合分类与代理的混合个体选择机制获取最优解.基于混合个体选择机制设计了新颖的多目标进化算法.本节首先简单介绍了基本的 MOEA/D 框架;然后详细论述了融合分类与代理的混合个体选择机制的具体实现过程,包括如何利用非支配排序策略<sup>[4]</sup>与  $k$  近邻(KNN)<sup>[29]</sup>算法构建有监督分类器模型以及相似性度量方法构建轻量级代理模型;最后介绍

了本文设计的融合分类与代理个体选择机制的 MOEA/D-CS 算法框架及其时间复杂度。

## 2.1 基于分解的多目标进化算法框架

MOEA/D 将 MOP 问题分解为一组子问题  $\{g_1, g_2, \dots, g_N\}$ , 并且同时优化这些子问题. 每个子问题的最优解有可能成为 MOP 问题的 Pareto 最优解<sup>[32]</sup>. 本文使用切比雪夫(Tchebycheff)方法作为 MOEA/D 的分解方法. 每个子问题  $i$  被定义为:

$$\min g^i(x) = g(x | \lambda^i, z^*) = \max\{\lambda_j^i | f_j(x) - z_j^* | \} \quad (1)$$

其中,  $\lambda^i = (\lambda_1^i, \dots, \lambda_m^i)^T$  是权重向量,  $z^* = (z_1^*, \dots, z_m^*)^T$  是参考点,  $z_j^*$  是  $f_j$  在搜索空间的最小值. 在大多数情况下, 两个具有相似权重向量的子问题具有相似的最优解. 基于权重向量之间的距离, MOEA/D 定义了子问题之间的邻域关系. 传统的 MOEA/D 算法使用交叉和变异算子生成后代解. 然而, 随着差分演化(DE)<sup>[33]</sup>的兴起, MOEA/D 算法开始与 DE 结合生成后代解, 称之为 MOEA/D-DE<sup>[11]</sup>. 本文利用 DE<sup>[33]</sup>算子和多项式变异<sup>[34]</sup>为每个子问题生成 3 个后代候选解<sup>[16]</sup>.

## 2.2 融合分类与代理的混合个体选择机制

基于分解的多目标进化算法中, 如果每个子问题产生的后代候选解越多, 那么获得最优解的几率就越大. 但是, 计算多个后代候选解花费的时间代价也越高. 因此, 传统方式利用实际目标值评价不适用于这类问题. 目前, 基于分类与代理获取最优解的相关算法越来越多. 分类算法普遍存在着难以构造准确的分类器模型; 利用局部信息构建的代理模型精度难以保证; 而利用全局信息构建的代理模型虽然精度较高, 但是需要花费的时间代价太大. 鉴于单独利用分类与代理获取最优解存在的诸多问题, 本文提出了融合分类与代理的混合个体选择机制实现从后代候选集中选择最优解, 在没有显著增加时间代价的前提下, 提升了获取最优解的概率.

下面详细介绍融合分类与代理的混合个体选择机制实现过程.

首先, 使用  $k$  近邻(KNN)<sup>[29]</sup>构建分类器<sup>[16]</sup>. 设置  $\langle x, y \rangle$  为一组训练数据, 其中  $x$  是数据点的个体向量,  $y \in L$  是数据点对应的标签,  $L$  是一组标签集合. 个体向量与标签之间的对应关系可以表示为  $y = \text{Class}(x)$ . 训练分类器的

目的是找到一个关系  $y = \hat{\text{Class}}(x)$ , 基于训练数据集可以近似地表示实际的对应关系  $y = \text{Class}(x)$ .

本文引入两个外部种群  $P_+$  和  $P_-$ ,  $P_+$  包含了迄今为止发现的标签为+1的优良解集合,  $P_-$  包含了标签为-1的不良解集合. 为了获得  $P_+$  和  $P_-$ , 本文使用非支配排序策略<sup>[4]</sup>. 令  $q = \text{NDS}(p, n)$  表示该方法从  $p$  中选择最好的  $n$  个解存储在  $q$  中. 这里所指的“最好的  $n$  个解”包含 3 种情况: (1) 非支配解的数量等于  $n$ . 最好的  $n$  个解就是全部非支配解的集合. (2) 非支配解的数量大于  $n$ . 算法利用拥挤距离<sup>[4]</sup>过滤掉多余的非支配解. (3) 非支配解的数量小于  $n$ . 此时根据序关系和拥挤距离<sup>[4]</sup>从支配解中找出一部分最靠近 Pareto 前沿的解, 再与非支配解集组成最好的  $n$  个解.

令  $P$  为初始种群,  $N$  为种群  $P$  的大小.  $P_+$  和  $P_-$  被初始化为:

$$P_+ = \text{NDS}\left(P, \left\lfloor \frac{N}{2} \right\rfloor\right) \quad (2)$$

$$P_- = P - P_+ \quad (3)$$

设  $x$  是一个个体向量,  $L = \{+1, -1\}$ , 其中+1 是优良解的标签, -1 是不良解的标签. 本文使用  $k$  近邻(KNN)<sup>[29]</sup>作为分类算法. 其中,  $k$  近邻(KNN)<sup>[29]</sup>的训练数据包括当前整个种群  $P$ .

$$y = \text{KNN}(x) = \text{sign}\left(\sum_{t \in N(x), |N(x)|=K} y_t\right) \quad (4)$$

$N(x)$  表示训练集中与  $x$  最近邻的  $K$  个邻居个体,  $K$  设置为奇数.  $\text{sign}(x)$  的执行过程为: 从  $N(x)$  中依次选择个体向量  $t$ , 将个体向量  $t$  对应的标签  $y_t$  累加求和, 累加求和的结果只能是正数或负数. 此时, 如果累加求和的结果为正数,  $\text{sign}(x)$  返回+1, 则  $x$  标记为优良解; 否则  $\text{sign}(x)$  返回-1,  $x$  标记为不良解.

令  $Q$  表示每一代中新产生的一组最优解集合,  $Q_+$  和  $Q_-$  分别表示包含非支配和被支配关系解集. 然后,  $P_+$  和  $P_-$  按照如下方式进行更新:

$$P_+ = NDS\left(P_+ \cup Q_+, \left\lfloor \frac{N}{2} \right\rfloor\right) \quad (5)$$

$$P_- = NDS\left(P_- \cup Q_-, \left\lfloor \frac{N}{2} \right\rfloor\right) \quad (6)$$

其次,利用相似性度量方法构建轻量级代理模型.传统的代理模型时间代价较高,往往不适用于代价昂贵的多目标优化问题.本文构建了一个轻量级代理模型,能够以较小的时间代价保证代理模型具有较高的精度.文献<sup>[35][36]</sup>指出,连续 MOP 问题在决策空间  $PS$  可以定义为分段连续的流形结构,那么对于 MOP 来说,目标空间  $PF$  也应该是分段连续的.因此,连续多目标优化具有如图 1 所示的性质.图 1(a)展示了  $PS$  与  $PF$  之间的对应关系.图 1(b)中,  $x^1, x^2, x^3 \in D$  (决策向量空间),  $z^1, z^2, z^3 \in F(D)$  (目标向量空间),相比  $x^3, x^1$  与  $x^2$  更接近,因此  $z^1$  与  $z^2$  也就更接近.同理,对于  $D$  中的任意两个相近的解,相应的  $F(D)$  中目标值也应该越相近.

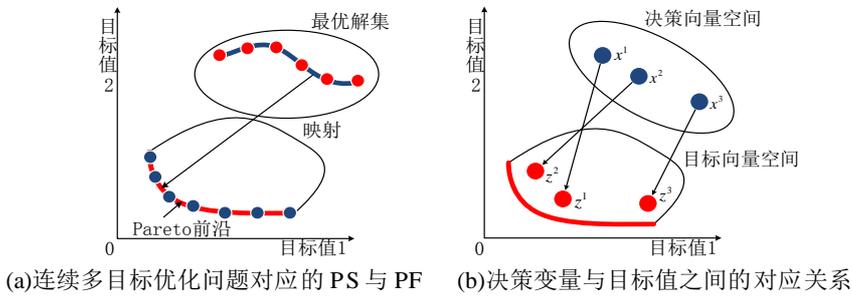


Fig. 1 The characteristics of continuous multiobjective optimization problem

图 1 连续多目标优化问题的性质

由于每个子问题的后代候选解都是由其邻域个体通过  $DE$ <sup>[33]</sup>算子和多项式变异<sup>[34]</sup>生成的,并且执行分类操作后只有优良解被保留下来.因此,本文利用相似性度量方法计算每个优良解与对应邻域个体之间的相似性,利用最近邻域个体的目标值作为每个优良解的估计目标值.目前,有许多方法用来度量相似性.本文利用余弦相似性度量每个优良解与邻域个体之间的相似程度.余弦相似性主要利用向量空间中两个向量的余弦夹角进行计算.余弦相似性的计算公式如下:

$$sim(x_1, x_2) = \frac{x_1^T x_2}{|x_1| |x_2|} \quad (7)$$

$x_1$  和  $x_2$  表示向量空间中具有相同维度的两个向量,  $sim(x_1, x_2)$  越接近 1,  $x_1$  和  $x_2$  就越相似.

### 2.3 算法框架与分析

算法 1 MOEA/D-CS 算法框架.

输入: 种群大小  $N$ , 邻域大小  $T$ , 子问题更新数目  $C$ ;

邻域搜索概率  $\delta$ , 变异概率  $p_m$ , 多项式变异算子  $\eta$ ;

权重向量集合  $\{w^1, w^2, \dots, w^N\}$ .

输出: MOP 问题的最优逼近  $P$ .

- 1 initialize  $x^j, B^i, z^*$ .
- 2 **while** not terminate **do**
- 3     calculate  $P_+, P_-$  by Eq. (3) and Eq. (4), and train  $classifier = \hat{Class}(x)$ .
- 4     **for** each  $i \in \text{perm}(1 \text{ to } N)$  **do**
- 5         generate offspring solution set  $Y = \{y^1, \dots, y^M\}$ .

```

6         set  $V = \{y \in Y \mid \hat{Class}(y) = +1\}$ , and reset  $V = Y$  if  $V == \text{null}$ .
7         if count( $V$ ) $\geq 2$ 
8             for each  $y \in V$  do
9                 calculate the most similar  $s \in \text{neighbor}$  to  $y$  by Eq. (7).
10                set  $y.objs = s.objs$ .//objs:目标值
11            end for
12            calculate each  $g^i(y)$  by Eq. (1), and  $Q=[Q, y]$  by  $\min g^i(y)$ .
13        end if
14        update  $z^*$  and  $P$ .
15    end for
16    update  $P_+$ ,  $P_-$  by Eq. (5) and Eq. (6).
17 end while
18 return  $P$ .

```

Fig. 2 MOEA/D-CS algorithm framework

图2 MOEA/D-CS 算法框架

MOEA/D-CS 算法框架如图 2 所示.MOEA/D-CS 算法流程主要包括 4 部分:

1)初始化工作:生成每个子问题对应的个体  $x^i$ ,邻域信息  $B^i$ ,初始化理想点  $z^*$ (第 1 行).

2)利用非支配排序<sup>[4]</sup>获取正例  $P_+$  和负例  $P_-$  样本,训练有监督分类器模型(第 3 行).

3)循环选择每一个子问题:首先利用 *classifier* 对候选解进行分类(第 6 行);然后,如果条件满足则利用轻量级代理模型估计每个优良解的目标值并保存最优解(第 7-13 行);最后,更新理想点  $z^*$ 和种群  $P$ (第 14 行).

4)更新正例  $P_+$  和负例  $P_-$  样本(第 16 行).

针对 MOEA/D-CS 算法的每一个子问题,其时间复杂度主要包含 3 部分:1)利用非支配排序<sup>[4]</sup>将初始种群划分为正例  $P_+$  和负例  $P_-$  样本.非支配排序<sup>[4]</sup>算法的时间复杂度是  $O(mN^2)$ ,其中  $m$  是目标个数, $N$  是种群大小.2)利用有监督分类器执行分类操作.对于每一个后代候选解,它需要与分类器中的每个样本计算欧式距离.因此比较次数是  $N$ ,所以时间复杂度是  $O(MN)$ ,其中  $M$  是后代候选解的数量.3)利用余弦相似性计算每个优良解与邻域个体的相似程度.邻域个体的数量上限是  $N$ , $M$  是优良解数量的上限,每个优良解最多需要计算  $N$  次,所以时间复杂度是  $O(MN)$ .本文主要针对 2 目标或 3 目标,每个子问题的后代候选解数量  $M=3$ ,所以有  $m \leq M$ .因为 MOEA/D-CS 算法需要同时优化  $N$  个子问题,所以 MOEA/D-CS 算法的时间复杂度为  $O(mN^3)+O(MN^2)+O(MN^2)$ ,近似为  $O(MN^3)$ .本文以函数评价次数作为终止条件,在这里假设函数评价次数为  $C$ ,那么终止条件折合成迭代次数就是  $C/N$ .所以 MOEA/D-CS 算法总的时间复杂度近似为  $O(CMN^2)$ .

### 3 实验

#### 3.1 实验设置

为了验证 MOEA/D-CS 算法的有效性,将其与当前多个有代表性的 MOEA 算法进行对比试验.

1)MOEA/D-CPS<sup>[16]</sup>.MOEA/D-CPS<sup>[16]</sup>首先利用 DE<sup>[33]</sup>算子和多项式变异<sup>[34]</sup>为每个子问题生成 3 个后代候选解,然后利用有监督分类器执行分类操作.

2)MOEA/D-DE<sup>[11]</sup>.MOEA/D-DE<sup>[11]</sup>利用 DE<sup>[33]</sup>算子和多项式变异<sup>[34]</sup>为每个子问题生成 1 个后代解.

3)MOEA/D-MO<sup>[37]</sup>.MOEA/D-MO<sup>[37]</sup>首先利用 DE<sup>[33]</sup>算子和多项式变异<sup>[34]</sup>为每个子问题生成 3 个后代候选解,然后直接利用实际目标值函数评价每个后代候选解.

4)MOEA/D-SIM.为了对比仅使用代理模型的后代个体选择机制,本文设计了 MOEA/D-SIM 算法.MOEA/D-SIM 首先利用 DE<sup>[33]</sup>算子和多项式变异<sup>[34]</sup>为每个子问题生成 3 个后代候选解;然后利用余弦相似性度量每个后代候选解与邻域个体的相似性;最后将最近邻域个体的目标值作为当前后代候选解的估计目标值.

本文采用文献<sup>[38]</sup>中使用的 9 个多目标优化测试函数 F1-F9 与文献<sup>[39]</sup>中使用的 10 个无约束多目标优化测试函数 ZZF1-ZZF10.

本文使用  $IGD^{[36]}$  指标评价算法的性能.令  $P^*$  是一组沿着 PF 均匀分布的点, $P$  在目标空间中近似于 PF. $IGD^{[36]}$ 定义为:

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

其中  $d(v, P)$  是  $v$  和  $P$  中任何一点的最短欧氏距离. $|P^*|$  表示  $P^*$  的基数.如果  $P^*$  足够大,它就可以更好地表示 PF. $IGD(P^*, P)$  同时衡量  $P$  的收敛性和多样性. $IGD(P^*, P)$  指标越小, $P$  就越接近于 PF\*.在本文的实验中,对于 2 目标函数,从 PF\* 中选择 500 个均匀分布的点代表 PF\*.对于 3 目标函数,选择 990 个均匀分布的点代表 PF\*.

相关参数设置如下:

1)对于所有测试函数,决策变量的维度  $n = 30$ .针对 2 目标函数,种群规模设置为  $N = 300$ ,针对 3 目标函数,种群规模设置为  $N = 595$ .

2)在每个测试函数上,所有算法独立运行 30 次,停止条件是函数评价次数(FES).针对 2 目标函数,算法执行 150000 次评价后停止;针对 3 目标函数,算法执行 297500 次评价后停止.

3)邻域大小  $T = 20$ ,邻域搜索概率  $\delta = 0.9$ ,子问题更新数目  $c = 2$ ,变异概率  $p_m = \frac{1}{n}$ ,多项式变异算子  $\eta = 20$ .

4)在 MOEA/D-CPS,MOEA/D-MO,MOEA/D-SIM 和 MOEA/D-CS 算法中,后代候选解个数  $M=3$ ,DE 算子中参数  $F = \{0.5, 0.7\}$ ;在 MOEA/D-DE 算法中,DE 算子中参数  $F = 0.5$ .

5)在 MOEA/D-CPS,MOEA/D-SIM 和 MOEA/D-CS 算法中,2 目标函数的邻域大小  $T = 15$ ,3 目标函数的邻域大小  $T = 30$ .在 MOEA/D-CPS 和 MOEA/D-CS 算法中,最近邻的数量  $K=3$ .

### 3.2 有效性实验

本文分别在 F1-F9 和 ZZF1-ZZF10 测试函数集上做了对比试验.所有算法均独立运行 30 次.在每个测试函数上,分别计算每个算法的  $IGD$  均值和标准差. $IGD$  均值越小,说明算法的收敛性和多样性越好;标准差越小,说明算法稳定性越好.同时,本文还分别统计了每个算法在 F1-F9 和 ZZF1-ZZF10 两个测试函数集上的平均运行时间.所有统计结果分别展示在表 1 与表 2 中.

**Table 1** The statistical results of mean  $IGD$  and std obtained by five algorithms on F1-F9 over 30 runs. ‘+’ and ‘-’ denote each algorithm performs better(equal) or worse than other algorithms

**表 1** 5 种算法在 F1-F9 测试函数上独立运行 30 次获得的  $IGD$  均值与标准差统计结果.‘+’与‘-’分别表示每一个算法优(等于)与劣于其他算法

实例	MOEA/D-CPS		MOEA/D-DE		MOEA/D-MO		MOEA/D-SIM		MOEA/D-CS	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差
F1	<b>0.0013</b>	0.0000	<b>0.0013</b>	0.0000	0.0014	0.0000	0.0014	0.0000	<b>0.0013</b>	0.0000
F2	0.0038	0.0011	<b>0.0032</b>	0.0004	0.0044	0.0007	0.0037	0.0006	0.0038	0.0009
F3	0.0034	0.0012	0.0134	0.0268	0.0037	0.0007	0.0028	0.0006	<b>0.0027</b>	0.0006
F4	0.0038	0.0011	0.0114	0.0138	0.0048	0.0012	0.0046	0.0017	<b>0.0030</b>	0.0004
F5	0.0068	0.0014	0.0110	0.0027	0.0078	0.0013	0.0065	0.0016	<b>0.0064</b>	0.0011
F6	0.0598	0.0088	0.0594	0.0098	<b>0.0569</b>	0.0067	0.0636	0.0085	0.0598	0.0084
F7	0.1577	0.0961	0.2314	0.1347	<b>0.1137</b>	0.1067	0.1960	0.1042	0.2167	0.1080
F8	0.0236	0.0211	0.0273	0.0237	<b>0.0130</b>	0.0096	0.0274	0.0197	0.0245	0.0210
F9	0.0046	0.0009	0.0052	0.0031	0.0063	0.0013	0.0051	0.0012	<b>0.0043</b>	0.0007
+/-	1/8		2/7		3/6		0/9		5/4	
平均运行时间	7.4001E+03		7.1433 E+03		1.9470E+04		2.6216E+04		2.1790E+04	

通过深入分析表 1 和表 2,可以发现如下的现象:

1)MOEA/D-CS 算法在大多数情况下表现出最好的性能.在 F1-F9 测试函数集中,MOEA/D-CS 算法有 5 个表现最好;在 ZZF1-ZZF10 测试函数集中,MOEA/D-CS 算法有 6 个表现最好.然后,在 F1-F9 与 ZZF1-ZZF10 测试

函数集中,本文进一步比较 MOEA/D-CS 与其他算法之间的性能差异.(1)MOEA/D-CS 与 MOEA/D-CPS 算法相比,MOEA/D-CS 算法有 15 个表现最好,说明增加轻量级代理模型提高了获取最优解的概率.MOEA/D-CS 利用分类获得的优良解更靠近  $P_+$ ,而  $P_+$ 更靠近 PS,所以优良解也靠近 PS.由于 PS 是近似于分段连续的,目标空间也是分段连续的,所以可以通过估计优良解的目标值近似代替优良解的实际目标值.因为优良解是由邻域个体生成的,此时利用相似性度量方法作为轻量级代理模型估计每个优良解最相似邻域个体,然后可以用最近邻域个体的目标值作为优良解的估计目标值.对于所有优良解,只需要利用其近似目标值计算优良解的排序关系,就可以从所有优良解中找出一个最优解.虽然 MOEA/D-CPS 算法设计的分类器是有效的,然而 MOEA/D-CPS 算法采用随机选择优良解作为最优解,这种方式不能保证以较高的概率获得最优解.因此,MOEA/D-CS 要比 MOEA/D-CPS 更有效.(2)MOEA/D-CS 与 MOEA/D-DE 算法相比,MOEA/D-CS 算法有 13 个表现最好.MOEA/D-CS 基于 MOEA/D 框架,说明融合分类与轻量级代理模型的 MOEA/D-CS 算法,确实提升了 MOEA/D 的性能.(3)MOEA/D-CS 与 MOEA/D-MO 算法相比,MOEA/D-CS 算法有 14 个表现最好.因为 MOEA/D-MO 算法对所有后代候选解均采用实际目标值评价,那么在相同评价次数作为终止条件下,MOEA/D-MO 算法的运行代数要远远少于 MOEA/D-CS 算法,这说明 MOEA/D-CS 要比 MOEA/D-MO 算法更有优势,表现也会更好.(4)MOEA/D-CS 与 MOEA/D-SIM 算法相比,MOEA/D-CS 算法有 16 个表现最好.MOEA/D-CS 算法首先采用有监督分类能够很好的区分优良解与不良解,然后利用轻量级代理模型可以从优良解中找出最优解.而 MOEA/D-SIM 直接估计每个候选解与最近邻域个体的相似性,如果候选解不是优良解也很可能通过估计目标值被误认为是最优解,这样 MOEA/D-SIM 算法的误差就会非常大.所以 MOEA/D-CS 算法的效果也会优于 MOEA/D-SIM 算法.从表 1 与表 2 还可以看出,本文提出的 MOEA/D-CS 算法在大多数测试函数上标准差都是最小的,说明 MOEA/D-CS 算法是非常稳定的.

**Table 2** The statistical results of mean *IGD* and std obtained by five algorithms on ZZF1-ZZF10 over 30 runs. '+' and '-' denote each algorithm performs better(equal) or worse than other algorithms

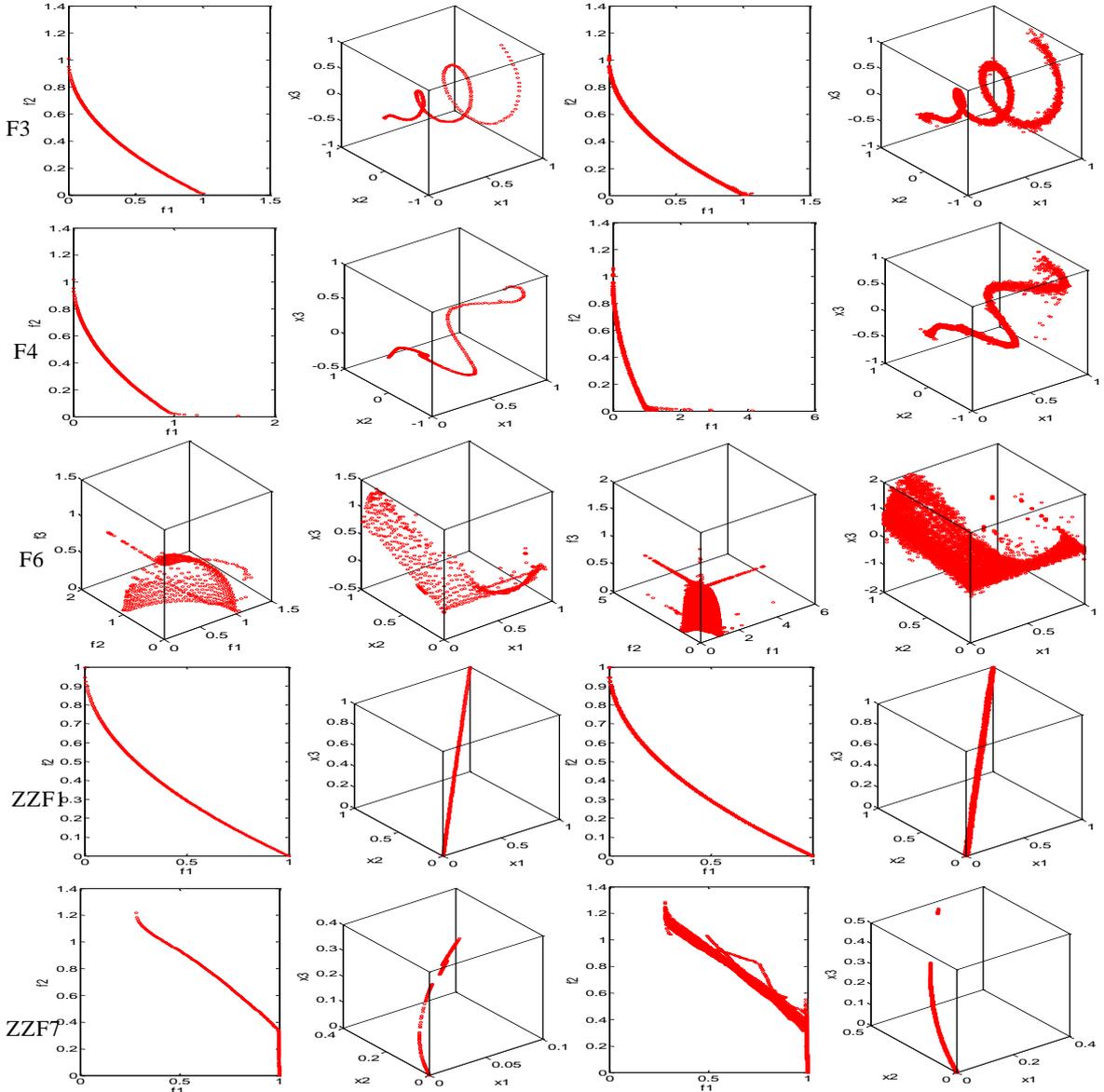
**表 2** 5 种算法在 ZZF1-ZZF10 测试函数上独立运行 30 次获得的 *IGD* 均值与标准差统计结果.'+'与 '-' 分别表示每一个算法优(等于)与劣于其他算法

实例	MOEA/D-CPS		MOEA/D-DE		MOEA/D-MO		MOEA/D-SIM		MOEA/D-CS	
	均值	标准差								
ZZF1	<b>0.0013</b>	0.0000								
ZZF2	<b>0.0013</b>	0.0000								
ZZF3	0.1088	0.0203	0.1419	0.0095	0.1247	0.0350	0.1034	0.0158	<b>0.1026</b>	0.0156
ZZF4	0.0220	0.0002	<b>0.0212</b>	0.0002	0.0217	0.0002	0.0221	0.0003	0.0221	0.0002
ZZF5	0.0014	0.0000	<b>0.0013</b>	0.0000	0.0014	0.0000	0.0014	0.0000	0.0014	0.0000
ZZF6	0.0928	0.2230	<b>0.0013</b>	0.0000	0.0015	0.0000	0.0014	0.0000	0.0319	0.3161
ZZF7	0.1409	0.0156	0.1597	0.0289	0.1535	0.0214	0.1251	0.0229	<b>0.1224</b>	0.0092
ZZF8	<b>0.0256</b>	0.0013	0.0425	0.0812	0.0260	0.0007	0.0262	0.0004	<b>0.0256</b>	0.0009
ZZF9	<b>0.0057</b>	0.0034	0.0076	0.0064	0.0079	0.0071	0.0092	0.0059	0.0079	0.0059
ZZF10	6.4666	2.6957	4.4941	1.9224	8.4900	4.6036	4.0950	3.8456	<b>3.8176</b>	1.8750
+/-	4/6		5/5		2/8		2/8		6/4	
平均运行时间	1.0190E+04		7.3299E+03		1.7735E+04		3.0278 E+04		2.7341E+04	

2)MOEA/D-CS 算法花费的时间代价低于 MOEA/D-SIM,但是要高于其他算法.由于 MOEA/D-CS 算法的性能最好,在权衡精度与时间代价的情况下,MOEA/D-CS 算法花费的时间代价是可以接受的.(1)MOEA/D-DE 算法利用 DE 算子生成 1 个后代解,然后利用实际目标值评价与更新操作,因此 MOEA/D-DE 算法在两个测试函数集上花费的时间代价最小.(2)MOEA/D-CPS 算法首先利用 KNN<sup>[29]</sup>构建分类器并执行分类操作;然后随机选择一个优良解作为后代解;最后利用实际目标值评价后代解并执行更新操作.因此 MOEA/D-CPS 算法花费的时间代价要高于 MOEA/D-DE 算法.(3)MOEA/D-MO 算法利用 DE 算子生成 3 个后代候选解,每个候选解均使用实际目标值进行评价.因此 MOEA/D-MO 算法花费的时间代价要高于 MOEA/D-DE 与 MOEA/D-CPS 算法.(4)MOEA/D-SIM 直接利用余弦相似性估计每个候选解的目标值,这需要花费相当大的时间代价.因此,MOEA/D-SIM 花费的时间代价最高.(5)MOEA/D-CS 算法融合分类与轻量级代理模型执行后代个体选

择.MOEA/D-CS 算法花费的时间代价高于前 3 种算法.其主要原因有两个:第一,本文利用余弦相似性构建轻量级代理模型,因为每个优良解都需要与邻域个体计算相似度,所以会耗费一定的运行时间;第二,本文使用的测试函数都相对简单,每个子问题产生的后代解数量较少,评价个体目标值花费的运行时间要少于计算相似度花费的运行时间.MOEA/D-CS 算法花费的时间代价低于第 4 种算法.其主要原因是 MOEA/D-CS 算法采用先分类后估计优良解目标值的方式,相比于 MOEA/D-SIM 算法采用估计全部候选解目标值的方式时间代价要小.

为了深入分析不同类型多目标测试函数对算法性能的影响,本文从 F1-F9 与 ZZF1-ZZF10 测试函数集中选择了一部分有代表性的测试函数 F3,F4,F6,ZZF1,ZZF7 和 ZZF9.在这些测试函数中,包含 MOEA/D-CS 算法性能明显优于其他算法的测试函数(F3,F4,ZZF7),也包含其性能逊于其他算法(F6,ZZF9)或者与其他算法取得相同性能(ZZF1)的测试函数.其中,F3,F4,ZZF1,ZZF7 和 ZZF9 是 2 目标函数,它们具有相似的 PF,而它们的 PS 形状是不同的非线性曲线.F3 和 F4 具有比其他 2 目标函数更复杂的 PS 形状.F6 是 3 目标函数,PF 和 PS 都是最复杂的.图 3 展示了 MOEA/D-CS 算法在这些有代表性的测试函数上获得的最终解集.



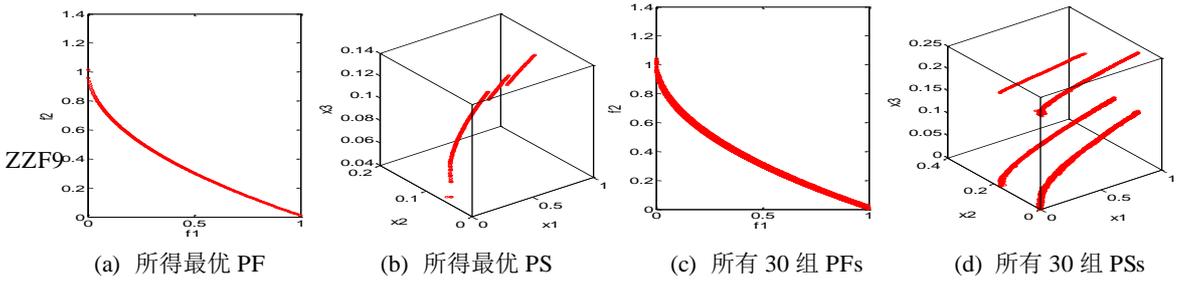


Fig.3 Final obtained approximations by MOEA/D-CS  
图 3 MOEA/D-CS 算法获得的最终解集

通过深入分析图 3,可以发现如下现象:

1)MOEA/D-CS 算法非常适合处理 PS 相对平滑的多目标测试函数.针对 F3,F4 和 ZZF7 等 3 个测试函数,PS 与 PF 都比较平滑,在这几个测试函数上获得的 IGD 均值为最小值.MOEA/D-CS 算法首先利用分类获取优良解;然后,如果条件满足则利用余弦相似性计算每个优良解与邻域个体的相似程度,此时获得的最近邻域个体是非常接近的,那么对应 PF 空间的目标值也应该是非常接近的.算法利用近似目标值估计优良解的优劣,很容易从优良解中找出最优解.因为 MOEA/D-CS 算法融合了分类与代理的混合个体选择机制,所以它提高了获得最优解的概率.

2)MOEA/D-CS 算法处理 PS 非常平滑的多目标测试函数,与其它流行的基于分解多目标进化算法取得了相同的性能.针对 ZZF1 测试函数,其对应的 PS 是非常平滑的,此时不管利用哪一种多目标进化算法,只要评价次数足够多,最终的 IGD 均值都是相等的.

3)MOEA/D-CS 算法处理 PS 具有复杂曲线形状的多目标测试函数,表现不如其他算法.针对 F6 与 ZZF9 测试函数,越是具有复杂曲线形状的多目标测试函数,利用余弦相似性估计每个优良解的目标值所产生的偏差就越大,所以 MOEA/D-CS 算法不适合处理 PS 具有复杂曲线形状的多目标测试函数.这也是该算法需要进一步改进的地方.

### 3.3 收敛性试验

为了验证 MOEA/D-CS 算法的收敛性,本文从 F1-F9 与 ZZF1-ZZ10 测试函数集中选取了 F3,F4,ZZF3 与 ZZF7 等 4 个多目标测试函数,这 5 种算法利用函数评价次数(FES)作为终止条件获得的平均 IGD(log)指标值如图 4 所示.

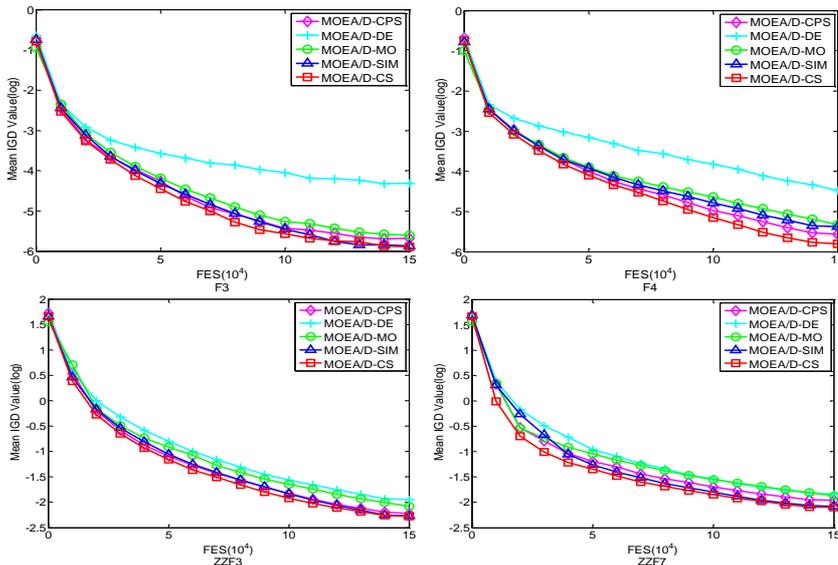


Fig. 4 The mean  $IGD(\log)$  values versus numbers of FES obtained by five algorithms over 30 runs

图 4 5 种算法利用评价次数作为终止条件独立运行 30 次获得的平均  $IGD(\log)$  值

从图 4 中可以发现 MOEA/D-CS 算法在不同运行阶段收敛性都是最好的.在迭代优化过程中,对应相同的函数评价次数(FES),MOEA/D-CS 算法的平均  $IGD(\log)$  指标均为最小值,并且 MOEA/D-CS 算法对应的平均  $IGD(\log)$  曲线始终呈现递减的趋势,这说明 MOEA/D-CS 算法取得了比其他算法更好的收敛性.

### 3.4 敏感性参数分析

本文深入分析敏感性参数对 MOEA/D-CS 算法性能的影响.在 MOEA/D-CS 算法中,有两个敏感性参数:DE 算子中的参数  $F$  和种群规模  $N$ .

1)本文分别设置了参数  $F$  为单值与组合的形式.如图 5(a)和(b)所示,在随机选取的测试函数 F4 和 ZZ7 上, $F$  以组合形式选取  $\{0.5, 0.7\}$  均取得了最好的效果.这样选取参数  $F$  可以保证 MOEA/D-CS 算法具有更好的性能.因此,对于 MOEA/D-CS 算法来说,选择合适的参数  $F$  值是非常重要的.

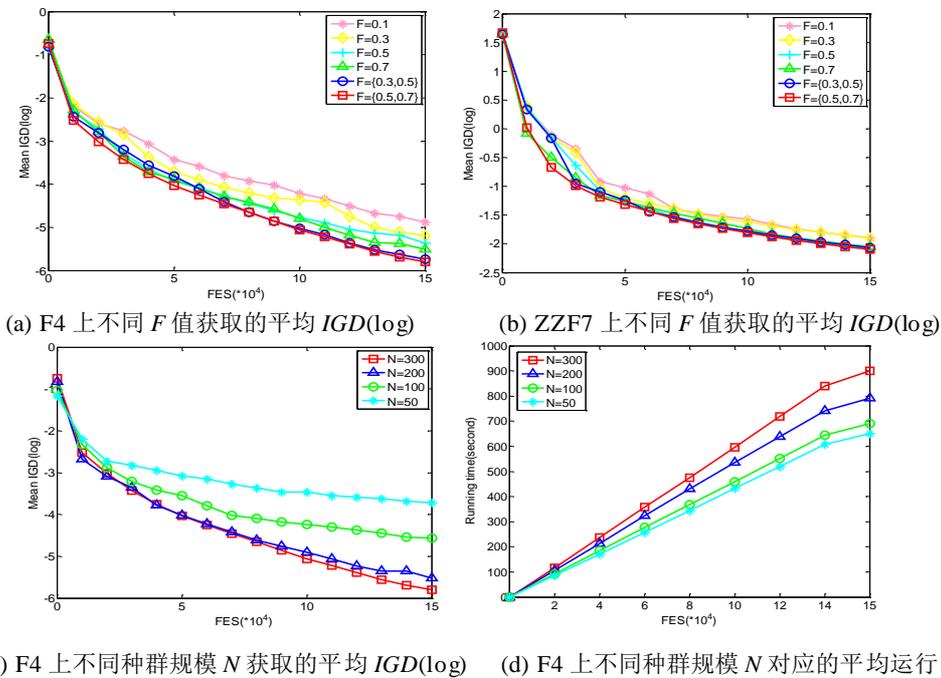


Fig. 5 The performances of MOEA/D-CS with setting different parameters

图 5 设置不同参数情况下 MOEA/D-CS 的性能

2)本文设计了不同种群规模情况下的参数实验,研究不同种群规模对 MOEA/D-CS 算法性能的影响.在这里,设置种群规模  $N=[50,100,200,300]$ .如图 5(c)所示,种群规模对平均  $IGD(\log)$  指标值产生重要的影响.随着种群规模的不断增加,平均  $IGD(\log)$  指标值越来越小.因此,PF 越来越接近了最优边界 PF\*.如图 5(d)所示,随着种群规模的不断增加,MOEA/D-CS 算法运行时间也不断增加.所以必须同时兼顾算法的精度和时间代价,合理地设置种群规模以满足实际需求.

## 4 结论

本文提出了一种基于混合个体选择机制的多目标进化算法,融合有监督分类与轻量级代理模型实现后代个体选择.大量实验结果表明:与当前流行的基于分解多目标进化算法相比,在没有显著增加时间代价的情况下,本文所提出的 MOEA/D-CS 算法能取得较好性能.未来工作包括:(1)设计适用于处理具有复杂 PS 曲线形状的多目标进化算法;(2)将 MOEA/D-CS 算法应用于实际多目标优化问题.

**References:**

- [1] Zhou AM, Qu BY, Li H, Zhao SZ, Ponnuthurai Nagarathnam, Zhang QF. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm & Evolutionary Computation*, 2011,1(1):32-49. [DOI:10.1016/j.swevo.2011.03.001]
- [2] Shi C, Li QY, Shi ZZ. A Quick Multi-objective Evolutionary Algorithm Based on Dominating Tree. *Ruan Jian Xue Bao/Journal of Software*, 2007,18(3):505-516(in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/505.htm> [DOI:10.1360/jos180505]
- [3] Knowles J D, Corne D W. Approximating the nondominated front using the Pareto Archived Evolution Strategy. *Evolutionary computation*, 2000,8(2):149-172. [DOI:10.1162/106365600568167]
- [4] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 2002,6(2):182-197. [DOI:10.1109/4235.996017]
- [5] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *Evolutionary Methods for Design, Optimization, and Control. ETH, TIK*, 2001. 95-100.
- [6] Yang DD, Ma JJ, Jiao LC, Gong MG, Si XY. Improved  $\epsilon$  dominance by ISO map. *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]
- [7] E Zitzler, S Künzli. Indicator-based selection in multiobjective search. In: *International Conference on Parallel Problem Solving from Nature*. Berlin: Springer, 2004. 832-842. [DOI:10.1007/978-3-540-30217-9\_84]
- [8] Basseur M, Zitzler E. A Preliminary Study on Handling Uncertainty in Indicator-Based Multiobjective Optimization. In: *Workshops on Applications of Evolutionary Computation*. Berlin: Springer, 2006. 727-739. [DOI:10.1007/11732242\_71]
- [9] 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]
- [10] Zhang QF, Li H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on evolutionary computation*, 2007,11(6):712-731. [DOI:10.1109/TEVC.2007.892759]
- [11] Li H, Zhang QF. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on evolutionary computation*, 2009,13(2):284-302. [DOI:10.1109/TEVC.2008.925798]
- [12] Zhang JY, Zhou AM, Zhang GX. A classification and Pareto domination based multiobjective evolutionary algorithm. In: *Evolutionary Computation (CEC)*. Sendai: IEEE, 2015. 2883-2890. [DOI:10.1109/CEC.2015.7257247]
- [13] Zhang QF, Liu WD, Tsang E, Virginas B. Expensive Multiobjective Optimization by MOEA/D With Gaussian Process Model. *IEEE Transactions on evolutionary computation*, 2010,14(3):456-474. [DOI:10.1109/TEVC.2009.2033671]
- [14] Zapotecas Martínez S, Coello Coello C A. MOEA/D assisted by RBF Networks for Expensive Multi-Objective Optimization Problems. In: *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. Amsterdam: ACM, 2013. 1405-1412. [DOI:10.1145/2463372.2465805]
- [15] WY G, AM Zhou, ZH Cai. A Multioperator Search Strategy Based on Cheap Surrogate Models for Evolutionary Optimization. *IEEE Transactions on evolutionary computation*, 2015,19(5):746-758. [DOI:10.1109/TEVC.2015.2449293]
- [16] Zhang JY, Zhou AM, Zhang GX. A Multiobjective Evolutionary Algorithm Based on Decomposition and Preselection. In: *Bio-Inspired Computing-Theories and Applications*. Berlin: Springer, 2015. 631-642. [DOI:10.1007/978-3-662-49014-3\_56]
- [17] Lin X, Zhang QF, Kwong S. A decomposition based multiobjective evolutionary algorithm with classification. In: *Evolutionary Computation (CEC)*. Vancouver: IEEE, 2016. 3292-3299. [DOI:10.1109/CEC.2016.7744206]
- [18] Schaffer JD. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, 1985,2(1):93-100.
- [19] N Srinivas, K Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 1994,2(3):221-248. [DOI:10.1162/evco.1994.2.3.221]
- [20] Zitzler E, Thiele L. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In: *international conference on parallel problem solving from nature*. Berlin: Springer, 1998,1498(3):292-301.
- [21] Qian H, Yu Y. Solving High-Dimensional Multi-Objective Optimization Problems with Low Effective Dimensions. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. San Francisco: AAAI Press, 2017. 875-881.

- [22] Laumanns M, Thiele L, Zitzler E. Running Time Analysis of Multi-objective Evolutionary Algorithms on a Simple Discrete Optimization Problem. In: International Conference on Parallel Problem Solving from Nature. Berlin: Springer, 2002. 44-53. [DOI:10.1007/3-540-45712-7\_5]
- [23] Laumanns M, Thiele L, Zitzler E. Running Time Analysis of Multiobjective Evolutionary Algorithms on Pseudo-Boolean Functions. IEEE Transactions on evolutionary computation, 2004,8(2):170-182. [DOI:10.1109/TEVC.2004.823470]
- [24] Qian C, Yu Y, Zhou ZH. An analysis on recombination in multi-objective evolutionary optimization. Conference on Genetic & Evolutionary Computation. New York: ACM, 2051-2058. [DOI:10.1145/2001576.2001852]
- [25] Lu XF, Tang K, Yao X. Classification-assisted Differential Evolution for computationally expensive problems. In: Evolutionary Computation (CEC). New Orleans: IEEE, 2011. 1986-1993. [DOI:10.1109/CEC.2011.5949859]
- [26] Lu XF, Tang K. Classification- and Regression-Assisted Differential Evolution for Computationally Expensive Problems. Journal of Computer Science and Technology, 2012,27(5):1024-1034. [DOI:10.1007/s11390-012-1282-4]
- [27] M Pil á, R Neruda. A surrogate based multiobjective evolution strategy with different models for local search and pre-selection. In: Tools with Artificial Intelligence (ICTAI). Athens: IEEE, 2012. 215-222. [DOI:10.1109/ICTAI.2012.37]
- [28] Yu Y, Qian H, Hu YQ. Derivative-Free Optimization via Classification. In: AAAI. Phoenix: AAAI Press, 2016. 2286-2292.
- [29] Song Y, Huang J, Zhou D, Zha H, Giles C L. IKNN: Informative K-Nearest Neighbor Pattern Classification. European Conference on Principles and Practice of Knowledge Discovery in Databases. Berlin: Springer, 2007. 248-264. [DOI:10.1007/978-3-540-74976-9\_25]
- [30] Loh W Y. Classification and regression trees. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 2011, 1(1):14-23. [DOI:10.1002/widm.8]
- [31] Tarabalka Y, Fauvel M, Chanussot J. SVM-and MRF-Based Method for Accurate Classification of Hyperspectral Images. IEEE Geoscience & Remote Sensing Letters, 2010, 7(4):736-740. [DOI:10.1109/LGRS.2010.2047711]
- [32] Zhou AM, Zhang QF, Zhang GX. A Multiobjective Evolutionary Algorithm based on Decomposition and Probability Model. Evolutionary Computation. Brisbane: IEEE, 2012. 1-8. [DOI:10.1109/CEC.2012.6252954]
- [33] Bhattacharya A, Chattopadhyay P K. Hybrid Differential Evolution With Biogeography-Based Optimization for Solution of Economic Load Dispatch. IEEE Transactions on Power Systems, 2010, 25(4):1955-1964. [DOI:10.1109/TPWRS.2010.2043270]
- [34] Gaspar-Cunha A, Covas J A. Robustness in multi-objective optimization using evolutionary algorithms. Computational Optimization & Applications, 2008, 39(1):75-96. [DOI:10.1007/s10589-007-9053-9]
- [35] Hillermeier C. Nonlinear multiobjective optimization. Basel: Springer, 2001. 66-74.
- [36] 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]
- [37] Li Y, Zhou AM, Zhang GX. An MOEA/D with Multiple Differential Evolution Mutation Operators. In: Evolutionary Computation (CEC). Beijing: IEEE, 2014. 397-404. [DOI:10.1109/CEC.2014.6900339]
- [38] 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 with English abstract). <http://www.jos.org.cn/1000-9825/4514.html> [DOI:10.13328/j.cnki.jos.004514]
- [39] Zhang QF, Zhou AM, Jin YC. RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm. IEEE Transactions on evolutionary computation, 2008,12(1):41-63. [DOI:10.1109/TEVC.2007.894202]

#### 附中文参考文献:

- [2] 石川, 李清勇, 史忠植. 一个快速的基于占优树的多目标进化算法. 软件学报, 2007,18(3):505-516. <http://www.jos.org.cn/1000-9825/18/505.htm> [DOI:10.1360/jos180505]
- [6] 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]
- [38] 周爱民, 张青富, 张桂成. 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.html> [DOI:10.13328/j.cnki.jos.004514]