

西安交通大学

硕士学位论文

高性能差分进化算法研究及其在波形反演的应用

学位申请人：梁浩

指导教师：潘志斌教授

学科名称：信息与通信工程

2018 年 5 月

Research on High-performance Differential Evolution Algorithm and Application to Waveform Inversion

A thesis submitted to
Xi'an Jiaotong University
in partial fulfillment of the requirements
for the degree of
Master of Engineering

By
Hao Liang
Supervisor: Prof. Zhibin Pan
(Information and Communication Engnieering)
May 2018

摘要

论文题目：高性能差分进化算法研究及其在波形反演的应用

学科名称：信息与通信工程

学位申请人：梁浩

指导教师：潘志斌教授

摘要

随着科学技术的发展，优化问题在数学和工程上越来越普遍。而差分进化（DE）算法作为一种简单而高效的优化算法，自提出以来，因其结构简单、搜索能力强，受到国内外研究者的广泛关注，并在信号处理、模式识别与图像处理、人工神经网络、油气勘探和生物信息学等各个领域得到深入的应用。其中，差分进化算法作为一种全局优化算法，因其不受初始模型限制的特点，是求解波形反演这种强非线性问题的一种较好的优化算法。在此背景下，本论文针对差分进化算法存在的问题对其进行改进，从平衡好差分进化算法开发性与探索性这个目标出发，设计了几种性能更好的差分进化算法变体。本文主要包括以下几方面的工作：

(1) 针对差分进化算法应用到波形反演问题中耗时过久的问题，提出了一种基于个体到种群中心的欧氏距离进行种群结构划分的多子种群差分进化算法 SpDE (Differential Evolution with Subpopulations)，最内层的子种群负责开发性，而最外层子种群则承担探索性，并提出一个分两阶段的变异策略，以求达到开发性与探索性的最佳平衡，进而提高差分进化算法应用到波形反演问题中的效率。在合成和实际的地震数据上的实验结果表明，SpDE 不仅能加快收敛速度，而且还在收敛精度上有很大的提升。

(2) 针对传统的差分进化算法的变异操作中的个体都是随机选出的问题，本论文提出了一种综合考虑个体适应度排名和个体对种群多样性贡献的方法来对参与变异操作的个体进行选择。变异操作是差分进化算法中最重要的操作，而变异操作中的基个体又是重中之重。传统的差分进化算法要么随机选择个体参与变异操作，要么选择当前代最优个体或者目标个体作为基个体，而这并没有很好地利用种群的进化信息。在我们的方法中，个体适应度排名要顾及开发性，而个体对种群多样性的贡献则考虑到探索性，综合考虑两者则可在开发性与探索性之间取得平衡，使差分进化算法性能更佳。在 CEC2005 测试函数集上的实验结果表明，我们提出的方法取得了很大的性能提升。

(3) 针对传统差分进化算法大都集中于改善在适应度上表现较好的个体而忽视占种群绝大部分的一般或者较差个体的问题，我们提出了一种基于适应度进行邻域划分的差分进化算法变体。每个个体的邻域大小因其适应度排名而不同，优秀的个体的邻域非常大，几乎囊括整个种群，而较差个体的邻域则相当小，只包括那些较为优秀的一部分。并相应地提出一种新的变异策略，使每个目标个体的变异操作中所涉及的个

体都来自于其邻域之内，因此，对于优秀个体而言，新的变异策略退化为传统的变异策略，而对于较差个体来说，则参与其变异操作的个体都是较优秀的那一部分，因此很大概率上能提升较差个体的表现，进而整体上提升种群的进化程度。在 CEC2005 测试函数集上的实验结果表明，我们的方法在最终收敛精度上有很大的提升。

关 键 词：差分进化算法；波形反演；子种群；变异策略个体排序；适应度排名

论文类型：应用基础

Title: Research on High-performance Differential Evolution Algorithm and Application to Waveform Inversion

Discipline: Information and Communication Engineering

Applicant: Hao Liang

Supervisor: Prof. Zhibin Pan

ABSTRACT

With the development of science and technology, optimization problems are becoming more and more popular in mathematics and engineering. As a simple and efficient optimization algorithm, differential evolution algorithm has been widely investigated by researchers worldwide. It has been deeply used in different fields, such as signal processing, pattern recognition and image processing, artificial neural network, oil and gas exploration and bioinformatics. As a global optimization algorithm, differential evolution algorithm is very suitable for solving the strong nonlinear problems such as waveform inversion because it is not restricted to initial models. In this context, this thesis aims at improving the problems in differential evolution algorithm. We propose several differential evolution variants to improve the performance of differential evolution algorithm. The main work of this paper includes the following aspects:

(1) To tackle the problem that it is too time-consuming when differential evolution algorithm is applied into waveform inversion, this thesis proposed a new subpopulation differential evolution (SpDE) which divides the whole population into three subpopulations based on the Euclidean distance between each individual and the population center. The main task of inner subpopulation is to exploit and the main task of the outer subpopulation is to explore. And then we proposed a two-phase mutation strategy to balance the exploration and exploitation. Experimental results on synthetic and field seismic data show that SpDE can not only speed up convergence, but also greatly improve convergence accuracy.

(2) In the conventional differential evolution algorithm, the individuals involved in mutation operator are selected randomly in population. To tackle this problem, this thesis proposed a new method, which considers both the fitness and contribution to the diversity of population, to select the individuals involved in mutation operator. The mutation operator is the most important operator in the differential evolution, and the base individual in the mutation operator has the top priority. The traditional differential evolution algorithm either randomly selects the individual to participate in the mutation operation or chooses the best individual or the target individual as the base individual, therefore it does not make good use of the information of population. In our method, we use the ranking based on fitness to maintain exploitation and use the ranking based on the contribution to the diversity of population to maintain the exploration. We consider both of them in order to achieve balance between exploitation and exploration, which improves the performance of differential evolution. The

experimental results on 25 benchmark functions from CEC2005 show that our method has achieved great improvements.

(3) Most of the traditional differential evolution algorithm focus on improving the fitness of the individuals with better performance while ignoring the normal or poor individuals that occupy a vast majority of the population. We proposed a new differential evolution variant with fitness-based neighborhood, in which the size of the individual's neighborhood is different according to its ranking based on fitness. The neighborhood of the very good individual is very large, including almost all of the population. While the neighborhood of the very poor individual is very small, including only the most excellent part. A new mutation strategy is proposed accordingly. The individual involved in the mutation operation of each target individual comes from its neighborhood. Therefore, for the very good individual, the new mutation strategy is degenerated into a traditional mutation strategy. And for the very poor individual, the individuals that participate in the variant operation are the most excellent part. Therefore, it has a high probability that it can enhance the performance of poor individuals and improve the evolution of population. Experimental results on the benchmark functions from CEC2005 show that our method has greatly improved the final convergence accuracy.

KEY WORDS: Differential evolution algorithm; Waveform inversion; Subpopulation; Ranking of individuals involved in mutation; Ranking based on fitness

TYPE OF THESIS: Application Fundamentals

目 录

1 绪论	1
1.1 课题的研究背景与意义	1
1.2 差分进化算法的研究现状	1
1.3 论文的主要研究内容	2
1.4 论文组织结构	3
2 相关工作回顾	5
2.1 波形反演问题	5
2.2 差分进化算法	6
2.2.1 经典差分进化算法	6
2.2.2 协同变异差分进化算法（CCDE）	10
2.3 CEC2005 测试函数集	11
2.4 本章小结	13
3 基于子种群的差分进化算法	15
3.1 引言	15
3.2 应用在波形反演的差分进化算法	15
3.3 基于多子种群的差分进化算法	16
3.3.1 子种群的划分	17
3.3.2 两阶段变异策略	18
3.4 算法在波形反演中的应用	20
3.4.1 SpDE 应用于合成地震数据	21
3.4.2 SpDE 的稳定性	24
3.4.3 SpDE 应用于实际地震数据	26
3.5 本章小结	28
4 变异策略的个体选择	29
4.1 引言	29
4.2 变异策略的改进方法	30
4.3 基于适应度和多样性的变异策略	31
4.3.1 算法动机	31
4.3.2 算法流程	32
4.4 实验结果与分析	35
4.5 本章小结	38
5 基于邻域的差分进化算法	39
5.1 引言	39

5.2 经典的基于邻域的差分进化算法	39
5.3 基于适应度划分邻域的差分进化算法	41
5.3.1 算法动机	42
5.3.2 基于适应度的邻域划分	43
5.4 实验结果与分析	44
5.5 本章小结	47
6 结论与展望	48
6.1 论文研究总结	48
6.2 展望	49
致 谢	50
参考文献	51
攻读学位期间取得的研究成果	54
声明	

CONTENTS

1 Preface	1
1.1 The Background and Significance of the Subject	1
1.2 Research Status.....	1
1.3 The Outline of the Thesis	2
1.4 The Organazition of the Paper.....	3
2 Related Work	5
2.1 Waveform Inversion	5
2.2 Differential Evolution Algorithms.....	6
2.2.1 The Classical Differential Evolution Algorithms	6
2.2.2 Differential Evolution with Cooperative Coevolutuionary Mutation Operator	10
2.3 The Benchmark Functions from CEC2005	11
2.4 Brief Summary	13
3 Differential Evolution Algorithm with subpopulations	15
3.1 Introduction	15
3.2 The Classical Differential Evolution Algorithm Applied to Waveform Inversion	15
3.3 Differential Evolution with Subpopulations.....	16
3.3.1 The Division of Subpopulations.....	17
3.3.2 The Two-phase Mutation Strategy	18
3.4 Experimental Results and Analysis	20
3.4.1 Numerical Examples	21
3.4.2 Stability of SpDE.....	24
3.4.3 Field Data Examples	26
3.5 Brief Summary	28
4 The Selection of the Individuals in the Mutation Strategy.....	29
4.1 Introduction	29
4.2 The Improvement of Mutation Strategy	30
4.3 The Strategy based on Fitness and Diversity	31
4.3.1 Motivation	31
4.3.2 The Proposed Algorithm	32
4.4 Experimental Results and Analysis	35
4.5 Brief Summary	38
5 The Neighborhood-based Differential Evolution	39
5.1 Introduction	39
5.2 The Classical Neighborhood-based Differential Evolution	39
5.3 The Proposed Algorithm	41
5.3.1 Motivation	42
5.3.2 The Fitness-based Neighborhood.....	43
5.4 Experimental Results and Analysis	44

5.5 Brief Summary	47
6 Conclusions and Suggestions	50
6.1 Conclusions on Completed Work	50
6.2 Suggestions for Future Work.....	49
Acknowledgements	50
References	51
Achievements	54
Declarations	

主要符号表

DE	差分进化算法(Differential evolution)
SpDE	多子种群差分进化算法(Differential evolution with subpopulations)
CEC	进化计算大会(Congress on evolutionary computation)
CCDE	协同变异差分进化算法 (DE with cooperative coevolutionary mutation operator)
NP	种群大小
F	变异系数
CR	交叉率
jDE	一种参数自适应的差分进化算法
SaDE	自适应差分进化算法(self-adaptive DE)
HEDE	高效差分进化算法(Highly efficient differential evolution)
MMDE	多组变异差分进化算法(Multimutation differential evolution)
rank-DE	基于排序变异策略的差分进化算法(DE with ranking-based mutation operators)
d	地震数据
g	正演操作
m	地下介质模型的物理参数
f()	目标函数, 适应度函数
D	个体维数
rand	0 到 1 之间均匀分布的随机数
X_{best}^G	当前代数 G 代中的最优个体
$win_j(t)$	第 j 个子成分对应的窗函数
floor()	下取整函数
$normrnd(a, b)$	期望为 a, 标准差为 b 的高斯分布
SNR	信噪比(Signal to noise ratio)
Maxgen	算法预先设置的最大进化代数
Max_NFFEs	最大的适应度函数计算次数
FDDE	基于适应度和多样性的变异策略的差分进化算法
FNDE	基于邻域的差分进化算法

1 绪论

本章首先简要地介绍差分进化算法的研究背景与研究意义，接着介绍差分进化算法的研究现状，最后介绍本论文的主要研究内容以及本论文的组织结构。

1.1 课题的研究背景与意义

从大到石油天然气勘探这些关系到人类社会持续发展的科技应用，小到我们平常生活中出去旅游如何按照预算安排行程等，优化问题广泛地存在于我们生活中的方方面面。当前，随着科学技术的日新月异，优化问题也越发地复杂。而传统的局部优化方法，如梯度下降等，因其受限于初始模型，而无法很好地应用到强非线性的复杂优化问题中。因此，为了解决这类强非线性的复杂优化问题，差分进化算法这种全局优化算法应运而生。

差分进化算法是一种群体智能的随机搜索算法，受启发于达尔文的生物进化理论，将自然界中各物种间自然选择、优胜劣汰的生存规律引入算法中。差分进化算法由一个初始种群开始，通过不断的变异、交叉、选择，最终选出一个最优的个体作为问题的最终解，而这与自然界中的基因变异、优胜劣汰是相同的。

将自然界中的自然选择看作是优化过程，并将这种优胜劣汰原则引进优化算法的最早研究是1975年Holland提出的遗传算法。在此基础上，Store和Price在1996年首先提出了差分进化算法^[1]。而在接下来的二十多年间，不断有研究者提出新的差分进化算法变体，并将其运用到各个领域中。石油天然气勘探就是一个典型应用案例。石油和天然气是人类社会持续发展的基石，关乎到人类社会的方方面面。然而随着石油天然气的勘探程度越来越大，勘探目标也就随着越发复杂。而如何得到勘探目标复杂地下构造的精细刻画则成了石油天然气勘探的难题。而地震波形反演技术能利用地震波形中含有的一些关于地下介质的信息来构建一个精细的地下介质模型，因此能很好的指导油气勘探。然而在地震波形反演问题中，数据与模型之间的关系往往是强非线性的，而一般的牛顿法、最速下降法等局部优化方法因其受限于初始模型而不能很好地应用到波形反演问题中^[2]。而差分进化算法作为一种全局优化算法，在整个搜索空间内进行搜索，不再受限于初始模型，因此在波形反演问题中得到广泛应用^{[3][4][5][6]}。

差分进化算法因其结构简单易于实现且搜索能力强大，因此被众多研究者广泛应用到许多领域。而且随着越来越多的研究者对差分进化算法的深入研究，差分进化算法必将会被更广更深地运用。

1.2 差分进化算法的研究现状

差分进化算法因其简单高效，自 Store 和 Price 在 1996 年提出差分进化算法后的这二十多年间，越来越多的研究者不断地深入研究差分进化算法，因此提出了许多新颖

而且更为高效的差分进化算法变体，并运用到越来越多的领域，并取得了很好的结果。而这些改进的差分进化算法变体主要可以分为以下三个类型：控制参数自适应，变异策略改进和其他优化算法进行融合。

差分进化算法的参数少，只有种群大小 NP 、变异系数 F 和交叉率 CR 。虽然算法的参数数量少，但每一个参数对算法的性能表现都有着重大影响。早期的差分进化算法凭借经验选择一个固定的参数，整个种群的所有个体共享同一个参数。然而不同的问题所需要的最优参数不一定相同；而且就算是同一个问题，在算法运行的不同阶段，所需的最佳参数都不一定相同；最后，在同一个问题的同一阶段，种群中每个个体的表现迥异，给所有的个体分配同一个参数显然是不合理的。因此选择固定参数时，算法性能表现不佳。Brest 等人在 2006 年提出的 jDE^[7]将变异系数 F 和交叉率 CR 编码进个体，让每个个体拥有自己的参数并让这些参数随着个体进化而进化。Qin 等人在 2009 年提出的 SaDE^[8]让控制参数 F 和 CR 通过前代成功信息不断地自适应调整。Gao 等人在 2012 年提出的 HEDE^[9]根据个体的适应度值将部分个体进行死亡和复活操作，进而控制种群大小 NP 自适应调整。Tanabe 在 2013 年提出的 SHADE^[10]利用历史成功信息对 F 和 CR 进行自适应更新。Tanabe 随后在 2014 年对 SHADE 作出改进，提出了 L-SHADE^[11]，在其中对种群大小 NP 引入了线性递减机制。Wu 等人于 2015 年提出的 CRsADE^[12]将交叉率 CR 从个体层面扩散到子成分层面，每个个体的不同子成分拥有自己的交叉率。

变异操作是差分进化算法中最重要的一项操作，该操作通过将差分向量作为扰动加到一个基个体上，进而产生一个新的变异个体，进而为整个种群引入新的信息，对种群接下来的进化有着重要影响。因此，设计一个高效的变异策略至关重要。Mallipeddi 在 2011 年提出的 EPSDE^[13]中，每个个体从一个策略池中选择一种变异策略，下一代的变异策略根据上一代的进化是否成功而定。Wang 在 2011 年提出的 CoDE^[14]选择了三种变异策略，每种变异策略独立产生一个变异个体，从中选择最优的进入下一代。Gong 在 2013 年提出的 rank-DE^[15]根据个体的适应度值来确定该个体被选择参与变异策略的概率。

差分进化算法与其他优化算法进行融合，顾名思义，就是将差分进化算法的长处与其他优化算法的长处进行融合，以求能在实际应用问题或者测试函数集中取得更好的表现。Zhang 在 2003 年提出的 DEPSO^[16]将差分进化算法与粒子群优化算法进行结合。Das 等人在 2008 年将差分进化算法与模拟退火算法进行结合^[17]。

1.3 论文的主要研究内容

本课题的主要研究内容是差分进化算法的性能改进研究与运用，针对差分进化算法中的关键问题——如何平衡好算法中的开发性与探索性展开，对此进行了深入的学习与研究，并提出了相应的差分进化算法改进变体，并将改进算法运用到波形反演问题和 CEC2005 测试函数集中。该论文的主要研究内容有以下三个方面：

(1) 首先，针对将差分进化算法应用到波形反演问题中耗时过久的问题，我们提出了根据个体到种群中心的距离进行种群结构划分的方法。该算法将整个种群划分为三个子种群，每个子种群肩负起相应的开发性和探索性任务，最后提出一个变异策略来均衡开发性和探索性。在波形反演实验中，该改进算法不仅加快了收敛速度，而且还提升了收敛精度；

(2) 其次，针对传统差分进化算法的变异策略只是从种群中随机选择个体参与变异策略，而没有很好地利用当前的进化信息这个问题，我们提出了综合考虑个体适应度和个体对种群多样性贡献来确定每个个体参与变异策略的概率。考虑的这两方面分别是开发性与探索性的代表，综合考虑这两方面则是平衡算法的开发性与探索性，提升了算法的性能；

(3) 最后，针对传统的差分进化算法只关注占种群小部分的优秀个体，而忽视占种群大多数的一般乃至较差个体，我们提出了一种根据个体适应度值进行个体邻域划分的方法。该方法保证优秀个体的邻域几乎为整个种群，而较差个体的邻域则只由优秀的那一小部分组成，最终每个个体变异策略的个体都从该个体的邻域中选取，这样则能保证把占种群绝大多数的个体进化得更好，以期使整个种群进化得更快，而实验结果也证明了这一点。

1.4 论文组织结构

本论文主要从差分进化算法的性能改进出发，通过设计各种方法以平衡差分进化算法的开发性与探索性，希望提出简单高效的差分进化算法变体，并将其运用到波形反演问题和 CEC2005 测试函数集中。

本论文共有六章，主要组织结构如下：

第一章 绪论。首先介绍了差分进化算法的研究背景与研究意义，接着介绍了差分进化算法的研究现状，最后介绍了本论文的主要研究内容以及组织结构。

第二章 相关工作的回顾。先介绍了波形反演问题的相关原理，然后引出经典差分进化算法，介绍了差分进化算法的变异，交叉和选择三个操作。随后针对经典差分进化算法应用在波形反演问题中的效率不高问题，介绍了解决高维波形反演问题的协同变异差分进化算法（CCDE）。最后介绍了经典的测试函数集——2005 年 IEEE 进化计算大会公布的 CEC2005 测试函数集。

第三章 基于子种群的差分进化算法。首先介绍了应用在波形反演问题中的几种差分进化算法变体，针对差分进化算法应用到波形反演问题中的收敛速度慢问题，我们提出了基于多子种群的差分进化算法。在合成地震数据与实际地震数据的实验中，该算法都能取得较快的收敛速度和较高的收敛精度。

第四章 变异策略的个体选择。介绍了经典的变异策略改进方法，并分析了这些改进算法的不足，提出了一种基于个体适应度和对种群多样性贡献的变异策略，该算法在 CEC2005 测试函数集上优胜于其他同类型的比较算法。

第五章 基于邻域的差分进化算法。介绍了经典的基于邻域的差分进化算法，并分析了这些改进算法的不足。提出了一种基于适应度划分邻域的差分进化算法，每个个体的邻域大小根据个体适应度进行相应的变化。该改进算法在 CEC2005 测试函数集上的实验结果优胜于其他同类型的比较算法。

第六章 结论与展望。简要地总结了本论文所做的主要研究内容，并根据个人对差分进化算法的理解与认识，对下一步的研究工作进行了规划。

2 相关工作回顾

本论文的主要研究内容是高效差分进化算法。差分进化算法最初由 Storn 和 Price^[1]于1996年提出,作为一种简单高效的全局优化算法,吸引了越来越多研究者参与研究,并被应用到众多领域。本论文主要将研究得到的高效差分进化算法应用到波形反演问题和 CEC2005 测试函数集中。因此,在本章接下来将顺序地简要介绍波形反演问题、差分进化算法和 CEC2005 函数集。

2.1 波形反演问题

石油和天然气是人类社会持续发展的基石,关乎着国计民生。随着社会的发展,石油和天然气的需求不断增长,然而随着油气开采程度的越加增大,油气勘探的目标也逐渐向复杂地质结构地区转移,如何精细刻画复杂地下介质则成了制约油气勘探的关键问题。而地震波形反演则能利用地震数据中包含的与地下介质相关的信息来构造一个精准的地下介质模型,因此可以指导油气勘探。

地震波形反演的目的就是构建一个合理的地下介质模型,使得该地下介质模型计算得到的计算地震数据与真实环境中测量得到的实际地震数据拟合最佳。正演是反演的基础,正演从一个给定的模型得到计算地震数据,正演可以表示如下:

$$\mathbf{d} = \mathbf{g}(\mathbf{m}) \quad (2-1)$$

式中, \mathbf{d} 代表了计算地震数据, \mathbf{g} 是正演操作,而 \mathbf{m} 则是地下介质模型的物理参数。

我们的目标就是使得上式求得的计算地震数据与实际测得的地震数据拟合最佳,因此我们设定一个目标函数来评价两者之间的拟合程度,目标函数可以表示如下:

$$f(\mathbf{m}) = \sum_{r=1}^{NR} \int |d_{o,r}(t) - d_{c,r}(t, \mathbf{m})| dt \quad (2-2)$$

式中, NR 表示接收器的数目,而 $d_{o,r}(t)$ 表示第 r 个接收器上的实际地震数据, $d_{c,r}(t, \mathbf{m})$ 则表示通过模型 \mathbf{m} 计算得到的计算地震数据, t 是旅行时。

由 (2-2) 可知, 波形反演问题已经转化为一个已知目标函数的优化问题了, 目标函数的数值越低, 就说明反演出的模型计算得到的计算地震数据与实际数据拟合得越好, 也即反演出的模型越接近真实地下介质模型。但由于目标函数的强非线性, 局部优化方法(牛顿法, 最速梯度下降法等)因其受限于初始模型, 容易陷入局部极值点而无法求得全局最优, 因此局部优化方法在此问题上表现不好。而差分进化算法这类全局优化算法因其在整个搜索空间内搜索, 摆脱了初始模型的限制, 因此在波形反演问题中有很好的性能表现。下一节将要介绍全局优化算法中的差分进化算法。

2.2 差分进化算法

如前所述，波形反演问题可以转化为一个强非线性的优化问题，而全局优化算法在整个搜索空间搜索最优解，不再依赖目标函数对参数的梯度，摆脱了初始模型的限制，理论上可以搜索到全局最优点，因此非常适合求解波形反演问题。差分进化算法作为一种全局优化算法，因其简单高效而被广泛应用，因此本小节主要介绍经典差分进化算法的基本原理以及一种求解高维波形反演问题的改进算法。

2.2.1 经典差分进化算法

差分进化算法于 1996 年被 Storn 和 Price^[1]提出，是一种简单高效的全局优化算法。它由一个随机初始化的种群开始，该种群有 NP 个个体，每个个体是待求解问题的一个解。每个个体有 D 维，第 G 代的第 i 个个体可以表示如下：

$$X_i^G = \{x_{i,1}^G, x_{i,2}^G, \dots, x_{i,D}^G\}, \quad i = 1, 2, \dots, NP \quad (2-3)$$

差分进化算法在整个搜索空间内搜索最优解，我们可以根据待求解问题的相关先验知识将算法的搜索空间设为一个 D 维的参数空间，该空间的上下限由待解问题的先验知识设为 X_{max} 和 X_{min} ：

$$X_{max} = \{x_{max,1}, x_{max,2}, \dots, x_{max,D}\} \quad (2-4)$$

$$X_{min} = \{x_{min,1}, x_{min,2}, \dots, x_{min,D}\} \quad (2-5)$$

因为差分进化算法在整个搜索空间搜索以找到最优解，为了保证后期种群中拥有足够的多样性，在初始化阶段要确保整个种群能均匀分布在整个搜索空间，因此可以按以下方式进行初始化：

$$x_{i,j}^0 = x_{min,j} + rand \times (x_{max,j} - x_{min,j}) \quad (2-6)$$

式中：rand 是 $[0,1]$ 之间的随机数，确保每一维都能均匀地分布在整个搜索空间。

初始化后，种群内的所有个体经过变异、交叉和选择操作的不断迭代，满足停止条件后，种群内最优的个体作为问题的最优解输出。变异、交叉和选择操作表示如下^[18]：

(1) 变异操作

在差分进化算法中，变异操作是最重要的一个操作，它通过在一个基个体上加上扰动向量，最终产生一个新的个体，这个新的个体就为整个种群引入新的信息。而产生新个体的方法叫变异策略，通常以“DE/x/y”的形式进行命名，其中 DE 代表的是差分进化算法（Differential Evolution），x 表示的则是基个体的类型，而 y 则表示差分向量的个数。目前，在差分进化算法中被广泛应用的有以下五种变异策略：

1) DE/rand/1:

$$V_i^G = X_{r1}^G + F \times (X_{r2}^G - X_{r3}^G) \quad (2-7)$$

2) DE/best/1:

$$V_i^G = X_{best}^G + F \times (X_{r1}^G - X_{r2}^G) \quad (2-8)$$

3) DE/rand/2:

$$V_i^G = X_{r1}^G + F \times (X_{r2}^G - X_{r3}^G) + F \times (X_{r4}^G - X_{r5}^G) \quad (2-9)$$

4) DE/best/2:

$$V_i^G = X_{best}^G + F \times (X_{r1}^G - X_{r2}^G) + F \times (X_{r3}^G - X_{r4}^G) \quad (2-10)$$

5) DE/current-to-best/1:

$$V_i^G = X_i^G + F \times (X_{best}^G - X_i^G) + F \times (X_{r1}^G - X_{r2}^G) \quad (2-11)$$

式中, X_i^G 是指当前代数中的目标个体, V_i^G 则是由变异策略新产生的变异个体, X_{best}^G 是当前代数 G 代中的最优个体, $r1,r2,r3,r4,r5$ 是[1,NP]中的互不相等的随机数, 并且与 i 不相等。F 是变异系数, 对差分进化算法的性能有着重要影响。

下图为标准变异策略“DE/rand/1”在二维空间的示意图:

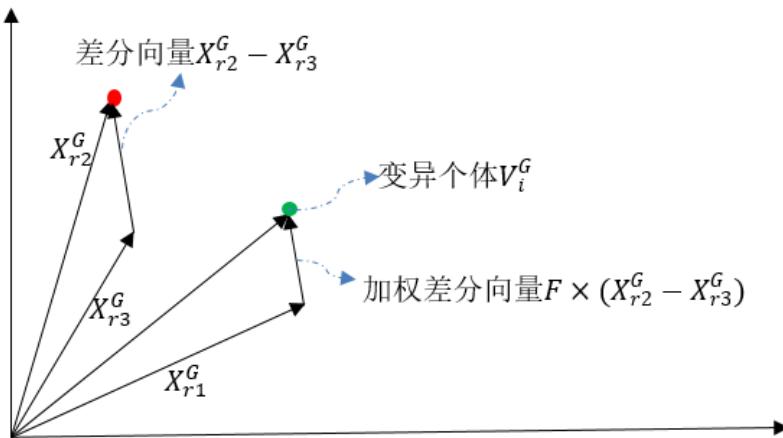


图 2-1 “DE/rand/1” 在二维空间的示意图

(2) 交叉操作

交叉操作主要是用于在原先的目标个体和新产生的变异个体之间交换信息, 它通过一定的概率随机地将目标个体 X_i^G 和变异个体 V_i^G 的某些子成分组合起来, 产生一个新的试验个体 U_i^G 。常用的交叉操作有两种: 均匀交叉与指数交叉。它们可以表示如下:

1) 均匀交叉:

$$u_{i,j}^G = \begin{cases} v_{i,j}^G, & \text{if } (\text{rand} \leq CR \text{ or } j = j_{rand}) \\ x_{i,j}^G, & \text{otherwise} \end{cases} \quad (2-12)$$

式中, $rand$ 是一个[0,1]之间的随机数, 每个个体的每一维都会随机产生一个; j_{rand} 是一个[1,D]之间的随机数, 确保试验个体中至少有一维来自于变异个体。 CR 是交叉率, 控制了试验个体中的维数分别来自目标个体和变异个体的比例, 对算法性能有着重要影响。

为了进一步说明均匀交叉操作, 我们在二维搜索空间中把所有可能的试验个体都

表示在图 2-2，如下图所示：

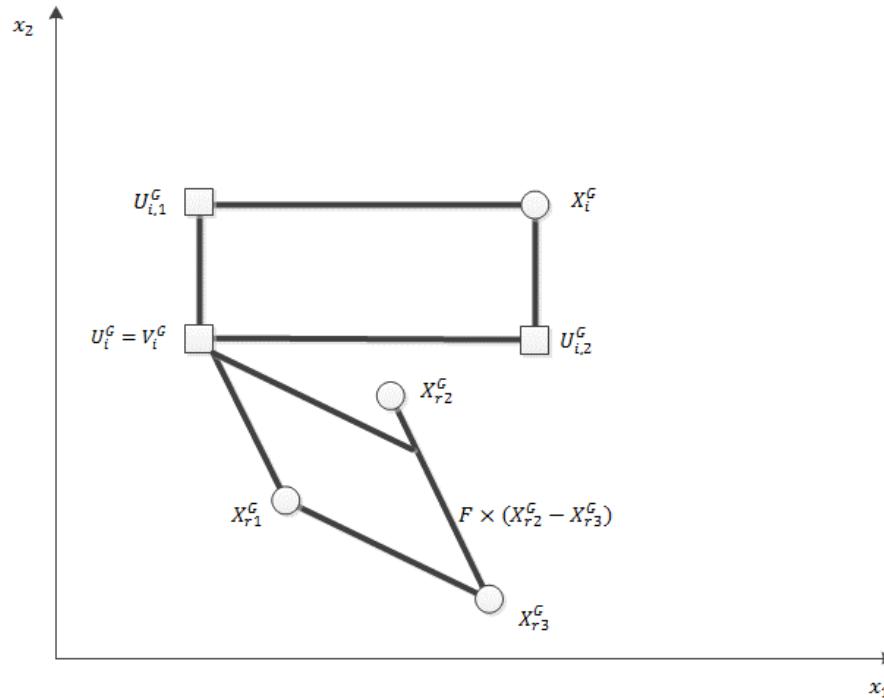


图 2-2 二维搜索空间中均匀交叉可能产生的试验个体示意图

在均匀交叉中，变异个体 V_i^G 和目标个体 X_i^G 按照一定的概率交换子成分以组成试验个体 U_i^G 。按照均匀交叉的操作流程，二维空间中的试验个体 U_i^G 只能有以下三种情况：

- $U_i^G = V_i^G$ ，也即试验个体中的两个子成分都来自变异个体；
- $U_i^G = U_{i,1}^G$ ，也即试验个体中的第一个子成分来自变异个体 V_i^G ，第二个子成分来自目标个体 X_i^G ；
- $U_i^G = U_{i,2}^G$ ，也即试验个体中的第一个子成分来自目标个体 X_i^G ，第二个子成分来自变异个体 V_i^G ；

2) 指数交叉：

$$u_{i,j}^G = \begin{cases} v_{i,j}^G, & \text{if } j = \langle n \rangle_D, \langle n+1 \rangle_D, \dots, \langle n+L-1 \rangle_D \\ x_{i,j}^G, & \text{otherwise} \end{cases} \quad (2-13)$$

其中， L 由以下伪代码产生：

```

L=0;
DO
{
    L=L+1;
} WHILE (rand<CR) AND (L<D)

```

式中： n 是[1,D]中的随机数， D 则是个体的维数， $\langle \rangle_D$ 表示以 D 为底取模。

(3) 选择操作:

选择操作参照自然界中优胜劣汰的自然法则，将上面通过变异操作和交叉操作产生的试验个体与目标个体进行比较以决定哪个个体能进入下一代的进化。选择操作是通过比较两者在目标函数的适应度值来决定哪个进入下一代的，具体如下：

$$X_i^{G+1} = \begin{cases} U_i^G, & \text{if } f(U_i^G) \leq f(X_i^G) \\ X_i^G, & \text{otherwise} \end{cases} \quad (2-14)$$

式中的 $f()$ 即为待最小化的目标函数，也即是差分进化算法中的适应度函数（fitness function）。

差分进化算法的停止条件可以由通过以下几种方式来确定：1) 设置一个固定的最大进化代数 G_{max} ，而这个最大进化代数具体取值多少则根据目标函数的复杂程度来决定；2) 当种群中最好个体的适应度函数值连续多代没有变化；3) 预先设置一个具体的适应度函数值，当最好的适应度函数值达到这个值后即可停止算法。

经典的差分进化算法的流程图如下：

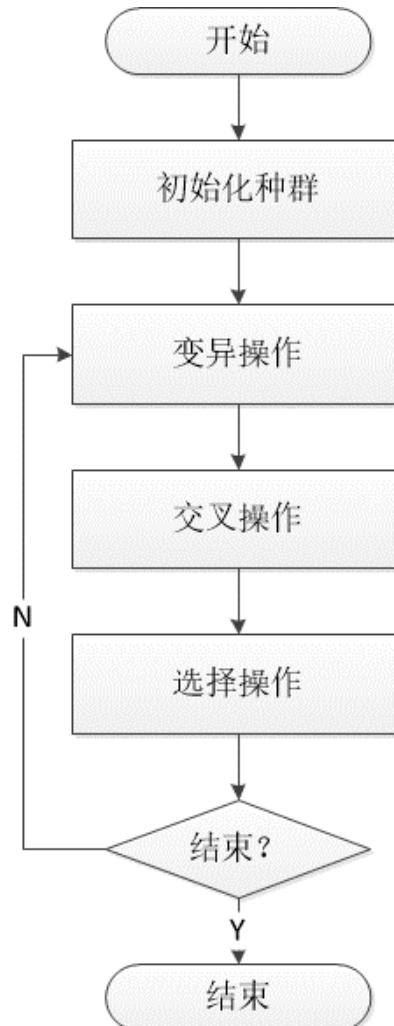


图 2-3 差分进化算法流程图

2.2.2 协同变异差分进化算法 (CCDE)

传统差分进化算法常常忽略了个体不同子成分间的巨大差异，而将所有的子成分都当作一个个体整体看待。当将经典的差分进化算法运用到复杂的、高维的波形反演问题中时，其算法性能会随着搜索空间的维数增加而急速下降，因此很难取得很好的结果。为了能将差分进化算法高效地应用到波形反演问题中，Wang 和 Gao 在 2012 年提出了协同变异差分进化算法 (CCDE) [19]。CCDE 借鉴了协同进化的思想，将整个个体划分为若干个子成分，并设计了局部适应度函数来评价每个子成分的表现。当 CCDE 应用到波形反演问题中时，局部适应度函数表示如下：

$$Lf_j(m) = \sum_{r=1}^{NR} \int |[d_{o,r}(t) - d_{c,r}(t, m)] \times win_j(t)| dt \quad (2-15)$$

式中： $win_j(t)$ 为和第 j 个子成分对应的窗函数。

最后将局部适应度函数的概念引入变异操作中，以局部适应度函数来引导每个子成分的进化方向，并将该变异策略命名为“DE/rand-local-best/1”。该变异策略主要流程如下：先如传统变异策略“DE/rand/1”所做的，从种群中随机选取三个个体 X_{r1}^G ， X_{r2}^G 和 X_{r3}^G ，传统的变异策略是在整个个体的层面上进行变异的，而“DE/rand-local-best/1”变异策略则扩散到子成分层面上进行变异，对于随机选出的三个个体的第 j 个子成分 $X_{r1,j}^G$ ， $X_{r2,j}^G$ 和 $X_{r3,j}^G$ ，先将它们进行先序排序得到序列 $[X_{r11,j}^G, X_{r21,j}^G, X_{r31,j}^G]$ ，再根据它们的局部适应度函数值 $Lf_j(X_{r11,j}^G)$ ， $Lf_j(X_{r21,j}^G)$ ， $Lf_j(X_{r31,j}^G)$ ，可以得到以下四种可能的分布情况：

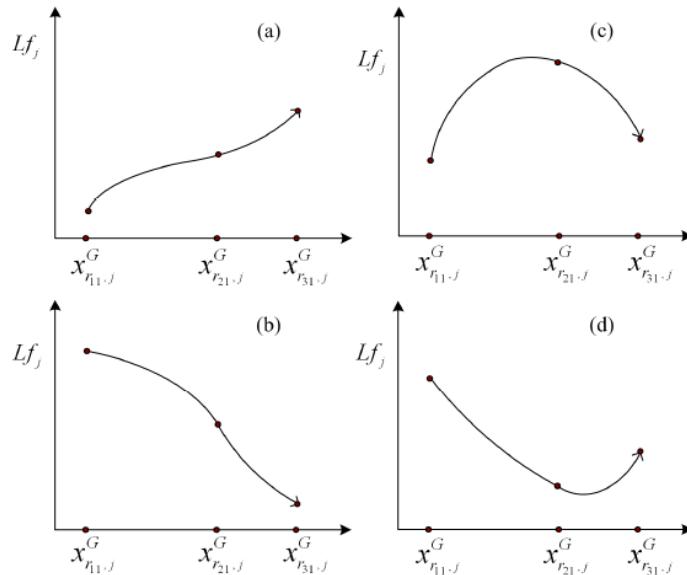


图 2-4 排序后的子成分按其局部适应度函数值可能的四种分布情况。

根据不同的分布情况，可以相应地引导子成分的变异方向。图 2-4(a)的分布情况可以用增函数来进行拟合，而增函数则指示了 $X_{r11,j}^G$ 的左边可能存在适应度函数值更低的子成分；图 2-4(b)的分布情况可以用减函数来进行拟合，而减函数则指示了 $X_{r31,j}^G$

的右边可能存在适应度函数值更低的子成分；图 2-4(c)的分布情况可以用凸函数来进行拟合，而凸函数则指示了 $X_{r21,j}^G$ 的左边或者右边都可能存在适应度函数值更低的子成分；图 2-4(d)的分布情况可以用凹函数来进行拟合，而凹函数则指示了其极小值点处可能存在适应度函数值更低的子成分，但出于简便性考虑，选取三个子成分的加权平均是一个更好的选择。因此“DE/rand-local-best/1”变异策略可以表示如下：

$$V_{i,j}^G = \begin{cases} X_{r11,j}^G - F \times |X_{r2,j}^G - X_{r3,j}^G|, & (a) \\ X_{r31,j}^G + F \times |X_{r2,j}^G - X_{r3,j}^G|, & (b) \\ X_{r21,j}^G + F \times (X_{r2,j}^G - X_{r3,j}^G), & (c) \\ p_1 X_{r11,j}^G + p_2 X_{r21,j}^G p_1 X_{r31,j}^G, & (d) \end{cases} \quad j = 1, 2, \dots, J \quad (2-16)$$

式中：

$$p_1 = \frac{p_4}{L f_j(X_{r11,j}^G)} \quad (2-17)$$

$$p_2 = \frac{p_4}{L f_j(X_{r21,j}^G)} \quad (2-18)$$

$$p_3 = \frac{p_4}{L f_j(X_{r31,j}^G)} \quad (2-19)$$

$$p_4 = 1 / (L f_j(X_{r11,j}^G)^{-1} + L f_j(X_{r21,j}^G)^{-1} + L f_j(X_{r31,j}^G)^{-1}) \quad (2-20)$$

CCDE 将局部适应度函数值引入变异操作中，根据不同的子成分分布情况相应地给出变异策略，为子成分指出了变异方向，大大提高了其变异成功率，提高了差分进化算法应用在波形反演问题中的效率。

2.3 CEC2005 测试函数集

CEC2005 测试函数集^[20]是 2005 年 IEEE 进化计算大会（IEEE International Conference on Evolutionary Computation, CEC）公布的 25 个标准测试函数，主要用来评价各种优化算法的性能。这 25 个函数主要分为四个种类：5 个单模函数（unimodal function）F1-F5，7 个基本多模函数（basic multimodal function）F6-F12，2 个扩展多模函数（expanded multimodal function）F13 和 F14，11 个混合组合函数（hybrid composition function）F15-F25。下面将在每个类型中选择一个函数进行描述，其余函数可以在相应文献^[20]中查看。

(1) 单模函数的 F1：

$$F_1(\mathbf{X}) = \sum_{i=1}^D z_i^2 + f_bias, \mathbf{Z} = \mathbf{X} - \mathbf{O}, \mathbf{X} = [x_1, x_2, \dots, x_D] \quad (2-21)$$

式中：D 是变量的维数，全局最优点 $\mathbf{X}^* = \mathbf{O} = [o_1, o_2, \dots, o_D]$ ，该处的全局最优值为 $F_1(\mathbf{X}^*) = f_bias$ 。下图为 D=2 时的 F1 函数图像：

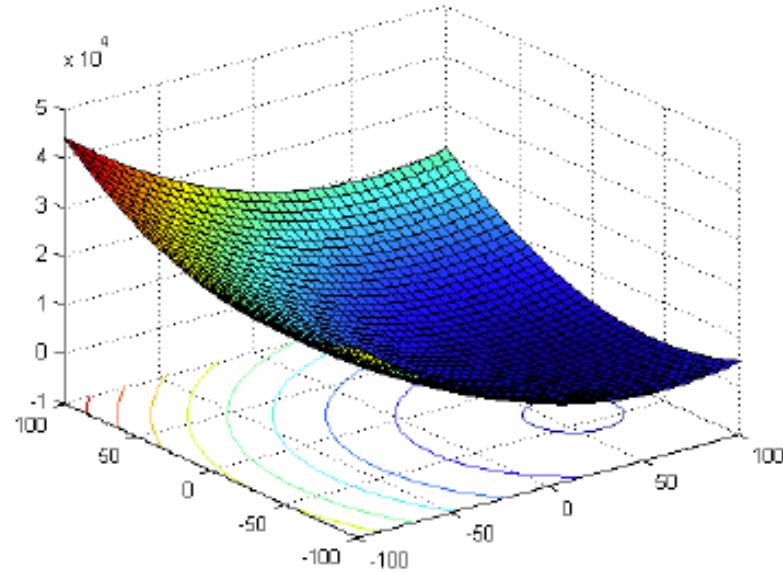


图 2-5 F1 的函数图像

(2) 基本多模函数 F6:

$$F_6(\mathbf{X}) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1}^2)) + (z_i - 1)^2 + f_bias, \mathbf{Z} = \mathbf{X} - \mathbf{O} + \mathbf{1}, \quad (2-22)$$

式中: D 是变量的维数, 全局最优点 $\mathbf{X}^* = \mathbf{O} = [o_1, o_2, \dots, o_D]$, 该处的全局最优值为 $F_6(\mathbf{X}^*) = f_bias$ 。下图为 D=2 时的 F6 函数图像:

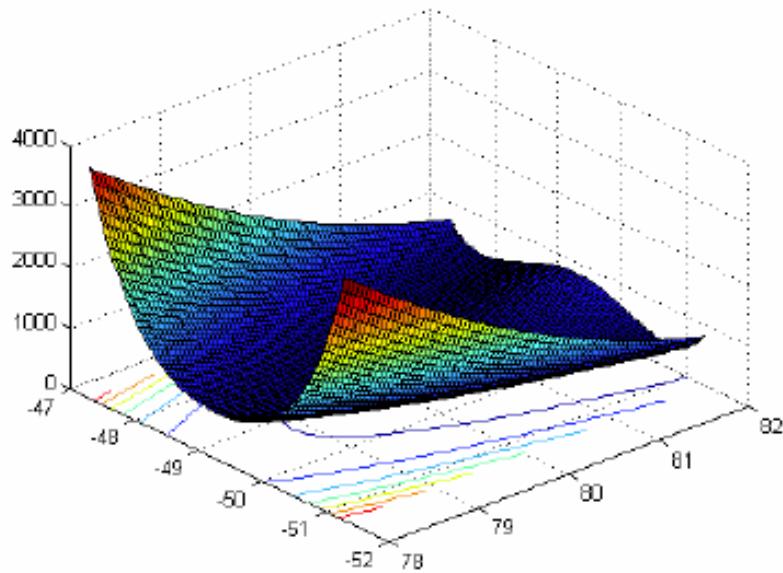


图 2-6 F6 的函数图像

(3) 扩展多模函数 F13:

$$\begin{aligned} F_{13}(\mathbf{X}) = & F8(F2(z_1, z_2)) + F8(F2(z_2, z_3)) + \dots \\ & + F8(F2(z_D, z_1)) + f_bias \end{aligned} \quad (2-23)$$

式中: $\mathbf{Z} = \mathbf{X} - \mathbf{O} + \mathbf{1}$, D 是变量的维数, 全局最优点 $\mathbf{X}^* = \mathbf{O} = [o_1, o_2, \dots, o_D]$,

该处的全局最优值为 $F_{13}(\mathbf{X}^*) = f_bias$ 。下图为 D=2 时的 F13 函数图像：

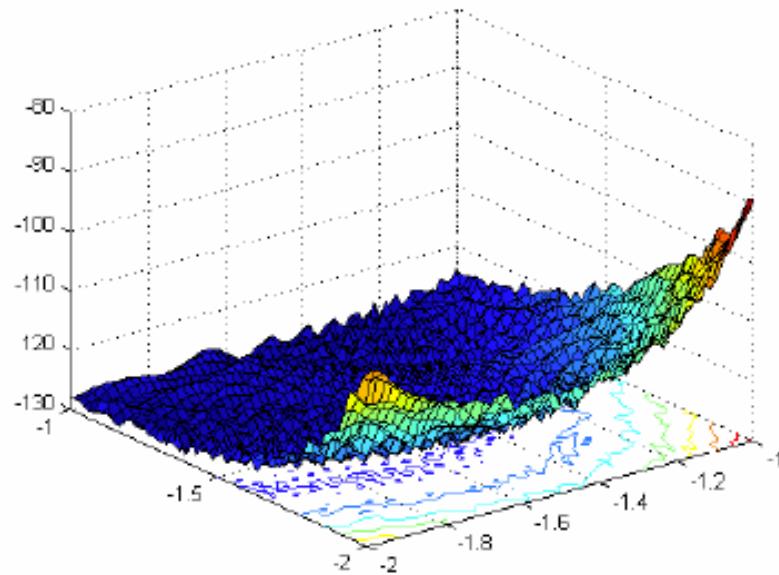


图 2-7 F13 的函数图像

(4) 混合组合函数 F15:

F15 由 10 个基本函数加权组合而成, 具体描述可参考相关文献, 下图为 D=2 时的 F15 函数图像:

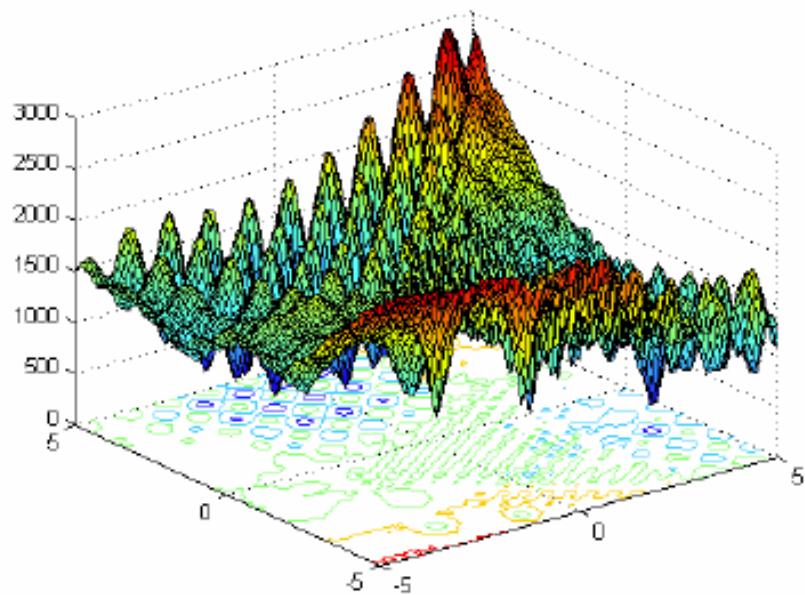


图 2-8 F15 的函数图像

我们将在 CEC2005 测试函数集上验证我们的改进算法的性能。

2.4 本章小结

本章首先回顾了波形反演问题的基本概念, 指出波形反演问题的目的就是构建一

个地下介质模型使得计算地震数据与实际地震数据拟合最佳，进而介绍地震波形反演中的目标函数来评价两者之间的拟合程度，由此波形反演问题即可转化为求解目标函数极值的最优化问题。

接着，本章简要介绍了经典差分进化算法，它主要由变异、交叉和选择三个操作组成，其中变异操作是其中最重要的一个操作，它产生了一个新的个体进而为种群引入新的信息。但由于波形反演问题是高维的复杂优化问题，经典差分进化算法并不能很好地应用于波形反演问题中，因此接着介绍聚焦于高维波形反演问题的协同差分进化算法（CCDE），CCDE 将个体分解为若干个子成分，并设计一个局部适应度函数来评价每个子成分的具体表现，并用局部适应度函数在变异策略中引导子成分的变异方向。

最后，本章简要介绍了 2005 年 IEEE 进化计算大会公布的 CEC2005 测试函数集，它由 25 个函数组成，分别属于四种类型，且函数维数可以扩展，是评价优化函数性能的很好的测试函数集，本论文接下来的工作也在该测试函数集中进行测试。

3 基于子种群的差分进化算法

差分进化算法作为一种简单高效的全局优化算法，被广泛地应用到众多领域，其性能与算法开发性与探索性的平衡密切相关。因此，提出相关方法平衡差分进化算法的开发性与探索性能极大地提高差分进化算法的优化效率，特别是应用在波形反演问题中。本章首先介绍一些已有的应用在波形反演问题中的差分进化算法，随后介绍本章提出的基于个体与种群中心的距离进行子种群划分的改进差分进化算法，并将其应用到波形反演问题中。

3.1 引言

地震波形反演是油气勘探中解决复杂储存层勘探的核心技术之一，然而由于波形反演问题中的模型与数据之间的强非线性关系，导致牛顿法、最速梯度下降等局部优化方法因其受限于初始模型而不能很好地解决该问题^[21]。而差分进化算法作为一种全局优化算法，在整个搜索空间寻优，不再受限于初始模型，因此是解决波形反演问题的一个很好解决方案。然而要得到一个精细的地下介质模型，则必须对地下介质进行尺度细小的划分，而这必然使得地下介质模型的物理参数的数量非常巨大，也即待优化问题的维数非常高，进而差分进化算法的搜索空间也就随着成几何级数增加。而差分进化算法是在整个搜索空间中寻优，搜索空间的变大也就意味着计算量的变大和收敛速度的变慢。因此如何提升差分进化算法的收敛速度一直是大家的研究重点。基于此，许多研究者都提出了改进收敛速度的差分进化算法变体，并将其运用到波形反演问题中。

同时，探索性与开发性的平衡一直是影响差分进化算法性能的关键因素^[22]。通常而言，探索性即是算法搜索当前搜索空间的不同区域以期能找到尚未搜到的更好解的能力，而开发性则是算法在当前已有解的基础上提升当前解质量的能力。当差分进化算法偏向开发性时，算法会比较容易陷入局部极值点，进而导致早熟。另一方面，若算法过于偏向探索性，虽然算法不会陷入局部极值点，但算法的收敛速度会极慢。因此为了取得更好的性能，我们必须给差分进化算法提供开发性与探索性的较好平衡。开发性在已有解的基础上提升质量，而探索性则在尚未搜索区域寻找更好的解，本章依据个体到种群中心的距离将整个种群划分为三个子种群，由内而外地分别负责开发性与探索性，最终提出一个新的变异策略将三个子种群联系起来最终实现开发性与探索性的平衡。

3.2 应用在波形反演的差分进化算法

由于差分进化算法是一种简单高效的全局优化算法，特别适合求解波形反演这种强非线性的优化问题，因此有许多研究者将差分进化算法应用到波形反演问题中。为

了将差分进化算法应用到波形反演问题中，地下介质模型的物理参数表示为差分进化算法中的一个 D 维个体，每个个体都是问题的一个解，第二章提到的目标函数则用来衡量每个模型参数的表现。Kvasnicka^[23]等在 2001 年利用差分进化算法来对地震震源进行定位。Steve^[24]等在 2010 年利用差分进化算法来反演地壳结构。针对高维波形反演问题，Wang^[19]等人在 2012 年提出协同变异差分进化算法（CCDE），CCDE 借鉴了协同进化的思想，将高维复杂问题分解为一系列互相关联的子问题，将算法中每个个体分解为若干个子成分，并设计局部适应度函数来评价每个子成分的质量，并在变异策略中用局部适应度函数值来引导每个子成分的进化方向。CCDE 有效地解决了高维波形反演问题，但其收敛速度依旧不太理想。因为随着进化进程的推进，每个个体的进化情况都不同，因此每个个体对种群进化的贡献有所不同，适量删减对种群进化贡献少的个体有利于减少计算量，而且进化的进程是动态的，有可能之前被删除的个体在种群后续的进化中会发挥很大的作用，因此有必要在被删除的个体中进行复活操作使其继续加入种群的进化。因此 Gao 等人在 CCDE 的基础上提出的 HEDE^[9]引入了根据个体进化信息来对个体进行删除和复活的机制以自适应调整种群大小，进而减少计算量以提高收敛速度。Pan 等人考虑到在高维反演问题中由于种群中个体维数一般较高，而常用的差分进化算法都以个体为单位分配 F 和 CR 等参数，然而高维反演问题中个体每一维之间的差异较大，进化程度也不同，因此他们在 2015 年将交叉率 CR 扩散到子成分层面上，进一步提升差分进化算法应用到波形反演问题中的收敛速度^[12]。常用的差分进化算法中的每个个体在一次变异操作时都只产生一个变异个体，而 Gao 等人在 2016 年提出的 MMDE^[25]指出多组变异机制比现在通用的单组变异机制好，MMDE 中每个个体在变异操作阶段都会产生多个临时变异个体，最后将它们加权相加以得到最终的变异个体，在波形反演问题取得了很好的收敛速度和收敛精度。

3.3 基于多子种群的差分进化算法

在本小节，我们将提出一种基于多子种群的差分进化算法，。该算法依据个体到种群中心的距离来划分子种群，因此子种群的划分能很好地表示整个种群在搜索空间的分布，每个个体在所属的子种群内进行变异，最后由一个变异策略将所有子种群的变异个体进行加权相加，实现开发性与探索性的平衡。

对于差分进化算法而言，开发性与探索性的平衡是影响算法性能的最重要因素^[26]。开发性是在已知解的基础上进行解质量的提升，而探索性则是搜寻搜索空间中尚未搜索到的区域以期能找到更好的解。而这两个性能与种群进化过程的两个方面极为相似，随着算法进化进程的推进，种群慢慢收缩，而种群中心慢慢地向全局最优点移动。于是我们可以让靠近种群中心的个体负责开发性，它们在现有解的基础上进行解质量的提升；远离种群中心的个体则负责探索性，它们在种群外围进行搜索以期找到更好的解。于是就有了下面的基于个体到种群中心的距离来进行子种群划分的多子种群差分进化算法（Differential Evolution with Subpopulations, SpDE）。接下来会分为两部分对我们算

法进行描述，第一部分是关于子种群的划分，而第二部分则是关于两阶段的变异策略。

3.3.1 子种群的划分

如上所述，开发性与探索性是差分进化算法性能的重要影响因素，我们根据个体到种群中心的距离将整个种群划分为三个子种群，每个子种群都在平衡开发性与探索性中发挥自己的作用。

为了划分子种群，我们将种群中心 X_C^G 定义如下：

$$X_C^G = \frac{1}{NP} \times \sum_{i=1}^{NP} X_i^G \quad (3-1)$$

式中： NP 为整个种群的个体数目。由上式可以看出，种群中心即为所有个体的平均。

然后我们计算每个个体 X_i^G 到种群中心 X_C^G 的欧氏距离 D_i ，计算公式如下：

$$D_i = \sqrt{\sum_{j=1}^D (X_{i,j}^G - X_{C,j}^G)^2} \quad (3-2)$$

当所有个体到种群中心的距离 D_i 计算完毕后，我们计算所有个体的距离的平均值 D_{av} 。然后以该距离平均值 D_{av} 为阈值可以将整个种群划分为两部分，在每一部分，我们计算该部分个体到种群中心距离的平均值，分别记为 D_{av1} 和 D_{av2} ，其中 D_{av1} 是较靠近中心的那一部分的平均值。然后可以根据每个个体到种群中心的距离将整个种群划分为三个子种群，相应的划分规则如下：

$$X_i^G \rightarrow \begin{cases} S_1, & \text{if } D_i < D_{av1} \\ S_2, & \text{if } D_{av1} \leq D_i \leq D_{av2} \\ S_3, & \text{if } D_i > D_{av2} \end{cases} \quad (3-3)$$

子种群的结构示意图如图 3-1 所示，同心圆的圆心即为种群中心 X_C^G 。在子种群 S_1 的个体是整个种群中最为靠近种群中心的个体，因此他们可能有最小的适应度函数值，也即表现最佳。当该子种群的个体参与变异策略时，新产生的变异个体会在现有最好的解（ S_1 中的个体）的附近进行搜索，因此子种群 S_1 中的个体肩负着开发性的任务，希望能在现有解的基础上提升解的质量。而在子种群 S_3 中的个体则是整个种群中离种群中心最远的个体，因为它们远离种群中心，因此在变异操作中有很大的概率能搜索种群以外的区域，因此它们肩负着探索性的任务，以期能在尚未搜到的区域找到更好的解。子种群 S_2 的个体则两者都兼顾到。而将在下一节介绍的两阶段变异策略则联合这三个子种群来更好地平衡开发性与探索性。

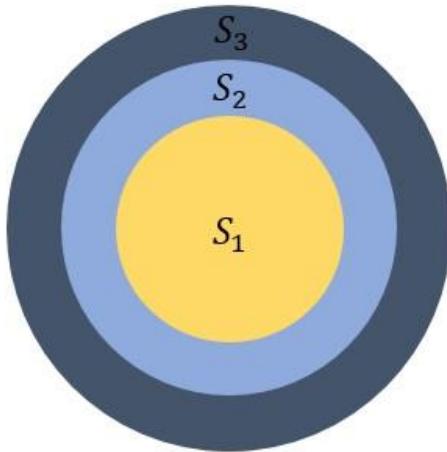


图 3-1 子种群结构示意图

我们在每一代都根据 (3-1)、(3-2) 和 (3-3) 来更新种群中心和子种群的划分。

在传统的差分进化算法中，参与变异策略的个体都是从整个种群中随机选出的，而且每次变异时只产生一个变异个体，而这没能有效地利用整个种群的进化信息。因此接下来我们将介绍新的分两阶段的变异策略。

3.3.2 两阶段变异策略

变异操作是差分进化算法中最重要的一个操作，它为种群产生新的变异个体，提供新的信息，上一节划分好的子种群结构天然地与开发性和相关性的平衡相匹配。因此这一小节，主要针对已划分好的子种群结构，从平衡开发性和探索性这个目标出发，为子种群结构设计一套匹配的变异策略。如上小节所述，划分好子种群后，我们将进入变异操作，我们改进的变异策略可以分为以下两个阶段：

1) 合作变异阶段

Gao 在 MMDE^[25]中指出，多变异机制比传统的单变异机制更为有效。在这个变异阶段，为了提高搜索效率，所有的子种群一起合作以产生最终的变异个体。首先，我们在每个子种群 S_j 中随机选择三个个体 $X_{j,r1}^G, X_{j,r2}^G, X_{j,r3}^G, j=1,2,3$ 。然后，运用协同变异差分进化算法 (CCDE)^[19]中的变异策略，我们可以在每个子种群 S_j 得到一个临时变异个体 $V_j^G, j=1,2,3$ 。最后我们按照下式得到最终的变异个体：

$$V_i^G = (V_1^G + V_2^G + V_3^G)/3 \quad (3-4)$$

当当代最好个体的适应度函数值与上代最好个体的适应度函数值的差值小于 0.4 时，变异策略将从当前阶段转到下一个阶段。我们之所以选择 0.4 作为阶段转换的阈值，是因为阶段转换应该发生在一个合理的时间。首先，我们应该确保子种群 S_1 中的个体应该足够优秀才转向第二阶段，否则它只是浪费 CPU 时间去搜索一些适应度函数值不太好的个体的邻域，而这是事倍功半的，因此阶段转换不能过早。其次，如果阶段转换发生得太晚，则算法的收敛精度可能不太好，因为留给变异策略第二阶段的进化代数太少了。

2) 独立变异阶段

在这个阶段，我们不再用 (3-4) 来产生最终变异个体，而是每个子种群独立地产生变异个体。经过第一个变异阶段后，子种群 S_1 中的个体应该足够优秀地定位到接近全局最优点的区域。因此，对于子种群 S_1 中的个体 X_i^G ，我们从中随机选择 z 维，然后在选定的 z 维中加以一个很小的扰动以来搜索当前解的附近以期能提升当前解的质量，我们按照下式来决定 z 的取值：

$$z = 1 + \text{floor}(\alpha \times \text{rand} \times D) \quad (3-5)$$

式中： $\text{floor}()$ 为下取整函数， rand 是 $[0,1]$ 之间的随机数， D 是个体的维数。为了保证 z 是一个较小的数值，我们设置 $\alpha = 0.1$ 。因此在这个变异策略中，子种群 S_1 中的个体 X_i^G 只会在很少的若干维上有所改变，我们的变异策略可以表示如下：

$$v'_{i,j} = x_{i,j}^G + \text{normrnd}(0, 0.05) \quad (3-6)$$

$$v_{i,j}^G = \begin{cases} v'_{i,j}, & j \text{ is one of the chosen } z \text{ dimensions} \\ x_{i,j}^G, & \text{otherwise} \end{cases} \quad (3-7)$$

式中： $\text{normrnd}(0, 0.05)$ 是期望为 0，标准差为 0.05 的高斯分布。

对于子种群 S_2 和 S_3 中的个体，我们沿用 CCDE 中的变异策略来产生变异个体。

另外，我们在独立变异阶段对交叉操作做了一个改进。首先，在每一代我们创建一个新的个体 $X_{bestsub}^G$ ，这个个体由当代最优秀的子成分构成，而优秀与否则是由式 (2-15) 定义的局部适应度函数来评价。在传统差分进化算法中，交叉操作通过混合目标个体和变异个体的子成分来产生试验个体。然而，传统的交叉操作并不能高效地利用当代进化信息，因为它只是简单地通过一定的概率随机抽取目标个体或者变异个体的子成分组成试验个体而已。而 $X_{bestsub}^G$ 则是包含当代最好的子成分的个体，如果它被运用到交叉操作时，则这些最优秀的子成分则会快速地扩散到整个种群，并因此而加快整个种群的进化进程，加快收敛速度。因此我们将利用 $X_{bestsub}^G$ 的信息来创建一个新的交叉操作，并将其命名为 bin-bestsub，该交叉操作可以表示如下：

$$u_{i,j}^G = \begin{cases} v_{i,j}^G, & \text{if } (\text{rand} \leq CR \text{ or } j = j_{rand}) \\ x_{bestsub,j}^G, & \text{otherwise} \end{cases} \quad (3-8)$$

式中： $x_{bestsub,j}^G$ 是指 $X_{bestsub}^G$ 的第 j 维， $v_{i,j}^G$ 则是指第 G 代中第 i 个变异个体的第 j 维， $u_{i,j}^G$ 则是指第 G 代中第 i 个试验个体的第 j 维， rand 是 $[0,1]$ 之间的一个随机数， j_{rand} 是 $[1,D]$ 之间的一个随机数，主要是为了保证试验个体中至少有一维是来自变异个体。与标准的均匀交叉 (2-12) 相比，我们提出的 bin-bestsub 交叉操作只是将原先的目标个体替代为包含当代最好子成分的 $X_{bestsub}^G$ ，而这能加快最优秀的子成分在整个种群的扩散速度，进而加快种群收敛速度。

尽管如此，为了保持种群的多样性防止算法早熟，我们依旧采用传统的均匀交叉 (2-12)。因此，对于每次交叉操作，我们采用标准均匀交叉和 bin-bestsub 交叉操作的机会

均为 50%。

综合以上算法描述内容，SpDE 的算法流程图可以表示如下：

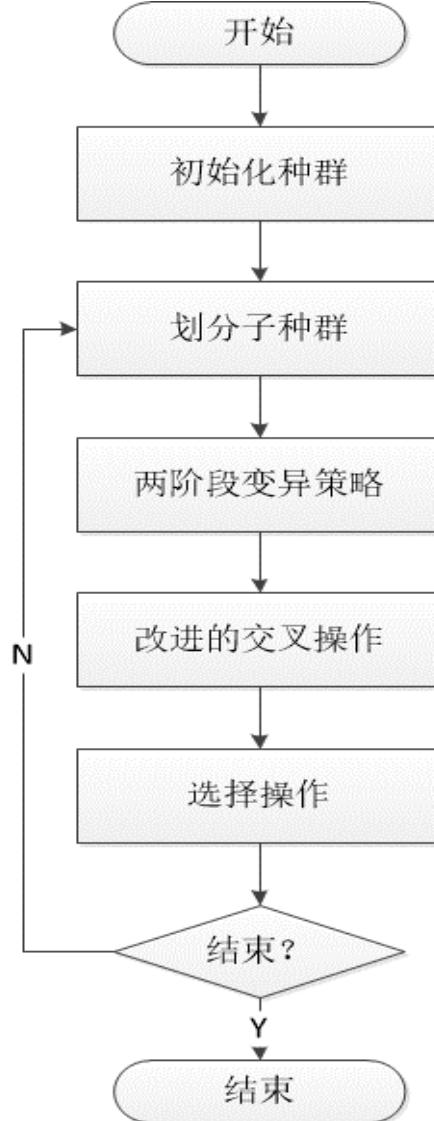


图 3-2 SpDE 的算法流程图

由上图的 SpDE 算法流程图可以看出，相对于经典的差分进化算法来说，SpDE 只是增加了划分子种群这一步，依然保持着差分进化算法本身结构简单的优点。下面将 SpDE 应用到波形反演问题中以验证其高效性。

3.4 算法在波形反演中的应用

为了验证我们算法 SpDE 的性能，我们将在合成地震数据和实际地震数据中进行实验，在这一节中，所有的实验都采用由式（2-2）定义的全局适应度函数和由式（2-15）定义的局部适应度函数。

3.4.1 SpDE 应用于合成地震数据

本小节实验中采用的速度模型如图 3-3 中的红线所示，共有 200 层，只对速度进行估计，因此共有 200 个速度参数需要估计，也即差分进化算法中的个体为 200 维。对真实速度模型进行滤波，并在其低频成分上 $\pm 0.8\text{km/s}$ 作为 SpDE 的搜索空间，正如图 3-3 中的蓝线所示。所有的实验都在一个具有 200 层的模型上运行，而且 SpDE 中个体的子成分由一层组成。种群大小设置为 $NP=100$ ，算法终止条件设置为最大进化代数 $G_{max} = 500$ 。

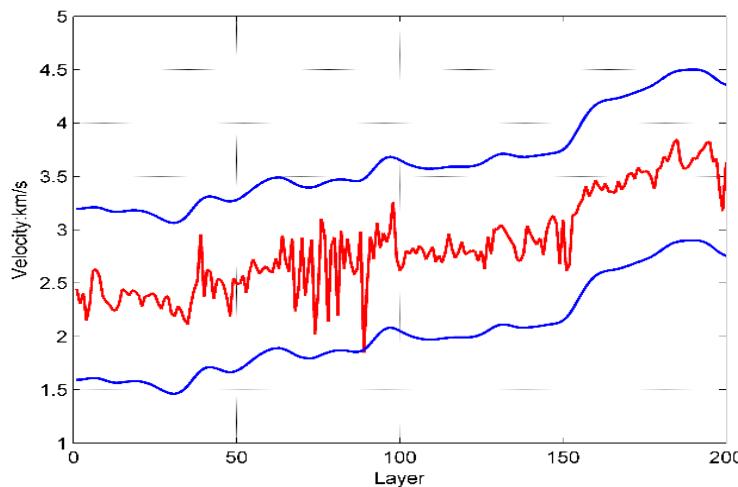


图 3-3 真实速度模型（红线）和搜索空间的上下限（蓝线）

差分进化算法的收敛速度一直是研究者的研究重点，特别是将算法运用到波形反演问题中，收敛速度更显得重要。为了验证 SpDE 的性能改进，我们将与 MMDE (multi-mutation differential evolution)^[25], CCDE (cooperative coevolutionary differential evolution)^[19] 和 SaDE-MMTS (self-adaptive differential evolution with modified multi-trajectory search)^[27] 进行比较，所有实验结果都是 35 次独立实验后的平均结果。我们同时测试了收敛速度和收敛精度，实验结果分别表示在图 3-4 和表 3-1 中。

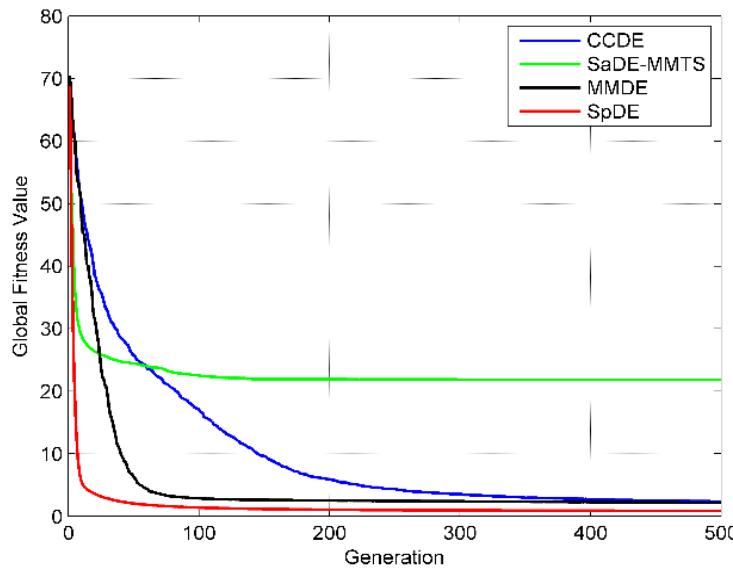


图 3-4 CCDE, MMDE, SaDE-MMTS 和 SpDE 的收敛曲线图, 纵轴是每一代的最优适应度值, 横轴是代数。

表 3-1 CCDE, SaDE-MMTS, MMDE 和 SpDE 在不同代数时的最优适应度值

	第 50 代	第 500 代
CCDE	25.6788	2.3715
SaDE-MMTS	24.4501	21.8296
MMDE	6.3191	2.1747
SpDE	2.0560	0.8746

图 3-4 显示的是每种算法的收敛曲线表示图, 它表现了每种算法的最优适应度值随着进化代数的增加而如何变化。由图中可以看出, 我们的算法 SpDE 是所有比较算法中收敛速度最快的, 而且 SpDE 的最终适应度函数值比其他算法要好。由图中可以看出 SaDE-MMTS 的最终适应度函数值非常大, 这意味着它并不能很好地解决波形反演问题。由表 3-1 可以看出, MMDE 和 CCDE 在第 50 代时的最佳适应度函数值分别为 6.3191 和 25.6788, 而 SpDE 在第 50 代时的最佳适应度函数值已经是 2.0560, 甚至比 CCDE 和 MMDE 在第 500 代的最佳适应度函数值还要好。而这意味着 SpDE 能在较短时间内取得更高的收敛精度, 算法的收敛速度大大地提升了。

我们的目标是构建一个精确的地下介质模型, 因此接下来我们将比较各种算法反演出来的速度模型与真实速度模型的拟合程度, 它们的比较图如图 3-5 所示。

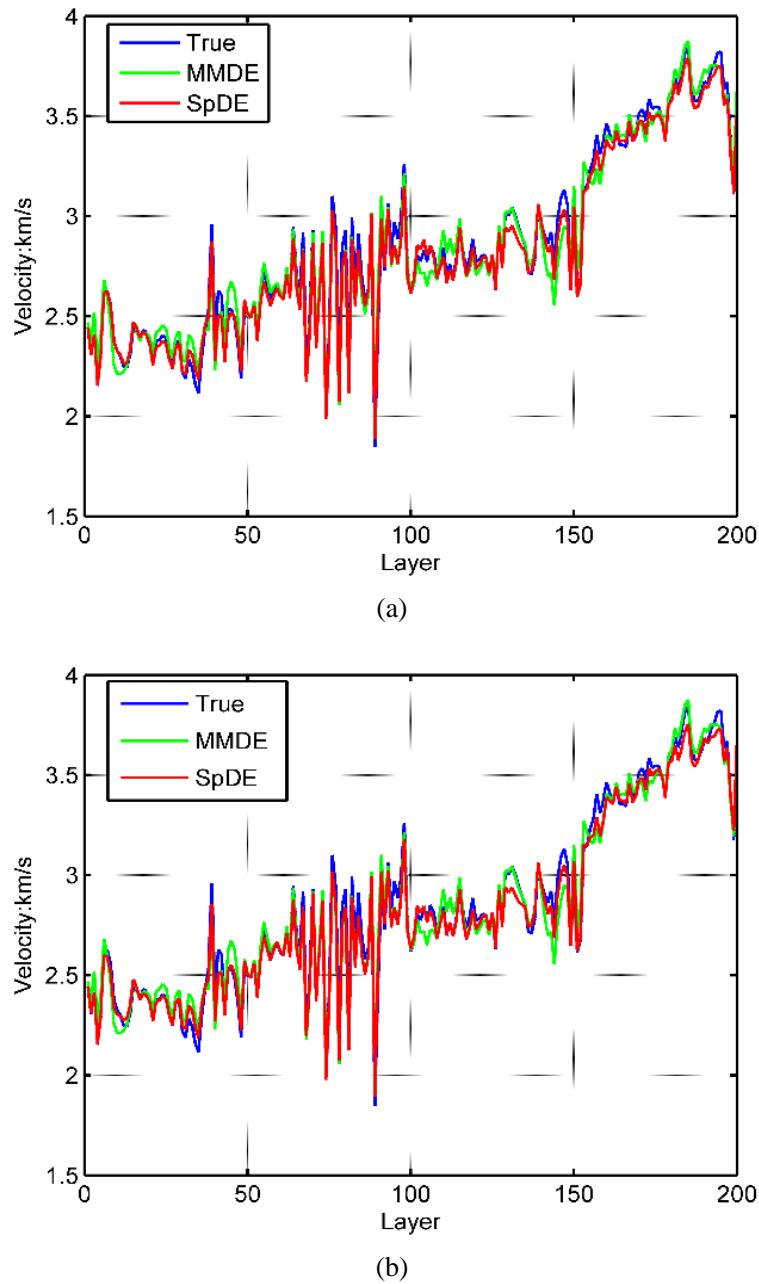


图 3-5 不同终止条件下，MMDE 和 SpDE 反演出的速度模型与真实模型的比较。(a) MMDE 和 SpDE 都终止于 500 代。(b) MMDE 终止于 500 代，而 SpDE 终止于 50 代。

图 3-5 为 MMDE 和 SpDE 在不同终止条件下反演出的速度模型与真实速度模型的拟合程度比较示意图。由图可以看出，两者反演出的速度模型都能很好地拟合真实速度模型。然而，从图 3-5(a)可以看出，SpDE 反演出的速度模型在很多层上都比 MMDE 反演出的更为拟合真实速度模型。而且从图 3-5(b)可以看出，虽然 SpDE 只进化了 50 代，但其反演出的速度模型几乎与 MMDE 进化 500 代后反演出的速度模型一样。也就是说 SpDE 只需要 MMDE 的十分之一的进化代数即可以取到相同的结果，极大地提升了算法的收敛速度以及应用在波形反演问题中的效率。

3.4.2 SpDE 的稳定性

为了验证 SpDE 的稳定性，我们在实验中向观测数据中添加不同信噪比的白高斯噪声，其中的信噪比按照下式进行定义：

$$\text{SNR} = \frac{\max_j |d(t_j)|}{\sigma} \quad (3-9)$$

式中， $d(t_j)$ 代表观测数据，而 σ 则是白高斯噪声的标准差，我们设置 $\sigma = 1$ 。我们分别选取 SNR=10 和 SNR=20 的白高斯噪声去测试 SpDE 的稳定性。图 3-6 是添加 SNR=20 的白高斯噪声的真实观测数据与 SpDE 反演出的速度模型得到的计算数据，而添加 SNR=20 的白高斯噪声后 SpDE 反演出的速度模型与真实速度模型的比较图则在图 3-7。图 3-8 是 SNR=10 时 SpDE 反演出的速度模型得到的计算数据与添加噪声后的观测数据比较图，而相应的速度模型比较图则放在图 3-9。

由图 3-6 到图 3-9 可以看出，在 SNR=10 和 SNR=20 时，SpDE 在添加噪声后的观测数据中依旧能反演出很好拟合真实情况的速度模型，也即是 SpDE 对数据中的噪音是不敏感的，有着很好的稳定性。

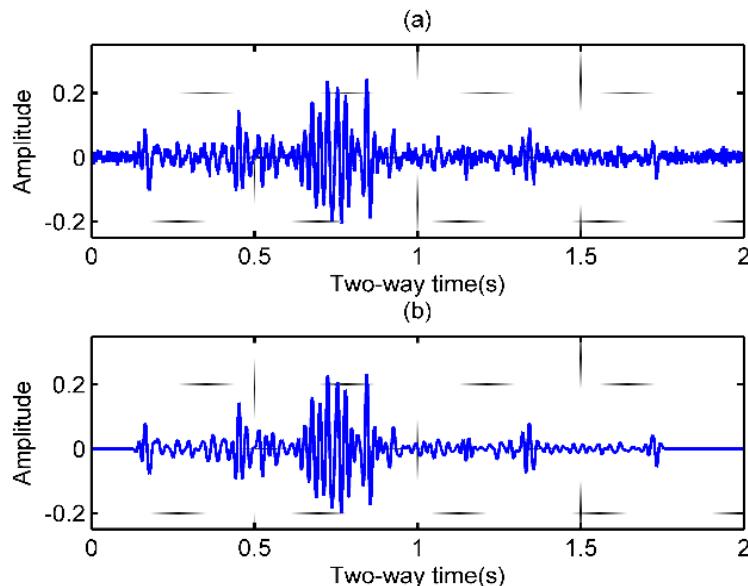


图 3-6 (a)添加 SNR=20 的噪声后的观测数据，(b)SpDE 反演模型得到的计算数据。

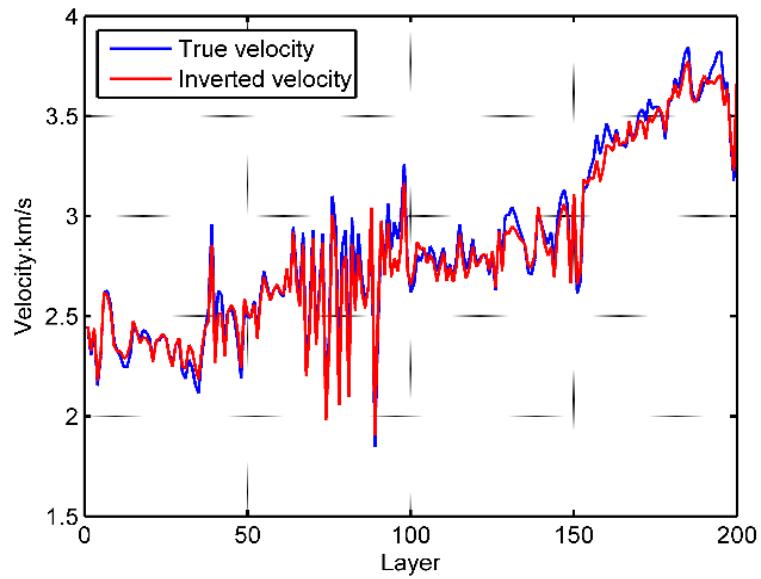


图 3-7 真实速度模型与 SNR=20 时 SpDE 反演出的速度模型比较。

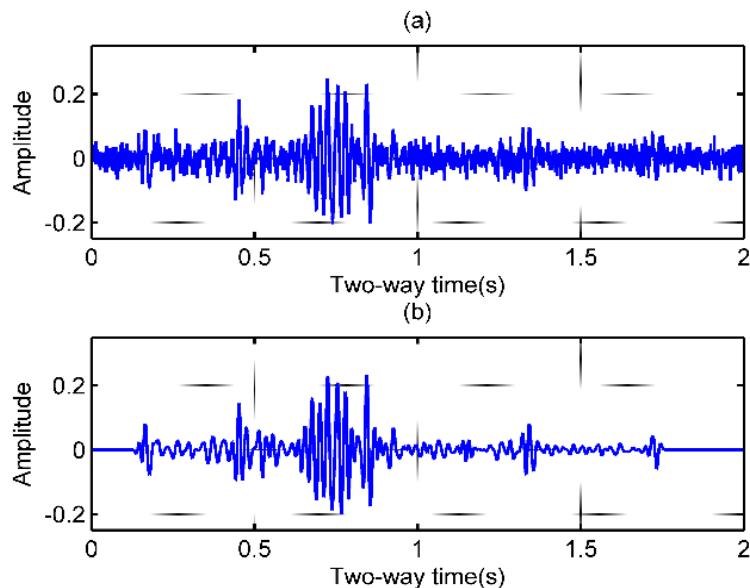


图 3-8 (a)添加 SNR=10 的噪声后的观测数据, (b)SpDE 反演模型得到的计算数据。

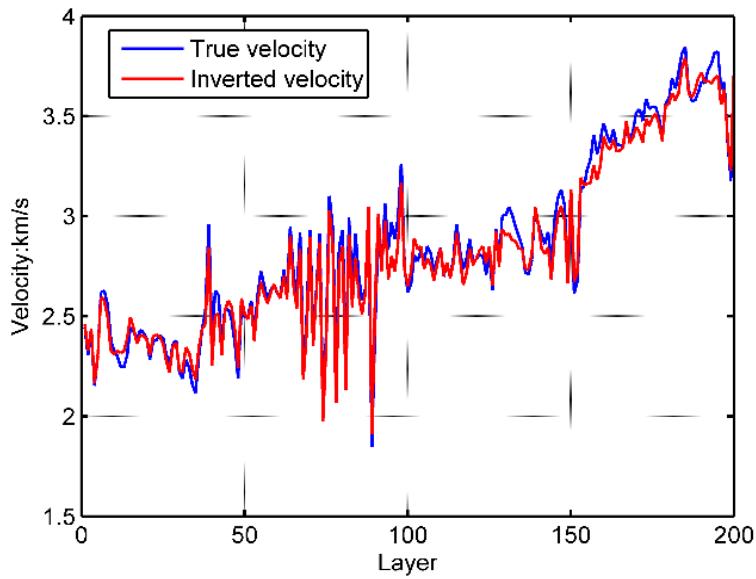
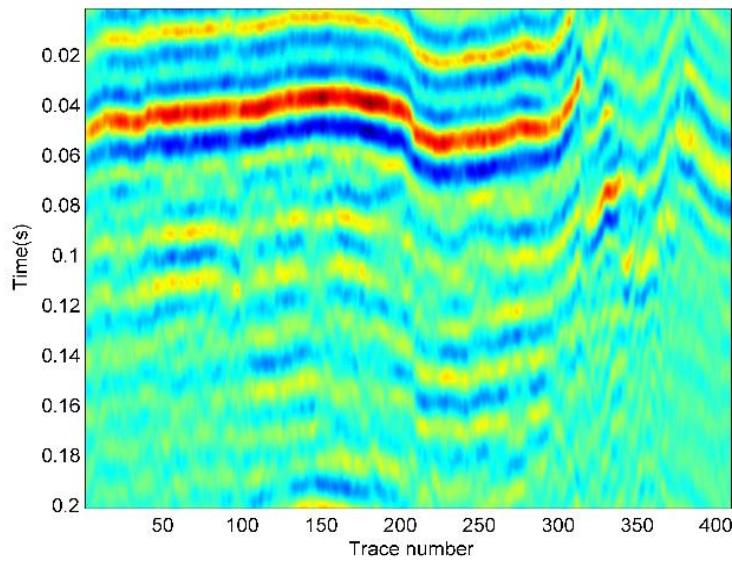


图 3-9 真实速度模型与 SNR=10 时 SpDE 反演出的速度模型比较。

3.4.3 SpDE 应用于实际地震数据

为了进一步验证 SpDE 的性能改进, 我们将 SpDE 应用到实际地震数据中, 并将其与 MMDE 进行比较, 比较结果显示在图 3-10 中。图 3-10(a)显示的是实际叠后地震剖面, 一共有 430 道记录, 每一道记录的时间长度为 201ms。我们将整个地下介质模型分为 67 层, 而每一层的厚度是 3ms。在实验中, 密度被假定为常数, 只对速度参数进行反演。差分进化算法的种群大小设置为 NP=50, 因为 3.4.1 小节中, SpDE 用 50 代反演出的速度模型与 MMDE 用 500 代反演出的速度模型几乎一样, 所以 SpDE 的算法终止条件设置为最多进化 50 代, 而 MMDE 的算法终止条件则设为最多进化 500 代。两者的实验结果分别显示在图 3-10(b)和图 3-10(c)中。



(a)

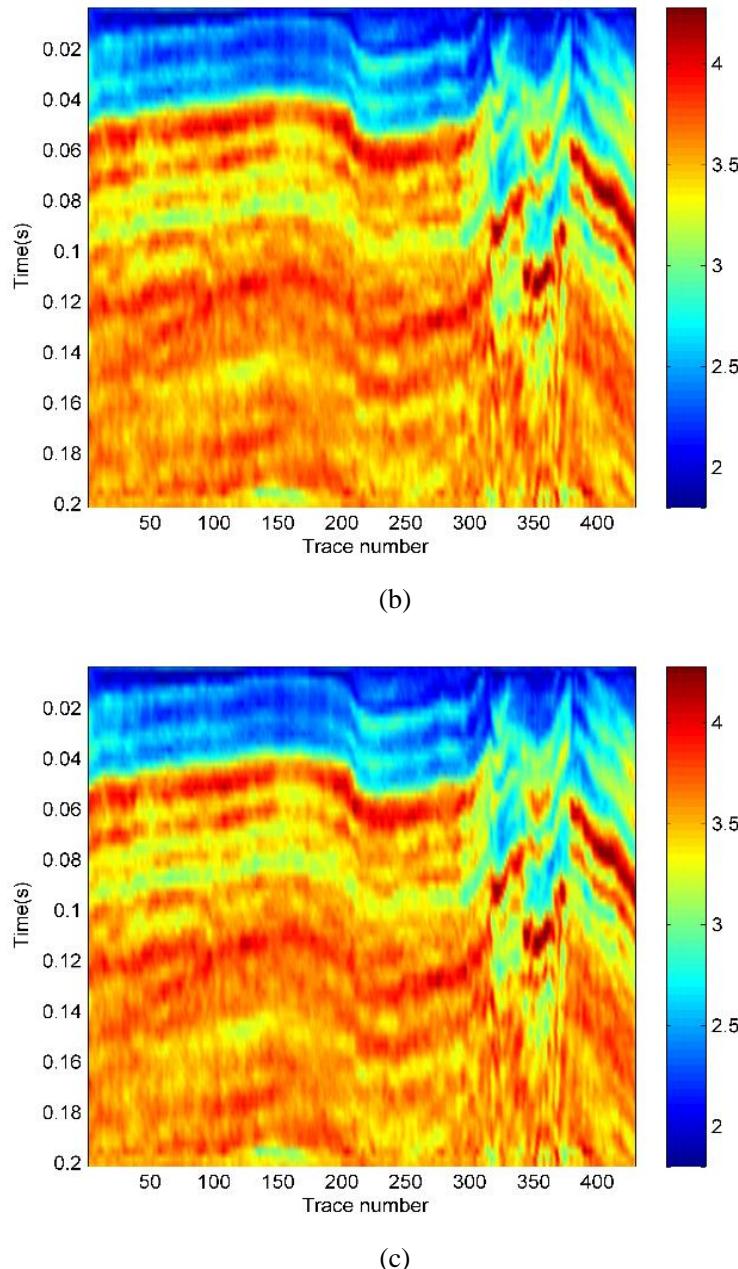


图 3-10 (a) 实际地震数据, (b) SpDE 进化 50 代得到的速度模型, (c) MMDE 进化 500 代得到的速度模型

由图 3-10 可以看出, 两个算法反演出的速度模型都能很好地反映了实际地震数据的特征, 例如速度在横向上的变化。但由于 SpDE 仅用了 MMDE 十分之一的进化代数就能得到相似的结果, 因此 SpDE 更为高效。

为了进一步验证 SpDE 的高效率, 我们统计了 SpDE 和 MMDE 运行时所耗费的 CPU 时间, 具体记录如表 3-2 所示。所有的实验都是在实验室电脑完成 (CPU: Intel Core i5-6500, 3.19GHz)。

表 3-2 MMDE 和 SpDE 运行时间比较表

算法	MMDE	SpDE
运行时间/s	151841	25044

由上表可以看出，SpDE 只用了 MMDE 六分之一的运行时间就可以反演出一个跟 MMDE 相似的速度模型，因此 SpDE 应用到波形反演问题时是非常高效的。

3.5 本章小结

在本章，我们介绍了一种基于多子种群的差分进化算法 SpDE，并将其运用到波形反演问题中。我们依据个体到种群中心的欧氏距离将整个种群划分为三个子种群，为了更好地平衡开发性和探索性，每个子种群肩负起各自的任务。并设计一个两阶段的变异策略来联结各个子种群，第一阶段各子种群独立变异，然后将各个子种群的临时变异个体加权相加得到最终的变异个体，当最内层的子种群中的个体变得足够好时，变异转换到第二阶段，最内层的个体在其附近搜索以提升现有解的质量，另外两层的个体则按照 CCDE 的变异策略进行变异。最后将当前种群中最好的子成分组成一个新的个体，并将其加入交叉操作，以期能加快优秀子成分在种群内的扩散速度，加快算法的收敛速度。

在应用于合成地震数据的波形反演的实验中，SpDE 不仅收敛速度比 MMDE 和 CCDE 等算法快，而且收敛精度也比这些比较算法的高。SpDE 仅用 50 代反演出的速度模型与 MMDE 进化 500 代反演出的速度模型极为相似，这说明 SpDE 应用到波形反演问题中的效率很高。在随后的稳定性实验中，我们分别向观测数据中加入 SNR=20 和 SNR=10 的白高斯噪声，但 SpDE 依旧能反演出拟合真实速度模型的模型，说明 SpDE 对噪音不敏感。在最后的实际地震数据实验中，我们比较了 SpDE 和 MMDE 的实际运行时间，在反演出相似的速度模型的前提下，SpDE 仅用了 MMDE 六分之一的时间。合成地震数据和实际地震数据实验都证明了 SpDE 不仅提升了收敛速度，还提高了收敛精度。

4 变异策略的个体选择

差分进化算法在种群初始化后，经过变异操作、交叉操作和选择操作的不断迭代，在整个搜索空间中搜寻最优解。传统变异操作随机选择三个个体，一个作为基个体，另外两个作差作为扰动加在基个体上，最终产生新的变异个体。交叉操作则在变异因子 CR 的控制下，将目标个体和变异个体的子成分按照一定的概率组合起来，形成试验个体。而选择操作则在比较目标个体和试验个体的适应度函数值后，将较为优秀的个体保留到下一代的进化中。由此可以看出，交叉操作和选择操作并没有对种群产生新的信息，它们只是在变异个体的基础上进行深加工。而变异操作则通过对基个体进行差分向量的扰动，产生一个原种群没有的个体，为种群引入新的信息，增加种群的多样性。因此，变异操作是差分进化算法中最重要最基本的一个操作，一个好的变异策略是差分进化算法性能改善的必要条件，而这也是众多研究者深入研究变异操作的原因之一。

变异操作主要是在一个基个体上加上一个或多个差分向量作为扰动，以产生新的个体。因此，如何选择个体参加变异策略以及如何安排这些个体的位置则是改进变异操作的重点。本章将提出一种基于新的变异策略个体选择方法的差分进化算法，称为 FDDE。在 FDDE 算法中，我们先提出一种方法来衡量个体对种群多样性的贡献，接着我们会综合考虑个体适应度排序以及个体对种群多样性贡献两方面来决定每个个体在变异策略中的位置。

4.1 引言

变异操作是差分进化算法中最为重要的一个操作，为整个种群产生新的信息。变异操作通过选择个体，在基个体上加上差分向量作为方向扰动，使基个体往扰动方向进行搜索，以期能找到更好的个体。因此，如何选择个体参与变异策略则极为重要，关系着开发性与探索性的平衡。如基本的变异策略“DE/rand/1”，它在整个种群中随机选择三个个体，一个作为基个体，另外两个作差加到基个体上。正如它的名字“rand”一样，它是完全随机的，个体选择是随机的，哪个个体作为基个体也是随机的。该策略放弃了开发性，而探索性极强，因此导致该策略的收敛速度极慢。而另一个极端例子，则是基本变异策略“DE/best/1”，该策略直接选取当前种群中最优秀的个体（best 个体）作为基个体，在 best 个体附近进行搜索，以期能进一步提升 best 个体的质量。该策略完全放弃了探索性，而开发性极强，因为变异策略的基个体只能是 best 个体，因此种群很容易陷入局部极值点而跳不出来，容易导致算法早熟。

平衡差分进化算法的开发性与探索性就是在基本变异策略“DE/rand/1”和“DE/best/1”之间进行折中以取得一个较好的平衡点。而这些年来，许多关于差分进化算法的改进也都是基于这点出发的，在接下来的小节将会简要介绍。本章会从适应度

的角度来衡量个体对种群多样性的贡献，并联合个体适应度排名而提出一个新的变异策略个体选择方法。

4.2 变异策略的改进方法

变异操作作为差分进化算法中最重要的一个操作，对算法的性能有着重要影响。因此如何改进变异操作一直是研究者的关注重点，本小节主要介绍相关的变异策略改进方法。在我看来，这些改进方法主要可以粗略地分为两个方向：1) 多种变异策略的集成学习，2) 对参与变异策略的个体进行选择。

不同的变异策略的特性各有不同，对开发性和探索性的偏向不同，如上小节介绍的“DE/rand/1”和“DE/best/1”，因此不同的问题需要的变异策略也就不同，而且即使在同一个问题的不同变异阶段需要的变异策略也可能不同。然而为了一个特定的问题去选择一个最优的变异策略并不是一件容易的事。因此将多种变异策略集成起来，让算法根据特定问题特定进化阶段去自适应地选择适合的变异策略，则是一个很自然且合理的想法。Qin 等人在 2009 年提出的 SaDE^[8]在候选变异策略池中加入几种基本变异策略，在经过一段学习期（learning period）后，每种变异策略的被选择概率会根据其之前的进化成功与否进行自适应调整，以此让算法选择当前问题当前阶段最适合的变异策略。Mallipeddi 等人于 2011 年提出的 EPSDE^[28]同样选择一个由几种特性不同的变异策略组成变异策略候选池，初始化阶段每个个体都会绑定一种变异策略，若是该个体产生的新个体在随后的选择操作中优胜过目标个体，则该个体继续在下一代沿用该变异策略，否则在候选池中为该个体重新随机选择一种变异策略。Wang 等人在 2011 年提出的 CODE^[14]选择“DE/rand/1”，“DE/rand/2”和“DE/current-to-rand/1”这三种基本变异策略组成候选池，每个个体在变异阶段，都会用这三种变异策略各产生一个变异个体，再通过交叉操作产生三个试验个体，分别计算这三个试验个体的适应度函数值，选择最好的试验个体参与最终的选择操作。CODE 因为要在每个个体进化时需要计算三次适应度函数，所以耗费的计算量是传统差分进化算法的三倍。类似的改进算法还有 SaJADE^[30], DE-SG^[31]。

除了多种变异策略集成自适应学习外，也有许多研究者^{[32][33]}在深入研究如何在单一的已有变异策略中选择参与变异策略的个体，因为变异操作的本质就是在一个基个体的基础上加入差分向量来引导搜索方向。如“DE/rand/1”和“DE/best/1”的本质差异就在于如何选择基个体，前者在整个种群随机选择一个个体，因此其探索性极强而开发性极弱；而后者则直接选取当前种群的 best 个体作为基个体，所有的变异操作都是在 best 个体附近进行搜索以期能提高 best 个体的质量，因此其开发性极强而探索性很弱。同理，传统变异策略“DE/current-to-best/1”因其中直接运用 best 个体，因此开发性极强。为了降低传统变异策略“DE/current-to-best/1”的贪婪性，Zhang 等人在 2009 年提出的 JADE^[34]中将该策略改进为“DE/current-to-pbest/1”，该改进策略不再如传统策略那样只是选择 best 个体，而是新增了一个新参数 p, $p \in (0,1]$ ，从最好的 $100*p\%$ 个

体中随机选择一个个体替代原策略中的 best 个体参与变异策略。“DE/current-to-pbest/1”变异策略不是单单只选择最好的个体，而是从最好的小部分中随机选择一个参与变异，选择范围扩大了，增加了多样性，也相应地增强了探索性。但是该算法新增加了一个参数 p ，如何选择 p 的值也得慎重考虑。Gong 等人在 2013 年提出的 rank-DE^[15]利用个体适应度信息来为变异策略选择个体。rank-DE 首先对个体从差到好排序，每个个体得到一个排名 R_i ，个体越好，其 R_i 值越大。随后决定每个个体参与变异策略的被选概率 $p_i = R_i/NP$, NP 为种群个体数目，也即个体越优秀其被选入参与变异策略的概率也就越大。即使如此，较差的个体也有机会被选入参与变异策略，虽然概率较小。这也是在传统变异策略“DE/rand/1”和“DE/best/1”之间进行折中，平衡算法的开发性与探索性。rank-DE 虽然在“DE/best/1”策略的基础上引入一定的随机性来增强探索性，但究其本质，该算法依旧是哪个个体的个体适应度高，该个体就会大概率地被选入变异策略，而这也还是有可能导致多样性的损失。还有研究者直接在最优秀的一部分中选择个体参与变异策略^[35]，其余的更多有关变异策略个体选择的算法可参考文献^{[36][37][38][39][40]}。

本章的工作主要集中于在单一的已有变异策略基础上选择参与变异策略的个体，为了增强种群的多样性，本章接下来将介绍的 FDDE 会综合考虑个体适应度和个体对种群多样性贡献来确定每个个体在变异策略的位置。

4.3 基于适应度和多样性的变异策略

针对差分进化算法变异策略个体选择的问题，从平衡开发性与探索性的目的出发，本小节主要提出一种基于适应度和多样性的变异策略，从适应度出发是为保证开发性，而从多样性出发则是为了保证探索性。两者综合考虑，则是在开发性与探索性之间取得平衡。

4.3.1 算法动机

如 4.2 小节所说，变异操作是差分进化算法中最重要的操作。因为变异策略主要通过在一个基个体上加扰动来产生新个体，因此如何选择个体参与变异策略以及决定它们的位置则是改进差分进化算法的关键。如在基本变异策略“DE/rand/1”中：

$$V_i^G = X_{r1}^G + F \times (X_{r2}^G - X_{r3}^G) \quad (4-1)$$

图 4-1 是变异策略“DE/rand/1”在二维参数空间的变异示意图：

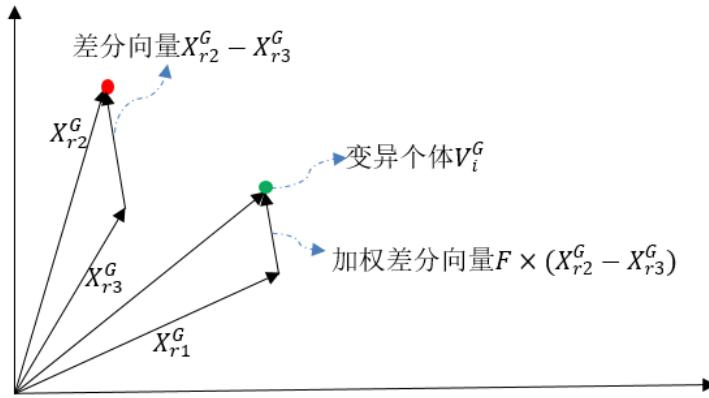


图 4-1 “DE/rand/1”在二维搜索空间的示意图

由图 4-1 可以明显看出变异策略在基个体的附近按照差分向量的引导进行搜索，最终产生变异个体。因此基个体极为重要，它决定了将来的变异个体将会落在哪块区域。而差分向量将决定往基个体附近哪个方向搜索，因此差分向量的首与尾也对算法性能有重大影响。因此对于被选上参与变异策略的个体，如何安排它们在变异策略中的位置对算法性能有着重要影响。

在自然界中，优秀的生物个体往往有着良好的遗传信息，能更好地在自然界存活下去并有更大的机会将自身优秀的遗传信息一代代地往后代遗传。相应地，在差分进化算法中，优秀的个体也往往包含着较多优秀的子成分，用其作基个体产生的变异个体也有更大的机会能进入下一代的进化。但同时，作为一种群体智能的优化算法，差分进化算法如果在选择基个体时过于依赖优秀的个体，有可能导致在多代进化后，整个种群的个体都集中在同一个狭小区域内，丧失了多样性，进而导致差分向量中两个个体差异不大，差分向量极小，最后整个种群没法产生新的个体，算法停滞，陷入早熟。因此在选择参与变异策略的个体时，不仅要考虑个体适应度（开发性），还得考虑该个体对种群多样性的贡献（探索性）。因此，通过对这两方面的综合考虑，本章提出一种新的差分进化算法变体 FDDE，具体算法流程将在以下小节进行介绍。

4.3.2 算法流程

开发性是在已有的较好解附近搜索，以期能提升解的质量，而这是可以通过个体的适应度来体现的。相反，探索性是算法搜索未知区域以期找到更好解的能力，而这正好可以由多样性来评估，因为只有足够的多样性，种群才能尽可能地在整个搜索空间进行搜索。因为开发性和探索性可以间接地由适应度和多样性来体现，因此平衡开发性与探索性也可以由适应度和多样性来引导，因此 FDDE 主要通过综合考虑这两方面来达到开发性与探索性的平衡。

(1) 适应度排序

如上一小节所述，一个好的个体可能包含较好的子成分，用其作为基个体所产生的变异个体有更大的机会进入下一代的进化，而这正保证了算法的开发性。因此，为了更好地利用种群中较好个体的信息，我们先对整个种群中的个体按照适应度从好到差

进行排序，每个个体均有自己的适应度排名 $fitrank_i$, $fitrank_i \in [1, NP]$ 。个体越优秀，其 $fitrank_i$ 则越小。

(2) 个体对种群多样性贡献排序

在我看来，多样性并不仅仅是种群在整个搜索空间广泛分布，我们关注多样性也并不是简单地让种群个体均匀地分散在整个搜索空间，否则我们只是简单地保留初始化后的种群就好了，因为它是多样性最强的，我们更应关注那些能帮我们找到更优个体的有用的多样性。在我看来，差分进化算法的进化过程可以看作由两方面组成：一方面，种群在不断收缩；另一方面，种群中心不断向最优点移动。因此，在算法进化的过程中，以种群在搜索空间的分布情况来判断多样性肯定有点片面。因此 FDDE 主要通过计算个体适应度对中间个体适应度的偏离来评估该个体对种群多样性的贡献，主要方法如下：

在前面计算每个个体适应度排名时已经将整个种群按适应度从优到差进行了排序，在此我们选择在适应度排名第 $NP/2$ 的个体，也即适应度排序居中的个体，取其适应度 f_{mid} 作为基准，按下式计算每个个体 X_i^G 对基准个体的偏离 $f_{de,i}$:

$$f_{de,i} = (f_i - f_{mid})^2 \quad (4-2)$$

式中， f_i 是种群中第 i 个个体 X_i^G 的适应度函数值。

将 $f_{de,i}$ 从大到小进行排序，每个个体 X_i^G 得到自己对种群多样性贡献的排名 $divrank_i$ ，排名越高说明对种群贡献越大。

为了综合考虑以上两方面的信息，我们设置一个参数 α 来得到每个个体的最终排名 $rank_i$ ，可以表示如下式：

$$rank_i = \alpha \times divrank_i + (1 - \alpha) \times fitrank_i \quad (4-3)$$

$$\alpha = gen / Maxgen \quad (4-4)$$

$$\alpha = \begin{cases} 0.2, & \alpha < 0.2 \\ 0.8, & \alpha \geq 0.8 \end{cases} \quad (4-5)$$

式中： gen 为当前进化的代数，而 $Maxgen$ 则为算法预先设置的最大进化代数。由上式可以看出，在前期主要偏向开发性，因为前期多样性足够，偏向开发性可以加快算法收敛速度。随着进化进程的推进，算法渐渐偏向保持多样性，免得算法因多样性不足而陷入局部极值点导致早熟。之所以将 α 限制在[0.2,0.8]之间，主要是考虑到在算法不能过于偏向开发性和探索性。

当所有个体的最终排名 $rank_i$ 确定后，算法进入变异操作阶段，首先如传统变异策略所做的，在整个种群中随机选择三个个体 X_{r1}^G , X_{r2}^G 和 X_{r3}^G 。由前面的 4.3.1 小节可知，在变异策略中，基个体是最重要的，差分向量是引导算法在基个体附近搜索的搜索方向，因此差分向量的终点比起点重要。因此可以对 X_{r1}^G , X_{r2}^G 和 X_{r3}^G 这三个个体按照他们的最终排名 $rank_i$ 从小到大进行排序，得到排序序列 $[X_1^G, X_2^G, X_3^G]$ 。最终，对变异策略“DE/rand/1”的改进如下：

$$V_i^G = X_1^G + F \times (X_2^G - X_3^G) \quad (4-6)$$

综合以上内容，FDDE 的算法流程如下所示：

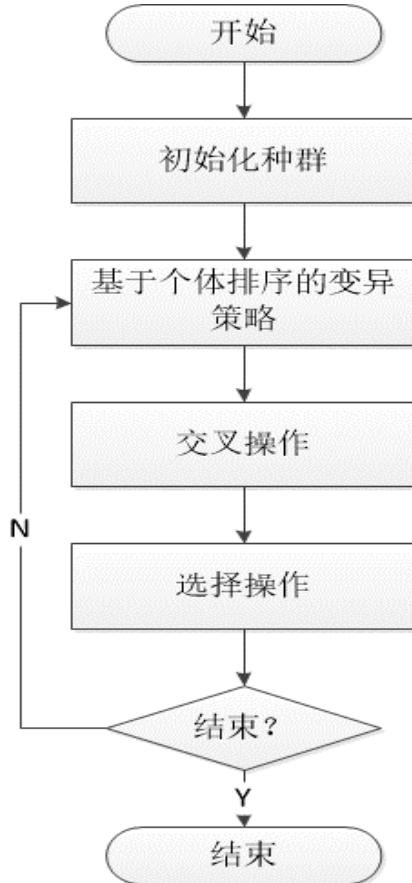


图 4-2 FDDE 的算法流程图

由上图可以看出，FDDE 的算法流程图基本与经典差分进化算法的一致，只在变异操作中做了改进。

与 rank-DE 不同，我们提出的 FDDE 主要在以下两个方面比 rank-DE 有所改进：
 1) rank-DE 仅仅采用了适应度对个体进行排序，并根据每个个体的排名给它们相应的被选概率。rank-DE 完全忽略了多样性，而多样性对增强探索性有着重要影响。而 FDDE 采用了一个新的方法衡量个体对种群多样性的贡献，并综合考虑该贡献的排名与适应度的排名，通过在进化的不同阶段给两者不同的权重以实现两者的加权平均以得到个体的最终排名；2) rank-DE 分配给每个个体的被选概率在变异操作选择个体时发挥作用，适应度越好的个体其被选入的概率越大，这是通过牺牲探索性来增强算法的开发性，因为较差的个体有可能多次进化都不一定能被选入变异操作。而 FDDE 的选择个体阶段跟传统的变异策略一样，在整个种群中随机选择三个个体。然后利用之前计算得到的最终排名，将这三个个体进行排序，并将它们分别分到基个体和差分向量中，也即是在 FDDE 中，每个个体被选入变异策略的概率都是一样的，只是根据个体的排名而将其分在重要性不同的基个体和差分向量中，这无疑比 rank-DE 多了更多的随机性，

增强了探索性，但同时它是通过个体排名来决定该个体是在基个体还是在差分向量中，因此它的开发性也很足，做到了两者的平衡。

4.4 实验结果与分析

在本节的实验中，我们将通过一系列的实验来证明 FDDE 的有效性，我们将选取 CEC2005 测试函数集作为测试集。

在第一个实验中，我们将 FDDE 与 jDE^[7]和 rank-jDE^[15]相比较。其中 jDE 是一个参数自适应的差分进化算法，它将变异系数 F 和交叉率 CR 绑定到个体上，并通过两个参数 τ_1 和 τ_2 进行自适应控制，在很多变异策略上都取得很好的结果。而 rank-jDE 是在 rank-DE 的基础上借用 jDE 的参数自适应方法的差分进化算法变体。三种比较函数都在基础变异策略“DE/rand/1”上进行比较。所有比较算法的种群大小都设置为 NP=100，最大的适应度函数计算次数（Max_NFFEs）设置为 D × 10000，其中 D 为待求解问题的维数，所有实验结果均为独立运行 25 次后得到的平均结果。另外，边界处理问题对差分进化算法的性能有着重要影响，因此对本论文接下来在 CEC2005 测试函数集上的所有实验，都采取如下边界处理原则：若个体的某一维超出搜索空间的上下限时，我们将对这一维进行重新初始化。下表为 D=30 时三种比较算法在所有测试函数的比较结果。

表 4-1 D=30 时，jDE, rank-jDE 和 FDDE 在所有测试函数的比较结果

	jDE	rank-jDE	FDDE
F1*	7.37e+00	8.93e-02	3.40e-03
F2	1.08e-05	1.44e-11	2.88e-13
F3	1.89e+05	8.12e+04	7.14e+04
F4	2.98e-01	7.98e-04	5.06e-04
F5	1.10e+03	1.11e+03	2.56e+02
F6	2.46e+01	5.74e-01	4.87e-01
F7	1.31e-02	9.75e-03	9.20e-03
F8	2.09e+01	2.09e+01	2.09e+01
F9*	7.64 e+01	6.42 e+01	6.29e+01
F10	5.86e+01	4.71e+01	4.65e+01
F11	2.80e+01	2.79e+01	2.71e+01
F12	1.16e+04	1.65e+03	2.43e+03
F13	1.70e+00	1.60e+00	1.60e+00
F14	1.30e+01	1.30e+01	1.27e+01
F15	3.40e+02	3.66e+02	3.00e+02
F16	7.56e+01	6.12 e+01	7.30e+01
F17	1.33e+02	1.06 e+02	1.23 e+02
F18	9.07e+02	9.08 e+02	9.03 e+02

表 4-1 (续) D=30 时, jDE, rank-jDE 和 FDDE 在所有测试函数的比较结果

	jDE	rank-jDE	FDDE
F19	9.06e+02	9.08 e+02	9.03 e+02
F20	9.06e+02	9.08 e+02	5.54 e+02
F21	5.00e+02	5.00e+02	5.00e+02
F22	9.04e+02	8.97 e+02	8.82 e+02
F23	5.34e+02	5.34e+02	5.34e+02
F24	2.00e+02	2.00e+02	2.00e+02
F25	2.10e+02	2.09 e+02	2.09 e+02

上表显示的是 D=30 时, jDE, rank-jDE 和 FDDE 在 CEC2005 测试函数集上所有函数的比较结果, 其中 F1 和 F9 由于三种比较算法都能在最大适应度函数计算次数内找到全局最优, 因此表中的数据为三种算法在最大适应度函数计算次数 Max_NFFEs=20000 时的结果。三者中的最好结果由**粗体字**显示, 由表中数据可以看出 FDDE 在绝大多数的函数中都能取得最好的结果, 仅在 F12, F16 和 F17 这三个函数上有所不如 rank-jDE。接下来我们将比较 D=50 时, jDE, rank-jDE 和 FDDE 在测试函数集上的表现, 具体比较结果显示在表 4-2。

表 4-2 D=50 时, jDE, rank-jDE 和 FDDE 在所有测试函数的比较结果

	jDE	rank-jDE	FDDE
F1*	4.88e-03	2.29e-06	1.93e-08
F2	8.99e-02	3.46e-05	3.27e-06
F3	5.30e+05	3.25e+05	3.20e+05
F4	8.31e+02	2.87e+02	9.67e+01
F5	3.39e+03	3.63e+03	3.02e+03
F6	3.98e+01	7.65e+00	1.65e+00
F7	4.13e-03	4.82e-03	7.20e-03
F8	2.11e+01	2.11e+01	2.11e+01
F9*	7.69e+01	5.66e+01	5.55e+01
F10	5.86e+01	7.66e+01	8.60e+01
F11	1.00e+02	5.31e+01	5.41e+01
F12	3.71e+04	6.20e+03	1.60e+04
F13	2.90e+00	2.81e+00	2.75e+00
F14	2.26e+01	2.26e+01	2.21e+01
F15	3.32e+02	3.16e+02	2.50e+02
F16	8.54e+01	6.93e+02	6.42e+01
F17	1.75e+02	1.32e+02	1.30e+02
F18	9.25e+02	9.31e+02	9.22e+02

表 4-2(续) D=50 时, jDE, rank-jDE 和 FDDE 在所有测试函数的比较结果

	jDE	rank-jDE	FDDE
F19	9.25e+02	9.30e+02	9.23e+02
F20	9.25e+02	9.30e+02	5.74e+02
F21	5.00e+02	5.00e+02	5.00e+02
F22	9.43e+02	9.40e+02	9.13e+02
F23	5.39e+02	5.39e+02	5.39e+02
F24	2.00e+02	2.00e+02	2.00e+02
F25	2.14e+02	2.14e+02	2.17e+02

上表显示的是 D=50 时, jDE, rank-jDE 和 FDDE 在 CEC2005 测试函数集上所有函数的比较结果, 其中表中 F1 和 F9 的数据为三种算法在最大适应度函数计算次数 Max_NFFEs=50000 时的结果。三者中的最好结果由粗体字显示, 由表中数据可以看出 FDDE 在绝大多数的函数中都能取得最好的结果, 仅在 F7, F10 这两个函数不如 jDE, 在 F11, F12, 和 F25 这三个函数上有所不如 rank-jDE。

由 4.3.2 节可知, FDDE 综合考虑个体适应度和个体对种群多样性贡献两方面来对个体进行排序, 并最终决定个体在变异策略中的位置。为了验证适应度和个体对种群多样性贡献这两方面各自对算法性能的影响, 我们将比较 FDDE_fit, FDDE_div, rank-jDE 和 FDDE 在 D=30 时在测试函数集上的表现, 其中, FDDE_fit 为仅用个体适应度对个体进行排序, 而 FDDE_div 指仅用个体对种群多样性的贡献对个体进行排序。四者的比较结果如下表所示:

表 4-3 D=30 时 FDDE_fit, FDDE_div, rank-jDE 和 FDDE 在所有测试函数的比较结果

	FDDE_fit	FDDE_div	rank-jDE	FDDE
F1*	1.40e-03	1.84e-01	8.93e-02	3.40e-03
F2	5.20e-13	1.30e-11	1.44e-11	2.88e-13
F3	1.01e+05	1.05e+05	8.12e+04	7.14e+04
F4	2.00e-03	1.23e-02	7.98e-04	5.06e-04
F5	1.80e+02	4.57e+02	1.11e+03	2.56e+02
F6	6.22e-01	9.39e-01	5.74e-01	4.87e-01
F7	7.20e-02	1.79e-02	9.75e-03	9.20e-03
F8	2.09e+01	2.09e+01	2.09e+01	2.09e+01
F9*	5.45 e+01	6.80e+01	6.42 e+01	6.29e+01
F10	4.66e+01	5.39e+01	4.71e+01	4.65e+01
F11	2.79e+01	2.74e+01	2.79e+01	2.71e+01
F12	3.74e+03	2.61e+03	1.65e+03	2.43e+03
F13	1.67e+00	1.70e+00	1.60e+00	1.60e+00
F14	1.29e+01	1.29e+01	1.30e+01	1.27e+01

表 4-3(续) D=30 时 FDDE_fit, FDDE_div, rank-jDE 和 FDDE 在所有测试函数的比较结果

	FDDE_fit	FDDE_div	rank-jDE	FDDE
F15	4.00e+02	3.00e+02	3.66 e+02	3.00 e+02
F16	1.13e+02	5.90e+01	6.12 e+01	7.30e+01
F17	2.64e+02	1.47e+02	1.06 e+02	1.23 e+02
F18	9.04e+02	9.03e+02	9.08 e+02	9.03 e+02
F19	9.05e+02	9.05e+02	9.08 e+02	9.03 e+02
F20	5.54e+02	5.58e+02	9.08 e+02	5.54 e+02
F21	5.00e+02	5.00e+02	5.00 e+02	5.00 e+02
F22	8.72e+02	8.84e+02	8.97 e+02	8.82 e+02
F23	5.34e+02	5.34e+02	5.34 e+02	5.34 e+02
F24	2.00e+02	2.00e+02	2.00 e+02	2.00 e+02
F25	2.10e+02	2.10e+02	2.09 e+02	2.09 e+02

由上表可以看出, FDDE 仅在 F1, F5, F9 和 F22 这四个函数不如 FDDE_fit, 其余 21 个函数都胜于或者至少不差于 FDDE_fit; 而与 FDDE_div 相比, FDDE 仅在 F16 不如 FDDE_div, 在其余 24 个测试函数都优于或者不差于 FDDE_div。而如果将 FDDE_fit 和 FDDE_div 分别与 rank-jDE 相比, 则 rank-jDE 在 8 个函数上优胜于 FDDE_fit, 在 11 个函数上优胜于 FDDE_div, 而 rank-jDE 仅只能在 3 个函数上优胜于 FDDE。因此个体按个体适应度排序和按个体对种群多样性贡献排序两者是相辅相成的, 只有综合考虑两方面才能取得开发性和探索性之间更好的平衡。

4.5 本章小结

在本章, 我们提出了一种基于变异策略个体选择的 DE 算法变体——FDDE 算法。在传统变异策略中, 要么从种群中随机选择个体, 要么直接选择当前种群最优个体作为基个体, 这两种方法只能在探索性和开发性中选择一方, 并不能做到两者平衡。在自然界中, 优秀的生物个体往往包含较好的遗传信息, 并有较大的概率能将自身较好的遗传信息传给后代, 受其启发, 我们在 FDDE 中引入适应度值排名来控制变异策略的个体位置。但仅利用适应度只能保证算法的开发性, 为此, FDDE 还重新定义了个体对种群多样性的贡献, 种群有了足够的多样性, 才能保证较强的探索性。综合考虑适应度和多样性, FDDE 给每个个体分配了一个最终排名, 因为变异策略中基个体和差分向量的重要性不同, 因此 FDDE 根据该最终排名将每个个体分派到变异策略的不同位置。

为了测试 FDDE 的性能, 我们在 CEC2005 测试函数集上进行了测试, 并与 jDE, rank-DE 等近年来竞争力很强的差分进化算法变体进行比较。我们分别在 D=30 和 D=50 时对这三种算法在测试函数集上的表现进行了测试, 由实验结果可以看出, FDDE 在不同的维数下, 能在绝大多数的测试函数中取得最好的结果。在接下来的实验中, 我们将按个体适应度排序和按个体对种群多样性贡献排序分别独立进行实验, 实验结果表明两个独立实验都不如两者综合考虑的结果好, 因为这两者分别代表了开发性和探索性, 只有综合考虑这两方面才能得到开发性与探索性更好的平衡, 取得更好的实验结果。

5 基于邻域的差分进化算法

差分进化算法是一种基于群体智能的全局优化算法，以一个初始种群为起点，通过变异，交叉和选择不断迭代，在整个搜索空间中寻优。变异操作为整个种群引入新的信息，是差分进化算法中最重要的一一个操作。然而，在绝大多数的差分进化算法中，变异操作并没有高效地利用邻域关系，而邻域关系往往包含着引导差分进化算法搜索进程的有用信息。

在本章，我们将提出一种新的基于个体适应度来划分邻域的差分进化算法 FNDE (Differential Evolution with Fitness-based neighborhood)，该算法将关注点投向大多差分进化算法常常会忽视的较差个体和一般个体，按照个体的适应度函数值，我们将较好个体的邻域划分得足够大，而较差个体的邻域则相对的很小，而且较差个体的邻域几乎只由较优秀的那一小部分个体组成，以期能提高整个种群的进化成功率，提高算法的收敛精度。

5.1 引言

差分进化算法是一种简单高效的全局优化算法，通过变异，交叉和选择不断迭代，在整个搜索空间中寻优。变异操作是差分进化算法中一个最为关键的操作，它通过在一个基个体上加上差分向量作为扰动以产生新的个体，为种群引入新的信息。然而在大多数的差分进化算法中，变异策略中的个体要么都从种群中随机选取，而这必然导致算法因该选法过于随机而收敛过慢；要么直接选取当前种群中的 best 个体作为基个体，而这又可能因其过强的导向性而使算法陷入局部极值点而导致算法早熟。另一方面，邻域关系包含着个体之间的联系关系和整个种群的结构信息，对种群进化有着引导作用，而大多数的差分进化算法的变异策略都没用上个体的邻域关系，这无疑会妨碍算法性能的进一步提升。

再者，由第 4 章知道，变异操作是差分进化算法最重要的一个操作，而如何选择参与变异策略的个体则又是其中重中之重。因此本章提出一种按个体适应度进行邻域划分的方法，并改进相应的变异策略，变异策略中的个体从目标个体的邻域内选择，以期能提高种群的进化成功率，进而提高算法性能。

5.2 经典的基于邻域的差分进化算法

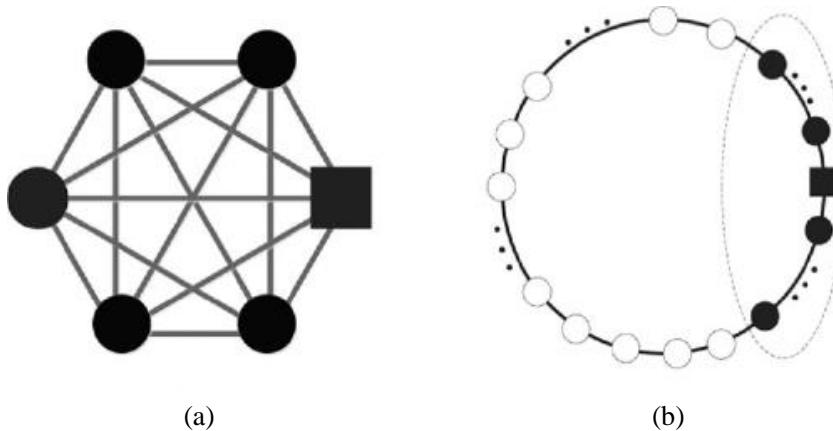
邻域关系包含着个体之间的联系信息，反映了种群的结构信息，能很好地指导种群的进化方向^[41]。现今，越来越多的差分进化算法变体采用了邻域关系的信息去提升算法性能，进而衍生出很多的定义邻域关系的方法^{[42][43][44][45]}。在我看来，这些方法主要可以粗略地分为两类：1) 基于距离来划分邻域；2) 基于个体下标来划分邻域。

因为个体与个体之间的距离可以明确地刻画两个个体在整个搜索空间中的真实远

近，因此基于距离来进行邻域划分能真实反映种群在搜索空间的分布情况，给种群进化提供引导作用。Epitropakis 等人在 2011 年提出的 ProDE(a proximity-based Differential Evolution) [46] 根据到目标个体的距离为目标个体选择参与变异策略的个体。Cai 等人于 2012 年提出的 LeDE^[47] (Learning-enhanced Differential Evolution) 基于划分好的簇 (cluster) 来划分个体的邻域。Sun 等人于 2012 年提出的 APTDE (Adaptive Population Topology Differential Evolution) [48] 从几种基本的邻域关系中自适应地选择一种给当前种群。更多的基于距离划分邻域的改进算法可以参考文献^{[49][50][51][52]}。

虽然基于距离来划分个体邻域能间接地反映种群在搜索空间的分布情况，然而计算种群中个体两两之间的距离的计算量是极大的，这与提升差分进化算法的收敛速度的目的相悖。为了避免计算距离所需的巨大计算量，基于个体下标进行个体邻域划分的方法被提出了。主要有以下四种高效的邻域关系拓扑结构：

- 1) Panmictic 型^{[53][54]}: 这是最基本的也是最广泛应用的，在该类型邻域关系中，每个个体的邻域由整个种群组成。在整个算法进化过程，每个个体可以自由地与种群其它个体进行联系，其结构示意图如图 5-1(a)所示。
- 2) Ring 型^{[42][52]}: 种群中所有的个体按照下标关系被排成一个环形，每个个体以一定的半径长度把位于该个体下标前后该半径长度内的所有个体划分进该个体的邻域。如某个个体的下标为 10，以 5 为半径长度，则下标为 5 到 15 的个体均为这个下标为 10 的个体的邻域。其结构示意图如图 5-1(b)所示。
- 3) Cellular 型^{[55][56]}: 种群中的个体被重排为一个二维结构图，每个个体只与二维结构图中相邻的个体产生邻域关系。其结构示意图如图 5-1(c)所示。
- 4) Distributed 型^{[57][58]}: 整个种群被划分为多个子种群，所有的子种群被重排为一个环形。同个子种群的个体互为邻域关系，每个子种群独立进化，在一定条件下，子种群之间可以进行信息交换。其结构示意图如图 5-1(d)所示。



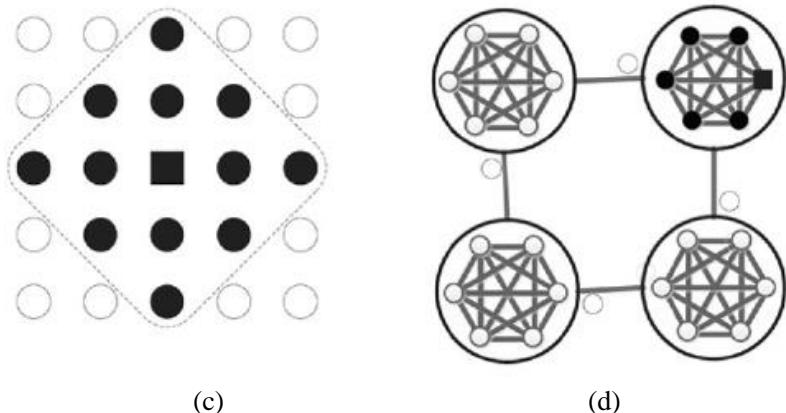


图 5-1 不同邻域关系的结构示意图。■表示当前个体，而●则表示当前个体邻域内的个体。(a) Panmictic 型, (b) Ring 型, (c) Cellular 型, (d) Distributed 型。

Peng 等人于 2017 年提出的 RNDE^[59]为每个个体从种群中随机选择一些个体作为其邻域。Cai 等人于 2017 年提出的 NaDE^[60]将多种基于下标的邻域关系集合成一个候选池，并根据每种邻域关系下个体的进化成功率来决定该邻域关系的被选概率。并利用个体的邻域关系改进了变异策略。例如，对于标准变异策略“DE/rand/1”：

$$V_i^G = X_{r1}^G + F \times (X_{r2}^G - X_{r3}^G) \quad (5-1)$$

在标准的“DE/rand/1”变异策略中， X_{r1}^G 、 X_{r2}^G 和 X_{r3}^G 这三个个体都是从种群中随机选出的。而在 NaDE 中，首先，变异策略的基个体 X_{r1}^G 从目标个体 X_i^G 的邻域中随机选出。随后目标个体的邻域中其他个体以基个体 X_{r1}^G 的适应度为阈值分为两部分，一组适应度均好于基个体的 superior 组，另一组的适应度均不如基个体的 inferior 组。接着，差分向量的终点 X_{r2}^G 从 superior 组随机选出，而差分向量的起点 X_{r3}^G 从 inferior 组随机选出。其示意图可以表示如下：

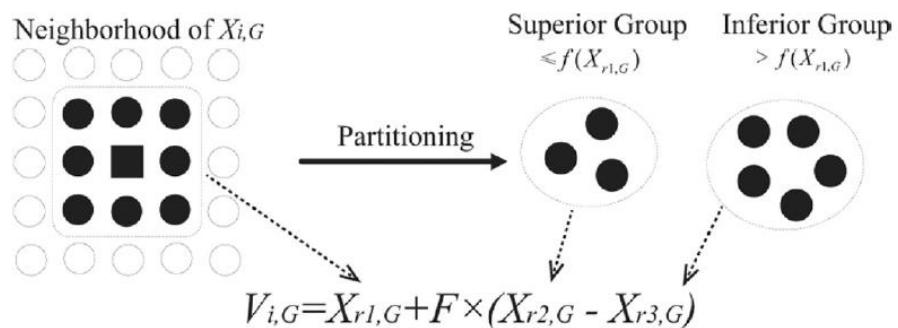


图 5-2 NaDE 中“DE/rand/1”的示意图

5.3 基于适应度划分邻域的差分进化算法

传统的基于邻域的差分进化算法要么通过个体之间的距离进行邻域划分，要么基于个体下标的方法进行邻域划分。通过距离进行邻域划分要计算整个种群中所有个体两两之间的距离，计算量巨大；而通过下标划分邻域则直接根据个体在初始化阶段分

配得到的下标按照一定规则进行划分，这个划分在整个进化进程中都不会改变，而这无疑无法适应种群进化时多变的状况。为了提高差分进化算法对邻域信息的利用效率，提高算法的收敛性能，本小节提出了一种全新的基于适应度进行个体邻域划分的差分进化算法变体 FNDE。

5.3.1 算法动机

邻域关系包含了个体之间的联系信息以及种群在搜索空间中的分布结构信息，对进化方向提供了引导信息。然而目前主流的划分邻域的两种方法：基于距离划分邻域和基于下标划分邻域，都有着自身的缺点，没法高效地提升算法性能。而且大多数的差分进化算法都集中关注于那些适应度较好的个体，迫切地想方法提升这些个体的质量，比如标准的“DE/best/1”变异策略就是一个最为极端的例子，所有个体的进化都只能在 best 个体附近进行搜索。的确，差分进化算法在算法终止后选择最后的 best 个体作为问题的解输出，关注种群最优的那一部分个体的确无可厚非。但是差分进化算法是一种基于群体智能的全局优化算法，只关注最好的那一个或者那一小部分个体，并不能在整体上提升种群质量。在进化前期，这种方法可能会得到一个极快的收敛速度，但在后期，由于忽视了占种群绝大多数的一般或者较差个体，种群有可能陷入局部极值点而无法跳出来。

在常用的 Panmictic 型邻域关系中，每个个体的邻域由种群所有个体组成，如果运用“DE/rand/1”变异策略，则参与变异策略的个体均在整个种群中随机选择，而这与上面提到的“DE/best/1”完全相反，这个策略的随机性极强，在整个种群搜索，探索性极强，而后者在 best 个体附近搜索，以期能提高现有解的质量，开发性极强。“DE/best/1”变异策略可以看作在选择基个体时每个个体的邻域只有一个 best 个体，可将其标记为 best_n 型邻域关系。如果变异策略的个体只在目标个体的邻域内选择，那么 Panmictic 型邻域关系和 best_n 型邻域关系就是两个极端，前者拥有极强的探索性而后者则有着极强的开发性。为了平衡开发性和探索性，对这两种邻域关系模型进行折中是一个很自然且合理的想法。

适应度衡量了一个个体对待求解问题的求解状况，是衡量一个个体优劣的最重要指标，本章将提出一种基于适应度划分邻域的差分进化算法变体。在该算法内，我们更加关注一般或者较差个体，努力使他们进化得更好，从而提高种群的整体质量，提升算法的进化成功率和算法效率。我们从 Panmictic 型邻域关系开始，每个个体的邻域为种群所有个体，同时，为了关注占种群绝大多数的一般个体和较差个体，我们按照适应度从好到差进行排序，不断剔除较差个体的邻域中的较差个体。最终，我们算法划分出的邻域有如下分布规律：较优的个体拥有较大的邻域，几乎包括整个种群，而较差的个体的邻域较小，只含有较好的个体。个体邻域的大小随着该个体的适应度排名下降而不断缩小。按照适应度的分布，个体邻域大小介于 Panmictic 型邻域关系和 best_n 型邻域关系之间，实现了开发性与探索性的平衡。

5.3.2 基于适应度的邻域划分

为了保证邻域关系能及时地反映种群的变化信息，我们在算法的每一代进化之前都对个体邻域信息进行更新。邻域划分方法如下：

- 1) 在每一代进化之前，将整个种群按照适应度从小到大排序，每个个体 X_i^G 得到其相应的排名 r_i ，例如 best 个体的排名为 1。
- 2) 对于某个个体 X_i^G ，其邻域由以下个体组成：

$$\{X_j^G | \text{rand} < \max\left(1 - \frac{r_i}{NP}, 1 - \frac{r_j}{NP}\right)\}, \quad (5-2)$$

其中， r_i, r_j 分别为 X_i^G, X_j^G 的排名， rand 为 [0,1] 间的随机数，NP 为整个种群的个体数目， $\max(x, y)$ 返回 x 和 y 中的较大者。

由 $\{X_j^G | \text{rand} < \max\left(1 - \frac{r_i}{NP}, 1 - \frac{r_j}{NP}\right)\}$ 可知， X_j^G 是否在 X_i^G 的邻域内，主要是看两者适应度排名谁高，由较高者决定是否在邻域内。假如 X_i^G 是个较为优秀的个体，则其排名 r_i 比较小， $1 - \frac{r_i}{NP}$ 较大，因此在上式中， \max 函数返回的值只能大于或等于 $1 - \frac{r_i}{NP}$ 这个较大的值。所以对于较优秀的个体，其邻域内包含的个体较多，几乎包含整个种群。而假设 X_i^G 是个较差的个体，则其排名 r_i 比较大， $1 - \frac{r_i}{NP}$ 较小，因此在上式中， X_j^G 是否在 X_i^G 的邻域内取决于 X_j^G 自身是否足够优秀。因此对于那些较差的个体，其邻域内包含的个体较少，几乎只包含那些比较优秀的个体。但随机数 rand 给这种分布增加了随机性，增强了多样性。再者，由上面的邻域划分方式可以知道，最优秀的那一部分个体几乎在所有个体的邻域内，而且随着个体的变差，这部分优秀个体在邻域内的占比越来越大，而优秀个体往往包含着很多优秀的子成分，这有利于优秀子成分在整个种群内快速扩散，提升算法的收敛速度。

因为邻域关系提供了个体之间的联系信息，因此可以利用邻域关系来指导变异策略。在基本的“DE/rand/1”变异策略中：

$$V_i^G = X_{r1}^G + F \times (X_{r2}^G - X_{r3}^G) \quad (5-3)$$

所有的个体从整个种群中随机选择，而划分邻域后，参与变异策略的个体则从目标个体的邻域内选择。选择方法如下：

- 1) 基个体 X_{r1}^G 从目标个体的邻域内随机选出；
- 2) X_{r2}^G 也从目标个体的邻域内选出，但其适应度排名 r_{r2} 要满足 $\text{rand} < (1 - \frac{r_{r2}}{NP})$ ；
- 3) X_{r3}^G 也从目标个体的邻域内选出，但其适应度排名 r_{r3} 要满足 $\text{rand} > (1 - \frac{r_{r3}}{NP})$ 。

改进的变异策略对于那些较优秀的个体而言，就相当于退化回基本变异策略“DE/rand/1”，因为这些优秀个体的邻域几乎包含整个种群，这保证了算法的探索性。而对那些较差个体来说，该策略就相当于在一个由优秀个体组成的小群体内选择个体参与变异，因此有很大的概率能选上一个很优秀的个体做基个体。而基个体在变异策略中相当重要，引导变异策略在其附近搜索，而这提升了这些较差个体的进化成功率，提高种群的质量，加快算法的收敛速度，而这确保了算法的开发性。综合来说，该邻域

划分方法平衡了开发性和探索性。

综上所述, FNDE 的算法流程图表示如下:

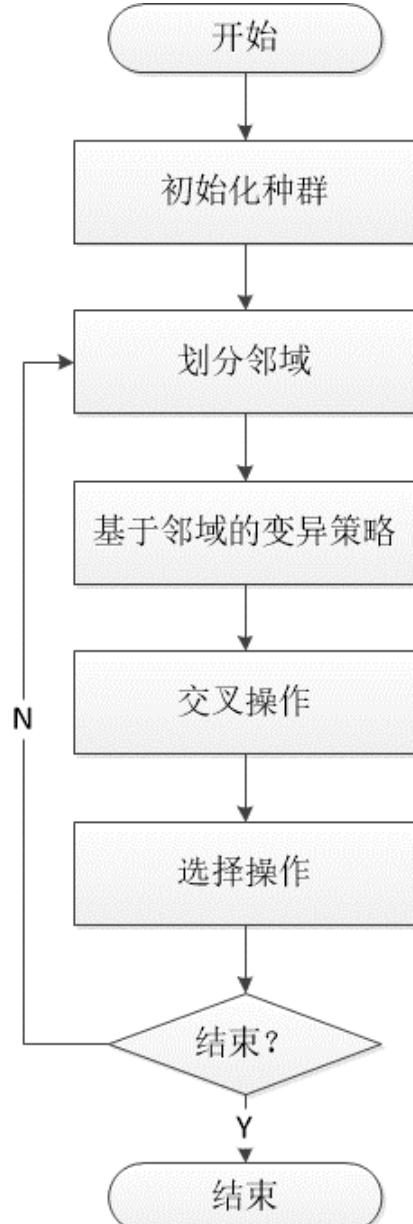


图 5-2 FNDE 算法流程图

5.4 实验结果与分析

在本节的实验中, 我们将通过一系列的实验来证明 FNDE 的有效性。我们将选取 CEC2005 测试函数集作为测试集, 该测试函数集一共有 25 个函数, 可以分为以下四类: 5 个单模函数(unimodal function)F1-F5, 7 个基本多模函数(basic multimodal function) F6-F12, 2 个扩展多模函数(expanded multimodal function) F13 和 F14, 11 个混合组合函数(hybrid composition function) F15-F25。在本小节中, 我们将 FNDE 与 jDE, NaDE^[60] 在 D=30 和 D=50 时进行比较, 其中 NaDE 为 5.2 小节中介绍的多种邻域关系自适应选

择的差分进化算法变体。三种比较函数都在基础变异策略“DE/rand/1”上进行比较。所有比较算法的种群大小都设置为 NP=100, 最大的适应度函数计算次数(Max_NFFEs)设置为 $D \times 10000$, 其中 D 为待求解问题的维数, 所有实验结果均为独立运行 25 次后得到的平均结果。另外, 边界处理问题对差分进化算法的性能有着重要影响, 因此对本论文接下来在 CEC2005 测试函数集上的所有实验, 都采取如下边界处理原则: 若个体的某一维超出搜索空间的上下限时, 我们将对这一维进行重新初始化。其中最好的实验结果将由粗体标出。下表为 D=30 时三个算法在 25 个函数的结果比较:

表 5-1 D=30 时, jDE, NaDE 和 FNDE 在所有测试函数的比较结果

	jDE	NaDE	FNDE
F1	0	0	0
F2	1.08E-05	2.12E-06	4.02E-13
F3	1.89E+05	2.06E+05	5.50E+04
F4	2.98E-01	9.35E-01	3.96E-04
F5	1.10E+03	1.06E+03	6.99E+02
F6	2.46E+01	6.14E+00	1.59E-01
F7	1.31E-02	9.16E-03	1.02E-02
F8	2.09E+01	2.10E+02	2.09E+01
F9	0	0	0
F10	5.86E+01	4.75E+01	4.86E+01
F11	2.80E+01	2.76E+01	2.05E+01
F12	1.16E+04	4.43E+03	2.51E+03
F13	1.70E+00	1.66e+00	2.46E+00
F14	1.30E+01	1.29E+01	1.28E+01
F15	3.40E+02	3.19E+02	4.00E+02
F16	7.56E+01	7.13E+01	4.99E+01
F17	1.33E+02	1.18E+02	1.03E+02
F18	9.07E+02	9.06E+02	9.05E+02
F19	9.06E+02	9.06E+02	9.03E+02
F20	9.06E+02	9.06E+02	5.53E+02
F21	5.00E+02	5.00E+02	5.00E+02
F22	9.04E+02	9.01E+02	9.01E+02
F23	5.34E+02	5.34E+02	5.34E+02
F24	2.00E+02	2.00E+02	2.00E+02
F25	2.10E+02	2.09 E+02	2.09E+02

由上表可以看出, 在 D=30 时, FNDE 在 F2 到 F6 这五个函数的结果有数量级的提升, 虽然在 F7, F10, F13 和 F15 这四个函数有所不如, 但相差的幅度并不大。可以说 FNDE 在大部分的函数都能取得很好的结果, 远远优胜于 jDE 和 NaDE。下表是 D=50 时三个比较函数在所有测试函数的结果。

表 5-2 D=50 时, jDE, NaDE 和 FDDE 在所有测试函数的比较结果

	jDE	NaDE	FDDE
F1	0	0	0
F2	8.99e-02	2.29e-01	1.06e-08
F3	5.30e+05	1.53e+06	1.23e+05
F4	8.31e+02	9.18e+02	1.78e+02
F5	3.39e+03	3.59e+03	3.78e+03
F6	3.98e+01	3.58e+01	2.37e+00
F7	4.13e-03	4.03e-03	7.40e-03
F8	2.11e+01	2.11e+01	2.11e+01
F9	0	0	3.18e-14
F10	5.86e+01	8.49e+01	9.00e+01
F11	1.00e+02	5.50e+01	4.38e+01
F12	3.71e+04	1.59e+04	1.30e+04
F13	2.90e+00	3.06e+00	4.33e+00
F14	2.26e+01	2.26e+01	2.21e+01
F15	3.32e+02	2.96e+02	2.50e+02
F16	8.54e+01	7.25e+01	6.42e+01
F17	1.75e+02	1.53e+02	1.27e+02
F18	9.25e+02	9.26e+02	9.22e+02
F19	9.25e+02	9.25e+02	9.33e+02
F20	9.25e+02	9.25e+02	5.73e+02
F21	5.00e+02	5.00e+02	5.00e+02
F22	9.43e+02	9.39e+02	9.15e+02
F23	5.39e+02	5.39e+02	5.39e+02
F24	2.00e+02	2.00e+02	2.00e+02
F25	2.14e+02	2.14e+02	2.14e+02

由上表可以看出 FDDE 仅在 F5、F7、F9、F10、F13 和 F19 这六个函数有所不如，另外两个比较算法，在其余 19 个函数都取得了较好的结果，在 F2、F6 等函数中还取得了数量级的突破。

虽然 NaDE 是多种基于下标的邻域关系的自适应选择，但基于下标的邻域划分始终只是单纯地按照随机分配的下标进行划分，而且已经划分好就无法更改。而 FNDE 基于适应度划分邻域，随时跟踪种群的变化，每代都在更新邻域关系，且更为关注种群中一般和较差的个体，努力提升这些个体的质量，进而提升种群的整体质量，加快算法的收敛速度。为此，我们将在 D=30 时，在 F1 函数上分别统计运行 FNDE 和 jDE 时每个个体的进化成功率，最大的适应度函数计算次数 (Max_NFFEs) 设置为 20000，独立运行 50 次后求平均得到每个个体的进化成功率如下图所示：

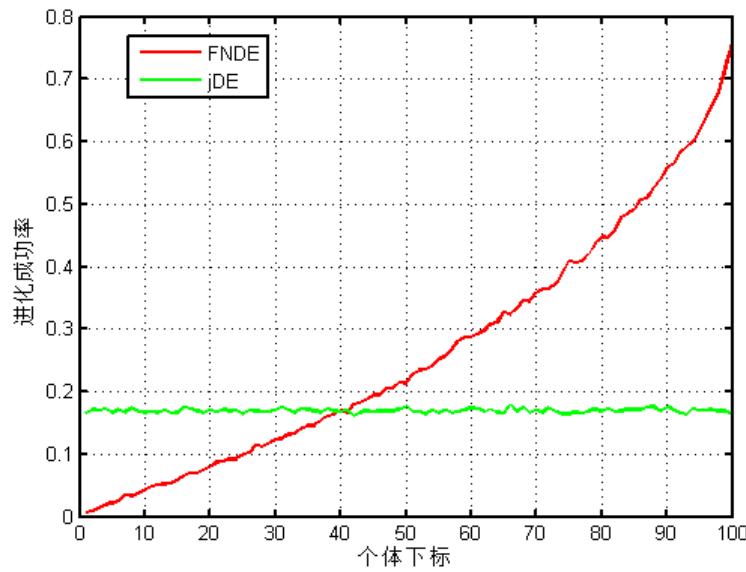


图 5-2 FNDE（红线）和 jDE（绿线）每个个体进化成功率比较图

若目标个体经过变异操作和交叉操作产生的试验个体最终在选择操作中打败目标个体，成功进入下一代，则认为该个体进化成功了。为了方便比较，图 5-2 中的个体下标是根据个体适应度从小到大排序的，也即下标越小，该个体越优秀。由上图可以看出 jDE 中每个个体的进化成功率都差不多，种群总体的进化成功率为 16.9%，而 FNDE 的个体进化成功率则随着下标的递增而越来越高，而且在下标为 40 之后就完全超过了 jDE，下标为 100 的个体的进化成功率几乎已经达到 80%，而且算法的总体进化成功率为 26.5%，几乎比 NaDE 高 10%。这验证了我们算法关注一般和较差个体，以期提高他们的质量进而提高种群整体质量，以提升种群进化成功率和收敛速度的目的。由上图可以看出我们的算法的确能在提升一般和较差个体的基础上提升算法整体的进化成功率。

5.5 本章小结

本章提出了一种新的基于邻域关系的差分进化算法 FNDE，与传统的基于距离或者个体下标划分邻域的方法不同，FNDE 主要基于个体适应度进行邻域划分，在邻域划分时根据每个个体的适应度排名而进行相应的自适应调整。FNDE 更为关注在适应度上表现一般或者较差的弱势个体，对较优秀的个体会将种群绝大部分的个体都划进其邻域，而对那些弱势个体，则将它们的邻域划得较小，并且邻域内都是些较为优秀的个体。在改进的变异策略，较优的个体就退化为与标准的变异策略差不多，足够大的邻域保证了算法的探索性，而对于弱势个体，由优秀个体组成的较小邻域则保证了算法的开发性，两者结合一起平衡了算法的开发性与探索性。而且 FNDE 更为关注弱势个体可以从整体上提高算法的进化成功率，取得更好的结果，而这些都有实验结果加以佐证。

6 结论与展望

本论文主要研究了高性能差分进化算法，在近年有关差分进化算法的一些代表性算法的基础上，进行了相应的分析与研究，并提出了自己的改进算法，其中主要涉及种群结构划分，变异策略个体选择方法和个体邻域划分等。在差分进化算法中，开发性和探索性是算法中最为重要的特性，两者的平衡与否对算法性能有着重要影响。本论文所有的工作都是基于平衡开发性和探索性这个目标出发的，提出新的改进方法以提高差分进化算法的性能。

6.1 论文研究总结

经典的差分进化算法因其简单高效、易于实现的特点，在众多领域有着广泛应用。特别是在复杂优化问题中，因其不受限于初始模型而在很多情况下表现出优越的优化性能，因而受到越来越多研究者的关注。但是，差分进化算法本身依然存在许多问题，本论文主要从平衡开发性和探索性的目的出发，对差分进化算法进行了相应的研究分析并提出了三种简单高效的差分进化算法变体。具体的工作内容包括以下三个方面：

(1) 经典的差分进化算法运用到高维波形反演问题时，由于搜索空间急剧增大，计算量也急剧增长，经典算法一般很难在高维波形反演问题中取得较好的结果。为了提高差分进化算法应用到波形反演问题中的效率，我们提出了一种基于个体到种群中心的距离来划分种群结构的多子种群差分进化算法 SpDE (Differential Evolution with Subpopulations)。SpDE 根据个体到种群中心的距离将整个种群划分为三个子种群，每个个体在所属的子种群内进行变异，内层子种群负责开发性，外层子种群负责探索性。接着提出一种分两阶段的变异策略将三个子种群联结起来，实现开发性与探索性之间的平衡。最后，我们还对交叉操作进行改进，将当前种群内最优的子成分组成一个新的个体，并将该个体加入到交叉操作中以加快这些最优子成分在种群内的扩散速度以加快算法收敛速度。在随后的应用于合成地震数据和真实地震数据的波形反演实验中，SpDE 在收敛速度和收敛精度上都优胜于 MMDE 和 CCDE，能用 MMDE 十分之一的进化代数反演出相似的速度模型，大大提升了差分进化算法在波形反演问题中的应用效率，而且 SpDE 对噪声具有良好的稳定性。

(2) 变异操作是差分进化算法中最为重要的一个操作，它为整个种群产生新的个体，引入新的信息。而基个体又是整个变异操作中最为重要的个体，它为变异操作指明搜索区域。因此如何安排个体在变异策略中的位置对差分进化算法的性能有着重要影响。因此本论文第四章提出了一种基于变异策略个体选择的差分进化算法变体FDDE 算法。FDDE 定义了个体对种群多样性的贡献，并综合考虑个体适应度和对种群多样性贡献来决定每个个体的排名。因为变异策略中基个体和差分向量的重要性不同，因此 FDDE 根据个体排名来安排个体在变异策略中的位置。为了验证FDDE的优化性能，我

们在CEC2005测试函数集上进行了测试，实验结果显示FDDE在绝大多数的函数都能取得最好的结果。

(3) 邻域关系能反映个体之间的联结信息，从侧面反映了种群在整个搜索空间的分布情况，能给种群进化提供指导信息。传统的邻域划分主要是基于距离和个体下标，然而两者都受限于自身的不足而无法进一步提升算法性能。本论文第五章提出了一种新的基于邻域关系的差分进化算法变体 FNDE，与传统的基于距离或者个体下标划分邻域的方法不同，FNDE 主要基于个体适应度进行邻域划分，在邻域划分时根据每个个体的适应度排名而进行相应的自适应调整。FNDE 更为关注在适应度上表现一般或者较差的弱势个体，对较优秀的个体将种群绝大部分个体都划进其邻域，而对那些弱势个体，则将它们的邻域划得较小，并且邻域内都是些较为优秀的个体。在改进的变异策略中，较优个体的就退化为与标准的变异策略差不多，足够大的邻域保证了算法的探索性，而对于弱势个体，由优秀个体组成的较小邻域则保证了算法的开发性，两者结合一起平衡了算法的开发性与探索性。该改进能有效地提高差分进化算法的优化性能。

6.2 展望

本论文基于经典的差分进化算法，从平衡算法的开发性与探索性这个目的出发，从算法种群结构划分，变异策略个体选择和个体邻域划分这三个方面对差分进化算法做出了改进。当前研究者们对差分进化算法的研究越来越深入，本论文的研究内容仅仅只是该领域很小的一部分，还有许多研究内容尚待进一步的研究与完善，基于我的理解，主要还有以下几个方面可以作进一步深入研究：

(1) 差分进化算法是一种参数少而高效的优化算法，它只有三个参数：变异系数 F 、交叉率 CR 和种群大小 NP 。已经有许多研究论文对前两个参数进行了深入分析与研究，提出来很多改进算法。然而对于种群大小 NP ，却没有多少研究论文对其进行改进，大多数的差分进化算法都是直接选取一个固定值。因为差分进化算法的比较是在一定的函数计算次数 Max_NFEs 下进行的， NP 越大，算法的进化代数就会越小，每个个体能迭代的次数就变少。因此种群大小自适应也能很好的提升算法性能，在算法进化前期，我们需要一个较大的种群来保证足够的探索性以让算法能定位到最优解所在的区域，在算法后期，我们可能需要减少种群数量以增加进化代数，让较少的较优个体能进化多些代数以提高解的质量。因此这种种群数量随着进化代数递减的方法是有利于算法进化，然而这种递减要如何更好地设计，则需要进一步的研究。

(2) 本论文的三个研究内容都是基于 jDE 的参数自适应选择，而 F 和 CR 的参数自适应选择是当前比较热门的研究方向而且对算法性能有着很大的提升。优秀的参数有很大的概率能产生优秀的个体，因此利用个体的进化成功信息对参数进行自适应调整，并将该参数控制方法加入到论文的研究内容中，这能进一步提升算法性能。因此找到一种更为有效的基于个体成功进化历史的参数自适应调整方法是需要进一步研究的内容。

致 谢

本论文是在导师潘志斌教授的悉心指导下完成的。潘老师严谨的科学态度，精益求精的工作作风，诲人不倦的高尚师德，朴实无华、平易近人的人格魅力深深地感染和激励着我。从本科大四开始进入实验室到如今硕士毕业这四年里，潘老师都始终给予我细心的指导和不懈的支持。在科研生活中，潘老师以他开阔的视野给我指明了研究的方向，当我在科研上遇到苦难时，潘老师总能以他渊博的学识给予我及时的指导与帮助，帮我克服科研上一个又一个的难题。在生活中，潘老师谦逊有礼的行事作风和平易近人的人格魅力让我们由衷地尊敬与热爱。值此毕业之际，谨向我们敬爱的潘老师致以深深的敬意与由衷的感激。

感谢吴俊杰师兄在本科毕设期间对我的指导，引领我跨入了地震波形反演和差分进化算法研究的大门。感谢高照奇师兄和程建超师弟在学术问题上对我的帮助，使我在课题研究中不断进步。感谢一起进组的杜东、李正伊和张蕊，相互陪伴，一起走过了这精彩的三年研究生生活。感谢实验室的师兄弟姐妹们，在课题科研中，我们互相讨论共同进步，在日常生活中，我们互助互爱，有了你们，我这三年里才能如此多姿多彩。

感谢班里、学校里的同学们和朋友们陪伴我度过了三年快乐的时光，感谢你们这三年来的鼓励与支持。祝你们将来工作顺利、实现理想。

最后，我要感谢我的爸爸妈妈，是你们起早摸黑含辛茹苦地培养了我们兄妹三人，是你们赋予了我前进的动力。在今后的工作和生活中，我会带着你们的期望和鼓励努力拼搏，开拓进取。

参考文献

- [1] Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces[J]. *Journal of Global Optimization*, 1997, 11(4):341-359.
- [2] 王家映. 地球物理资料非线性反演方法讲座(一) 地球物理反演问题概述[J]. *工程地球物理学报*, 2007, 4: 1-3.
- [3] Semnani A, Kamyab M, Rekanos IT. Reconstruction of one-dimensional dielectric scatterers using differential evolution and particle swarm optimization[J]. *IEEE Geoscience & Remote Sensing Letters*, 2009, 6 (4): 671-675.
- [4] Dehmollaian M. Through-wall shape reconstruction and wall parameters estimation using differential evolution[J]. *IEEE Geoscience & Remote Sensing Letters*, 2011, 8 (2): 201-205.
- [5] Wang C, Gao J. A new differential evolution algorithm with Cooperative Coevolutionary selection operator for waveform inversion[C]: 30th IEEE International Geoscience and Remote Sensing Symposium, 2010: 688-690.
- [6] Wang C, Gao J, Yang H, et al. Waveform inversion of cross-well data with cooperative coevolutionary differential evolution algorithm[C]: SEG, 2011: 2752-2756.
- [7] Brest J, Greiner S, Boskovic B, et al. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems[J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(6):646-657.
- [8] Qin A K, Huang V L, Suganthan P N. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2):398-417.
- [9] Gao Z, Pan Z, Gao J. A New Highly Efficient Differential Evolution Scheme and Its Application to Waveform Inversion[J]. *IEEE Geoscience & Remote Sensing Letters*, 2017, 11(10):1702-1706.
- [10] Tanabe R, Fukunaga A. Success-history based parameter adaptation for Differential Evolution[C] Evolutionary Computation. IEEE, 2013:71-78.
- [11] Tanabe R, Fukunaga A S. Improving the search performance of SHADE using linear population size reduction[C] Evolutionary Computation. IEEE, 2014:1658-1665.
- [12] Pan Z, Wu J, Gao Z, et al. Adaptive Differential Evolution by Adjusting Subcomponent Crossover Rate for High-Dimensional Waveform Inversion[J]. *IEEE Geoscience & Remote Sensing Letters*, 2015, 12(6):1327-1331.
- [13] Mallipeddi R, Suganthan P N, Pan Q K, et al. Differential evolution algorithm with ensemble of parameters and mutation strategies[J]. *Applied Soft Computing*, 2011, 11(2):1679-1696.
- [14] Wang Y, Cai Z, Zhang Q. Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters[J]. *IEEE Transactions on Evolutionary Computation*, 2011, 15(1):55-66.
- [15] Gong W, Cai Z. Differential Evolution With Ranking-Based Mutation Operators[J]. *IEEE Transactions on Cybernetics*, 2013, 43(6):2066-2081.
- [16] Zhang W J, Xie X F. DEPSO: Hybrid particle swarm with differential evolution operator[J]. 2003, 4:3816 - 3821.
- [17] Das S, Konar A, Chakraborty U K. Annealed Differential Evolution[C] Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2008:1926-1933.
- [18] Das S, Suganthan P N. Differential Evolution: A Survey of the State-of-the-Art[J]. *IEEE Transactions on Evolutionary Computation*, 2011, 15(1):4-31.
- [19] Wang C, Gao J. High-Dimensional Waveform Inversion With Cooperative Coevolutionary Differential Evolution Algorithm[J]. *IEEE Geoscience & Remote Sensing Letters*, 2012, 9(2):297-301.

- [20] Suganthan P N, Hansen N, Liang J J, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization[J]. KanGAL report, 2005, 2005005: 2005.
- [21] Belina F A, Ernst J R, Holliger K. Inversion of crosshole seismic data in heterogeneous environments: Comparison of waveform and ray-based approaches[J]. Journal of Applied Geophysics, 2009, 68(1): 85-94.
- [22] Pan Z, Liang H, Gao Z, et al. Differential evolution with subpopulations for high-dimensional seismic inversion[J]. Geophysical Prospecting, 2018.
- [23] B. Růžek, M. Kvasnička. Differential Evolution Algorithm in the Earthquake Hypocenter Location[J]. Pure & Applied Geophysics, 2001, 158(4):667-693.
- [24] Roecker S. A global optimizing approach for waveform inversion of receiver functions[J]. Computers & Geosciences, 2010, 36 (7): 871–880.
- [25] Gao Z, Pan Z, Gao J. Multimutation Differential Evolution Algorithm and Its Application to Seismic Inversion[J]. IEEE Transactions on Geoscience & Remote Sensing, 2016, 54(6):3626-3636.
- [26] Das S, Mullick S S, Suganthan P N. Recent advances in differential evolution – An updated survey[J]. Swarm & Evolutionary Computation, 2016, 27:1-30.
- [27] Zhao S Z, Suganthan P N, Das S. Self-adaptive differential evolution with multi-trajectory search for large-scale optimization[J]. Soft Computing, 2011, 15(11):2175-2185.
- [28] Mallipeddi R, Suganthan P N, Pan Q K, et al. Differential evolution algorithm with ensemble of parameters and mutation strategies[J]. Applied Soft Computing, 2011, 11(2):1679-1696.
- [29] Wang Y, Cai Z, Zhang Q. Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1):55-66.
- [30] Gong W, Cai Z, Ling C X, et al. Enhanced differential evolution with adaptive strategies for numerical optimization[J]. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 2011, 41(2):397-413.
- [31] Adam P. Piotrowski, Jaroslaw J. Napiorkowski, Adam Kiczko. Differential Evolution algorithm with Separated Groups for multi-dimensional optimization problems[J]. European Journal of Operational Research, 2012, 216(1):33-46.
- [32] Price K, Storn R M, Lampinen J A. Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)[M]. Springer-Verlag New York, Inc. 2005.
- [33] Kaelo P, Ali M M. A numerical study of some modified differential evolution algorithms[J]. European Journal of Operational Research, 2006, 169(3):1176-1184.
- [34] Zhang J, Sanderson A C. JADE: Adaptive Differential Evolution With Optional External Archive[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5):945-958.
- [35] Huo P, Qiu C, Gong W. Differential Evolution with Mutation Operators Based on Truncation[C] International Conference on Computational & Information Sciences. IEEE, 2013:742-745.
- [36] Wang J, Liao J, Zhou Y, et al. Differential evolution enhanced with multiobjective sorting-based mutation operators[J]. IEEE Transactions on Cybernetics, 2017, 44(12):2792-2805.
- [37] Sharifi-Noghabi H, Mashhadi H R, Shoaee K. A novel mutation operator based on the union of fitness and design spaces information for Differential Evolution[J]. Soft Computing, 2016:1-8.
- [38] Cai Y, Chen Y, Wang T, et al. Improving differential evolution with a new selection method of parents for mutation[J]. 中国计算机科学前沿:英文版, 2016, 10(2):246-269.
- [39] Jiang L, Qiang H. Differential evolution algorithm with base vector group[C]. Control Conference. IEEE, 2013:8006-8009.
- [40] Gong W, Cai Z, Liang D. Adaptive ranking mutation operator based differential evolution for constrained optimization.[J]. IEEE Transactions on Cybernetics, 2014, 45(4):716-727.
- [41] Falco I D, Cioppa A D, Maistro D, et al. Impact of the Topology on the Performance of Distributed Differential Evolution[C] European Conference on the Applications of Evolutionary Computation.

- Springer, Berlin, Heidelberg, 2014:75-85.
- [42] Das S, Abraham A, Chakraborty U K, et al. Differential evolution using a neighborhood-based mutation operator[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(3):526-553.
- [43] Piotrowski A P. Adaptive Memetic Differential Evolution with Global and Local neighborhood-based mutation operators[J]. Information Sciences, 2013, 241(12):164-194.
- [44] Hu Z, Cai X, Fan Z. An improved memetic algorithm using ring neighborhood topology for constrained optimization[M]. Springer-Verlag, 2014.
- [45] Dorronsoro B, Bouvry P. Improving Classical and Decentralized Differential Evolution With New Mutation Operator and Population Topologies[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1):67-98.
- [46] Epitropakis M G, Tasoulis D K, Pavlidis N G, et al. Enhancing Differential Evolution Utilizing Proximity-Based Mutation Operators[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1):99-119.
- [47] Cai Y, Wang J, Yin J. Learning-enhanced differential evolution for numerical optimization[M]. Springer-Verlag, 2012.
- [48] Sun Y, Li Y, Liu G, et al. A Novel Differential Evolution Algorithm with Adaptive of Population Topology[C]. International Conference on Information Computing and Applications. Springer Berlin Heidelberg, 2012:531-538.
- [49] Biswas S, Kundu S, Bose D, et al. Synchronizing Differential Evolution with a modified affinity-based mutation framework[C]. Differential Evolution. IEEE, 2013:61-68.
- [50] Liang J, Qu B, Mao X, et al. Differential Evolution Based on Fitness Euclidean-Distance Ratio for Multimodal Optimization[J]. Neurocomputing, 2014, 137(15):252-260.
- [51] Sarkar S, Mukherjee R, Biswas S, et al. An Adaptive Clustering and Re-clustering Based Crowding Differential Evolution for Continuous Multi-modal Optimization[C]. Asia Pacific Symposium on Intelligent and Evolutionary Systems. 2015:373-388.
- [52] Liao J, Cai Y, Chen Y, et al. Differential evolution enhanced with composite population information based mutation operators[J]. Journal of Digital Information Management, 2015, 13(6):1051-1056.
- [53] Wang S C. Differential evolution optimization with time-frame strategy adaptation[J]. Soft Computing, 2016:1-22.
- [54] Zhou Y Z, Yi W C, Gao L, et al. Adaptive Differential Evolution With Sorting Crossover Rate for Continuous Optimization Problems[J]. IEEE Transactions on Cybernetics, 2017, PP(99):1-12.
- [55] Noman N, Iba H. Cellular Differential Evolution Algorithm[C] Ai 2010: Advances in Artificial Intelligence -, Australasian Joint Conference, Adelaide, Australia, December 7-10, 2010. Proceedings. DBLP, 2010:293-302.
- [56] Noroozi V, Hashemi A B, Meybodi M R. CellularDE: a cellular based differential evolution for dynamic optimization problems[C] International Conference on Adaptive and Natural Computing Algorithms. Springer-Verlag, 2011:340-349.
- [57] Weber M, Neri F, Tirronen V. A study on scale factor in distributed differential evolution[J]. Information Sciences, 2011, 181(12):2488-2511.
- [58] Neri F. Disturbed Exploitation compact Differential Evolution for limited memory optimization problems[J]. Information Sciences, 2011, 181(12):2469-2487.
- [59] Peng H, Guo Z, Deng C, et al. Enhancing differential evolution with random neighbors based strategy[J]. Journal of Computational Science, 2017.
- [60] Cai Y Q, Sun G, Wang T, et al. Neighborhood-adaptive differential evolution for global numerical optimization[J]. Applied Soft Computing, 2017, 59: 659-706.

攻读学位期间取得的研究成果

- [1] Pan Z, Liang H, Gao Z, et al. Differential evolution with subpopulations for high-dimensional seismic inversion[J]. Geophysical Prospecting, 2018.

声明

学位论文独创性声明（1）

本人声明：所呈交的学位论文系在导师指导下本人独立完成的研究成果。文中依法引用他人的成果，均已做出明确标注或得到许可。论文内容未包含法律意义上已属于他人的任何形式的研究成果，也不包含本人已用于其他学位申请的论文或成果。

本人如违反上述声明，愿意承担以下责任和后果：

1. 交回学校授予的学位证书；
2. 学校可在相关媒体上对作者本人的行为进行通报；
3. 本人按照学校规定的方式，对因不当取得学位给学校造成的名誉损害，进行公开道歉。
4. 本人负责因论文成果不实产生的法律纠纷。

论文作者（签名）： 日期： 年 月 日

学位论文独创性声明（2）

本人声明：研究生_____所提交的本篇学位论文已经本人审阅，确系在本人指导下由该生独立完成的研究成果。

本人如违反上述声明，愿意承担以下责任和后果：

1. 学校可在相关媒体上对本人的失察行为进行通报；
2. 本人按照学校规定的方式，对因失察给学校造成的名誉损害，进行公开道歉。
3. 本人接受学校按照有关规定做出的任何处理。

指导教师（签名）： 日期： 年 月 日

学位论文知识产权权属声明

我们声明，我们提交的学位论文及相关的职务作品，知识产权归属学校。学校享有以任何方式发表、复制、公開阅览、借阅以及申请专利等权利。学位论文作者离校后，或学位论文导师因故离校后，发表或使用学位论文或与该论文直接相关的学术论文或成果时，署名单位仍然为西安交通大学。

论文作者（签名）： 日期： 年 月 日

指导教师（签名）： 日期： 年 月 日

(本声明的版权归西安交通大学所有，未经许可，任何单位及任何个人不得擅自使用)