2017 届研究生硕士学位论文

分类号: _____ 学校代码: 10269

密　级: _____ 学　　号: 51141201039

# 華東師範大學

# East China Normal University

# 硕士学位论文

# MASTER'S DISSERTATION

## 论文题目: 分布估计算法中基于差分采样策略研究

院　　系: 计算机科学与软件工程学院

专　　业: 计算机科学与技术

研 究 方 向: 演化计算

指 导 教 师: 周爱民　副教授

学 位 申 请 人: 董兵

2017 年 6 月 6 日

Dissertation for

Master degree, 2017

Student ID: 51141201039

University Code: 10269

# East China Normal University

**Title: The study of the sampling strategy based on differential evolution in estimation of distribution algorithms**


Department: School of Computer Science and Software Engineering

Major: Computer Science and Technology

Research direction: Evolutionary Computation

Supervisor: A. Prof. Aimin Zhou

Candidate: Bing Dong


June, 2017 · Shanghai

# 华东师范大学学位论文原创性声明

郑重声明: 本人呈交的学位论文《演化计算中基于差分进化的采样策略》, 是在华东师范大学攻读硕士 / 博士 ( 请勾选 ) 学位期间, 在导师的指导下进行的研究工作及取得的研究成果. 除文中已经注明引用的内容外, 本论文不包含其他个人已经发表或撰写过的研究成果. 对本文的研究做出重要贡献的个人和集体, 均已在文中作了明确说明并表示谢意.

**作者签名:** _____ **日期:** 年 月 日

# 华东师范大学学位论文著作权使用声明

《演化计算中基于差分进化的采样策略》系本人在华东师范大学攻读学位期间在导师指导下完成的硕士/博士（请勾选）学位论文, 本论文的著作权归本人所有. 本人同意华东师范大学根据相关规定保留和使用此学位论文, 并向主管部门和学校指定的相关机构送交学位论文的印刷版和电子版; 允许学位论文进入华东师范大学图书馆及数据库被查阅、借阅; 同意学校将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索, 将学位论文的标题和摘要汇编出版, 采用影印、缩印或者其它方式合理复制学位论文.

本学位论文属于 ( 请勾选 )

( ) 1. 经华东师范大学相关部门审查核定的 " 内部 " 或 " 涉密 " 学位论文 *, 于 年 月 日解密, 解密后适用上述授权.

( ) 2. 不保密, 适用上述授权.

**导师签名:** _____ **本人签名:** _____

年 月 日

*** " 涉密 " 学位论文应是已经华东师范大学学位评定委员会办公室或保密委员会审定过的学位论文 ( 需附获批的《华东师范大学研究生申请学位论文 " 涉密 " 审批表》方为有效 ), 未经上述部门审定的学位论文均为公开学位论文. 此声明栏不填写的, 默认为公开学位论文, 均适用上述授权.**

# 董兵 硕士学位论文答辩委员会成员名单

| 姓 名 | 职 称 | 单　　位 | 备注 | |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |

# 分布估计算法中基于差分进化的采样策略

## 中　文　摘　要

演化算法是一种受生物进化启发的基于种群的启发式的优化算法。演化算法适用于解决各种问题，因为它并不需要复杂的假设条件。演化算法已经被广泛应用于工程、生物学、经济学、基因工程以及社会科学等。

分布估计算法是一种新型的演化算法。不同于传统的演化算法，分布估计算法中没有杂交变异操作。分布估计算法中主要由三个主要的步骤组成，即：建模，采样，选择。采样对于分布估计算法来说是至关重要的环节，它关系到能否产生更为优异的的子代种群。优异的子代种群对于最终求得最优解有着重大意义。

差分进化自从1995年被提出之后就受到研究者的广泛关注。差分进化是一种简单但却有效的随机优化算法，得益于它的诸多优点，它已经被广泛应用于各个领域。由于差分进化的诸多优点，差分进化在单目标全局优化，多目标优化，约束优化等领域受到了广大学者的研究和关注。

本文研究的是对于演化算法中采样策略的相关工作。受差分进化思想所启发，在本文中提出了一种基于差分进化的采样策略，即查分采样策略，并将这种采样策略应用于分布估计算法之中，从而提高算法性能和运行效率。本文将基于差分进化的采样策略和分布估计算法相结合，并且分别研究其在解决多目标以及单目标问题上的性能表现。通过综合的实验结果分析，可以看出基于差分进化的采样策略在解决分布估计算法中的采样问题具有较好的表现，并且具有很大的潜力。

本文的主要内容分别针对单目标优化问题和多目标优化问题。对于单目标优化问题，基于DE/EDA算法框架，通过结合基于特征向量的查分进化来采样，同时，利用expensive local search(LS)来进一步提高解集质量。另一方面，对于多目标优化问题，通过差分采样策略来改进RM-MEDA的采样，并且效果显著。通过将差分采样策略分别应用单目标和多目标分布估计算法，并且在综合的实验分析的基础上，有效地表明了差分采样策略对于分布估计算法中的重大意义。

**关键词：** 分布估计算法，差分进化，演化算法，全局优化

# Differential Evolution Sampling Strategy in Evolution Algorithms

## ABSTRACT

This paper mainly studies the differential evolution sampling strategy in evolutionary algorithms. Evolution algorithm is a kind of algorithm inspired by biological evolution. Evolutionary algorithms are utilized to solve diverse problems. As it does not need any complex assumption conditions. Evolutionary algorithms have been widely applied to engineer, biology, economy, gene engineer and social science.

Estimation of distribution algorithm is a novel evolutionary algorithm. Unlike traditional evolutionary algorithm, there is no crossover and mutation in EDA. EDA mainly consists of three steps, namely, modeling, sampling and selection. Sampling is crucial to EDA. Since it is significant to generate promising offspring generation, which will be useful to obtain the final solutions.

Differential evolution has attracted many researchers since proposed in 1995. Differential evolution is a simple but powerful random optimization algorithm. Due to the advantages of differential evolution, it has been widely applied to different areas. As it is easy to implement differential evolution, and it can combine with other evolutionary algorithm. Hence lots of researcher have proposed a number of hybrid algorithms based on differential evolution. This paper mainly study the sampling in EDA based on differential evolution.

This paper combine differential evolution sampling strategy with EDA. Moreover, we will study its performance on single object and multi-objective problems. And the experimental results have shown that it is significant to improve the performance of EDA.

**Key words:** estimation of distribution algorithm, differential evolution, evolutionary algorithm, eigenvector

# 目 录

# 1 绪论

演化计算 [1–3]是用于解决大量优化问题的通用的并且强大的优化算法框架。通过利用现有的解并且同时在解空间中继续搜索，演化计算能够在合理的时间内找到最优解。通过结合现有解的统计信息以及搜索的解空间的属性，演化计算可以有效地解决优化问题。演化计算通过种群个体的演化（即代表对于问题的候选解）的搜索来求解最终解。

演化计算，受自然界进化所启发，也就是对现有的解进行过滤并且产生新的解。个体中的基因相对于问题解中的组成部分。评价函数被用来评价个体对于问题解决的程度。种群中的个体通过在搜索空间中随机采样产生。在演化的过程中，现有的个体通过交叉变异来交换信息来产生新的子代种群。具有更好适应性的个体更有可能进行交叉变异，它们产生的子代也会更优。通过产生越来越好的解，演化计算最终就有可能找到全局最优解。演化计算的基本流程图如图 1所示。

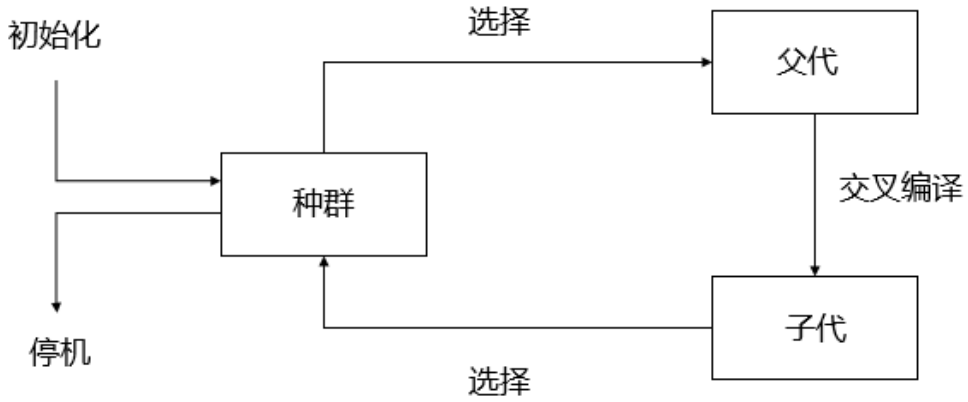



**图** 1: 演化计算的基本流程图

演化算法是隶属于演化计算的一种人工智能算法，是一种基于基因种群的启发式的优化算法。演化算法被广泛地应用于解决优化问题，包括单目标以及多目标优化问题 [4]。。现如今演化算法已经得到了众多研究学者的关注，演化算法已经分化成多个分支：遗传算法 [5]，生物地理学优化 [6]，遗传编程 [7]，演化编程 [8]，差分进化 [9]等等。演化算法被广泛地应用于工程实践和学术研究中。

## 1.1 研究目的和意义

分布估计算法是一种新兴的演化算法，它通过建立一个概率模型，然后通过在这个概率模型中采样来产生新的种群个体，通过提取当前精英中的统计信息来进一步地更新概率

模型来指导下一步的搜索 [10] [11]。不同于传统的演化算法，在分布估计算法中不存在变异和交叉，通过使用一个精确的概率模型来提取种群的统计信息。子代种群解集将会通过从建立的概率模型中采样产生，并且产生的解集会部分或者全部地取代父代中区解集。分布估计算法通过概率模型能够更好地进行全局搜索，提取全局统计信息。分布估计算法主要由三个步骤组成，即：建模，采样，选择。采样时分布估计算法的重要环节，本文主要针对分布股计算中的采样，提升采样的性能可以大大地提高算法的性能，更有可能产生更为优质的子代种群解集 [12]。

差分进化算法是一种有效并且简单的演化算法 [9, 13]。差分进化算法通过利用种群中的距离和方向信息来指导搜索的过程。差分进化算法通过提取种群中的差分信息来指导下一步搜素并且可以加快解集收敛的速度。得益于这一点，受到差分进化算法思想的启发，我们提出基于差分进化的采样策略（Differential Evolution based Sampling, DES)。通过利用分布估计算法中概率模型不断收集种群中的统计信息，不断地更新概率模型，指导搜索的进行，同时利用基于差分进化的采样策略，提高种群的多样性，在保证种群的多样的同时，也加速了最终结果的收敛，从而进一步提高算法的性能。

## 1.2 本文所做的工作和内容安排

针对分布估计算法的采样，提出一种基于差分进化的采样策略，用于提高分布估计算法的性能。我们针对单目标和多目标优化问题，通过一系列的综合实验分析来研究基于差分进化的采样策略对于提高演化算法性能的意义。在多目标优化问题上，DES被用于改进基于平滑模型的分布估计算法（A Regularity Model-based Estimation of Distribution Algorithm, RM-MEDA) [14] 中的采样策略，提高算法性能。在单目标优化问题上，基于差分进化和分布估计算法的混合算法（DE/DEA) [15]，通过改进DE/EDA 中的差分进化算法，并且同时提出一种基于特征向量的改进的差分进化和分布估计算法的混合算法（EDA/DE-EIG），研究其对于提高分布估计算法对于解决单目标优化问题上算法性能和速度提升的意义。

论文内容安排如下：

第一章 绪论，介绍演化算法以及本文的研究目的和论文内容安排。

第二章 研究背景，介绍了最优化问题以及演化算法，接着着重介绍了分布估计算法和差分进化的相关概念和当前的研究现状。

第三章 对于连续多目标优化问题的研究，提出一种基于差分进化的采样策略，通过这种新型的采样策略去改进RM-MEDA的采样环节，具体描述了相关的研究背景和算法框架和具体实现。最后，通过综合的实验对比来分析基于差分进化的采样策略的重要意义。

第四章 对于全局单目标优化问题的研究，基于DE/EDA算法框架，引进基于特征向量的差分进化算法，并且通过结合expensive LS来进一步提高算法性能，最终提出基于差分进化采样策略的单目标分布估计算法EDA/DE-EIG。通过和JADE，DE/EDA的综合实验分析对比，可以对于基于差分进化的采样策略在解决单目标优化问题的作用和意义。

第五章 对本文进行全面地总结，并且根据目前工作的不足之处提出未来的展望。

# 2 研究背景

本章首先介绍了演化算法中的单目标优化问题以及多目标优化问题，并给出了相关定义。接着分别简要地介绍分布估计算法和差分进化算法的理论知识和算法框架，并就差分进化和分布估计算法的研究现状做了基础性地介绍。

## 2.1 最优化问题

在数学以及计算机科学中，最优化问题就是对于一个问题在可能的解集中找到一个最优的解。可以将最优化问题使用以下公式描述：

$$\begin{aligned} min \quad & f(x) \\ s.t. \quad & f_i(x) \leq b_i, i = 1, \cdots, m. \end{aligned} \tag{2.1}$$

其中$x$是一组决策向量，$f_i(x) \leq b_i, i = 1, \cdots, m$是相应的约束条件。对于目标函数$f(x)$，在满足约束条件的情况下求得最小的解。

根据决策变量是离散的还是连续的，最优化问题可以分为组合优化问题和连续优化问题。本文的重点主要是连续优化问题。连续优化问题在工程实践和数学领域都有着广泛的应用，因此对于连续优化问题的研究具有重要意义。

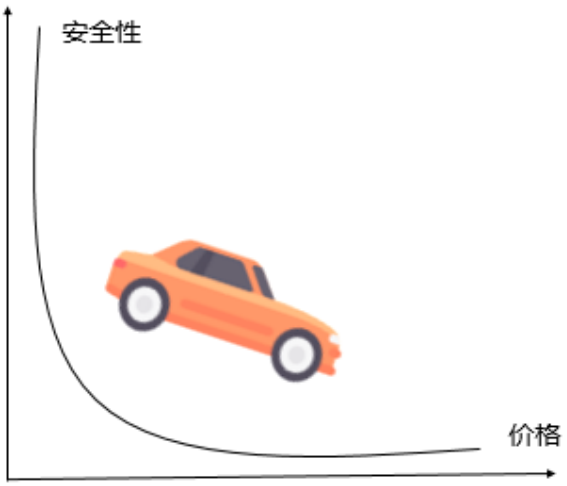



**图** 2: 将买车问题认为是一个多目标问题，在考虑价格的同时还需要考虑安全性

在本文中，主要针对的是单目标和多目标优化问题。单目标问题就是对于一个目标求得最优的解。在科学研究和工程应用中，在对于设计和策略的解决方案中往往涉及到对于

多个目标的优化问题，这就是本文中所说的多目标优化问题（Multiobjective Optimization Problems, MOPs）。拿一个我们日常生活中买车的例子，如图 2所示，在买车的时候我们既要考虑到价格同时也要考虑到安全，这可以理解成为一种多目标优化问题。

## 2.2 演化算法

对于最优化问题求解的方法是多种多样的。传统的优化方法包括梯度下降法、牛顿法以及共轭梯度法等。这些问题一般是利用问题的梯度信息或者其它信息来逼近问题的最优解。但是这些方法在靠近极小值或者在求解效率上还有较大的提升空间。

## 2.3 分布估计算法

分布估计算法（Estimation of Distribution Algorithm, EDA），也被称为基于概率模型的遗传算法（Probabilistic Model-based Genetic Algorithm, PMGA），是一种通过建模和采样来搜索最优解的随机优化方法 [10]。分布估计算法虽然属于演化算法，但是和传统的演化算法却有着较大的不同之处。传统的算法通过变量之间隐含的分布关系来产生新的子代种群，然而分布估计算法是通过概率模型建立的精确的分布关系来产生新的种群 [16]。

### 2.3.1 算法框架

分布估计算法主要由三个步骤组成：建模，采样和选择。传统的分布估计算法的算法框架如算法 1所示。

**算法 1:** 分布估计算法

**初始化**: 建立随机初始种群$Pop(t)$，$t$ 是相应的种群代数。
**while** *not terminate* **do**
  **建模：**根据种群$Pop(t)$中的统计信息建立概率模型$p(x)$。
  **采样：**通过从建立的概率模型$p(x)$中采样产生一个新的解集$Q$。
  **选择：**根据某个条件从$Q \cup Pop(t)$中挑选后代组建下一代种群$Pop(t+1)$。
  $t = t + 1$
**end**

### 2.3.2 分布估计算法的研究现状

分布估计算法自从1994年首次提出后，就得到了广泛地应用于解决各种大型的复杂问题，包括军事天线设计 [17]，多目标背包问题 [18]，地下水治理 [19]，森林管理 [20]等。值得强调的一点是，在解决相同等级大小和复杂度的问题的时候，分布估计算法往往就是最优的算法。

根据分布估计算法针对的不同问题，可以将分布估计算法分为以下几种类型：

基于离散变量的分布估计算法：对于分布估计算法，模型的种类是比较重要的环节，在此，可以分为，单变量模型，树形表示模型以及以及多元关系模型。

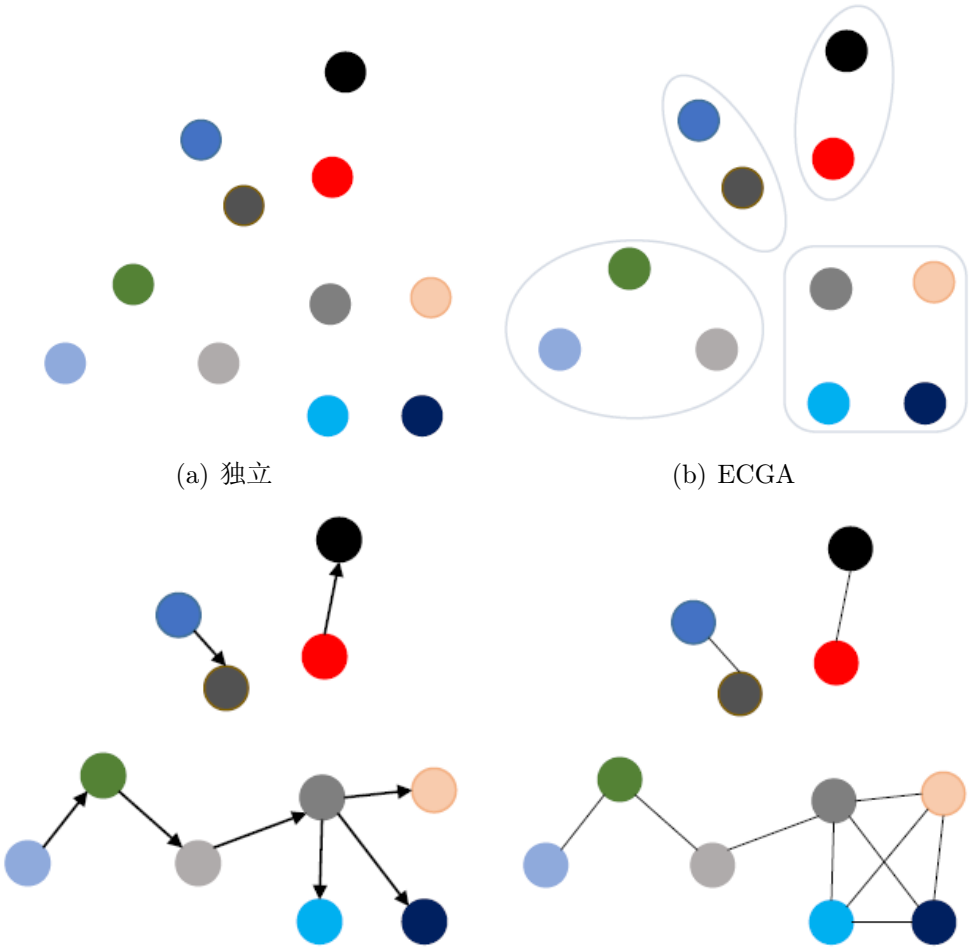
最简单的方法就是假设问题的变量之间是相互独立的。在这种假设下，个体变量的概率分布就和其他变量的没有关系。这种分布估计算法通常被称为单变量分布估计算法，图3表示这种类型的分布估计算法。从数学意义上来说，单变量模型可以将多个变量的概率进行分解成多个变量概率的乘积：

$$p(X_1, X_2, \cdots, X_n) = p(X_1)p(X_2), \cdots, p(X_n) \tag{2.2}$$

$p(X_i)$是变量$X_i$的概率分布，$p(X_1, X_2, \cdots, X_n)$是候选解集$(X_1, X_2, \cdots, X_n)$的概率分布。

对于单变量模型的例子，比如单变量边缘分布算法（UMDA） [21],用于求解onemax问题。然而大多数的分布估计算法是通过维护一个候选解种群来工作的。增量分布估计算法通过概率模型来替换种群。这个概率模型开始在解空间中服从均匀分布。在每一个迭代之中，都会产生新的解，最好的解或者相对来说比较好的解会被选择进入下一代迭代。这个概率模型就会慢慢地向解的类型变化。通过这种方式，概率模型在不断地改良，并且不需要将大量的种群全部存储起来。基于种群的增量学习（PBIL） [22]是一种作用于二进制串的单变量的增量分布估计算法。像UMDA 一样，PBIL也是通过概率模型来构建概率向量。为了防止种群过早收敛，会通过一个变异比率参数在每一次迭代中对概率向量进行微调。压缩遗传算法（cGA） [23]是另外一种增量的单变量增量分布估计算法。和PBIL类似，cGA也是通过使用概率向量来代表整个种群中的解集，使用指定长度的二进制串进行编码。cGA和PBIL最主要的区别就是这两者在每一次迭代中更新概率向量的方法。

单变量模型是基于变量之间是相互独立的，然而很多问题的变量往往是相互联系的。双变量依赖假设是比变量独立假设更为宽松的假设，假设随机变量的依赖性仅存在于两个变量之间。因此接着继续讨论使用树形表示模型的分布估计算法。MIMIC [24]



(a) 独立 (b) ECGA

(c) 贝叶斯 (d) 马尔科夫

**图** 3: 不同分布估计算法的模型图

是一种通过链分布来描述变量之间关系的分布估计算法。假设问题中n个变量的排列为，$\pi = i_1, i_2 \cdots i_n$，MIMIC可以将$p(X_1, X_2, \cdots, X_n)$的概率分布分解为：

$$p_\pi(X) = p(X_{i_1}|X_{i_2})p(X_{i_2}|X_{i_3}) \cdots p(X_{i_{n-1}}|X_{i_n})p(X_{i_n}) \tag{2.3}$$

$p(X_{i_j}|X_{i_{j+1}})$表示在给定$X_{i+j+1}$的条件下，$X_{i_j}$的条件概率。新的解集通过从这个概率模型中采样而产生。采样的过程会将$\pi$中的变量顺序逆置，以$X_{i_n}$开头，$X_{i_1}$ 结尾。

相对于MIMIC中的链分布模型，Baluja等人于1997年提出使用依赖关系树来进一步提升概率模型的表现力 [25]。在依赖关系树中，每一个父代都可以有多个子代。这个分估计算法依靠一个包含所有成对关系概率的概率矩阵来工作。概率模型中的关系树可以最大化相互联系变量之间的交互信息，根据真实分布的相对熵，这个模型被证明是最优的模型 [26]。这个概率矩阵是的初始化是相对应于所有解集的均匀分布。在算法的每一次的迭代中，依赖关系树都会被建立并且产生新的解。接着最好的解会被用来进一步地更新概率矩阵。二变量边缘分布算法（BMDA）使用一组相互独立依赖关系树来建模 [27]。通过从种群中获得的条件概率信息来建模并且采样。

对于多元变量的问题，依赖关系树模型可能就没有办法很好地描述。拓展压缩遗传算法（ECGA） [28]使用一种模型可以将变量划分成独立的聚类，每一个聚类被当做是一个单独的变量。这个模型的建立也是建立在所有问题变量都是独立的假设之上的。在模型建立的每一次迭代中，两个聚类会合并来进一步提升模型的质量。模型的质量是通过最小描述长度（MDL）来度量的。当聚类的合并没有办法提升模型质量的时候，建模过程就会终止。一旦模型的结构建立完成，就会计算出了一个概率表来描述在每个聚类中基于概率关系被挑选的解和采样产生的新解。在ECGA的每一个迭代中，这个建模过程都会被重复，因此在每次迭代中，概率模型都会包含不同的变量聚类。ECGA概率模型的例子如图3(b)所示。很多包含重叠子问题的问题没有办法通过将问题化成独立的聚类。贝叶斯优化算法通过使用贝叶斯网络来对候选解集进行建模。贝叶斯网络是一个无环的图，每一个结点代表一个变量，没一条边代表结点之间的条件依赖关系。一个具有n个结点的贝叶斯网络可以通过使用n个随机变量$X_1, X_2, \cdots, X_n$的联合分布来表示：

$$p(X_1, X_2, \cdots, X_n) = \prod^n{}_{i=1} p(X_i|\prod i) \tag{2.4}$$

$\prod_i$表示和$X_i$之间存在一条边的一组变量集合，$p(X_i|\prod_i)$表示在条件$\prod_i$下$X_i$的条件概率。图 3(c)是贝叶斯网络的图例。贝叶斯网络和依赖关系树的主要的区别是在贝叶斯网络

中，每一个变量可能依赖于多个变量。ECGA的概率模型和贝叶斯网络最主要的区别是贝叶斯网络能够获得问题中更复杂的子问题之间的交互信息。BOA中概率模型的简历是从一个没有边的网络开始的。使用一个贪婪算法向网络中添加边添加边的依据是BD度量。贝叶斯网络估计算法（EBNA） [29]和学习可分解分布算法（LFDA） [30]同样也是采用贝叶斯网络对解集进行建模。EBNA和LFDA都是利用贝叶斯信息准则（BIC）来评估利用贪婪网络算法构建的贝叶斯网络结构。通过对BOA在两个关键地方进行拓展，层次贝叶斯优化算法（hBOA） citehboa2006能够解决很多困难的层次分解问题。为了能够描述高层次之间的香菇关系，hBOA使用更为紧凑的贝叶斯网络。

另外一种可以对多元信息进行编码的概率模型是马尔科夫网络。除了在马尔科夫网络中，变量的之间的联系是无向的这一点，马尔可夫网络的结构和贝叶斯网络相似，如图3(d) 所示。对于一个给定的分解函数，马尔可夫网络相较于贝叶斯网络可能会更容易收敛到全局最优解。然而，在马尔科夫网络中采样往往是更加困难的。

基于遗传编程的分布估计算法：在分布估计算法的成功，多数研究者希望在遗传编程（GP）这一领域继续复制这一成功 [31]。在GP领域中，尽管存在着众多挑战，但是分布估计算法还是在GP中得到了成功的应用。

概率增强式编程进化（PIPE） [32]使用概率原型树（PPT）来存储概率分布的所有函数和编程树中每个结点的操作。对于PIPE的拓展算法是拓展压缩遗传编程（ECGP） [33]。受ECGA 所启发，ECGP将编程树里面的节点分割成独立的聚类，其工作方式和ECGA类似。由于编程空间中复杂性，在候选编程的整个搜索空间中找到准确的问题分解是十分困难的。MOSES通过将搜索空间分割成子编程空间并对这些空间保持维护。接着利用hBOA 对每一个子空间来产生新的编程，这样同时也能产生新的子空间。为了解决遗传编程的问题，有的分布估计算法通过语法规则来实现。基于语法规则的随机遗传编程（SG-GP） [34]从一个指定的与上下文无关的语法规则开始，然后为每一条规则添加默认概率。基于每一个迭代中采样产生的优质解，规则的概率也表现得越来越好。在这个基本算法里面，没有存储位置信息，可以通过跟踪规则的使用来进一步拓展该算法。

基于多目标的分布估计算法：上述介绍的分布估计算法基本上是解决单目标问题的。然而，现实世界中的问题往往是具有多个目标的。一种针对多目标问题的解决方案是根据目标权重将多目标转化为单目标。但是，更好的方式在目标之间找到一个最好的权衡从而构建一个帕累托最优 [35]。简而言之，帕累托最优要比在任意目标上的最优解要更好。因此，分布估计算法被用于在多目标优化问题中寻找多种帕累托最优解。

对于多目标问题，就不可能找到一个解可以使所有的目标达到最优解。最终的目标

是找到帕累托最优解集的宽分布。贝叶斯网络多目标优化算法（BMOA） [36]使用一个特别的选择选择算子$\epsilon$-archive来更新帕累托最优解集并且保证种群的多样性。这个选择算子会维护一个最小的解集，这个算子会支配其他产生的解。本质上来说，BMOA是将混合BOA [37]和SPEA2 [38]结合起来。

MIDEA算法 [39]是对于多目标优化问题上的IDEA算法框架的进一步拓展。通过使用参数$\delta$来指导搜索过程，通过这一特别的选择算子来保证种群的多样性。接着通过领导者算法对种群进行聚类，从而对于每一个聚类建立一个单变量模型并且采样产生新的解。多目标贝叶斯优化算法（mBOA） [40]通过使用NSGA-II选择算子来维持Pareto最优解前沿种群的多样性。这个选择算子会给种群中的每一个解等级和聚集距离。这个等级可以最大化所有的目标，反之聚集距离能够维持Pareto前沿的多样性和广覆盖。如果两个解的等级不同，等级优的进入下一代；如果两个解的等级相同，那么聚集距离则作为选择标准。对于这些被选择的解建立一个贝叶斯网络模型，然后对于模型进行采样产生新的解来构建下一代种群。多目标层次贝叶斯优化算法（mohBOA） [41]通过结合hBOA，NSGA-II和聚类来对hBOA做进一步的拓展。

## 2.4 差分进化

差分进化（Differential Evolution, DE）自从由Storn于1995年提出，就得到了广大学者的关注并且发展迅速 [9]。自从20 世纪90年代以来，差分进化算法就在多数科学工程领域得到广泛的应用 [42]。究其原因，可以总结如下：

- 与传统的演化算法相比，差分进化更简单也更容易实施。算法的代码往往只需要几行代码即可，因此它可以很好地应用于其他的领域。尽管粒子群优化算法（Particle Swarm Optimization, PSO）的代码也比较简单，但是差分进化算法在大多数问题上的表现都比粒子群优化算法要更加优秀 [43, 44]。

- 另一方面，差分进化中的控制参数和其他的演化算法相比，控制参数更少。在经典的差分进化中，一般只有控制参数Cr，缩放因子F以及种群大小NP三个参数。对于F和Cr的自适应规则的研究，在不给算法带来额外的负担的条件下，对于算法性能的提升意义重大 [45, 46]。

差分进化是一个基于种群的启发式优化算法。和其他的演化算法类似，差分进化也包含三个基本的操作：变异，交叉以及选择。差分进化通过变异操作产生变异向量，然后通过交叉操作产生交叉向量，最后在交叉向量和种群中选择个体进入下一代种群中。

### 2.4.1 算法框架

**算法 2:** 差分进化

随机初始种群$P_0$：$P_0 = \{x_{1,D}, x_{2,D}, x_{3,D}, \cdots, x_{N,D}\}$
**while** *not terminate* **do**
  $v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} + x_{r3,G})$
  **if** $rand_j(0,1) \leq CR$ *or* $j = j_{rand}$ **then**
    $u_{i,j,G} = v_{i,j,G}$
  **else**
    $u_{i,j,G} = x_{i,j,G}$
  **end**
  **if** $f(u_{i,G}) \leq f(x_{i,G})$ **then**
    $x_{i,G+1} = u_{i,G}$
  **else**
    $x_{i,G+1} = x_{i,G}$
  **end**
**end**

差分进化主要由三步组成，包括：变异，交叉和选择，经典的差分进化的算法框架如算法 2所示。

其中$F$是缩放因子，$CR$是交叉概率因子。$v_{i,G}$是变异向量，$u_{i,G}$是试验向量，$x_{i,G+1}$是目标向量。$r1$，$r2$和$r3$是从$[1, N]$中挑选出来的整数，且它们也不同于$i$。

在传统意义中的演化计算中，变异是通过一个随机元素从而使种群发生改变或者扰动从而产生变异向量来找到更好的解。在差分进化中，对于每一个目标向量$x_{i,G}$都会使用某一种变异策略来产生相应的变异向量$v_{i,G}$。下面是p六种文献中常用的变异策略：

1. DE/rand/1策略

$$v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} + x_{r3,G}) \tag{2.5}$$

2. DE/best/1策略

$$v_{i,G} = x_{best,G} + F \cdot (x_{r1,G} + x_{r2,G}) \tag{2.6}$$

3. DE/rand/2策略

$$v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}) + F \cdot (x_{r4,G} - x_{r5,G}) \tag{2.7}$$

4. DE/best/2策略

$$v_{i,G} = x_{best,G} + F \cdot (x_{r1,G} - x_{r2,G}) + F \cdot (x_{r3,G} - x_{r4,G}) \tag{2.8}$$

5. DE/current-to-best/1

$$v_{i,G} = x_{r1,G} + F \cdot (x_{best,G} - x_{i,G}) + F \cdot (x_{r1,G} - x_{r2,G}) \tag{2.9}$$

6. DE/current-to0rand/1

$$u_{i,G} = x_{i,G} + K \cdot (x_{r1,G} - x_{i,G}) + F' \cdot (x_{r2,G} - x_{r3,G}) \tag{2.10}$$

其中$x_{best,G}$是指当前种群中的最优个体，$x_{i,G}$为目标向量（父代种群个体），$u_{i,G}$ 是父代种群个体对应的变异向量。不同的策略往往针对不同类型的问题更为有效，一般来说，DE/rand/1 是最常用的策略。

差分进化利用交叉算子来结合变异向量$v_{i,G}$和目标向量$x_{i,G}$来产生试验向量$u_{i,G}$。$CR$是交叉控制参数，$j_{rand}$是1到$D$之间的随机整数，为了使试验向量至少在一个维度上与目标向量不同。$rand_j(0,1)$是一个0到1之间服从均匀分布的随机数。

经典的差分进化利用一对一的竞争机制从目标向量和试验向量中挑选个体。如果试验向量的目标函数值小于或者等于目标向量的目标函数值，则试验向量进入下一代，否则目标向量进入下一代。

### 2.4.2 差分进化算法的研究现状

差分进化算法自从提出之后，就受到了工业界以及学术界的广泛关注，针对差分进化的研究百花齐放。根据差分进化解决的问题类型，可以将差分进化分为以下几种类型,对于下面每种类型的差分进化算法下面也分为多种类型的算法。

**针对连读单目标优化问题的差分进化算法：**Fan等人在提出了一种三角变异算子来提高差分进化的性能。为了实现这种方法，对于对于每一个目标向量，会在种群中随机选择三个不同的向量。对于第$i$个目标向量$X_{i,G}$而选择的种群个体是$X_{r1,G}$，$X_{r2,G}$和$X_{r3,G}$。三个下标$r1$，$r2$和$r3$ 是从[1, NP]区间选择的三个互不相同的整数，同时它们也不同于$i$。那

么这三个加权系数就可以根据下面的公式生成：

$$\begin{aligned} p^{'} &= |f(X_{r1})| + f(X_{r2})| + |f(X_{r3})| \\ p_1 &= \frac{|f(X_{r1}|}{p^{'}} \\ p_2 &= \frac{|f(X_{r2}|}{p^{'}} \\ p_3 &= \frac{|f(X_{r3}|}{p^{'}} \end{aligned} \tag{2.11}$$

令Γ为0到1之间的三角变异比率，那么三角变异可能会按照以下方式实现：

$$V_{i,G+1} = \begin{cases} \frac{X_{r1}+X_{r2}+X_{r3}}{3} + (p_2-p_1)\cdot(X_{r1}-X_{r2}) + (p_3-p_2)\cdot(X_{r2}-X_{r_3}) \\ +(p_1-P_3)\cdot(X_{r3}-X_{r1}) \qquad \text{如果} \quad rand[0,1] \le \Gamma \\ X_{r1}+F\cdot(X_{r2}-X_{r3}) \qquad\qquad \text{其他} \end{cases} \tag{2.12}$$

因此，这个方法会在Γ概率下执行三角变异，在$(1-\Gamma)$概率下执行DE/rand/1变异策略。

Price在2006年又提出一种先进的DE/rand/1-either-or算法 [47]。这个算法中，在概率$p_F$下，试验向量全都是突变体，而在概率$1-p_F$下，试验向量全都是重组体。这一新的方法相对于传统的rand/1/bin和target-to-best/1/bin有着较大的竞争力。试验向量的产生方式如下：

$$U_{i,G} = \begin{cases} X_{r1,G} + F\cdot(X_{r2,G}-X_{r3,G}) \quad \text{如果} rand_i(0,1) < p_F \\ X_{r0,G} + k\cdot(X_{r1}+X_{r2}-2\cdot X_{r0} \qquad \text{其它} \end{cases} \tag{2.13}$$

参数$k$是对于算术交叉的加权系数，对于它的推荐设置是对于给定的$F$,将其设置为$k = 0.5\cdot(F+1)$。参数$p_F$决定了变异和算术交叉的重要性，推荐设置是0.4。根据目标函数的一些特殊属性从而自适应地设置参数$p_F$的值，这也是一个值得研究的话题。

反向差分进化（ODE） [48]是一种基于反向学习的用于快速全局搜索和优化的算法。这个算法在噪音优化方面得到了重要的应用。通过对利用反向数对传统差分在三个层次上进行进化优化，即，种群初始化，迭代数跳跃和种群最优解局部改善。

大多数的演化算法的运行效率取决于它们在搜索时利用和开发的趋势。为了更好地平

衡演化算法中的利用和开发的趋势，Das提出了两种邻域拓扑模型 [49]。根据这一原理，提出了基于全局和局部邻域的差分进化（DEGL），对于DE/target-to-best/1策略做出了不小的改进。

差分进化中试验向量的产生策略多种多样，每一个策略可能只对特定的问题比较有效。Qin等人第一次提出了自适应的差分进化算法SaDE [50]，根据之前解的产生过程来调整参数从而达到试验向量的产生策略的自适应。在SaDE使用了四个有效地试验向量产生策略，包括：DE/rand/1/bin，DE/rand-to-best/2/bin，DE/rand/2/bin以及DE/current-to-rand/1策略，通过这些策略来构建一个策略候选池。前三个策略使用二项式交叉，最后一个使用算术重组。在SaDE算法中，对于当前种群中的每一个试验向量，会根据之前能够进入下一迭代节的成功比率从而在参数池中选择合适的策略。接着这个策略会被用于目标向量来产生试验向量。更特别的是，在每一次迭代中，四个参数的选择概率的总和为1。最初每个策略的概率都是相同的，然后对着演化过程的推进，会自适应地做出相应的调整。JADE [51]是一种新型的差分进化算法，在JADE中提出了一种新的变异策略"DE/current-to-$p$best"，它通过使用一种自适应的方式来进行种群的更新，从而提高算法的自适应性。对于种群中的每一个目标向量，JADE都会对F和CR 进行更新，同时这些信息也将用于更新F和CR，从而作用于新的种群。

相对于共通启发式算法，混合技术是将两个或者多个算法的特点结合起来构建一个新的混合算法，这个混合算法比原始任意算法在解决特别的应用或者相应的测试集上表现得更佳。一般来说，差分进化的混合算法包括将差分进化和全局优化算法相结合以及将差分进化和局部搜索方法两个种类。粒子群算法（PSO）最初的提出是模仿人类的社交行为，现在已经被广泛地应用于全局搜索和优化。Hendtlass第一次将差分进化和粒子群优化算法结合起来，提出一个混合优化器 [52]。在这个优化器中，只有后代的适应度更好的时候粒子的位置才会发生改变。差分进化只会在特定的间隔的迭代中才会对粒子群算法起作用。Zang等人提出了另外一种流行的粒子群差分进化混合算法（DEPSO） [53]。在这个算法中，粒子群算法和差分进化分别作用于奇数迭代和偶数迭代中。DEPSO相对于原始的算法在某些约束优化问题上能够达到更好的收敛。对于粒子群算法和差分进化，还有更多的研究，在此就不一一叙述。差分进化还与其他的全局优化算法相结合，比如：菌群优化算法（BFOA） [54]，蚁群优化算法（DEACO） [55]，以及人工免疫系统（AIS） [56]等。局部搜索算法是在解空间中候选解的小范围内搜索，直到找到局部最优解或者达到限制时间。一般在局部搜索算法中，每一个候选解都有多个邻域解；向哪一个邻域解移动取决于当前解邻域中的解集的信息。Noman等人提出了一个基于自适应局部搜

索算子的交叉算子从而来进一步提高传统差分进化的性能 [57]。Yang等人提出的基于邻域搜索的差分进化（NSDE） [58]被认为是对于演化编程协助的主要策略之一。NSDE通过向每个目标向量的成分添加一个服从正态分布的随机数来执行变异操作：

$$V_{i,G} = X_{r^i{}_1,G} + \begin{cases} d_{i,G} \cdot N(0.5, 0.5) & \text{如果} \quad rand_i(0,1) < 0.5 \\ d_{i,G} \cdot \delta & \text{其他} \end{cases} \tag{2.14}$$

$d_{i,G} = X_{r^i{}_2,G} - X_{r^i{}_3,G}$，$\delta$表示一个尺度参数$t$为1的柯西随机变量。Yang 等人后续又提出了基于协同演化框架的自适应的基于邻域搜索的差分进化，这一算法能够解决大规模不可分问题（达到1000维以上）。

**针对约束优化的差分进化算法：**现实生活中优化问题不仅涉及找到最优解，同时需要满足一个或者多个约束条件。边界限制在现实世界中是非常常见的，因为参数是和物理元素或者具有具有自然限制有关。为了处理这些边界限制，通过利用惩罚方法来根据目标函数的相关标准从而在有限的区域内寻找最优解。差分进化主要通过四种惩罚方法来处理违反边界限制，包括：1）砖墙惩罚 [47]：如果向量的任何参数草果事先定义的上限或者下限，那么目标函数值则会变得足够大从而保证个体不会被选择；2）自适应惩罚 [59]：和砖墙惩罚类似，不过个体目标函数的增长幅度取决于向量超过边界的程度；3）随机重新初始化 [60]：当参数超过界限的时候，从可行区间内重新挑选一个值。

Storn首次将差分进化加以拓展用来处理非等式约束优化问题，他提出了一种多成分差分进化算法CADE [61]对每一个个体产生$M$个儿子，接着在$M+1$个个体中只会有一个个体会存活到下一代。Zhang等人将动态随机分级和多成分差分进化框架相结合 [62]，这个算法在CEC 2006 的约束优化中的22个测试集上都取得了好的表现。

Lampinen通过使用在约束空间中的帕累托优势来将差分进化处理约束问题 [63]。Kukkonen 和Lampinen提出一种广义基于差分进化的算法来解决多目标约束优化问题 [64]。另外一些研究者也将差分进化中的参数控制作为约束优化。Brest 等人提出一种对于差分进化中交叉和变异算子的参数的自适应参数控制。

**针对多目标优化问题的差分进化算法：**得益于差分进化的简单但却强大的特点，差分进化同时也得到了多目标研究学者的广泛关注。Chang等人第一次将差分进化拓展用于解决多目标优化问题 [65]。Abbas和Sarkar提出了帕累托差分进化算法（PDE）通过连续变量来解决多目标优化问题 [66]，并且这个算法在和其它多目标优化算法相比具有很大的竞争力。Kukkonen 和Lampinen通过将DE/rand/a/bin进行拓展用于解决多目标优化

问题，提出广义差分进化（GDE） [67]。接着，他们也提出了一系列的改进算法。Xue等人提出了一个多目标差分进化 [68]（MODE），利用最好的个体来产生后代。通过基于Pareto的方法来挑选最好的个体。同时，还使用了$(\mu + \lambda)$选择算子，Pareto 分级以及聚集距离来产生新解并维持其良好的分布。Robic和Filipic提出了一种另外一种对于多目标优化问题的差分进化DEMO [69]，这个算法结合差分进化的优点以及基于Pareto 分级和聚集距离排序的机制。DEMO仅维持一个种群并且当产生新解的时候这个种群会被进行拓展。这能够确保种群向Pareto front 快速收敛，然而非劣排序和聚集距离的使用造成了解集的均匀扩撒。Iorio等人提出非劣排序差分进化算法（NSDE） [70]，这个算法是基于NSGA-II做出了一些改动。这两个算法的主要区别时产生新的个体的方式。NSGA-II使用实数编码交叉算子和变异算子，而NSDE则是使用差分进化中的交叉算子和变异算子。NSDE在解决旋转的多目标优化问题上要优于NSGA-II。

有一些研究者也一处采用非Pareto的方法来解决多目标优化问题。Babu等人提出一个结合两种函数的多目标差分进化算法 [71]。这个算法通过结合两种函数来构建新的机制来解决双目标问题，这两种函数包括约束函数和聚集函数。Li等人提出基于分解的多目标差分进化算法（MOEA/D-DE） [72]用于解决连续多目标问题。值得注意的是，MOEA/D是一个得到广泛应用的多目标优化算法，该方法得到研究者的广泛应用。在MOEA/D-DE之中，DE/rand/1/bin倍用于产生新的实验向量，并且所有子问题的邻域关系也会被定义从而这些子问题具有相似的最优解。

# 3 对于全局单目标优化问题的研究

在本章中，对于全局单目标优化问题，在基于DE/EDA算法框架的基础上，通过结合基于特征向量的差分进化和分布估计算法的建模来进行采样，并且通过expensive LS进一步提升算法性能。提出了基于差分进化采样策略的单目标分布估计算法EDA/DE-EIG。通过综合的实验分析，EDA/DE-EIG对于全局单目标优化问题具有较大潜力。

3.1节简要地介绍了全局单目标优化问题； 3.2节主要介绍了基于差分进化的采样策略的分布估计算法以及其相应的背景知识； 3.3节主要介绍了实验设置以及需要比较的算法，通过综合的实验分析比较，最终得出对于基于差分进化的采样的采样策略算法性能的综合评价； 3.4节做出了对本章的小节。

## 3.1 全局单目标优化问题

单目标优化问题基础出现在科学和工程应用的各个领域中。单目标优化问题的目标函数一般是非凸函数并且在可行区域内具有很多的局部极小值或者极大值。本文研究的单目标优化问题针对的是连续空间的全局优化问题，一般即是求目标函数的最小值或者最大值。全局优化问题是应用数学和数值分析的一个分支，用于解决在一定条件下求得一个或者一组函数的最优解 [73]。一般来说，对这些函数会有一系列的限制，并且这些决策变量会会依据这些限制做优化。全局优化问题和普通优化问题的区别是其是在所有的输入值中寻找最大值或者最小值，而不是寻找局部的最大值或者最小值。同样，对于全局优化问题在本文中做出以下定义：

$$\begin{aligned} &minf(x) \\ &s.t.x \in [a_i, b_i]^n \end{aligned} \tag{3.1}$$

其中$x = (x_1, x_2, \cdots, x_n)^T \in R^n$ 是决策变量向量，$[a_i, b_i]^n$是搜索空间区域，$f : R^n \to R$ 则是目标函数。

## 3.2 基于差分进化的采样策略的单目标分布估计算法

分布估计算法通过从建立的概率模型中采样从而产生新的子代种群。因此，通过结合差分进化中的建模采样和差分进化，毫无疑问能够给分布估计算法的采样带来提升。本章

节基于DE/EDA 框架，将DE-EIG和建模采样想结合，来改进分布估计算法中采样环节，同时利用expensive LS 来进一步提升算法性能。

### 3.2.1 新的交叉策略

DE-EIG是一种基于特征向量的差分进化算法 [74]，其主要贡献是在一个旋转的坐标空间对于个体进行交叉操作，这样可以利用旋转空间中种群的协方差矩阵的特征向量信息。为了避免失去种群的多样性，通过在原始的坐标空间中产生子代种群或者在旋转的坐标空间中产生子代种群的概率是随机的。通过设置合理的参数来控制种群是在原始的坐标空间中产生亦或是在旋转后的坐标空间中产生，这样既可以增加种群的多样性，也可以避免过早收敛。

DE-EIG最主要的贡献就是提出一种在旋转的坐标空间中对种群个体进行交叉操作。这样可以有效地引导种群向全局最优演化，同时也不会失去差分进化的搜索能力。在二项式交叉操作中，CR控制种群个体中变量变化的数量。当CR=1的时候，经典的差分进化的性能就和坐标空间没有关系了 [42]。相反，如果CR的值小于1 的时候，差分进化的性能就和坐标空间的旋转有着紧密的联系。

### 3.2.2 DE-EIG算法框架

DE-EIG算法是基于上文提出的新的交叉策略提出的一种新的差分进化。ED-EIG和经典的差分进化的主要区别是使用基于特征向量的交叉策略。同时为了保证种群的多样性，DE-EIG是将基于特征向量的交叉策略和经典的交叉策略相结合。这样在保证解快速收敛的同时，也保证了种群的多样性，避免种群的过早收敛。

参数$p$是用来DE-EIG是在原始坐标空间或者旋转的坐标空间做交叉操作。$rand()$是服从均匀分布的0到1之间的小数。通过转化种群的坐标空间，可以避免种群统计信息的丢失，加速种群的收敛。

### 3.2.3 DE/EDA算法

差分进化是一种非常成功的用于解决全局连续问题优化的算法。它主要是利用当前种群中的距离和方向信息来指导未来的搜索。分布估计算法则是通过从建立的概率模型中采样来产生新的种群个体。DE/EDA 就是将差分进化和分布估计算法结合起来，用于解决全局连读问题优化。通过利用分布估计算法可以提取种群全局信息和差分进化可以

**算法 3:** DE-EIG

初始化种群$Pop(t) = \{x_1, x_2, x_3, \cdots, x_N\}$ ($N$是种群大小)
**while** *not terminate* **do**
  $v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G})$
  **if** $rand() < p$ **then**
    **if** $rand() \leq CR$ **then**
      $u_{i,G} = v_{i,G}$
    **else**
      $u_{i,G} = x_{i,G}$
    **end**
  **else**
    求得$x_{i,G}$的特征向量矩阵$E$，令$E'$为特征向量矩阵的逆矩阵。
    $x'_{i,G} = E' \cdot x_{i,G}$
    $v'_{i,G} = E' \cdot v_{i,G}$
    **if** $rand() \leq CR$ **then**
      $u'_{i,G} = v'_{i,G}$
    **else**
      $u'_{i,G} = x'_{i,G}$
    **end**
    $u_{i,G} = E \cdot u'_{i,G}$
  **end**
  **if** $f(u_{i,G}) \leq f(x_{i,G})$ **then**
    $x_{i,G+1} = u_{i,G}$
  **else**
    $x_{i,G+1} = x_{i,G}$
  **end**
  $t = t + 1$
**end**

提取种群差分信息的优点，DE/EDA 是一种非常具有研究前景的算法。本章节将会基于DE/EDA这一算法框架作进一步的研究。

**算法 4:** DE/EDA

在可行的搜索空间内构建一个随机种群$Pop(t)$
**while** *not terminate* **do**
  构建概率模型：
  $p_k(x) = \prod_{i=1}^{n} \mathcal{N}(x_i; \mu_i, \sigma_i)$
  对于所有$j = 1, 2, \cdots, n$，产生一个试验向量$u = (u_1, u_2, \cdots, u_n)$
  **if** $rand() < CRP$ **then**
    $u_j = \frac{(x_i)_j + (x_d)_j}{2} + F \cdot [(x_d)_j - (x_i)_j + (x_b)_j - (x_c)_j]$
  **else**
    $u_j$根据$\mathcal{N}(x_i; \mu_i, \sigma_i)$采样产生
  **end**
  **if** $f(u) < f(x_i)$ **then**
    $x_i^{t+1} = u$
  **else**
    $x_i^{t+1} = x_i^t$
  **end**
  $t = t + 1$
**end**

### 3.2.4 复杂的局部搜索

复杂的局部搜索（expensive LS）是EDA/LS [75]中一种区别于cheap LS的一种演化策略。演化算法通常对于已经比较良好的解一般难以进一步优化，特别是在演化过程的后面的阶段。因此，通过利用expensive LS来对于已经收敛的解进行进一步的优化也显得尤为必要。

问题的关键是如何判断一个解是否收敛，通过函数$Converge(\theta, t, t_e)$来判断解是否收敛。令

$$\Delta f = \frac{|f_{t-50}^1 - f_t^1|}{max\{|f_{t-50}^1|, |f_t^1|\} + \varepsilon} \tag{3.2}$$

$$\Delta x = \frac{|c_{t-50} - c_t|}{max\{c_t, c_{t-50}\} + \varepsilon} \tag{3.3}$$

在这，$\Delta f$表示在近50代中最好的评价值的比率的降低，$f_t^1 = min_{x \in pop_t}$ 是在代数t时最好的目标函数值。$\Delta x$表示在近50代中种群覆盖区域的变化比率，$c_t = \frac{1}{n} \sum^{n}_{i=1} (max_{x \in pop} - min_{x \in pop})$，$\varepsilon = 1.0 \times 10^{-50}$。

通过将$min\{\Delta f, \Delta x\}$和给定的阈值$\theta$进行比较，皆可以判断解是否收敛。另外，为了提高算法的运行效率，每两次expensive LS至少间隔50代。

一旦种群被认为是收敛的，就是用改进的鲍威尔方法 [76]来进一步优化。鲍威尔方法是一种基于置信域的优化方法。它不使用任何的梯度信息，通过估计海塞矩阵来定位局部最优解。

### 3.2.5 EDA/DE-EIG算法

基于DE/EDA框架，将DE-EIG算法引入到之中，来进一步改善分布估计算法的采样环节。同时为了保持种群的多样性，通过一个随机参数来控制种群是通过概率模型采样产生亦或是通过DE-EIG来产生。这样，可以增加种群的多样性，同时也可以提高差分进化的搜索能力。同时，expensive local search 也将会被用于进一步提升算法求解的性能。

**算法 5:** EDA/DE-EIG

初始化种群$Pop(t) = \{x_1, x_2, x_3, \cdots, x_N\}$ ($N$是种群的大小)
**while** *not terminate* **do**
    构建概率模型：
    $p(x) = \prod_{i=1}^{n} \mathcal{N}(x_i; \mu_i, \sigma_i)$
    根据下面的流程产生新的实验向量$u_{i,G}$：
    **if** $rand() < CRP$ **then**
        根据DE-EIG得到$u_{i,G}$
    **else**
        根据概率模型$p(x)$采样得到$u_{i,G}$
    **end**
    **if** $f(u_{i,G}) < f(x_{i,G})$ **then**
        $x_{i,G+1} = u_{i,G}$
    **else**
        $x_{i,G+1} = x_{i,G}$
    **end**
    **if** *Coverage*$(\theta, G, G_e)$ **then**
        执行expensive LS
    **end**
    $t = t + 1$
**end**

## 3.3 实验分析

在本章节，我们将EDA/DE-EIG和JADE [51]和 [15]相比较。JADE的源代码是从其作者手中获得，DE/EDA是由我们自己实现的。本章节还会介绍后面的测试题以及算法的参

数设置。对于算法性能的比较会做一个综合全面的分析。

### 3.3.1 比较算法

JADE是一种自适应的差分进化算法，它通过使用一种新型的变异策略"DE/current-to$p$best"以及额外的空间来更新控制参数。JADE 和多个一流算法比较，都比较具有优势。DE/EDA是一种混合算法，结合了差分进化和分布估计算法的思想。这两个算法将会和EDA/DE-EIG在同样的测试题上进行比较。

### 3.3.2 测试集

所有的算法都会根据其在YYL测试集 [77]中的前13个测试集上的性能进行比较。所有测试集的全局最优解都是0。所有的测试集可以被分为4类：f1-f5是单峰函数；f6是阶梯函数；f7 是具有白噪音的函数；f8-f13是具有局部最优解的多模函数。因此，这些测试集能够全面地反映算法的性能。

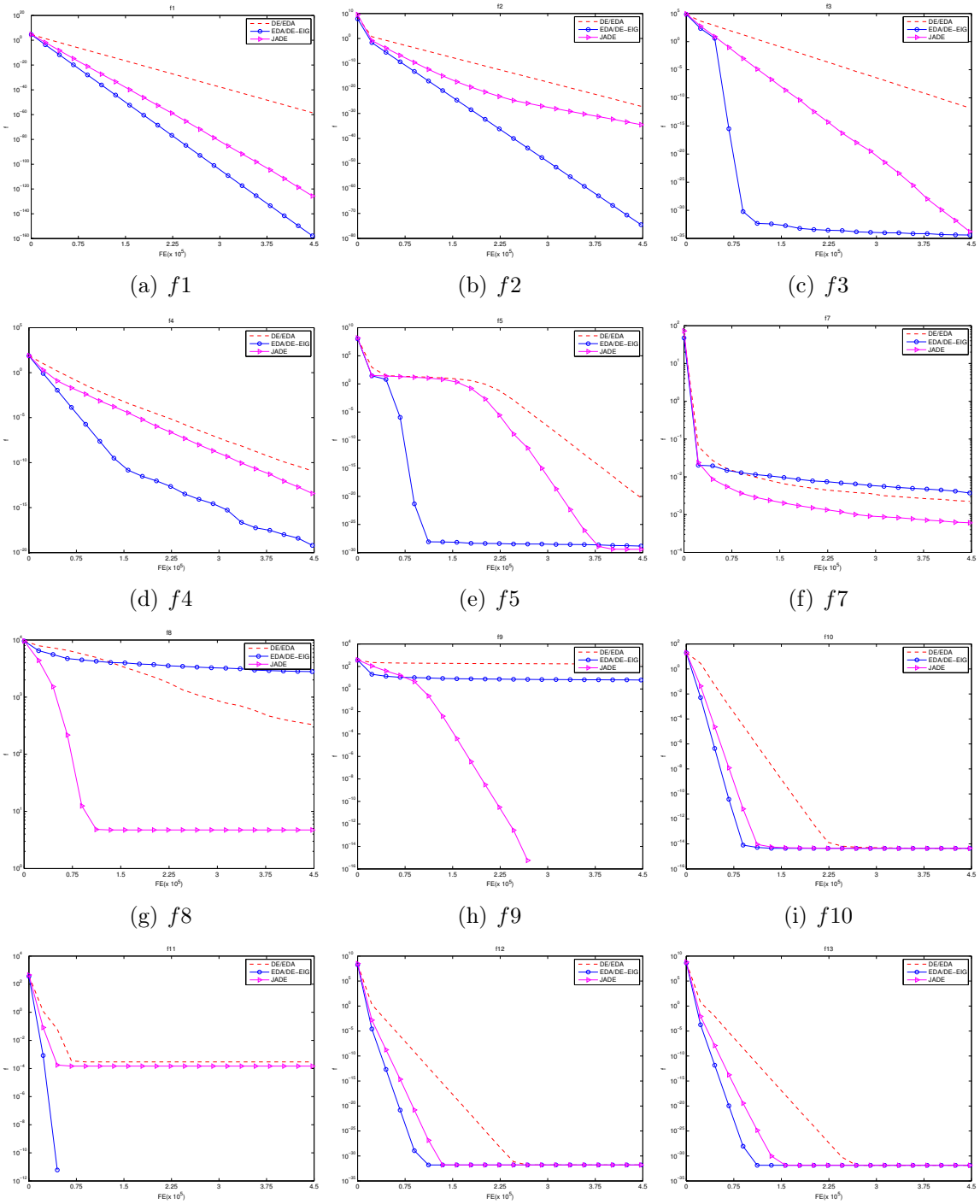
### 3.3.3 参数设置

为了公平地比较这几个算法的性能，参数设置将会根据其相应论文里面的参数。所有的算法都是通过Matlab来实现的，并且是在同一台计算机上运行。试验参数设置如下：

1. 测试集中所有种群维度都设置为30.所有算法都会在每一个测试题上独立运行50 次，体积条件是450000函数评估。
2. JADE：参数设置为：$N = 150, p = 0.05, c = 0.1, F = 0.5$ and $CR = 0.9$
3. DE/EDA: $N = 150, F = 0.5$ and $CRP = 0.9$。
4. EDA/DE-EIG: $CRP = 0.5, F = 0.5, CR = 0.6, \theta = 0.1$；控制坐标旋转的参数$p$ 设置为0.5；种群的大小$N$设置为150。对于expensive LS的相关参数的设置，与EDA/LS中的参数设置相同。

### 3.3.4 实验分析

表 1统计了各个算法的最终结果的均值和标准差。通过利用威尔克松统计实验来计较EDA/DE-EIG和其他算法。"+"，"-"以及"∼"分别表示其他算法的最终解大于，小于以及近似于EDA/DE-EIG的最终解。




(a) $f1$ (b) $f2$ (c) $f3$

(d) $f4$ (e) $f5$ (f) $f7$

(g) $f8$ (h) $f9$ (i) $f10$

(j) $f11$ (k) $f12$ (l) $f13$

**图** 4: 三个算法在$f1 - f13$中除了$f6$的12个测试集上目标函数平均值的折线图

**表** 1: 三个算法在测试题$f1-f13$上对于（平均值±标准差）的统计结果

| | EDA/DE-EIG | JADE | DE/EDA |
|---|---|---|---|
| $f1$ | **1.54e-159 ± 5.11e-159** | $3.90e-127 \pm 2.74e-126(+)$ | $1.39e-59 \pm 2.58e-59(+)$ |
| $f2$ | **1.02e-75 ± 7.46e-76** | $2.60e-35 \pm 1.64e-34(+)$ | $5.15e-28 \pm 4.68e-28(+)$ |
| $f3$ | **4.01e-35 ± 8.47e-35** | $7.79e-35 \pm 2.51e-34(\sim)$ | $1.23e-12 \pm 1.20e-12(+)$ |
| $f4$ | **5.01e-20 ± 3.06e-19** | $3.15e-14 \pm 6.42e-14(+)$ | $9.90e-12 \pm 2.69e-11(+)$ |
| $f5$ | $1.46e-29 \pm 2.62e-29$ | **3.85e-30 ± 9.58e-30**$(-)$ | $3.37e-21 \pm 8.66e-21(+)$ |
| $f6$ | **0.00e+00 ± 0.00e+00** | **0.00e+00 ± 0.00e+00**$(\sim)$ | **0.00e+00 ± 0.00e+00**$(\sim)$ |
| $f7$ | $3.60e-03 \pm 1.00e-03$ | **6.01e-04 ± 2.23e-04**$(-)$ | $2.20e-03 \pm 5.59e-04(-)$ |
| $f8$ | $2.79e+03 \pm 5.02e+02$ | **4.74e+00 ± 2.34e+01**$(-)$ | $1.82e+03 \pm 6.72e+02(-)$ |
| $f9$ | $6.23e+00 \pm 2.21e+00$ | **0.00e+00 ± 0.00e+00**$(-)$ | $1.54e+02 \pm 1.96e+01(+)$ |
| $f10$ | **4.44e-15 ± 0.00e+00** | **4.44e-15 ± 0.00e+00**$(\sim)$ | **4.44e-15 ± 0.00e+00**$(\sim)$ |
| $f11$ | **0.00e+00 ± 0.00e+00** | $1.48e-04 \pm 1.05e-03(\sim)$ | $2.96e-04 \pm 1.46e-03(\sim)$ |
| $f12$ | **1.57e-32 ± 5.53e-48** | **1.57e-32 ± 5.53e-48**$(\sim)$ | **1.57e-32 ± 5.53e-48**$(\sim)$ |
| $f13$ | **1.35e-32 ± 1.11e-47** | **1.35e-32 ± 1.11e-47**$(\sim)$ | **1.35e-32 ± 1.11e-47**$(\sim)$ |
| | | 3(+)6(∼)4(−) | 6(+)5(∼)2(−) |

[1] 粗体的表示更好的

从表 1可以看出，EDA/DE-EIG在和其它两个算法比较具有不错的表现。EDA/DE-EIG在9个测试题上取得了最好的表现，除了f5，f7，f8和f9。接着对EDA/DE-EIG和其它两个算法分别做进一步的比较。对于DE/EDA，EDA/DE-EIG具有相当不俗的表现。EDA/DE-EIG 在7个测试机上都要优于DE/EDA。EDA/DE-EIG 在测试集上的提升是非常显著的，值得注意的是，这两个算法之间的性能差异还是比较显著的。

将EDA/DE-EIG和JADE相比较，EDA/DE-EIG在5个测试集上取得了更好的结果。这两个算法在测试集f6，f10，f12以及f13上都取得了最好的结果。总而言之，EDA/DE-EIG的性能表现还是比较具有竞争力的。值得注意的是，在f1-f4以及f11这5个测试题上，EDA/DE-EIG 都具有显著的性能提升。对于测试集f7，f8以及f9，EDA/DE-EIG可能是受到差分进化的局部最优地影响。在这些测试题上的提升，EDA/DE-EIF值得更加深入的研究。

同时，为了从一个更加客观的角度来比较最终的结果，威尔克松统计分析被用来比较这几个算法。对于JADE，EDA/DE-EIG在3个测试题上要优于JADE；这两个算法在6个测试集上取得了近似的结果；在4 个测试题上，JADE的最终结果更好。在与DE/EDA相比，EDA/DE-EIG有着不小的提升。同时，对于JADE，这两个算法从统计上来说在13个测试题上有着近似的表现。这表明DE-EIG的确取得了有意义的效果。这证明EDA/DE-EIG的性能表现还是具有潜力的同时也需要进一步的提升研究。

为了进一步诠释EDA/DE-EIG和其他两个算法之间的性能比较，图 4是这三个算法结果的趋势图。因为在测试集f6上，JADE和EDA/DE-EIG都收敛得非常早，为了更好地站线结果，在本章节就不做在f6上的趋势图比较。在13个测试题中，EDA/DE-EIG在8个测试题上取得了最好的结果。EDA/DE-EIG在收敛速度和最终的收敛结果上都取得了较好的结果。对于某些测试集，这个比较还是比较显著的，包括f1，f2，f3，f4和f11。对于DE/EDA而言，在11个测试题上，ED/EDA都表现得更差，除了测试题f7和f8。与JADE相比较，对于测试集f1-f4，EDA/DE-EIG在收敛速度和最终收敛结果都是领先的。对于测试集f5,JADE最终结果更好，但是EDA/DE-EIG具有更好地收敛趋势并且更早地收敛。对于测试集f7，f8和f9上，EDA/DE-EIG的表现不佳。在测试题f10，f12和f13上，EDA/DE-EIG 具有一个更好地收敛速度。总之，EDA/DE-EIG在与DE/EDA相比还是具有比较明显的优势。同时，和JADE相比较，EDA/DE-EIG 的表现还是具有相当的竞争力。

## 3.4 本章小结

DE/EDA是一种对于全局优化的具有很大潜力的方法，它结合了分布估计算法的全局建模信息和差分进化的局部搜索信息。在本章节中，通过引进一种基于特征向量的差分进化算法，DE-EIG。DE-EIG有利于更好地利用种群的统计信息并且加速收敛。同时，通过结合expensive LS来进一步提炼演化的解集，并且通过随机参数的设置，从而保证种群的多样性。综上，我们提出了一种基于差分进化的分布估计算法EDA/DE-EIG。实验结果表明，在和JADE 和DE/EDA 比较的过程中，EDA/DE-EIG 还是比较具有一定的优势，在解决单目标全局优化问题具有一定的竞争力。

当然，本章节的研究是非常基础的还有需要提升的地方。第一，EDA/DE-EIG的算法结构值得进一步的提炼和优化；第二，值得探究在DE-EIG 和EDA算法资源的分配，这涉及到演化计算中的利用和开发，是一个非常值得研究的话题。

# 4 对于连续多目标优化问题的研究

在本章中，我们基于RM-MEDA算法提出了一种基于差分进化的采样策略来解决连续多目标问题。通过采取差分进化中的变异策略，将种群在隐空间中进行转化，从而产生新的种群。

3.1节给出了连续多目标问题的相关背景知识，3.2节主要介绍RM-MEDA算法框架以及相应的背景知识，3.3节详细介绍基于差分进化的采样策略，3.4节主要讲述通过基于差分进化的采样策略来改进RM-MEDA 算法，3.5节是实验结果的分析。

## 4.1 连续多目标问题

为了不失一般性，在本文中，在本文中我们假设多目标优化问题中的每个问题都是最小化问题，则多目标优化问题可以由以下数学公式表达：

$$\begin{aligned} \min \quad & F(x) = (f_1(x), \cdots, f_m(x)) \\ \text{s.t} \quad & x \in \Omega \end{aligned} \tag{4.1}$$

其中，$x = (x_1, \cdots, x_n)^T \in R^n$ 是决策变量向量，$\Omega = \Pi_{i=1}^{n}[a_i, b_i] \subset R^n$ 表示可能的搜索空间区域，$f_i : R^n \to R, i = 1, \cdots, m$ 是一个连续的目标函数，$F(x)$则是相应的目标函数向量。

在多目标优化问题中，多个目标相互之间往往是冲突的，从而导致无法在满足所有约束条件下使得所有目标函数都能够达到全局最优解，但是存在一组Pareto最优解 [78]。对于此，做出以下定义：令$a, b \in R^n$，当$a_i \leq b_i \wedge a \neq b$，且$i = 1, \cdots, n$，则称$a$ 支配$b$。向量$x^* \in \Omega$ 即是公式 4.1*Pareto* 最优解，如果不存在$x \in \Omega$使得$F(X)$ 支配$F(x^*)$。$F(x^*)$被称为Pareto最优目标向量。所有的Pareto 最优解的集合就是Pareto最优解集（PS），对应的最优向量的集合则成为Pareto前端（PF）。

多目标问题在很多工程领域都普遍存在，通常来说，多目标问题中的各个目标之间往往都是相互冲突的在一般条件下，根据Karush-Kuhn-Tucker 可以推导出：连续多目标问题在决策空间中的Pareto set 是一个连续分段的（m-1）维的流形体（m是目标数）。对于一个成功的多目标演化算法（multiobjective estimation of distribution algorithm, MOEA）来说，独立的个体应该是在决策空间中分散在Pareto set附近，如图 5所示。

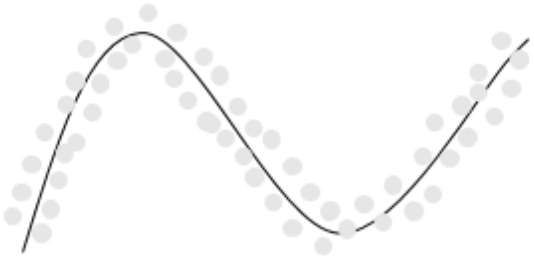


**图** 5: 连续多目标问题决策空间中个体的分布情况

## 4.2 基于差分进化的采样策略的多目标分布估计算法

根据连续多目标问题以上的特性，基于规律模型的多目标分布算法（RM-MEDA）算法被提出用于解决连续多目标问题。其通过在每一次迭代中，通过局部主成分分析（LPCA） [79]在决策空间中的区域建立概率分布模型，然后通过使用拉丁实验设计采样得到新的子代种群。RM-MEDA 采用基于非劣排序的方法 [5] 来挑选个体来产生新的种群。RM-MEDA 经提出后，就收到了广泛的关注。本小节基于RM-MEDA算法，提出了一种基于差分进化的采样策略，用于进一步提升RM-MEDA 中采样环节。

### 4.2.1 RM-MEDA算法

对于连续多目标问题，在决策空间中，种群体中的个体如果越接近Pareto set，则越容易进行问题的求解。因此，假设种群中的个体为随机向量$\xi \in R^D$ 的观测值，$\xi$的中央部分就是Pareto set。并且因为在连续多目标问题中，Pareto set是一个m-1维的流体，那么$\xi$则可以由公式 4.2表示：

$$\xi = \zeta + \epsilon \tag{4.2}$$

$\zeta$相当于是均匀分布在m-1维流体附近的个体，$\epsilon$是均值为0的n维的噪音向量。

RM-MEDA算法是基于分布估计算法的框架，主要也是包括建模、采样、选择三个环节。算法 6则是RM-MEDA的主要的算法框架。

建模是分布估计算法中关键环节，在RM-MEDA中，如何找到种群中的个体在决策空间中的中心曲线或曲面是建模的关键。在此假设$\xi$由$k$个流体$M^1, \cdots, M^k$ 组成，每一个流体是一个超矩形。一般来说，对于双目标，每一个流体$M^j$是一个线段，对于三个目标，每一个流体$M^j$是一个二维矩形。建模的重点就是估计流体$M^j$以及流体的主要组成部分$\zeta$。为了更好地描述这些流体，通过LPCA 将种群划分为$k$个聚类：$C_1, \cdots, \mathbb{C}_k$。

**算法 6:** RM-MEDA 算法框架

初始化一个随机种群$Pop(0)$，并且设置$t=0$。
**while** 没有达到停机条件 **do**
  **建模**：建立一个概率模型$\xi$来表示在随机种群$Pop(t)$中的个体。
  **采样**：通过上述的概率模型进行采样得到新的解集$Q$。
  **选择**：从$Q \bigcup Pop(t)$中挑选出$N$个个体来组成一个新的种群$Pop(t+1)$。
  $t=t+1$
**end**

令$S$为$R^n$ 的有限子集，则定义子集$C$ 的均值为：

$$\overline{x}=\frac{1}{|C|}\sum_{x\in C}x \tag{4.3}$$

接着求得集合$S$里面个体的协方差矩阵：

$$Cov=\frac{1}{|C|-1}\sum_{x\in C}(x-\overline{x})(x-\overline{x})^T \tag{4.4}$$

第$i$个主成分$U^i$是协方差矩阵$Cov$中第$i$个最大的特征值对应的特征向量。接着，$C$ 中个体的$(m-1)$维的主子空间仿射可以定义为：

$$\left\{x\in R^n, x=\overline{x}+\sum_{i=1}^{m-1}\theta_i\cdot U^i, \theta_i\in R, i=1,\cdots,m-1\right\} \tag{4.5}$$

LPCA通过最小化以下误差函数的最小值来将种群划分聚类：

$$\sum_{j=1}^{k}\sum_{x\in C^j}d(x,A_j^{m-1})^2 \tag{4.6}$$

其中$A_j^{m-1}$是聚类个体的$(m-1)$维主子空间仿射，$d(x,A_j^{m-1})$是$\overline{x}$ 和它在$A_j^{m-1}$上投影的欧式距离。

图 [79]形象地表示了LPCA划分聚类的过程。开始的时候，种群中的个体随机地分布，通过LPCA对种群中的个体进行聚类划分，直到每个聚类中的个体不会在发生变化，聚类划分就完成了。

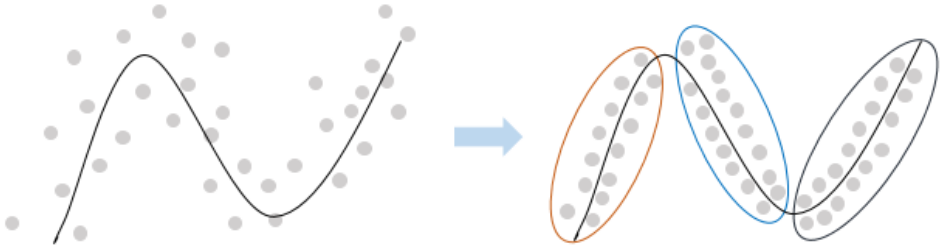


图 6: LPCA划分聚类过程

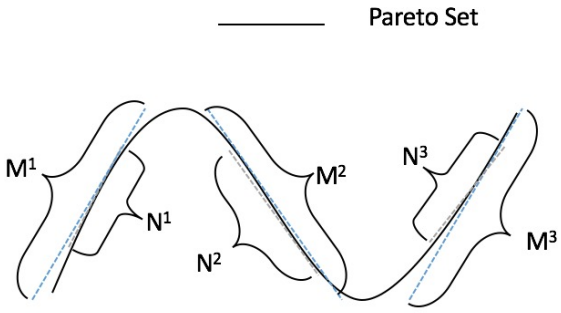



图 7: 通过缩放比例来覆盖Pareto set

### 4.2.2 RM-MEDA中存在的问题和解决方法

在RM-MEDA中，通过使用Local PCA将种群分成k个聚类。如图 7所示，在每一个聚类之中，$N^k$用来表示聚类中的Pareto set，而$M^k$则用来覆盖每个聚类中的Pareto set。为了能够覆盖聚类中的Pareto set，RM-MEDA通过设置一个缩放比例用来覆盖聚类中的Pareto set。但是这个缩放比例依赖于具体的问题，对于不同的问题，其表现也不禁相同。如果缩放比例设置过大，则对于实际的Pareto set则显得多余；如果设置的缩放比例过小，又不足以覆盖实际的Pareto set。因此，如何设置一个合适的缩放比例也成为了一个比较困难的问题。

### 4.2.3 基于差分进化的采样策略

为了避免在RM-MEDA中通过设置缩放比例来进行采样，我们提出了一种新型的基于差分进化的采样策略。通过改进rand-1-bin变异策略，这个变异策略的公式如 8所示：

$$X = X_{r_1} + rand \cdot (X_{r_2} - X_{r_3}) + F \cdot (X_{r_2} - X_{r_3}) \tag{4.7}$$

其中$X_{r_1}$, $X_{r_2}$, $X_{r_3}$是种群中的随机个体，$r_1, r_2, r_3$则是从1到NP(NP是种群的大小)之

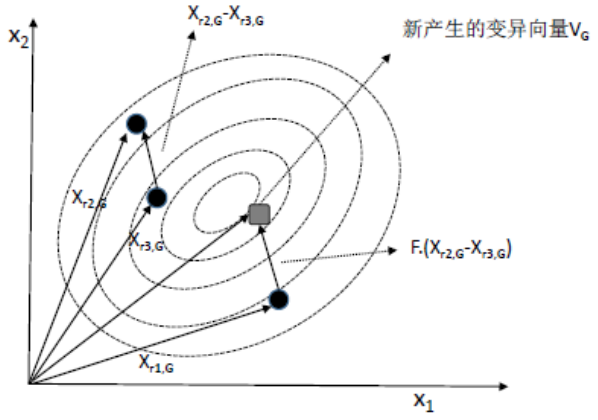



**图** 8: 通过缩放比例来覆盖Pareto set

间选择的三个互不相同的整数。F是变异策略中的缩放因子，rand是0 到1 之间符合均匀分布的随机数。

图 8诠释了在二维空间中这个变异策略是如何实现的。向量$rand \cdot (X_{r_2} - X_{r_3}) + F \cdot (X_{r_2} - X_{r_3})$对于增加种群的多样性具有重要意义。

对于种群中的每一个聚类中，DES将决策空间中的个体转化到隐空间中，通过上述变异策略产生新的个体，再将个体转换到正常的决策空间中。通过在隐空间中执行变异策略，可以有效地提取种群中的统计信息。DES的算法框架如 7所示：

将DES引入到RM-MEDA算法中来采样，那么DES-RM-MEDA算法框架如算法 8所示。在建模环节，还是基于连续多目标问题的特性，利用LPCA将种群中的个体划分为不同的聚类。同时，为了增加种群的多样性，使用基于差分进化的采样策略，在新型的隐空间中对于每个聚类中的个体进行变异操作，从而产生新的个体。这样保证收敛的同时，通过差分进化的样策略的变异算子来提高种群的多样性。

## 4.3 实验分析

### 4.3.1 实验设置

本章节主要针对ZZJ中的10个测试题做实验比较。

本章节主要采用IGD(inverted generational distance)指标 [80]来比较算法的性能。令$P^*$ 为Pareto front周围的目标向量空间中均匀分布的一组点。假设P是Pareto front的估

**算法 7:** DES

1 对于每个给定的聚类求得相应的协方差矩阵$C$并进行分解操作：

$$C = EDE^T$$

$E$是协方差矩阵$C$的特征向量矩阵，$D$是由特征值组成的对角矩阵。

2 对于聚类中每一个个体$x$，将其映射到隐空间中：

$$y = x \cdot R.$$

$R$是特征向量矩阵$E$中前$(m-1)$个主要成分。

3 在隐空间中对于种群个体进行变异操作：

$$y' = y_{r_1} + rand \cdot (y_{r_2} - y_{r_3}) + F \cdot (y_{r_2} - y_{r_3})$$

4 将$y'$映射到原始的决策空间

$$x' = y' \cdot R^T.$$

5 返回产生的新的个体

$$x'' = x' + \varepsilon'$$

6 $\varepsilon'$是一个服从分布$\mathcal{N}(0, \sigma_\tau I)$的高斯噪音（$\tau \in \{1, 2, \cdots, K\}$是一个随机产生的整数）

**算法 8:** DES-RM-MEDA

1 初始化一个随机种群$Pop(t)$，并且设置t为0。
2 **while** *not terminate* **do**
3 　**建模**：通过建立概率模型$\delta$来描述种群中个体的分布情况。
4 　**采样**：根据概率模型将种群划分为不同的聚类$C_i$。对于每一个聚类，分别应用DES来产生新的候选集合$Q_i$，最终生成集合$Q = \cup_i Q_i$。
5 　**选择**：从$Q \bigcup Pop(t)$中选择$N$个个体来构建新的种群$Pop(t+1)$。
6 　$t = t + 1$
7 **end**

计，那么$P$到$P^*$之间的IGD指标定义如下：

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{P^*} \tag{4.8}$$

其中,$d(v, P)$是$v$到$P$中任意一点的最小欧氏距离。如果$P^*$足够大到可以表示Pareto front，那么$IGD(P^*, P)$就可以用来测量$P$ 的多样性和收敛性。$IGD(P^*, P)$的值越小，则$P$距离$P^*$ 越近。

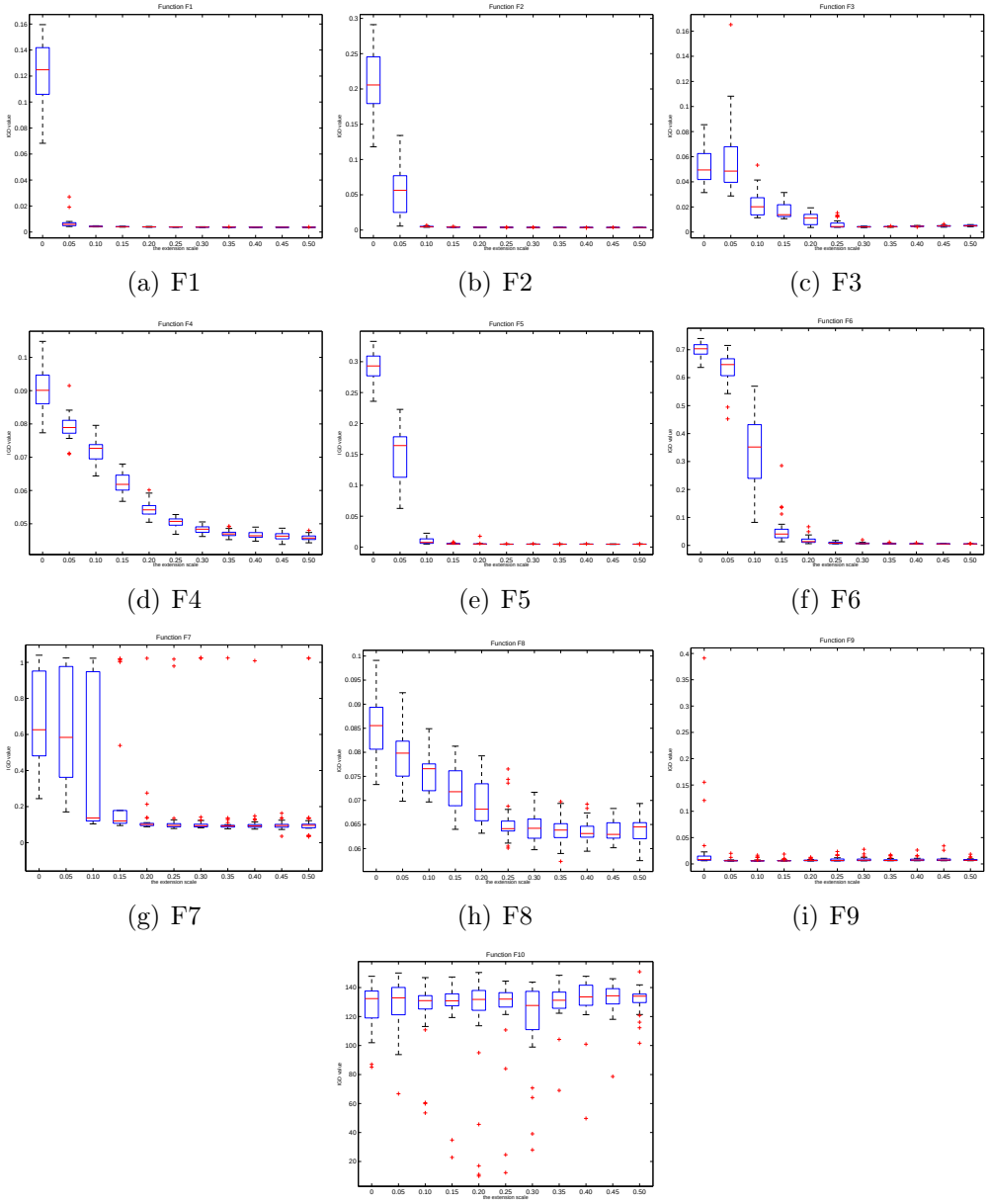
RM-MEDA和DES-RM-MEDA都是使用Matllab进行编程实现并且是在同一台计算机上运行程序。在这篇论文中的试验参数如下：

- 初始化种群：这两个算法中的初始种群是随机生成的。
- 种群大小：对于测试集（$F3$, $F7$, $F9$）每一代的种群大小设置成100，对于其它的测试题，每一代种群大小设置为200.
- 决策向量的大小: 决策向量的大小在这两个算法都设置成30。
- 聚类数量：聚类的数量设置成5。
- 缩放因子$F$：缩放因子$F$设置为0.4。
- 运行次数：对于算法中每一个测试题将单独运行30次。
- 迭代次数：对于测试题（$F1$, $F2$, $F5$, $F6$）迭代次数设置为100，对于测试题$F4$和$F8$设置为200，其它的设置为1000。p

### 4.3.2 RM-MEDA缩放因子的影响

为了研究缩放因子对于RM-MEDA性能的影响，在本章节中，从0到0.5之间以0.05为间隔设置10 个缩放因子，并比较RM-MEDA在这些缩放因子作用下的性能比较。图9是RM-MEDA在不同缩放因子的作用下，IGD指标的箱线图。

从这个箱线图中，如果不设置缩放因子，那么RM-MEDA表现则不是很好。一般来说，缩放因子设置的越大，则RM-MEDA表现得也更加优秀。但对于设置较大的缩放因子也可能会造成性能的不稳定，比如图 9(g)和(h)。总的来说，如果在实践中设置一个最佳的缩放因子还是比较困难的。



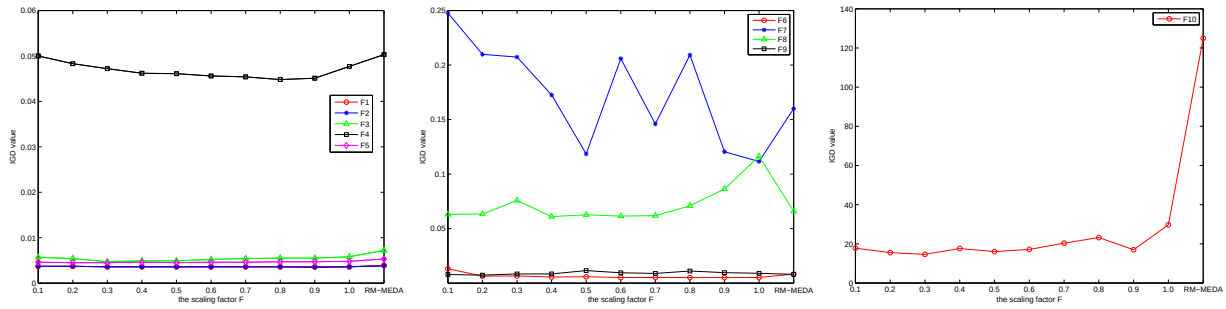

(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

(g) F7 (h) F8 (i) F9

(j) F10

**图** 9: 不同缩放因子下，RM-MEDA的IGD指标的箱线图




(a) (b) (c)

**图** 10: 不同缩放因子下，DES-RM-MEDA的性能比较

### 4.3.3 缩放因子F的敏感性

为了验证DES-RM-MEDA中缩放因子F的敏感性，在本节中，将F设置为从0到1之间，从而来比较在不同缩放因子F作用下，DES-RM-MEDA 的性能表现。

从图 10可以看出，缩放因子F的鲁棒性还是比较优秀的。对于大多数测试题，除了F7，DES-RM-MEDA的性能表现基本上都是非常平滑的。同时，将不同缩放因子下DES-RM-MEDA和RM-MEDA 进行了对比。

当缩放因子设置为0.2、0.5以及0.7的情况下，DES-RM-MEDA比RM-MEDA在9个测试题上都表现得更好。在缩放因子设置为0.1、0.4、0.6、以及0.9的时候，RM-MEDA在8个测试题上表现不如DES-RM-MEDA。在将缩放因子设置为其它参数的情况下，DES-RM-MEDA 的IGD指标要优于RM-MEDA 在7个测试题上。

对于测试题F1、F2、F3、F4以及F5，DES-RM-MEDA在使用任一缩放因子设置情况下，其性能表现都要优于RM-MEDA。除了将缩放因子设置为0.1，在测试题F6上，DES-RM-MEDA都比RM-MEDA 要表现得更好。对于F7、F8、F9，DES-RM-MEDA的优势怎没有这么多。对于测试题F10，可以看出RM-MEDA和DES-RM-MEDA 之间的差距还是比较明显的。综合来说，缩放因子的鲁棒性还是比较好的，同时对于算法性能的提升也有着重要的意义。

为了进一步比较DES-RM-MEDA在不同的缩放因子下的性能表现，我们将使用威尔科克森符号秩检验（Wilcoxon's rank sum test）来对于RM-MEDA和DES-RM-MEDA 之间的性能进行比较。层叠柱状图 11 可以表明在不同缩放因子下，DES-RM-MEDA的表现还是比较优异的。

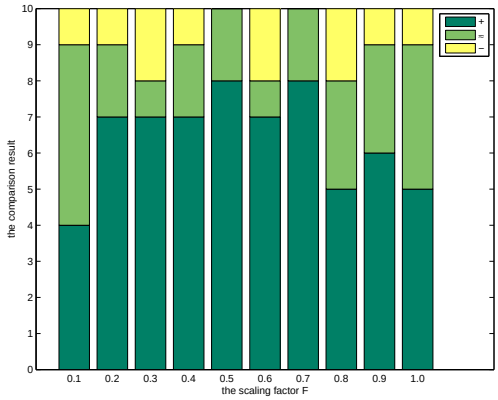



**图** 11: "+", "≈", 以及"−" 分别表示DES-RM-MEDA的性能要优于，相似于，差于RM-MEDA的性能

### 4.3.4 对比研究

从表 2可以看出，在8个测试题上，DES-RM-MEDA都要优于RM-MEDA。对于其它的2个测试题，这两个算法的性能比较接近。值得注意的是，在测试题F3 以及F10上，DES-RM-MEDA 在性能上的提升是非常显著的。

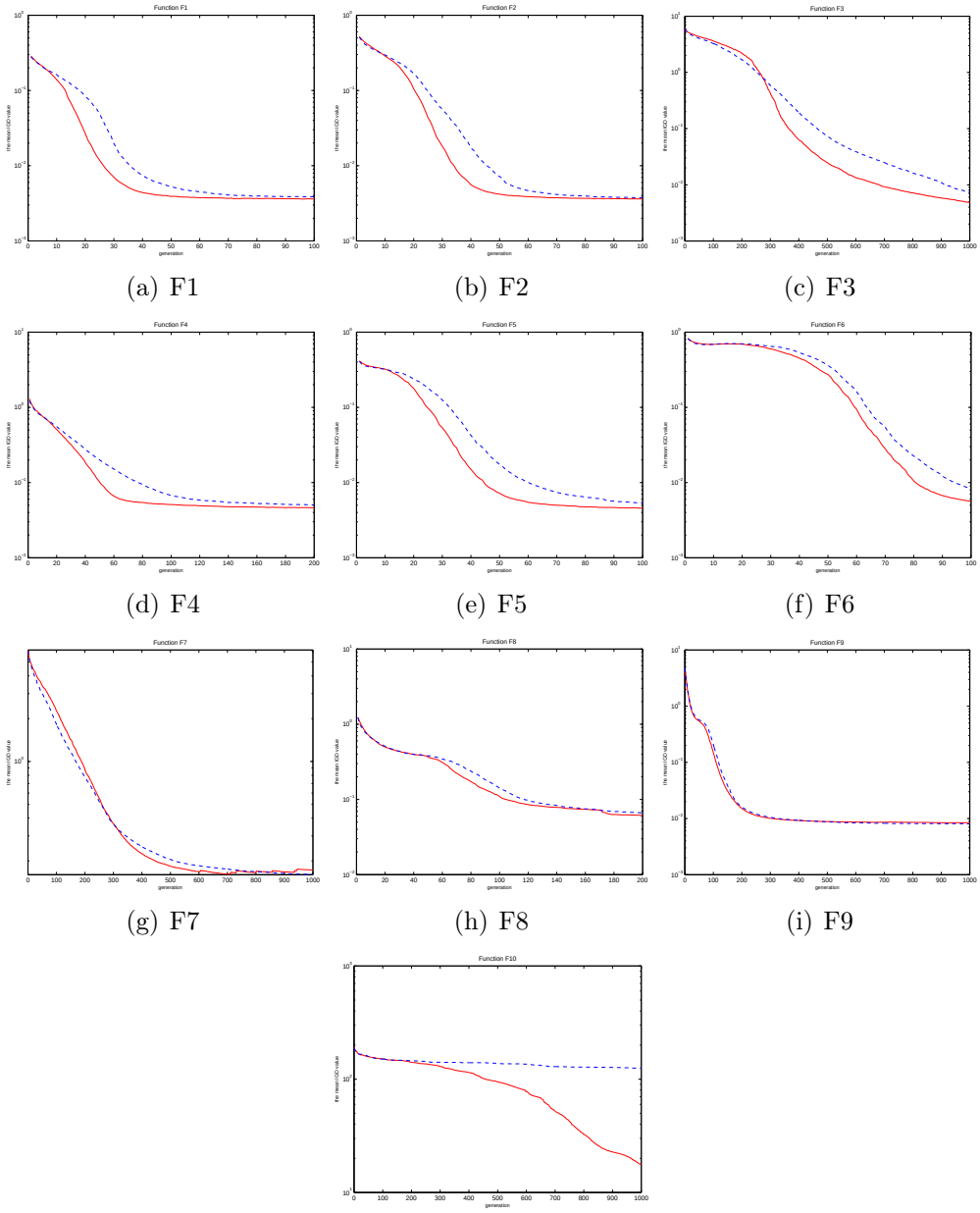
图 12可以用来比较DES-RM-MEDA和RM-MEDA在测试题上的收敛趋势。从图 12可以看出，对于大多数测试题，DES-RM-MEDA收敛的更快更好。这表示DES对于RM-MEDA采样的意义还是相当重要的。下面，对于实验结果进行深入的比较分析：

1. 对于具有两个目标的测试题，DES-RM-MEDA具有非常亮眼的表现。对于简单的测试题，比如F1和F2，以及困难的测试题，比如F3 和F10，无论是在收敛速度还是最终的收敛值，DES-RM-MEDA 都取得更好地表现。F9是一个相对比较简单的测试题，这两算法这个测试题上的表现近似。只有对于测试题F7，DES-RM-MEDA在前面的表现都不错，但是在后面的代数里面表现的不是足够稳定。

2. 对于具有三个目标的测试题，包括F4和F8。在这两个测试题上，DES-RM-MEDA的收敛速度更快，最终的收敛水平也要略优于RM-MEDA。

## 4.4 本章小结

在本章节我们提出了基于差分进化的采样策略，即在隐空间中产生新的种群。基本的思路是将父代种群映射到隐空间中，然后在隐空间中利用一种新型的变异策略来进行变异操作从而产生新的个体，接着将这些个体映射到原始的空间来产生子代种群。基于



(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

(g) F7 (h) F8 (i) F9

(j) F10

图 12: IGD指标均值趋势图。实线表示DES-RM-MEDA，虚线表示RM-MEDA。

表 2: DES-RM-MEDA和RM-MEDA的独立运行30次IGD指标

| | RM-MEDA | | | | DES-RM-MEDA | | | |
|---|---|---|---|---|---|---|---|---|
| | mean | std. | best | worst | mean | std. | best | worst |
| $F1$ | $3.90e-03$ | $1.39e-04$ | $3.70e-03$ | $4.20e-03$ | **3.60e-03** | $9.16e-05$ | **3.43e-03** | **3.77e-03** |
| $F2$ | $3.80e-03$ | $1.43e-04$ | $3.50e-03$ | $4.10e-03$ | **3.60e-03** | $1.01e-04$ | **3.35e-03** | **3.82e-03** |
| $F3$ | $7.20e-03$ | $3.90e-03$ | **3.60e-03** | $1.55e-02$ | **4.90e-03** | $8.09e-04$ | $3.90e-03$ | **7.31e-03** |
| $F4$ | $5.03e-02$ | $1.30e-03$ | $4.82e-02$ | $5.35e-02$ | **4.62e-03** | $9.32-04$ | **4.44e-02** | **4.85e-02** |
| $F5$ | $5.30e-03$ | $3.00e-03$ | $4.40e-03$ | $2.12e-02$ | **4.60e-03** | $1.49e-04$ | **4.33e-03** | **4.96e-03** |
| $F6$ | $8.30e-03$ | $2.10e-03$ | $5.70e-03$ | $1.50e-02$ | **5.60e-03** | $9.04e-04$ | **4.52e-03** | **8.30e-03** |
| $F7$ | **1.60e-01** | $2.35e-01$ | $7.96e-02$ | $1.03e+00$ | $1.73e-01$ | $2.28e-01$ | **3.20e-02** | **1.02e+00** |
| $F8$ | $6.59e-02$ | $3.50e-03$ | $6.05e-02$ | $7.69e-02$ | **6.10e-02** | $2.03e-03$ | **5.67e-02** | **6.39e-02** |
| $F9$ | **8.00e-03** | $2.80e-03$ | $5.80e-03$ | **1.48e-02** | $8.40e-03$ | $3.20e-03$ | **5.51e-03** | $2.12e-02$ |
| $F10$ | $1.25e+02$ | $2.35e+01$ | $2.27e+01$ | $1.44e+02$ | **1.76e+00** | $1.29e+01$ | **4.73e+00** | **7.15e+01** |

[1] 粗体的表示更好的结果

差分进化的采样策略被用于改进RM-MEDA的采样环节。在实验环节中，通过设置不同的缩放比例来研究RM-MEDA 在不同缩放比例下的性能表现。接着通过综合的实验对比，将DES-RM-MEDA和RM-MEDA进行了详细全面地综合对比，从而说明基于差分进化的采样策略对于提高RM-MEDA算法性能的重要意义。

这一章节的基本研究表明在隐空间中的采样还是相当具有潜力的，因为其维度大小于决策空间的维度，从而能够更好地利用种群的统计信息。同时，对于基于差分进化的采样策略依然还有别的工作需要完成，包括：(a)尝试在隐空间中采用别的采样策略。(b) 可以将基于差分进化的采样策略和别的多目标演化算法相结合。

# 5 总结与展望

## 5.1 本文工作总结

演化计算中的exploration和exploitation一直是一个比较困扰的话题。如何在演化计算中，既能在找到最优解的同时，也避免陷入局部最优解，这一直是演化算法中的一大难题。分布估计算法能够在概率模型的基础上提取全局信息从而找到最优解。差分进化则是一种利用局部搜索能力来找到最优解。本文在基于分布估计算法的基础上，通过利用差分进化来改进分布估计算法的采样过程，并分别研究在解决多目标和单目标优化问题上的性能表现。在阅读大量文献和实验探究的基础上，提出了基于差分进化的采样策略。通过实验对比，可以看出，基于差分进化的采样策略在解决演化计算中的优化问题上都有着很大的潜力，和其他的一流演化算法具有一定的可比性。下面对本文在作进一步的概括，对本文作进一步的总结：

本文首先介绍了在演化计算中单目标和多目标问题的定义，这也是本文提出算法主要解决的目标。接着，简要介绍了分布估计算法和差分进化，以及当前学术界的研究情况。

对于多目标优化问题，我们提出了一种基于差分进化的采样策略，在基于RM-MEDA这一经典的多目标分布估计算法的基础上，基于差分进化的采样策略被用于改进RM-MEDA的采样过程，提高了种群的多样性，加速了种群收敛。同时，也分析了RM-MEDA的一个问题，并提出了相应的改进。通过详细的实验对比分析，可以看出及于差分进化采样策略的优势。

对于单目标优化问题，在基于DE/EDA的算法框架上，我们希望通过利用差分进化来进一步提升分布估计算法的采样过程。通过一种新型的差分进化DE-EIG，来进一步改进采样，同时在不失去种群多样性的情况下，加快最终结果的收敛，从而提出了EDA/DE-EIG算法。通过和JADE和DE/EDA算法的实验结果分析，可以看出，差分进化对于在单目标问题上的优化具有一定的潜力。

## 5.2 工作展望

本文的工作是基于分布估计算法框架，提出了一种基于差分进化的采样策略，通过这种采样策略来改进分布估计算法中的采样过程。这项研究工作当前还是比较基础的，还有很多不完善的地方。对于未来工作的展望主要包括以下几个方面：

- 将差分进化的采样策略应用于其他的多目标演化算法，比如基于分解的多目标演化算法，从而进一步研究基于差分进化采样策略的潜力和价值。
- 进一步优化EDA/DE-EIG算法，简化算法框架。
- 对于DE/DEA算法框架中，DE和EDA资源的分配是一个有意思的话题，值得进一步地探索研究。

# 参考文献

[1] D. B. Fogel, *Evolutionary computation: toward a new philosophy of machine intelligence*. John Wiley and Sons, 2006, vol. 1.

[2] T. Bäck, D. Fogel, and Z. Michalewicz, "Handbook of evolutionary computation," *Release*, vol. 97, no. 1, p. B1, 1997.

[3] H. P. P. Schwefel, *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley and Sons, Inc., 1995.

[4] D. Simon, *Evolutionary optimization algorithms*. John Wiley and Sons, 2013.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[6] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[7] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, "Genetic programming: an introduction: on the automatic evolution of computer programs and its applications," *Genetic Programming*, vol. 19, no. 4, pp. 372–380, 1998.

[8] X. Yao and Y. Liu, "Fast evolutionary programming." in *Evolutionary Programming*, 1996, pp. 451–460.

[9] R. Storn and K. Price, *Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces*. ICSI Berkeley, 1995, vol. 3.

[10] P. Larranaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science and Business Media, 2002, vol. 2.

[11] M. Pelikan, D. E. Goldberg, and F. G. Lobo, "A survey of optimization by building and using probabilistic models," *Computational optimization and applications*, vol. 21, no. 1, pp. 5–20, 2002.

[12] A. Z. Bing Dong and G. Zhang, "Sampling in latent space for a multiobjective estimation of distribution algorithm," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016.

[13] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[14] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[15] J. Sun, Q. Zhang, and E. P. Tsang, "DE/EDA: A new evolutionary algorithm for global optimization," *Information Sciences*, vol. 169, no. 3, pp. 249–262, 2005.

[16] J. A. Lozano, *Towards a new evolutionary computation: advances on estimation of distribution algorithms*. Springer Science and Business Media, 2006, vol. 192.

[17] T.-L. Yu, S. Santarelli, and D. E. Goldberg, "Military antenna design using a simple genetic algorithm and hboa," in *Scalable Optimization via Probabilistic Modeling*. Springer, 2006, pp. 275–289.

[18] R. Shah, P. Reed, R. Shah, and P. Reed, "Comparative analysis of multiobjective evolutionary algorithms for random and correlated instances of multiobjective d-dimensional knapsack problems," *European Journal of Operational Research*, vol. 211, no. 3, pp. 466–479, 2011.

[19] R. Arst, B. Minsker, and D. E. Goldberg, "Comparing advanced genetic algorithms and simple genetic algorithms for groundwater management," in *Proceedings of the American Society of Civil Engineers (ASCE) Environmental & Water Resources Institute (EWRI) 2002 Water Resources Planning & Management Conference*, 2002.

[20] E. I. Ducheyne, B. D. Baets, and R. D. Wulf, *Probabilistic Models for Linkage Learning in Forest Management*. Springer Berlin Heidelberg, 2005.

[21] H. Mühlenbein, J. Bendisch, and H.-M. Voigt, "From recombination of genes to the estimation of distributions ii. continuous parameters," in *International Conference on Parallel Problem Solving from Nature*. Springer, 1996, pp. 188–197.

[22] S. Baluja, "Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning," DTIC Document, Tech. Rep., 1994.

[23] G. Harik, E. Cantú-Paz, D. E. Goldberg, and B. L. Miller, "The gambler's ruin problem, genetic algorithms, and the sizing of populations," *Evolutionary Computation*, vol. 7, no. 3, pp. 231–253, 1999.

[24] J. S. De Bonet, C. L. Isbell, P. Viola *et al.*, "Mimic: Finding optima by estimating probability densities," *Advances in neural information processing systems*, pp. 424–430, 1997.

[25] S. Baluja and S. Davies, "Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space." DTIC Document, Tech. Rep., 1997.

[26] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[27] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing*. Springer, 1999, pp. 521–535.

[28] G. Harik, "Linkage learning via probabilistic modeling in the ecga," *Urbana*, vol. 51, no. 61, p. 801, 1999.

[29] R. Etxeberria and P. Larranaga, "Global optimization using bayesian networks," in *Second Symposium on Artificial Intelligence (CIMAF-99)*. Habana, Cuba, 1999, pp. 332–339.

[30] H. Muhlenbein H and T. Mahnig, "Convergence theory and applications of the factorized distribution algorithm," *Journal of Computing and Information Theory*, vol. 7, no. 1, pp. 19–32, 1999.

[31] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111–128, 2011.

[32] R. Salustowicz and J. Schmidhuber, "Probabilistic incremental program evolution: Stochastic search through program space," in *ECML*, 1997, pp. 213–220.

[33] K. Sastry and D. E. Goldberg, "On extended compact genetic algorithm," *IlliGAL report*, no. 2000026, 2000.

[34] A. Ratle and M. Sebag, "Avoiding the bloat with stochastic grammar-based genetic programming," in *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer, 2001, pp. 255–266.

[35] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley and Sons, 2001, vol. 16.

[36] M. Laumanns and J. Ocenasek, "Bayesian optimization algorithms for multi-objective optimization," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2002, pp. 298–307.

[37] J. Ocenasek, S. Kern, N. Hansen, and P. Koumoutsakos, "A mixed bayesian optimization algorithm with variance adaptation," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 352–361.

[38] E. Ziztler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," *Evolutionary Methods for Design, Optimization, and Control*, pp. 95–100, 2002.

[39] P. A. Bosman and D. Thierens, "Multi-objective optimization with the naive\ mathbb {M} id\ mathbb {E} a," in *Towards a New Evolutionary Computation*. Springer, 2006, pp. 123–157.

[40] N. Khan, D. E. Goldberg, and M. Pelikan, "Multi-objective bayesian optimization algorithm," *Urbana*, vol. 51, p. 61801, 2002.

[41] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multiobjective hboa, clustering, and scalability," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM, 2005, pp. 663–670.

[42] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[43] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary computation*, vol. 12, no. 1, pp. 64–79, 2008.

[44] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.

[45] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE transactions on evolutionary computation*, vol. 10, no. 6, pp. 646–657, 2006.

[46] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, 2009.

[47] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science and Business Media, 2006.

[48] S. Rahnamayan, H. R. Tizhoosh, and M. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.

[49] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.

[50] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *2005 IEEE congress on evolutionary computation*, vol. 2. IEEE, 2005, pp. 1785–1791.

[51] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[52] T. Hendtlass, "A combined swarm differential evolution algorithm for optimization problems," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2001, pp. 11–18.

[53] W.-J. Zhang, X.-F. Xie *et al.*, "DEPSO: hybrid particle swarm with differential evolution operator," in *IEEE International Conference on Systems Man and Cybernetics*, vol. 4, 2003, pp. 3816–3821.

[54] A. Bisuias, S. Dasgupta, and S. D. A. Abrahaml, "A synergy of differential evolution and bacterial foraging optimization for global optimization," *Neural Network World*, vol. 6, no. 07, pp. 607–626, 2007.

[55] X. Zhang, H. Duan, and J. Jin, "DEACO: Hybrid ant colony optimization with differential evolution," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 921–927.

[56] X. He and L. Han, "A novel binary differential evolution algorithm based on artificial immune system," in *2007 IEEE Congress on Evolutionary Computation*. IEEE, 2007, pp. 2267–2272.

[57] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 107–125, 2008.

[58] Z. Yang, X. Yao, and J. He, "Making a difference to differential evolution," in *Advances in metaheuristics for hard optimization*. Springer, 2007, pp. 397–414.

[59] R. Storn, "On the usage of differential evolution for function optimization," in *Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American*. IEEE, 1996, pp. 519–523.

[60] J. Lampinen and I. Zelinka, "Mixed integer-discrete-continuous optimization by differential evolution," in *Proceedings of the 5th International Conference on Soft Computing*, 1999, pp. 77–81.

[61] R. Storn, "System design by constraint adaptation and differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 1, pp. 22–34, 1999.

[62] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Information Sciences*, vol. 178, no. 15, pp. 3043–3074, 2008.

[63] J. Lampinen, "A constraint handling approach for the differential evolution algorithm," in *Proceedings of the congress on evolutionary computation*, vol. 2. IEEE Computer Society Washington, DC, 2002, pp. 1468–1473.

[64] S. Kukkonen and J. Lampinen, "Constrained real-parameter optimization with generalized differential evolution," in *2006 IEEE International Conference on Evolutionary Computation*. IEEE, 2006, pp. 207–214.

[65] C.-S. Chang, D. Xu, and H. Quek, "Pareto-optimal set based multiobjective tuning of fuzzy automatic train operation for mass transit system," *IEE Proceedings-Electric Power Applications*, vol. 146, no. 5, pp. 577–583, 1999.

[66] H. A. Abbass and R. Sarker, "The pareto differential evolution algorithm," *International Journal on Artificial Intelligence Tools*, vol. 11, no. 04, pp. 531–552, 2002.

[67] S. Kukkonen and J. Lampinen, "An extension of generalized differential evolution for multi-objective optimization with constraints," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 752–761.

[68] F. Xue, A. C. Sanderson, and R. J. Graves, "Modeling and convergence analysis of a continuous multi-objective differential evolution algorithm," in *2005 IEEE Congress on Evolutionary Computation*, vol. 1. IEEE, 2005, pp. 228–235.

[69] T. Robič and B. Filipič, "DEMO: Differential evolution for multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 520–533.

[70] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2004, pp. 861–872.

[71] B. Babu and M. M. L. Jehan, "Differential evolution for multi-objective optimization," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 4. IEEE, 2003, pp. 2696–2703.

[72] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[73] R. Horst, P. M. Pardalos, and N. Van Thoai, *Introduction to global optimization*. Springer Science and Business Media, 2000.

[74] S.-M. Guo and C.-C. Yang, "Enhancing differential evolution utilizing eigenvector-based crossover operator," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 31–49, 2015.

[75] A. Zhou, J. Sun, and Q. Zhang, "An estimation of distribution algorithm with cheap and expensive local search," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 807–822, 2015.

[76] M. Okamoto, T. Nonaka, S. Ochiai, and D. Tominaga, "Nonlinear numerical optimization with use of a hybrid genetic algorithm incorporating the modified powell method," *Applied Mathematics and Computation*, vol. 91, no. 1, pp. 63–72, 1998.

[77] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

[78] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[79] N. Kambhatla and T. Leen, "Dimension reduction by local principal component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1493–1516, 1997.

[80] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

# 致　谢

白驹过隙，三年的研究生的学习生活转眼来到了尾声。想起当初的事情，依然都历历在目。回想这三年的点点滴滴，不禁感慨万千，既有泪水，也有汗水，当然也有更多的收获。在这三年的研究生生涯之中，自己收获了很多，得到了成长，结识了很多的好朋友，得到了老师们的帮助，使得我最终能够顺利地完成学业。

首先，我最想感谢的是我的导师周爱民老师。我觉得周老师对我的帮助是方方面面，不仅是学习上，学术上方面的内容，同时也包括在工作和生活中的方方面面。周老师教会我如何严谨地进行科学实验，如何写论文，让我在学术研究上成长了很多，我相信这必然是我人生中的一大裨益。另一方面，周老师也给予了我参加应用实践的机会，让我实际地提高了自己的动手编程能力，并且督促我形成良好的编程风格，这为我以后的工作也打下了良好的基础。周老师，也教会了了我在工作中如何和他人保持良好的沟通。周老师对我的帮助太多。

其次，感谢李阳师兄和张晋媛学姐。李阳师兄虽然我们只有短短一年的相处时间，但我觉得师兄给我的帮助却是很大的。不管是在论文写作上的问题，亦或是生活上的一些问题，李阳师兄都给了我很好的建议。那段时间和李阳师兄一起打羽毛球的时光，也让我怀念不已。张晋媛师姐给我的帮助无疑是巨大的。我觉得师姐是我最好的良师益友，在论文写作的过程中，和师姐讨论过很多问题，她给了我很多很有价值的建议。在我遇到困难的时候，她也给了我莫大的鼓励。感谢实验室的师弟师妹们，我很珍惜我们在一起学习玩耍的时光，我相信这是我人生最美好的时光之一。

感谢所有帮助过我的老师、同学和朋友，感谢所有关心和帮助过我的人。感谢我的家人，没有他们的帮助，我也不会有今天的成绩。感谢华东师范大学，这是我学生生涯的一个终点，但也是我崭新人生的新起来。华东师大，这个美丽的地方，给了我信心应对未来的挑战！

# 在校期间的学术论文和科研项目

[1] B. Dong, A. M. Zhou and G. X. Zhang, “Sampling in Latent Space for a Multiobjective Estimation of Distribution Algorithm,” IEEE Congress on Evolutionary Computation, 2016

[2] B. Dong, A. M. Zhou and G. X. Zhang, “A Hybrid Estimation of Distribution Algorithm with Differential Evolution for Global Optimization,” IEEE Symposium Series on Computational Intelligence, 2016

[3] 国家重大科学仪器设备开发专项子课题(2012YQ18013201): 拉曼光谱仪软件算法设计