

信息科学技术学术著作丛书

# 差分演化算法的理论与应用

熊盛武 胡中波 苏清华 著



科学出版社

信息科学技术学术著作丛书

# 差分演化算法的理论与应用

熊盛武 胡中波 苏清华 著

本书得到湖北省教育厅重点科研项目(D20161306)、  
湖北省自然科学基金重点项目(2015CFA059)、  
湖北省科技支撑计划项目(2014BAA146)和  
河南省科技开放合作项目(2106000048)资助

科学出版社

北京

## 内 容 简 介

全书内容分为差分演化算法(以下简称算法)的理论与应用两篇。理论篇主要内容包括算法的不确保依概率收敛性的理论分析、算法依概率收敛的充分条件、改进算法的收敛性分析、辅助算法收敛的子空间聚类算子的设计。应用篇主要内容包括收敛算法在螺旋压缩弹簧参数优化问题中的应用、算法在薄膜太阳能电池抗反射层微结构设计和在彩色图像颜色量化问题优化中的应用。

本书可作为高等院校数学、信息与计算机类专业的教材或参考书,也可供从事智能优化应用研究的相关工作人员参考。

### 图书在版编目(CIP)数据

差分演化算法的理论与应用/熊盛武,胡中波,苏清华著. —北京:科学出版社,2017.5

(信息科学技术学术著作丛书)

ISBN 978-7-03-052736-3

I. ①差… II. ①熊… ②胡… ③苏… III. 差分法-研究 IV. O241.3

中国版本图书馆 CIP 数据核字(2017)第 101427 号

责任编辑:魏英杰 / 责任校对:桂伟利

责任印制:张 伟 / 封面设计:陈 敬

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

北京教图印刷有限公司印刷

科学出版社发行 各地新华书店经销

\*

2017年5月第一版 开本:720×1000 1/16

2017年5月第一次印刷 印张:13 1/4

字数:268 000

定价:95.00元

(如有印装质量问题,我社负责调换)

## 《信息科学技术学术著作丛书》序

21世纪是信息科学技术发生深刻变革的时代,一场以网络科学、高性能计算和仿真、智能科学、计算思维为特征的信息科学革命正在兴起。信息科学技术正在逐步融入各个应用领域并与生物、纳米、认知等交织在一起,悄然改变着我们的生活方式。信息科学技术已经成为人类社会进步过程中发展最快、交叉渗透性最强、应用面最广的关键技术。

如何进一步推动我国信息科学技术的研究与发展;如何将信息技术发展的新理论、新方法与研究成果转化为社会发展的新动力;如何抓住信息技术深刻发展变革的机遇,提升我国自主创新和可持续发展的能力?这些问题的解答都离不开我国科技工作者和工程技术人员的求索和艰辛付出。为这些科技工作者和工程技术人员提供一个良好的出版环境和平台,将这些科技成就迅速转化为智力成果,将对我国信息科学技术的发展起到重要的推动作用。

《信息科学技术学术著作丛书》是科学出版社在广泛征求专家意见的基础上,经过长期考察、反复论证之后组织出版的。这套丛书旨在传播网络科学和未来网络技术,微电子、光电子和量子信息技术、超级计算机、软件和信息存储技术,数据知识化和基于知识处理的未来信息服务业,低成本信息化和用信息技术提升传统产业,智能与认知科学、生物信息学、社会信息学等前沿交叉科学,信息科学基础理论,信息安全等几个未来信息科学技术重点发展领域的优秀科研成果。丛书力争起点高、内容新、导向性强,具有一定的原创性;体现出科学出版社“高层次、高质量、高水平”的特色和“严肃、严密、严格”的优良作风。

希望这套丛书的出版,能为我国信息科学技术的发展、创新和突破带来一些启迪和帮助。同时,欢迎广大读者提出好的建议,以促进和完善丛书的出版工作。

中国工程院院士

原中国科学院计算技术研究所所长



## 前 言

差分演化算法是一类新兴的应用广泛的演化算法,算法实现简单、经验参数少、稳健性强。差分演化算法是在历届演化优化算法的国际竞赛中唯一一个一直排名前五的算法。自1995年提出以来的20年的时间里,差分演化算法吸引了越来越多不同工程领域的科技工作者的关注与研究,算法已经被应用到30多个领域,算法源码也被嵌入到MATLAB、OPTIMUS等十多个常用软件包。

与差分演化算法的应用研究相比,算法的理论研究进展缓慢,关于差分演化算法的收敛性理论研究成果更少;存在为数不多的依概率收敛的差分演化算法被提出,但该类算法往往因为求全能力与求精能力的不平衡导致算法效率不高,达不到理论上的预期效果,因此在理论上收敛的算法的工程应用研究进展缓慢。针对这些不足之处,团队围绕差分演化算法的收敛理论、依概率收敛差分演化算法的设计及其在压缩弹簧参数优化、太阳能电磁抗反射结构参数优化和彩色图像颜色量化等多个领域的应用,进行了多年的研究。基于本实验室的研究成果,本书讲述差分演化算法的收敛性理论及其应用的部分研究成果。

本书分两篇共9章,第1章综述差分演化算法的理论研究、算法设计研究和工程应用研究的相关成果。第一篇从第2章~第5章讲述差分演化算法的收敛理论与收敛算法的设计。其中,第2章和第3章讲述差分演化算法的收敛性理论研究成果;第4章和第5章讲述依概率收敛差分演化算法的设计。第二篇从第6章~第9章分别介绍差分演化算法在螺旋压缩弹簧参数优化、太阳能电磁抗反射结构参数优化和彩色图像颜色量化问题中的应用。

本书集中了团队多年的研究成果,得到湖北省教育厅重点科研项目(D20161306)资助。书中第1章~第6章包含胡中波博士阶段的部分研究成果,第7章包含赵永翔博士后阶段的部分研究成果,第8和9章包含苏清华博士阶段的部分研究成果。全书由长江大学胡中波博士统稿。

学识所限,不妥之处在所难免,敬请读者指正,不胜感激!



2016年5月于武汉理工大学

# 目 录

《信息科学技术学术著作丛书》序

前言

第 1 章 绪论	1
1.1 引言	1
1.2 差分演化算法	2
1.3 差分演化算法理论研究概述	4
1.3.1 差分演化算法算子的搜索机理	5
1.3.2 差分演化算法的渐近收敛性	5
1.3.3 差分演化算法的计算复杂度	6
1.3.4 依概率收敛的差分演化算法设计	6
1.4 差分演化的算法改进研究概述	7
1.4.1 差分演化算法控制参数的设置研究	7
1.4.2 差分演化算法的演化算子改进研究	10
1.5 差分演化算法在工程应用领域的研究概述	12
参考文献	14

## 第一篇 差分演化算法的收敛性理论与收敛算法设计

第 2 章 差分演化算法的不确保依概率收敛性	31
2.1 相关差分演化算法收敛性结论的分析	31
2.2 基于马尔可夫链的差分演化算法收敛性分析	33
2.2.1 相关定义与定理	33
2.2.2 连续解空间的离散化	34
2.2.3 差分演化算法的马尔可夫链建模	35
2.2.4 基于马尔可夫链的差分演化算法收敛性证明	35
2.3 基于随机漂移模型的差分演化算法收敛性分析	37
2.3.1 差分演化算法的随机漂移建模	37
2.3.2 线性欺骗函数的构造	38
2.3.3 基于随机漂移模型的差分演化算法收敛性证明	40
2.4 一类让经典 DE 算法不能确保收敛的函数	42
2.4.1 函数的结构特征分析	43

2.4.2	数值实验分析 .....	45
2.4.3	函数难优化的缘由分析 .....	46
2.5	本章小结 .....	48
	参考文献 .....	48
<b>第3章</b>	<b>差分演化算法依概率收敛的充分条件</b> .....	<b>50</b>
3.1	充分条件的推理 .....	50
3.2	充分条件的注记 .....	52
3.3	几个差分演化算法的收敛性分析 .....	52
3.3.1	DE-RW 算法的收敛性证明 .....	52
3.3.2	CCoDE 算法的收敛性证明 .....	54
3.3.3	msDE 算法的收敛性证明 .....	57
3.4	本章小结 .....	59
	参考文献 .....	59
<b>第4章</b>	<b>差分演化算法的依概率收敛模式及辅助算子</b> .....	<b>60</b>
4.1	一个依概率收敛的差分演化算法模式 .....	60
4.2	辅助差分演化算法收敛的常用繁殖算子 .....	61
4.2.1	均匀变异算子 .....	61
4.2.2	高斯变异算子 .....	62
4.3	常用繁殖算子的辅助效率测试 .....	63
4.3.1	实验设计与实验参数设置 .....	64
4.3.2	实验结果与分析 .....	64
4.4	本章小结 .....	77
	参考文献 .....	77
<b>第5章</b>	<b>依概率收敛差分演化算法的辅助算子设计</b> .....	<b>78</b>
5.1	子空间聚类算子 .....	78
5.1.1	子空间聚类算子的概率分析 .....	79
5.1.2	子空间聚类算子的统计分析 .....	82
5.1.3	子空间聚类算子的程序实现 .....	84
5.2	一类基于子空间聚类算子的收敛差分演化算法 .....	85
5.3	算法的收敛性证明 .....	86
5.4	数值实验分析 .....	88
5.4.1	测试函数集 .....	88
5.4.2	实验设计与参数设置 .....	89
5.4.3	实验结果与分析 .....	91

5.5 本章小结 .....	106
参考文献 .....	106
第二篇 差分演化算法的应用	
<b>第 6 章 依概率收敛差分演化算法在螺旋压缩弹簧参数优化中的应用</b> .....	109
6.1 螺旋压缩弹簧参数优化问题的模型建立 .....	109
6.2 面向 CCS 优化设计的子空间聚类差分演化算法 .....	111
6.2.1 面向 CCS 优化设计的罚函数约束处理技术 .....	111
6.2.2 CCS 优化中混合变量的处理技术 .....	112
6.3 实验设计与结果分析 .....	112
6.4 本章小结 .....	114
参考文献 .....	115
<b>第 7 章 薄膜太阳能电池抗反射层微结构设计</b> .....	116
7.1 薄膜太阳能电池抗反射层研究现状 .....	116
7.2 梯度氮化硅/氮氧化硅结构的光捕获设计与优化 .....	119
7.2.1 优化模型 .....	119
7.2.2 单因素分析 .....	121
7.2.3 基于差分演化算法的设计与优化 .....	124
7.2.4 与纳米粒子结构的性能比较 .....	128
7.3 多层梯度抗反射层结构设计 .....	130
7.3.1 优化模型 .....	131
7.3.2 基于差分演化算法的优化与设计 .....	132
7.3.3 与纳米粒子结构的性能比较 .....	135
7.4 介质纳米粒子与多层抗反射层的光捕获性能比较 .....	138
7.4.1 介质纳米粒子的等效模型 .....	138
7.4.2 与多层抗反射层比较 .....	141
7.5 石墨烯透明导电薄膜光捕获结构设计 .....	143
7.5.1 优化模型 .....	144
7.5.2 基于差分演化算法的优化与设计 .....	146
7.6 本章小结 .....	150
参考文献 .....	151
<b>第 8 章 彩色图像颜色量化问题的优化与算法参数设置</b> .....	153
8.1 彩色图像颜色量化问题的优化方法发展现状 .....	153
8.2 彩色图像颜色量化的基本优化策略 .....	155
8.2.1 基于基本差分演化算法的彩色图像颜色量化优化策略 .....	155



---

8.2.2 彩色图像颜色量化基本优化策略的参数设置 .....	172
8.3 彩色图像颜色量化的混合优化策略 .....	173
8.3.1 彩色图像颜色量化优化模型的元素置换等效性 .....	173
8.3.2 混合策略 .....	173
8.3.3 彩色图像颜色量化的混合差分演化算法 .....	174
8.3.4 数值实验 .....	176
8.3.5 实验结果分析 .....	177
8.4 本章小结 .....	183
参考文献 .....	184
<b>第9章 彩色图像颜色量化问题的自适应优化方法</b> .....	<b>185</b>
9.1 差分演化算法参数的自适应化 .....	186
9.2 彩色图像颜色量化的混合自适应差分演化算法 .....	186
9.3 数值实验 .....	188
9.4 实验结果分析 .....	194
9.5 彩色图像颜色量化自适应混合优化策略在数字油画制作软件中的 应用 .....	194
9.6 本章小结 .....	195
参考文献 .....	196
<b>附录</b> .....	<b>198</b>

# 第 1 章 绪 论

## 1.1 引 言

人工智能(artificial intelligence, AI)是创造智能机器的科学与工程<sup>[1]</sup>, 计算智能(computational intelligence, CI)是实现人工智能的技术与方法<sup>[2]</sup>, 基于自然精神的算法是计算智能的主要组成部分, 包括神经网络(neural networks)、模糊系统(fuzzy system)和演化计算(evolutionary computation, EC)等。演化计算方法又分为群体智能算法(swarm intelligence algorithms)和演化算法(evolutionary algorithms), 典型的演化算法有遗传算法(genetic algorithm, GA)、遗传程序设计(genetic programming, GP)、演化策略(evolution strategy, ES)、演化规划(evolution programming, EP)和差分演化(differential evolution, DE)算法等。

差分演化算法<sup>[3]</sup>是 Storn 和 Price 1995 年为求解切比雪夫多项式拟合问题提出的一种基于演化机理的随机搜索算法。国内一般有差分演化算法、差分进化算法和差异进化算法等三种称谓。近 20 年的数值模拟与工程应用研究表明, 差分演化算法是最强有力的智能计算方法之一。

差分演化算法具有优化效果稳健、控制参数少、易于编程实现等优点, 自提出以来迅速引起了众多学者关注。2005 年 Springer 出版差分演化算法的第一本专著 *Differential Evolution: A Practical Approach to Global Optimization*<sup>[4]</sup>。2007~2009 年, 汤姆森科技信息集团的科学引文检索收录有关差分演化算法的文章不少于 3964 篇<sup>[5]</sup>。2011 年 2 月, *IEEE Transaction on Evolutionary Computation* 出版了一期差分演化算法的专辑。目前, 该算法已经被应用到了组合优化<sup>[6]</sup>、多目标优化<sup>[7]</sup>、函数优化<sup>[8]</sup>和图像颜色量化<sup>[9]</sup>等二十多个领域, 算法源码已经被包含在诸如 Built in optimizer in MATHEMATICA's function Nminimize (since version 4. 2), MATLAB's GA toolbox contains a variant of DE, Digital Filter Design, Diffraction grating design, Electricity market simulation, Auto2Fit, LMS Virtual Lab Optimization, Optimization of optical systems, Finite Element Design, LabView, Microwave Office 10. 0 by AWR Corp. 和 OPTIMUS 等 12 个应用软件包中<sup>[10]</sup>。

众多比较性数值模拟与应用研究表明, 差分演化算法是最强有力的随机优化算法之一。文献<sup>[11]</sup>应用 34 个测试函数, 比较了差分演化算法、粒子群算法、一个改进的粒子群算法<sup>[12, 13]</sup>(attractive and repulsive PSO, arPSO)和基本演化算法<sup>[14]</sup>

(simple evolutionary algorithm, SEA)的性能,比较结果显示差分演化算法的整体性能优于被比较的其他三个算法。文献[15]运用差分演化算法和遗传算法优化给排水系统,在6个案例上的数值实验结果表明差分演化算法能得到更好的优化效果。文献[7]比较了差分演化算法和模拟退火算法(simulated annealing, SA)在辐射传递问题上的优化效果,结果表明差分演化算法和模拟退火算法都能得到较好的优化结果,但是差分演化算法相对模拟退火算法更稳健。

在历届演化优化的国际竞赛(international contest on evolutionary optimization, ICEO)中,差分演化算法是唯一的在每届比赛中都表现优异的算法<sup>[5]</sup>。在1996年的第一届ICEO竞赛中,差分演化算法虽然只获得第三名<sup>[16,17]</sup>,但却是表现最好的演化算法(排名在前两位的是非智能算法)。接下来,在1997年的第二届ICEO竞赛中,差分演化算法获得第一名<sup>[17,18]</sup>,数值实验结果表明差分演化算法是所有参赛算法中表现最优异的。表1.1给出了自2006~2013年各届ICEO竞赛的结果,表中“-V”表示基于某算法的改进算法。由此可见,ICEO的测试数据集包括约束与无约束的单目标优化、约束与无约束的多目标优化、大规模单目标优化和动态优化等。在这些数据集上,差分演化算法或者差分演化算法的改进算法(differential evolution variants, DE-V)是唯一一个能够一直排名前五的演化计算方法。差分演化算法是当前最有影响力的演化计算方法之一<sup>[19-22]</sup>。

表 1.1 历届演化计算国际竞赛(ICEO2006~2013)排名前五的算法

年份	排名第一	排名第二	排名第三	排名第四	排名第五	数据集
2006	DE-V <sup>[23]</sup>	PSO-V <sup>[24]</sup>	Other <sup>[25]</sup>	DE-V <sup>[26]</sup>	DE-V <sup>[27]</sup>	约束单目标
2007	GA-V <sup>[28]</sup>	GA-V <sup>[29]</sup>	DE-V <sup>[30]</sup>	DE-V <sup>[31]</sup>	DE-V <sup>[32]</sup>	多目标优化
2008	DE-V <sup>[33]</sup>	PSO-V <sup>[34]</sup>	Other <sup>[35]</sup>	MTS-V <sup>[36]</sup>	EDA-V <sup>[37]</sup>	大规模单目标
2009	DAS <sup>[38]</sup>	DE-V <sup>[39]</sup>	AIA-V <sup>[40]</sup>	EP-V <sup>[41]</sup>	PSO-V <sup>[42]</sup>	动态优化
2009	MOEA/D <sup>[43]</sup>	MTS-V <sup>[44]</sup>	Other <sup>[45]</sup>	MOEA/D <sup>[46]</sup>	DE-V <sup>[47]</sup>	多目标优化
2010	MA-V <sup>[48]</sup>	DE-V <sup>[49]</sup>	DE-V <sup>[50]</sup>	PSO-V <sup>[51]</sup>	DAS <sup>[52]</sup>	大规模单目标
2010	DE-V <sup>[53]</sup>	Other <sup>[54]</sup>	DE-V <sup>[55]</sup>	DE-V <sup>[56]</sup>	ABC-V <sup>[57]</sup>	约束单目标
2011	GA-V <sup>[58]</sup>	DE-V <sup>[59]</sup>	DE-V <sup>[60]</sup>	GA-V <sup>[61]</sup>	DE-V <sup>[62]</sup>	单目标优化
2013	CMA-ES-V <sup>[63]</sup>	Other <sup>[64]</sup>	MA-V <sup>[65]</sup>	DE-V <sup>[66]</sup>	CMA-ES-V <sup>[63]</sup>	单目标优化

注:“-V”表示基于某算法的改进算法,“Other”表示非演化计算方法。

## 1.2 差分演化算法

差分演化算法常被用来求解连续优化问题,不失一般性,最小化边界约束连续

优化问题的形式化表达式为

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s. t.} \quad & \mathbf{x} = (x_1, x_2, \dots, x_D) \in \psi \subset \mathbb{R}^D \end{aligned} \quad (1.1)$$

其中,  $\mathbb{R}^D$  是  $D$  维搜索空间;  $\psi = \prod_{j=1}^D [L_j, U_j]$  是解空间,  $U_j$  和  $L_j$  分别是  $x_j$  的上下界。

记该问题的最优解是  $\vec{x}^*$ , 最优解集是  $B^*$ 。除非特别说明, 接下来的讨论将基于上述最小化问题展开研究。

差分演化算法的基本思想是运用当前种群个体的差来重组得到中间个体, 即实验个体(trial individual), 然后运用父子个体适应值竞争来获得新一代个体。差分演化算法是最典型的演化算法之一, 与其他演化算法类似, 算法通过变异、交叉和选择等三步的循环来达到寻优目的。算法流程如下。

Step1, (初始化) 初始化种群规模(population size,  $N$ )、个体维数(individual dimension,  $D$ )、变异因子(mutation factor,  $F$ )、交叉概率(crossover probability,  $CR$ )、初始化种群  $X^t = (\vec{x}_1^t, \vec{x}_2^t, \dots, \vec{x}_N^t)$ 。这里迭代次数  $t=0$ , 表示第 0 代的初始化种群;  $\vec{x}_i^t, i=1, 2, \dots, N$  表示第  $t$  代的第  $i$  个个体, 每个个体都是  $D$  维向量。

Step2, (变异) 算法通过变异操作为每个个体产生一个对应的捐助向量  $\mathbf{v}$ (donor vector), 最常用的 5 个变异操作如下。

第一, DE/rand/1:  $\vec{v}_i = \vec{x}_{r_1}^t + F \cdot (\vec{x}_{r_2}^t - \vec{x}_{r_3}^t)$ 。

第二, DE/best/1:  $\vec{v}_i = \vec{x}_{\text{best}}^t + F \cdot (\vec{x}_{r_1}^t - \vec{x}_{r_2}^t)$ 。

第三, DE/current-to-best/1:  $\vec{v}_i = \vec{x}_i^t + F \cdot (\vec{x}_{\text{best}}^t - \vec{x}_i^t) + F \cdot (\vec{x}_{r_1}^t - \vec{x}_{r_2}^t)$ 。

第四, DE/best/2:  $\vec{v}_i = \vec{x}_{\text{best}}^t + F \cdot (\vec{x}_{r_1}^t - \vec{x}_{r_2}^t) + F \cdot (\vec{x}_{r_3}^t - \vec{x}_{r_4}^t)$ 。

第五, DE/rand/2:  $\vec{v}_i = \vec{x}_{r_1}^t + F \cdot (\vec{x}_{r_2}^t - \vec{x}_{r_3}^t) + F \cdot (\vec{x}_{r_4}^t - \vec{x}_{r_5}^t)$ 。

其中,  $\vec{x}_i^t, i=1, 2, \dots, N$  是第  $t$  代种群中的第  $i$  个个体, 称为目标向量(target vector);  $r_1, r_2, \dots, r_5$  是  $1 \sim N$  中不等于  $i$  的相异的随机整数;  $\vec{x}_{\text{best}}^t$  是第  $t$  代种群中的最优个体; 变异因子  $F$  是经验参数, 一般在区间  $(0, 1]$  上取值。

Step3, (交叉) 算法通过目标向量和捐助向量之间的交叉操作为每个目标个体产生一个实验向量  $\mathbf{u}$ , 差分演化算法经典的交叉操作有指数交叉(exponential crossover)和二项式交叉(binomial crossover), 最常用的二项式交叉可以表示为

$$u_{i,j}^t = \begin{cases} v_{i,j}^t, & \text{rand}(0, 1) \leq CR \text{ 或 } j = j_{\text{rand}} \\ x_{i,j}^t, & \text{其他} \end{cases}$$

其中,  $j=1, 2, \dots, D$ ;  $CR$  是算法的第二个经验参数, 一般在  $(0, 1)$  上取值;  $\text{rand}(0, 1)$  是在  $[0, 1]$  上服从均匀分布的随机数;  $j_{\text{rand}}$  是 1 和  $D$  之间的一个随机整数, 保证至少在某一维上实施交叉操作。

Step4, (选择) 差分演化算法通过在目标向量和实验向量之间实施贪婪的选

择操作来产生下一代种群,选择操作(针对最小化问题)可表示为

$$\vec{x}_i^{t+1} = \begin{cases} \vec{u}_i^t, & f(\vec{u}_i^t) < f(\vec{x}_i^t) \\ \vec{x}_i^t, & \text{其他} \end{cases}$$

这里  $f(\cdot)$  是最小化问题的目标函数值。

Step5, (终止) 循环执行 Step2~Step4, 直至达到设定的循环终止条件, 输出最优结果。常用的终止条件有两个: 一是设定最大迭代次数, 二是设置精度水平。

算法注解。

① 差分演化算法最具特色的是它自适应的变异操作。在演化的初期阶段, 因为种群中个体的差异较大, 因此用来作为变异扰动的差向量也较大, 个体的扰动就较大, 有利于算法的全局搜索; 随着演化的进行, 当算法趋于收敛的时候, 种群中个体的差异随之较小, 因此用来变异扰动的差向量也随之自适应地变小, 较小的扰动有利于算法的局部搜索。正是这种简单又独具特色的变异操作有效地平衡了差分演化算法的全局搜索能力和局部搜索能力。需要注意的是, 差分演化算法的变异操作对于搜索空间是不封闭的, 即变异后得到的捐助向量可能会溢出搜索空间。周期模式(Periodic Mo 差分演化)是常用的处理方法之一, 即

$$v_{i,j} = \begin{cases} U_j - (L_j - v_{i,j}) \% |U_j - L_j|, & v_{i,j} < L_j \\ L_j - (v_{i,j} - U_j) \% |U_j - L_j|, & v_{i,j} > U_j \end{cases}$$

其中, “%” 是“求余”运算。

② 相对其他演化算法, 差分演化具有算法简单、易于实现的优点。

③ 差分演化算法仅有三个经验控制参数, 即种群规模  $N$ 、变异因子  $F$  和交叉概率  $CR$ 。算法的表现对参数的设置是敏感的, 针对其中两个关键控制参数  $F$  和  $CR$ , 已经研究出了较多简单有效的自适应(或者适应)控制方法。

④ 相对其他有竞争力的同类算法, 较小的空间复杂度是差分演化算法的另一个优势, 如协方差矩阵自适应进化策略(covariance matrix adaptation-evolution strategy, CMA-ES)。在不高于 100 维的优化问题上, CMA-ES 表现出很强的竞争力, 但是在更高维的大规模优化问题上, 空间复杂度的优势让差分演化算法表现出更强的可拓展性。

### 1.3 差分演化算法理论研究概述

随着差分演化算法在数值模拟和众多工程应用领域的成功, 越来越多的学者开始关注差分演化算法的理论研究。演化计算方法的理论研究是一个研究的难点, 差分演化算法的理论研究更是进展缓慢。接下来, 从算子搜索机理、算法渐近

收敛性、算法复杂度和收敛算法设计等四个方面介绍当前的理论研究成果,然后分析在该领域可能存在的研究空间。

### 1.3.1 差分演化算法算子的搜索机理

差分演化算法是典型的演化算法之一,与传统的遗传算法类似,通过循环执行变异、交叉和选择操作来实现种群的逐步优化,算子搜索能力的强弱直接影响算法的整体优化性能。算子的搜索能力可分为全局搜索(global exploration)能力和局部搜索(local exploitation)能力,全局搜索能力可由种群的多样性反映,种群的多样性又可由种群的方差反映。沿着这一研究思路,Zaharie 应用基于统计量的方法从理论上研究了差分演化算法的算子搜索机理。

2001年,Zaharie<sup>[67]</sup>研究了差分演化算法的变异算子、交叉算子和种群方差的数学期望之间的关系,进而推演得到种群方差的数学期望与变异因子、交叉概率之间的函数关系,并证明差分演化算法(不含选择操作)的种群方差比演化策略的大,即种群多样性比演化策略的好,这也从某种程度上解释了差分演化算法在数值模拟上表现优越的内在原因。在此基础上,2002年 Zaharie<sup>[68]</sup>进一步从理论上探讨了参数的取值与算法早熟之间的关系;进而,Zaharie<sup>[69]</sup>给出一种基于上述理论研究的自适应参数策略,该策略通过控制种群方差来控制种群的多样性。2008年,Zaharie<sup>[70]</sup>提出一个不使用经典差运算算子的变异操作,并从理论上证明该算子对种群方差与经典变异算子和交叉算子有等同的影响力。同时,也对二进制差分演化的种群分布做了初步的分析。

差分演化算法的交叉算子最常用的有二项式交叉(binomial crossover)和指数型交叉(exponential crossover)。Zaharie<sup>[71,72]</sup>从理论上分析了差分演化算法交叉算子和交叉概率对算法行为的影响。理论和实验的研究结果表明,相对于常用的二项式交叉而言,指数型交叉对问题的规模更敏感,且两个交叉算子对差分演化算法行为的影响主要是通过控制变异组件来实现的。

与 Zaharie 基于统计量研究思路不同,Dasgupta 等<sup>[73]</sup>建立了差分演化算法的动力系统模型,论证了差分演化算法搜索过程有梯度下降特征,并应用模型研究了差分演化的收敛速度与变异因子、交叉概率的相关性,进而应用李雅普诺夫稳定性定理(Lyapunov's stability theorems)分析了差分演化算法的稳定性。

胡中波和熊盛武等<sup>[74]</sup>从代数系统是否同构的角度在理论上证明了 DE 算法不适合与 Koziel<sup>[75]</sup>的约束处理映射相结合,而遗传算法却适合与该约束处理映射相结合。该结论从理论上论证了差分演化算法和遗传算法搜索机理的差异。

### 1.3.2 差分演化算法的渐近收敛性

2005年,Xue 等<sup>[76]</sup>简化基本差分演化算法,只考虑算法的变异和交叉操作,假

设初始种群符合正态分布,建立了连续的多目标差分演化算法的数学模型,并证明在初始种群中包含问题的帕累托(Pareto)最优解的前提下,算法能够收敛到帕累托最优解。同年,Xue等<sup>[77]</sup>把上述工作推广到离散的多目标差分演化模型中,得到了类似的结论。

2010年,贺毅朝等<sup>[78]</sup>认为差分演化算法的演化算子可以看成是一个随机映射,然后运用压缩映像定理证明基本差分演化算法的渐近收敛性。同年,孙成富<sup>[79]</sup>运用马尔可夫链模型分析并证明基本差分演化算法的种群序列是有限齐次马尔可夫链,进而采用马尔可夫链的吸收态说明了基本差分演化算法无法保证全局依概率收敛。

2012年,Gohosh等<sup>[80]</sup>运用李雅普诺夫稳定性定理证明基本差分演化算法在一类特殊函数上的渐近收敛性。该类函数具有两个特征,即函数二阶连续可导,函数在可行域内只有唯一的全局最优值点。

### 1.3.3 差分演化算法的计算复杂度

因演化搜索的随机性带来了求解时间的不确定性,使得随机算法的算法复杂度的分析显得尤其重要。差分演化算法是随机搜索算法,2005年Zielinski等<sup>[81]</sup>从理论上和数值实验上研究了差分演化算法对不同终止准则的计算复杂度。Zielinski等指出,差分演化算法的两个繁殖操作——变异、交叉——在种群(记种群规模是 $NP$ )每一个个体(记个体维数是 $D$ )的每一维上执行,且变异操作的执行次数与循环的总数是成比例的。因此,如果终止准则是达到最大循环次数 $G_{max}$ ,则差分演化算法的计算复杂度是 $O(NP \cdot D \cdot G_{max})$ ,进而推出结论,从整体上看,最大距离终止准则对差分演化算法有更小的计算复杂度。这里的最大距离终止准则是指,当种群中每个向量到种群中最好的向量之间的最大距离小于给定值时终止循环。

### 1.3.4 依概率收敛的差分演化算法设计

考虑到收敛的算法具有较高的稳健性,随着差分演化算法理论研究成果的逐步显现,收敛差分演化算法的设计正在逐步成为差分演化算法设计的一个研究分支。

2006年, Ter Braak<sup>[82]</sup>设计了一个差分演化马尔可夫链算法,并证明该算法的种群序列收敛到一个平稳分布。2009年,Zhao等<sup>[83]</sup>利用停滞个体的重新抽样(抽样函数)机制改进了经典的差分演化算法,并证明改进的算法是依概率收敛的。2012年,Zhan等<sup>[84]</sup>基于随机游动机制(random walk)设计了一个差分演化随机游动算法,文章虽然没有给出算法的收敛性理论证明,但该算法的依概率收敛性不难证明。2013年,Li<sup>[85]</sup>设计了一个基于高斯变异(Gaussian mutation)和多样性逆转

抽样(diversity-triggered reverse)的差分演化算法,并运用马尔可夫链理论证明算法的依概率收敛性。

## 1.4 差分演化的算法改进研究概述

### 1.4.1 差分演化算法控制参数的设置研究

基本差分演化算法有三个经验控制参数,即种群规模  $NP$ 、变异因子  $F$  和交叉概率  $CR$ ,且算法的执行效果对参数的设置是敏感的,参数的设置问题是差分演化算法基础研究的一个重要内容。Storn 和 Price<sup>[3]</sup> 指出合理的  $NP$  取值应介于  $5D \sim 10D$  ( $D$  是实际问题的维数),并且  $F$  取值为  $0.5$  比较好。近期, Das 和 Suganthan 指出  $F$  的有效取值范围通常介于  $0.4 \sim 1$ 。

参数  $CR$  用以控制个体(向量)中变异的分量数目。以二维向量为例,对一个由  $10$  个二维向量构成的初始种群使用差分演化算法(不使用选择操作)进行  $200$  次迭代,图 1.1 显示当  $CR$  分别取值为  $0$ 、 $0.5$  和  $1$  时所有  $2000$  个实验向量的分布。由此可见,当  $CR$  取值较小时,每一代个体中所变异的分量较少,因此导致大多数个体都集中分布在与坐标轴垂直的方向上;当  $CR$  取值较大(接近  $1$ )时,每一代个体中变异的分量较多,这样种群的分布更均匀、种群的多样性更好。

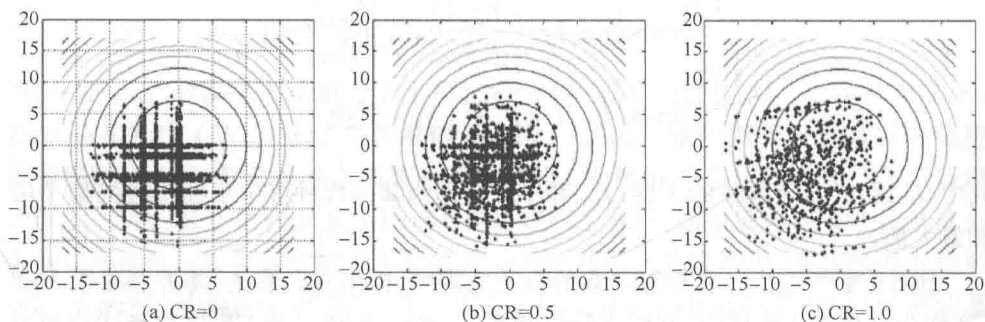


图 1.1 2000 个实验向量的分布

值得注意的是,对于传统的差分演化算法(DE/rand/1/bin),只有当  $CR=1$  时,算法才具备旋转不变性。此时,交叉操作是针对整个向量的操作,即该操作使得实验向量中的分量全部变异。只要  $CR=1$  且  $F$  是常数,或者按照某个分布对每个实验向量的分量的抽取不超过一次,那么变异实验向量在拓扑图中的位置就不会随着坐标的旋转而发生变化。如果  $CR$  取值较小(如  $0.1$ ),那么就需要通过依概率抽样对各个方向(或各个方向的一部分)进行改变。这个策略对可分离和可分解

的函数(如  $f(\mathbf{x}) = \sum_{i=1}^D f_i(x_i)$ ) 来说是一种有效策略。



实验结果表明,全局最优搜索能力和收敛速度对控制参数 NP、F 和 CR 的选择是敏感的,而较合理的参数设置是种群规模 NP 介于  $3D \sim 8D$ ,变异因子  $F = 0.6$ ,交叉概率 CR 介于  $0.3 \sim 0.9$ 。Ronkkonen 等<sup>[86]</sup>指出在一般情况下  $F = 0.9$  且  $0.4 < CR < 0.95$  是最好的选择,当函数可分离时 CR 应该介于  $(0, 0.2)$ ,而当函数的变量不是相互独立时 CR 应该介于  $(0.9, 1)$ 。

研究者考虑到使用某些自适应策略来调整控制参数。例如,在差分演化算法中,对控制参数通过自适应方法进行调整<sup>[87-91]</sup>。Liu 和 Lampinen<sup>[89]</sup>介绍了一种模糊自适应差分演化算法,该算法通过包含连续几代的相关函数值和个体的模糊逻辑控制器来自适应地调整变异因子 F 和交叉概率 CR。Qin 等<sup>[90]</sup>提出 SaDE 算法,通过学习前几次迭代中产生优选解的经验逐渐地对实验向量的产生策略和控制参数 F 和 CR 进行调整。在 SaDE 算法中虽然增加了其他的控制参数,但是算法的执行效果对新增加的控制参数不敏感,因此 SaDE 算法比经典的差分演化算法更稳健。

Ali 和 Törn 针对 F 提出一个基于个体适应值的自适应差分演化算法<sup>[87]</sup>。该算法使用含有两个演化种群的系统,交叉概率 CR 取值为 0.5,而 F 的取值在每一代都会自适应地进行更新,其更新规则为

$$F = \begin{cases} \max \left\{ l_{\min}, 1 - \left| \frac{f_{\max}}{f_{\min}} \right| \right\}, & \left| \frac{f_{\max}}{f_{\min}} \right| < 1 \\ \max \left\{ l_{\min}, 1 - \left| \frac{f_{\min}}{f_{\max}} \right| \right\}, & \text{其他} \end{cases}$$

其中,  $l_{\min}$  是目标函数  $f$  的下界;  $f_{\min}$  和  $f_{\max}$  分别为某一代种群中目标函数的最小值和最大值。

Brest 等<sup>[88]</sup>给出一种控制参数的自适应规则,提出 jDE 算法,将控制参数 F 和 CR 作为分量加入个体中,并引入新的参数  $\tau_1$  和  $\tau_2$  来对 F 和 CR 进行自适应地调整。每一代中控制参数取值较好的个体更具有竞争力,因此进入下一代种群,其对应的控制参数 F 和 CR 根据如下规则进行自适应地调整,即

$$F_{i,G+1} = \begin{cases} F_l + \text{rand}_1 * F_u, & \text{rand}(0,1) < \tau_1 \\ F_{i,G}, & \text{其他} \end{cases}$$

$$CR_{i,G+1} = \begin{cases} \text{rand}_3, & \text{rand}(0,1) < \tau_2 \\ CR_{i,G}, & \text{其他} \end{cases}$$

其中,  $F_l$  和  $F_u$  分别是 F 的上下限,且  $F_l \in [0, 1]$ ,  $F_u \in [0, 1]$ 。