

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: 演化算法中基于差分进化的采样策略

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 sampling strategy based on differential evolution
in evolutionary 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 | |
|-----|-------|--------|-----|--|
| | 教 授 | 华东师范大学 | 主 席 | |
| | 教 授 | 华东师范大学 | | |
| | 副 教 授 | 华东师范大学 | | |

演化算法中基于差分进化的采样策略

中 文 摘 要

本文研究的是对于演化算法中采样策略的改进工作。演化计算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。演化计算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十分强大的随机优化算法，得益于它的诸多优点，它已经被广泛应用于各个领域。由于差分进化的易于实现，它被用于与其他的演化算法进行结合，并且已经多数学者提出了基于差分进化的混合算法。本文将差分进化的思想引入到分布估计算法的采样中，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，并且研究其在解决多目标以及单目标问题上的性能表现。实现结果表明，这一研究对于提高分布估计算法性能有着重大意义。

关键词： 分布估计算法, 差分进化, 演化算法, 特征向量

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

| | |
|------------------------|----|
| 中文摘要 | i |
| 英文摘要 | ii |
| 1 绪论 | 1 |
| 1.1 研究的目的和意义 | 2 |
| 1.2 本文所做的工作和内容安排 | 3 |
| 2 理论基础 | 4 |
| 2.1 多目标优化问题 | 4 |
| 2.2 单目标优化问题 | 9 |
| 2.3 分布估计算法 | 11 |
| 2.4 差分进化 | 13 |
| 3 基于采样策略对于连续多目标优化问题的研究 | 12 |
| 3.1 连续多目标优化问题 | 12 |
| 3.2 RM-MEDA算法 | 13 |
| 3.3 基于差分进化的采样策略 | 13 |
| 3.4 算法框架 | 14 |
| 3.5 实验分析 | 14 |
| 4 基于差分进化的分布估计算法 | 16 |
| 4.1 差分进化和分布估计算法的研究 | 16 |
| 4.2 基于差分进化采样的分布估计算法 | 19 |

| | |
|------------------|----|
| 4.3 实验框架 | 20 |
| 4.2 实验结果分析 | 18 |
| | |
| 5 总结与展望 | 25 |
| 5.1 本文工作总结 | 25 |
| 5.2 工作展望 | 25 |
| | |
| 参考文献 | 27 |
| 致 谢 | 31 |

1 绪论

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 [1]。演化算法受1859年达尔文提出的物种起源的启发，吸取其中的生物进化和自然选择的思想。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 [2]，生物地理学优化 [3]，遗传编程 [4]，演化编程 [5]，差分进化 [6]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，它通过建立在概率模型中采样来产生新的种群个体，通过提取当前优秀中取得概率模型来指导下一步的搜索 [7] [8]。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，通过使用一个精确地概率分布模型来提取统计信息。子代种群解集将会通过从建立的概率分布模型中采样产生，并且产生的解集会部分或者全部地取代父代中解集。分布估计算法通过分布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。其中，采样环节是本文的重点，采样对于分布估计算法具有重大意义，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 [9]。

差分进化算法是一种有效并且简单的演化算法 [6, 10]。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索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）。DES对于提高种群的多样性，加速种群解集收敛都有着重大的意义。

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，差分进化的思想被引入到采样策略中，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，来验证DES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佈估计算法（A Regularity Model-based Estimation of Distribution Algorithm, RM-MEDA）[11]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) [12], 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安排。

第二章 理论基础, 介绍分布估计算法和差分进化的基本概念以及相应的算法框架。

第三章 提出基于差分进化的采样策略以及相应的算法框架。

第四章 将基于差分进化的采样策略应用到多目标优化问题上面, 并给出相应的算法框架和实验结果分析。

第五章 将基于差分进化的采样策略应用到单目标优化问题上, 并给出相应的算法框架和试验结果分析。

第六章 论文总结和展望。

2 理论基础

本章主要介绍演化算法中对于多目标优化问题以及单目标优化问题的相关定义，同时分别简要地介绍分布估计算法和差分进化算法的理论和算法框架。

2.1 多目标优化问题

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，这就是本文中所述的多目标优化问题（Multiobjective Optimization, MOP）。拿一个我们日常生活中买车的例子，如图所示，在买车的时候我们既要考虑到价格同时也要考虑到舒适性，这就可以理解成一种多目标优化问题。

为了不失一般性，在本文中，在本文中我们假设每个问题都是最小化问题，则MOP可以由以下数学公式表达：

$$\begin{aligned} \min \quad & F(x) = (f_1(x), \dots, f_m(x)) \\ \text{s.t.} \quad & x \in \Omega \end{aligned} \quad (2.1)$$

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

在MOP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 [13]。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 $i = 1, \dots, n$ ，则称 a 支配 b 。向量 $x^* \in \Omega$ 即是2.1 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）。

2.2 单目标优化问题

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，一般即是求得最小值或者最大值。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 [14]。同样，对于全局优化问题在本文中做出以下定义：

$$\begin{aligned} \min f(x) \\ s.t. x \in [a_i, b_i]^n \end{aligned} \quad (2.2)$$

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

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 [7]。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地分布来产生新的种群 [15]。

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Algorithm 1所示。

Algorithm 1: 分布估计算法

```

1 初始化: 建立随机初试种群  $Pop(t)$ ,  $t$  是相应的种群代数。
2 while not terminate do
3   建模: 根据种群  $Pop(t)$  中的统计信息建立概率模型  $p(x)$ 。
4   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  $Q$ 。
5   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  $Pop(t+1)$ 。
6    $t = t + 1$ 
7 end
```

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 [6]。自从20 是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 [16]。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 [17,18]。
- 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 [19,20]。

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

Algorithm 2: 差分进化

```

1 随机初始种群 $P_0$ :  $P_0 = \{x_{1,D}, x_{2,D}, x_{3,D}, \dots, x_{N,D}\}$ 
2 while not terminate do
3    $v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} + x_{r3,G})$ 
4   if  $\text{rand}_j(0, 1) \leq CR$  or  $j = j_{rand}$  then
5      $u_{i,j,G} = v_{i,j,G}$ 
6   else
7      $u_{i,j,G} = x_{i,j,G}$ 
8   end
9   if  $f(u_{i,G}) \leq f(x_{i,G})$  then
10     $x_{i,G+1} = u_{i,G}$ 
11  else
12     $x_{i,G+1} = x_{i,G}$ 
13  end
14 end

```

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 。

2.4.2 差分变异策略

1. DE/rand/1策略

$$v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} + x_{r3,G}) \quad (2.3)$$

2. DE/best/1策略

$$v_{i,G} = x_{best,G} + F \cdot (x_{r1,G} + x_{r2,G}) \quad (2.4)$$

3. DE/rand/2策略

$$v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}) + F \cdot (x_{r4,G} - x_{r5,G}) \quad (2.5)$$

4. DE/best/2策略

$$v_{i,G} = x_{best,G} + F \cdot (x_{r1,G} - x_{r2,G}) + F \cdot (x_{r3,G} - x_{r4,G}) \quad (2.6)$$

5. DE/current-to-best/1

$$v_{i,G} = x_{r1,G} + F \cdot (x_{best,G} - x_{i,G}) + F \cdot (x_{r1,G} - x_{r2,G}) \quad (2.7)$$

6. DE/current-to0rand/1

$$u_{i,G} = x_{i,G} + K \cdot (x_{r1,G} - x_{i,G}) + F' \cdot (x_{r2,G} - x_{r3,G}) \quad (2.8)$$

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。

2.4.3 主要的差分进化算法

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，各种各样基于差分进化的算法都涌现出来。下面就演化计算中广泛流行的几种差分进算法进行介绍。

JADE [21]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-pbest"，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。

CoDE(Composite DE) [22]算法通过结合三个不同的后代产生策略和三个常用的参数设置，后代产生策略和参数设置会随机组合在一起来产生新的解集。这个算法虽然比较简

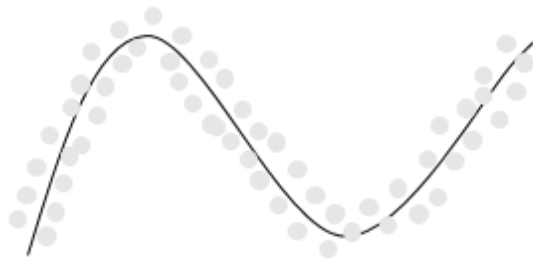


图 1: 连续多目标问题决策空间中个体的分布情况

单，但却十分有效，在CEC2005所有的测试题上经过运算，和其他差分进化算法相比都有着不小的提高。

3 基于采样策略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。

3.1 连续多目标问题

在第二章中，已经介绍过多目标问题的相关定义。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 1所示。

3.2 基于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Local PCA [23]在决策空间中的区域建立概率分布模型，然后通过使用拉丁采样得到新的子代种群。RM-MEDA采用基于非劣排序的方法 [24]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。

3.2.1 RM-MEDA算法

对于连续多目标问题，在决策空间中，种群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， ξ 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 ξ 则可以由公式 3.1表示：

$$\xi = \zeta + \epsilon \quad (3.1)$$

ζ 相当于是均匀分布在m-1维流体附近的个体， ϵ 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 3则是RM-MEDA的主要的算法框架。

Algorithm 3: RM-MEDA 算法框架

- 1 初始化一个随机种群 $Pop(0)$ ，并且设置 $t = 0$ 。
 - 2 **while** 没有达到停机条件 **do**
 - 3 **建模：** 建立一个概率模型 ξ 来表示在随机种群 $Pop(t)$ 中的个体。
 - 4 **采样：** 通过上述的概率模型进行采样得到新的解集 Q 。
 - 5 **选择：** 从 $Q \cup Pop(t)$ 中挑选出 N 个个体来组成一个新的种群 $Pop(t + 1)$ 。
 - 6 $t = t + 1$
 - 7 **end**
 - 8 返回最终的种群解集 $Pop(t)$ 。
-

3.2.2 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 2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问题，，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

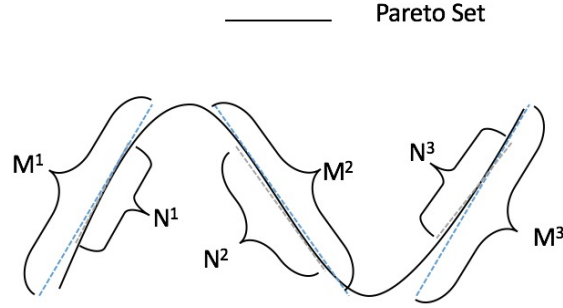


图 2: 通过缩放比例来覆盖Pareto setp

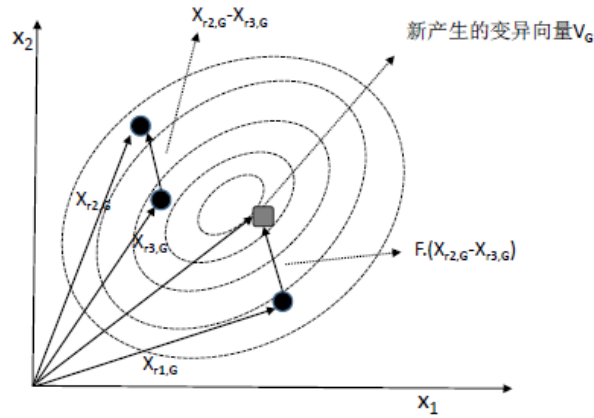


图 3: 通过缩放比例来覆盖Pareto setp

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Pareto set也成为了一个比较困难的问题。

3.2.3 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 3所示：

$$X = X_{r_1} + rand \cdot (X_{r_2} - X_{r_3}) + F \cdot (X_{r_2} - X_{r_3}) \quad (3.2)$$

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 3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 4所示：

Algorithm 4: 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

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

where ε' is the Gaussian noise subjects to the distribution $\mathcal{N}(0, \sigma_\tau I)$
 $(\tau \in \{1, 2, \dots, K\})$ is a randomly generated integer).

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 5所示。

3.3 实验分析

3.3.1 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 [25]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：

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{P^*} \quad (3.3)$$

Algorithm 5: DES-RM-MEDA

```

1 Initialize a population  $Pop(0)$ , and set  $t = 0$ .
2 while not terminate do
3   Modeling: Build the probabilistic model  $\delta$  in order to model the distribution of
     the solutions in  $Pop(t)$ .
4   Reproduction: Partition the population into different clusters  $C_i$  according to
     the probabilistic model. For each cluster, use DES to generate a set of candidate
     solutions  $Q_i$ . Set  $Q = \cup_i Q_i$ .
5   Selection: Select  $N$  solutions from  $Q \cup Pop(t)$  to construct a new population
      $Pop(t+1)$ .
6    $t = t + 1$ 
7 end
8 Return the solutions in  $Pop(t)$ .
```

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 Matlab 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。

3.4 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4是RM-MEDA在不同缩放因子的作用下，IGD指标的箱线图。

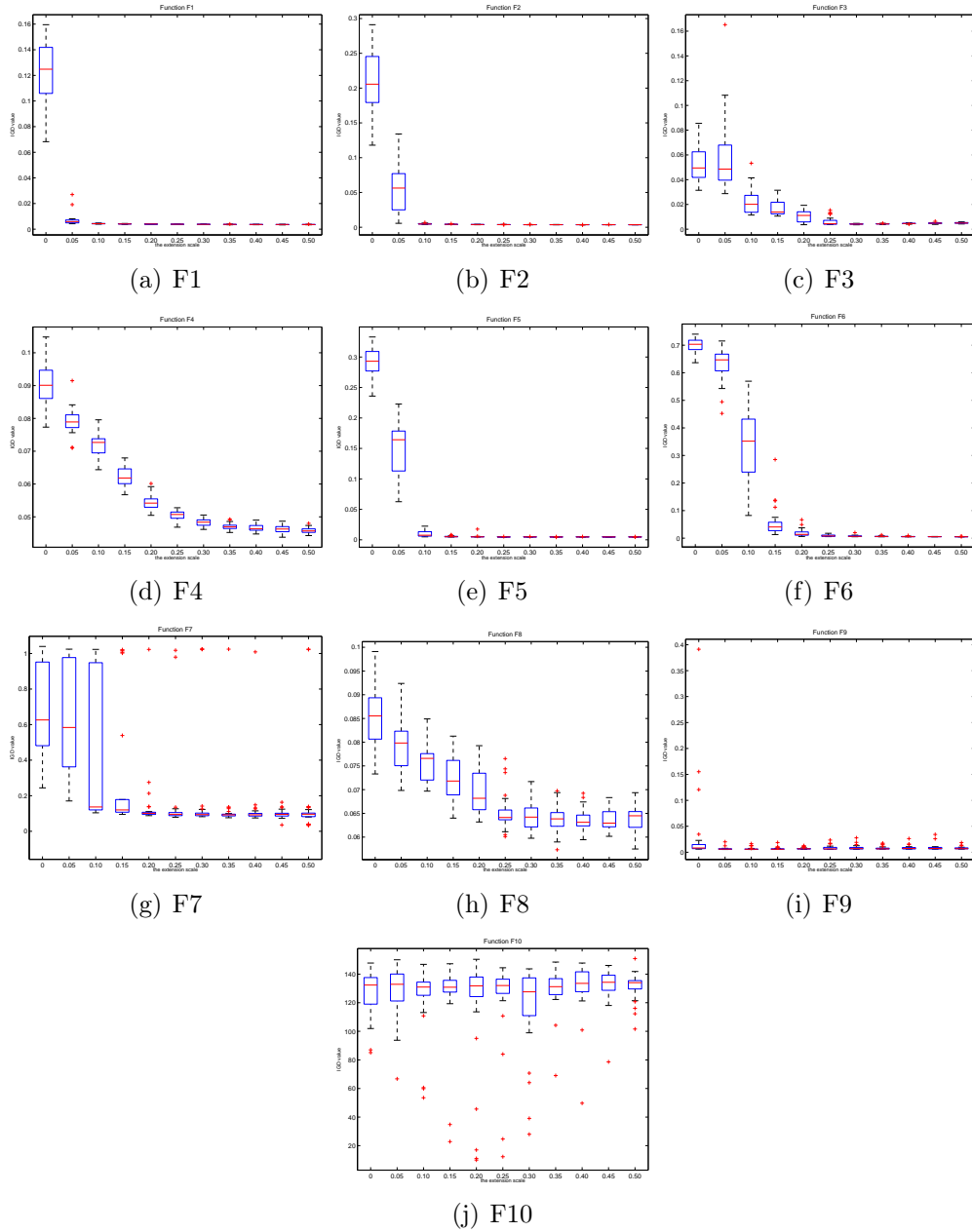


图 4: 不同缩放因子下，RM-MEDA的IGD指标的箱线图

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 4(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。

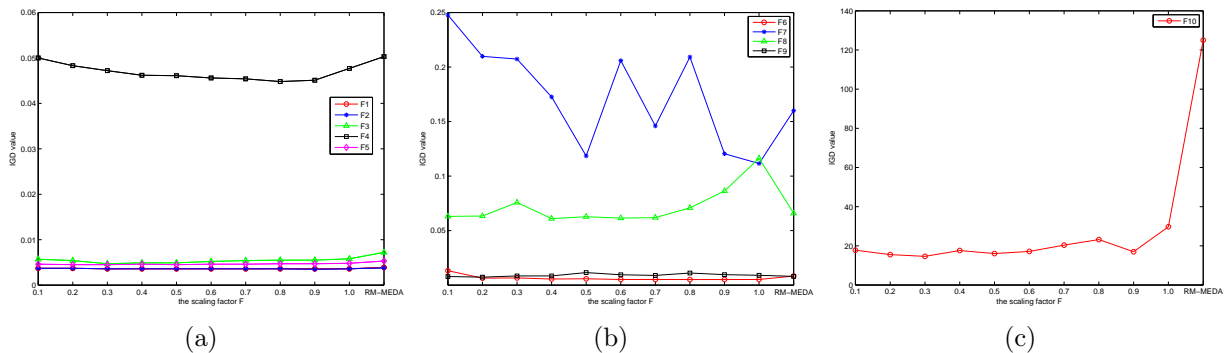


图 5: 不同缩放因子下, DES-RM-MEDA的性能比较

3.5 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 5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,我们将使用魏可送两样本检验法(Wilcoxon's rank sum tes)来对于RM-MEDA和DES-RM-MEDA之间的性能进行比较。层叠柱状图 6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。

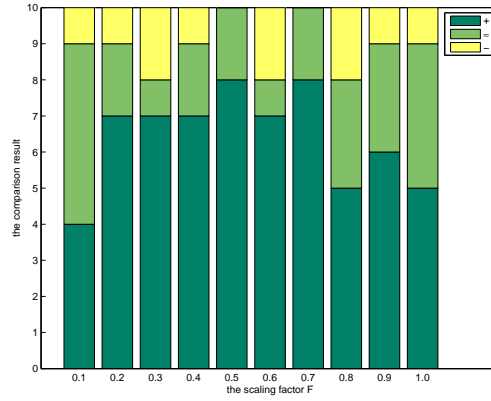


图 6: "+", "~", 以及 "-" 分别表示DES-RM-MEDA的性能要优于, 相似于, 差于RM-MEDA的性能

3.6 对比研究

从表 1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 7可以用来比较DES-RM-MEDA和RM-MEDA在测试题上的收敛趋势。从图 7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。

3.7 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编译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。基于差分进

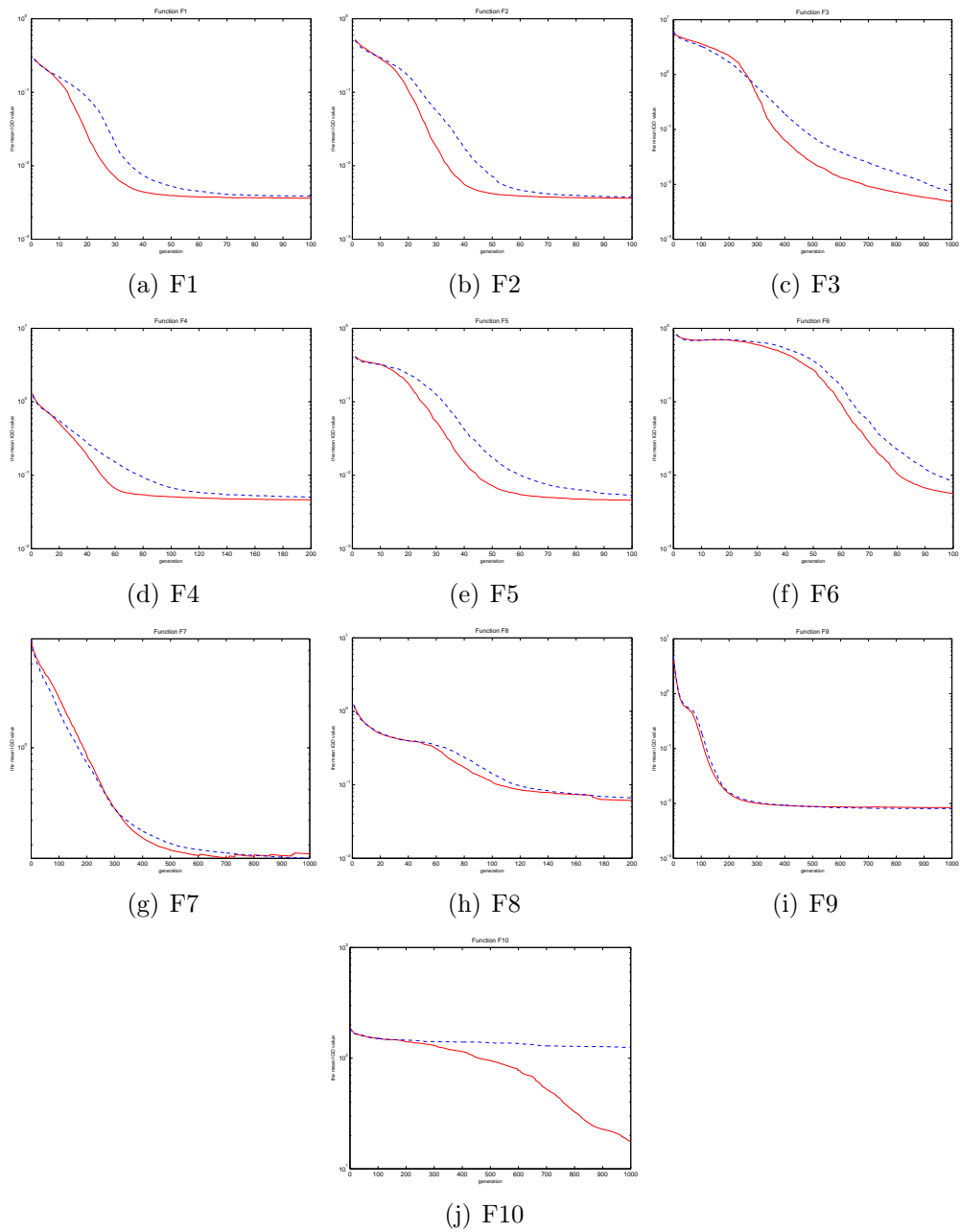


图 7: IGD指标均值趋势图。实线表示DES-RM-MEDA，虚线表示RM-MEDA。

表 1: 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 |
| <i>F1</i> | $3.90e-03$ | $1.39e-04$ | $3.70e-03$ | $4.20e-03$ | 3.60e-03 | $9.16e-05$ | 3.43e-03 | 3.77e-03 |
| <i>F2</i> | $3.80e-03$ | $1.43e-04$ | $3.50e-03$ | $4.10e-03$ | 3.60e-03 | $1.01e-04$ | 3.35e-03 | 3.82e-03 |
| <i>F3</i> | $7.20e-03$ | $3.90e-03$ | 3.60e-03 | $1.55e-02$ | 4.90e-03 | $8.09e-04$ | $3.90e-03$ | 7.31e-03 |
| <i>F4</i> | $5.03e-02$ | $1.30e-03$ | $4.82e-02$ | $5.35e-02$ | 4.62e-03 | $9.32e-04$ | 4.44e-02 | 4.85e-02 |
| <i>F5</i> | $5.30e-03$ | $3.00e-03$ | $4.40e-03$ | $2.12e-02$ | 4.60e-03 | $1.49e-04$ | 4.33e-03 | 4.96e-03 |
| <i>F6</i> | $8.30e-03$ | $2.10e-03$ | $5.70e-03$ | $1.50e-02$ | 5.60e-03 | $9.04e-04$ | 4.52e-03 | 8.30e-03 |
| <i>F7</i> | 1.60e-01 | $2.35e-01$ | $7.96e-02$ | $1.03e+00$ | $1.73e-01$ | $2.28e-01$ | 3.20e-02 | 1.02e+00 |
| <i>F8</i> | $6.59e-02$ | $3.50e-03$ | $6.05e-02$ | $7.69e-02$ | 6.10e-02 | $2.03e-03$ | 5.67e-02 | 6.39e-02 |
| <i>F9</i> | 8.00e-03 | $2.80e-03$ | $5.80e-03$ | 1.48e-02 | $8.40e-03$ | $3.20e-03$ | 5.51e-03 | $2.12e-02$ |
| <i>F10</i> | $1.25e+02$ | $2.35e+01$ | $2.27e+01$ | $1.44e+02$ | 1.76e+00 | $1.29e+01$ | 4.73e+00 | 7.15e+01 |

¹ 粗体的表示更好的结果

化的采样策略被用于改进RM-MEDA的采样环节。初步的研究表明，通过差分进化的采样策略比拉丁方设计能够有效地改进RM-MEDA的采样。

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。

4 基于采样策略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，我们引入了DE-EIG算法，从而提出EDA/DE-EIG算法。通过利用DE-EIG来提高EDA在单目标优化问题上的采样性能。实验表明，EDA/DE-EIG对于全局单目标优化问题具有较大潜力。

4.1 单目标优化问题

4.2 基于特征向量的差分进化

DE-EIG是一种基于特征向量的差分进化算法 [?], 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。

4.2.1 新的交叉策略

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群居最有演化，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 [16]。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。

4.2.2 DE-EIG算法框架

4.3 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，用于解决全局连续问题优化。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。

4.4 复杂的局部搜索

复杂的局部搜索（expensive local search）是EDA/LS [26]中一种进一步优化种群演化的策略。

4.5 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也将被用于进一步提升算法求解的性能。

Algorithm 6: DE-EIG

```
1 初始化种群  $Pop(t) = \{x_1, x_2, x_3, \dots, x_N\}$  ( $N$ 是种群大小)
2 while not terminate do
3    $v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G})$ 
4   if  $rand() < p$  then
5     if  $rand() \leq CR$  then
6        $u_{i,G} = v_{i,G}$ 
7     else
8        $u_{i,G} = x_{i,G}$ 
9     end
10  else
11    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。p
12     $x'_{i,G} = E' \cdot x_{i,G}$ 
13     $v'_{i,G} = E' \cdot v_{i,G}$ 
14    if  $rand() \leq CR$  then
15       $u'_{i,G} = v'_{i,G}$ 
16    else
17       $u'_{i,G} = x'_{i,G}$ 
18    end
19     $u_{i,G} = E \cdot u'_{i,G}$ 
20  end
21  if  $f(u_{i,G}) \leq f(x_{i,G})$  then
22     $x_{i,G+1} = u_{i,G}$ 
23  else
24     $x_{i,G+1} = x_{i,G}$ 
25  end
26   $t = t + 1$ 
27 end
```

Algorithm 7: DE/EDA

```
1 Generate population  $Pop(t)$  randomly consists of  $N$  solutions  $x_1, x_2, \dots, x_N$  from the
  feasible search space. while not terminate do
2   Construct the probabilistic model:
3    $p_k(x) = \prod_{i=1}^n \mathcal{N}(x_i; \mu_i, \sigma_i)$ 
4   For all  $j = 1, 2, \dots, n$ , produce a trial solution  $u = (u_1, u_2, \dots, u_n)$ 
5   if  $rand() < CRP$  then
6      $u_j = \frac{(x_i)_j + (x_d)_j}{2} + F \cdot [(x_d)_j - (x_i)_j + (x_b)_j - (x_c)_j]$ 
7   else
8      $u_j$  is sampled according to  $\mathcal{N}(x_i; \mu_i, \sigma_i)$ 
9   end
10  where  $CRP$  is the controlling parameter.
11  if  $f(u) < f(x_i)$  then
12     $x_i^{t+1} = u$ 
13  else
14     $x_i^{t+1} = x_i^t$ 
15  end
16   $t = t + 1$ 
17 end
```

Algorithm 8: EDA/DE-EIG

```
1 初始化种群  $Pop(t) = \{x_1, x_2, x_3, \dots, x_N\}$  ( $N$ 是种群的大小)
2 while not terminate do
3   构建概率模型:
4    $p(x) = \prod_{i=1}^n \mathcal{N}(x_i; \mu_i, \sigma_i)$ 
5   Generate a trial solution  $u_{i,G}$  as follows:
6   if  $rand() < CRP$  then
7     根据DE-EIG得到  $u_{i,G}$ 
8   else
9     根据概率模型  $p(x)$  采样得到  $u_{i,G}$ 
10  end
11  if  $f(u_{i,G}) < f(x_{i,G})$  then
12     $x_{i,G+1} = u_{i,G}$ 
13  else
14     $x_{i,G+1} = x_{i,G}$ 
15  end
16  if  $Converage(\theta, G, G_e)$  then
17    执行expensive local search.
18  end
19   $t = t + 1$ 
20 end
```

4.6 实验分析

在本章节，我们将EDA/DE-EIG和JADE [21]和 [?]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。

4.6.1 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-topbest"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。

4.6.2 测试集

所有的算法都会根据其在YYL测试集 [27]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。

4.6.3 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: The CRP is 0.5; F is set to be 0.5; CR is set to be 0.6; the parameter p to control the probability to operate crossover two coordinate systems is 0.5; the convergence threshold $\theta = 0.1$; the size of the population N is 150. For the parameter setting of expensive local search, it is same as that in [26].

4.6.4 实验结果和分析

表 2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比较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。

从表 2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-EIG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 8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。

4.7 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, 它结合了全局和局部的信息。在本章节中, 通过引进一种提升的差分进化算法, 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。实验结果表明, 在和JADE和DE/EDA比较的过程中, EDA/DE-EIG还是比较具有优势的。

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优化; 第二, 值得探究在DE-EIG 和EDA算法资源的分配, 这是一个有意思的话题。

表 2: Statistical results (*mean* \pm *std*) for the three algorithms on instances $f1 - f13$.

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

¹ The bold ones mean the best.

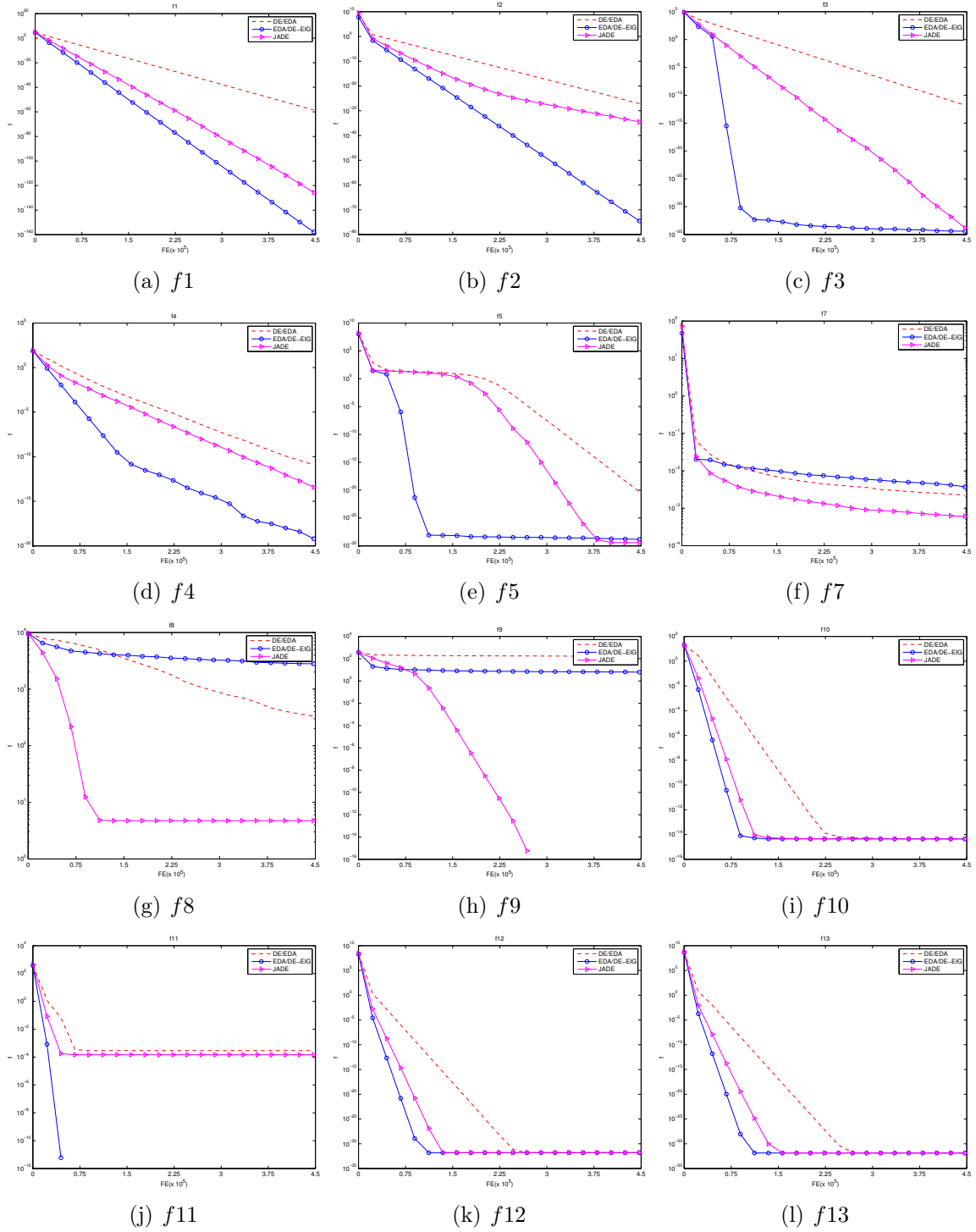


图 8: The mean function value versus on $f_1 - f_{13}$ except f_6 of the three algorithms.

5 总结与展望

5.1 本文工作总结

5.2 工作展望

参考文献

- [1] D. Simon, *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.
- [2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.
- [3] D. Simon, “Biogeography-based optimization,” *IEEE transactions on evolutionary computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [4] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic programming: an introduction*, 1998.
- [5] X. Yao and Y. Liu, “Fast evolutionary programming.” in *Evolutionary Programming*, 1996, pp. 451–460.
- [6] R. Storn and K. Price, *Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces*. ICSI Berkeley, 1995, vol. 3.
- [7] P. Larranaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science & Business Media, 2002, vol. 2.
- [8] 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.
- [9] A. Z. Bing Dong and G. Zhang, “Sampling in latent space for a multiobjective estimation of distribution algorithm,” 2016.
- [10] 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.
- [11] 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.

- [12] 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.
- [13] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.
- [14] R. Horst, P. M. Pardalos, and N. Van Thoai, *Introduction to global optimization*. Springer Science & Business Media, 2000.
- [15] *Towards a new evolutionary computation: advances on estimation of distribution algorithms*. Springer Science & Business Media, 2006, vol. 192.
- [16] S. Das and P. N. Suganthan, “Differential evolution: a survey of the state-of-the-art,” *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 4–31, 2011.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] J. Zhang and A. C. Sanderson, “JADE: adaptive differential evolution with optional external archive,” *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 5, pp. 945–958, 2009.

- [22] Y. Wang, Z. Cai, and Q. Zhang, “Differential evolution with composite trial vector generation strategies and control parameters,” *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 55–66, 2011.
- [23] N. Kambhatla and T. Leen, “Dimension reduction by local principal component analysis,” *Neural Computation*, vol. 9, no. 7, pp. 1493–1516, 1997.
- [24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.
- [25] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117–132, 2003.
- [26] A. Zhou, J. Sun, and Q. Zhang, “An estimation of distribution algorithm with cheap and expensive local search,” *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 6, pp. 807–822, 2015.
- [27] X. Yao, Y. Liu, and G. Lin, “Evolutionary programming made faster,” *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 2, pp. 82–102, 1999.

6 附录

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点。华东师大，这个美丽的地方，给了我信心应对未来的挑战！