

密级 _____

中国科学院研究生院

博士学位论文

无导数优化方法的研究

作者姓名 _____ 张在坤

指导教师 _____ 袁亚湘 (院士、研究员、博士)

中国科学院数学与系统科学研究院

学位类别 _____ 理学博士

学科专业 _____ 计算数学

培养单位 _____ 中国科学院数学与系统科学研究院

2012 年 4 月

Typeset by L^AT_EX 2 ϵ on April 7, 2012 (Revised on September 21, 2012)

With package `CASthesis` v0.1h by Lingyun Wu

On Derivative-Free Optimization Methods

By

Zaikun Zhang

Supervisor:

Prof. Ya-Xiang Yuan

*A Dissertation Submitted to
Graduate University of Chinese Academy of Sciences
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Science*

Institute of Computational Mathematics and Scientific/Engineering Computing
Academy of Mathematics and Systems Science
Chinese Academy of Sciences

April, 2012

献给我的父亲母亲

The subject of optimization is a fascinating blend of heuristics and rigour, of theory and experiment.

— R. Fletcher, *Practical Methods of Optimization*

In fact, we consider optimization without derivatives one of the most important, open, and challenging areas in computational science and engineering, and one with enormous practical potential.

— A. R. Conn, K. Scheinberg, and L. N. Vicente,
Introduction to Derivative-Free Optimization

摘 要

大多数优化方法都依赖问题的导数信息. 但是, 在实际应用中, 大量问题的导数信息都是不可用的. 这就要求我们研究不使用导数信息的方法, 这就是本文研究的无导数优化算法.

算法的评价是算法研究中的重要问题. 我们研究了如何客观可信地评价和比较不同的无导数优化算法. 我们用一个例子清楚地说明传统的评价方法对于无导数算法是不可靠的. 通过引入统计的方法, 我们建立了评价无导数方法的一套新体系. 与传统的评价方法相比, 新方法不但能够反映算法对计算机舍入误差的稳定性, 而且能更可靠的度量算法的计算开销.

最小 Frobenius 范数插值和对称 Broyden 修正是无导数信赖域方法中最有效的两种建立模型的方式. 我们第一次指出了这两种方式在一些情况下的等价性. 与这两种模型紧密相关的一个问题是 NEWUOA 算法的重开始技术. 通过修改 NEWUOA 源代码中的重开始条件, 我们给出了一个改进版本的 NEWUOA 代码. 新版代码仅仅删除了原始代码中的四个字母, 就显著降低了代码的计算开销, 并且明显提高了代码对计算机舍入误差的稳定性. 我们系统地比较了最小 Frobenius 范数模型和对称 Broyden 修正在 NEWUOA 算法框架下的表现, 并且指出, 当求解精度不太高时, 最小 Frobenius 范数模型比对称 Broyden 修正建立的模型表现更好. 这一事实对于实际应用领域很有意义, 因为实际的无导数优化问题对解的精度要求往往不高.

为了研究无导数优化中有广泛应用的最小范数插值, 我们率先将 PDE 理论中的 Sobolev 范数和半范数引入无导数算法的研究中. 我们用二次函数的系数给出了二次函数在 ℓ_p 球上的 H^0 范数和 H^1 半范数的显式表达式. 我们证明, 最小范数插值实际上是在一个 ℓ_2 球上极小化插值函数的 H^1 半范数. 这一观察为理解最小范数插值提供了有力的工具. 通过这一观察, 我们首次指出了最小范数插值中两个参量的几何意义. 我们将这些理论用于研究扩展的对称 Broyden 修正, 得到了简单并且有效的参数选取方式.

到目前为止, 无导数方法可求解的问题规模还十分有限. 为了求解大规模问题, 我们提出了两种无导数的子空间算法. 在第一种子空间方法中, 利用 Hooke-Jeeves 模式搜索的思想, 针对无导数信赖域方法, 我们提出在子空间上

求解信赖域子问题的策略. 这种子空间策略改善了 NEWUOA 算法的数值表现. 第二种子空间方法, 即 NEWUOAs (A NEW Unconstrained Optimization Algorithm with subspace technique based on NEWUOA) 算法, 是本文最大的亮点. 其基本想法是, 把一个大规模无导数优化问题转化为一系列低维子问题. 我们首先研究了一个一般性的子空间算法框架, 建立了其全局收敛性和 R-线性收敛速度. 然后, 使用 NEWUOA 算法作为子问题的求解器, 我们不依赖导数地实现了该框架, 得到了 NEWUOAs 算法. 我们证明了 NEWUOAs 算法在理论上的全局收敛性、R-线性收敛速度和计算上的有限终止性. 我们还提出了一项预条件技术, 显著改善了 NEWUOAs 算法对坏条件问题的表现. 据本文作者所知, 这是无导数算法中第一次引入预条件技术. 实验证明, NEWUOAs 算法不论在函数值计算次数、CPU 时间还是对计算机舍入误差的稳定性上都明显优于 NEWUOA 算法, 后者是目前最优秀的无导数算法之一. 我们还发现, NEWUOAs 算法很适合求解初始点质量较差的问题, 这对实际应用领域很有意义, 因为很多实际问题很难给出一个好的初始点. 不仅如此, 对于很多维数高达 2000 的测试问题, NEWUOAs 算法可以在几分钟内求到精度很高的解, 且使用的函数值计算次数不超过 50000 (相当于不到 25 个单纯形梯度). 这是一个突破, 因为目前大部分无导数算法 (包括 NEWUOA 算法) 至多可以求解几百维的问题; 对于它们, 2000 维的问题几乎是不可求解的.

关键词：无导数优化, 信赖域方法, 二次插值, 对称 Broyden 修正, Sobolev 半范数, 子空间方法, 大规模问题

Abstract

Most of the optimization algorithms depend on derivative information of the problem. However, there are numerous real-world problems where derivatives are unavailable. This motivates us to study derivative-free optimization methods.

The assessment and comparison of algorithms play important roles in the research of algorithms. We study how to assess derivative-free algorithms in a reliable way. Through an example, we show that it is not reliable to merely count the number of function evaluations. By introducing statistical method, we establish a new system for the assessment of derivative-free algorithms. The new system reflects the stability of algorithms with respect to computer rounding errors, and provides more convincing comparison of different algorithms.

Least Frobenius norm quadratic interpolation and symmetric Broyden update are the most successful methods of constructing models in derivative-free trust-region algorithms. We prove the equivalence between these two strategies in some cases. The restart technique in `NEWUOA` is closely related to these two strategies. We modify the restart criterion in the source code of `NEWUOA` by simply deleting four letters and obtain a new version of the source code. The modification brings considerable improvement. Under the framework of `NEWUOA`, we compare least Frobenius norm model and the model established by symmetric Broyden update. We point out that least Frobenius norm model works better if a low-precision minimizer is desirable, which is meaningful for applications, because many problems in practice do not require high-precision solutions.

To study the widely-used least norm quadratic interpolation in derivative-free methods, we introduce the Sobolev norms and seminorms, which are classical in PDE theory but rarely noticed in optimization. For the H^0 norm and H^1 seminorm of a quadratic function over an ℓ_p ball, we obtain explicit formulae in terms of the coefficients of the function. We prove that least norm quadratic interpolation seeks an interpolant with minimal H^1 seminorm over an ℓ_2 ball. This observation provides a new perspective to interpret the interpolation. Consequently,

we present the geometrical meaning of the parameters in the interpolation. We apply our theory to study the extended symmetric Broyden update, and propose a very simple but effective way of choosing the parameters in the update.

Until now, derivative-free methods can only solve problems with modest dimension. We study subspace techniques to attack large scale problems. We presents two derivative-free subspace methods. In the first method, we apply the idea of Hooke-Jeeves pattern-search to derivative-free trust-region method, and propose to solve the trust-region subproblem in a low-dimensional subspace of \mathbb{R}^n . This subspace strategy improves the performance of **NEWUOA**. The second subspace method, which is named as **NEWUOAs**, is the highlight of the thesis. Its basic idea is to divide a large scale problem into a sequence of low-dimensional subproblems. We first study a general subspace algorithm based on this idea, and present its convergence theory. Then using **NEWUOA** as the subproblem solver, we implement the algorithm without using derivatives and obtain **NEWUOAs**. For **NEWUOAs**, we establish its global convergence and R-linear convergence rate in theory, and prove its finite termination in computation. We propose a preconditioning technique, which improves the performance of **NEWUOAs** on ill-conditioned problems. As far as we know, this is the first derivative-free algorithm with preconditioning procedure. In our numerical experiments, **NEWUOAs** works evidently better than **NEWUOA**, in the number of function evaluations, CPU time, and stability. We also find that **NEWUOAs** is good at solving problems with bad starting points, which is favourable in real-world applications. Besides, **NEWUOAs** is capable of solving many 2000-dimensional test problems to high precision within several minutes, using not more than 50000 function evaluations (equivalent to less than 25 simplex gradients). It is a breakthrough, because most state-of-the-art derivative-free algorithms can only solve problems with not more than a few hundreds of variables, and 2000-dimensional problems are nearly unsolvable for them.

Keywords: derivative-free optimization, trust-region method, quadratic interpolation, symmetric Broyden update, Sobolev seminorm, subspace method, large scale problem

目 录

| | |
|--|-----------|
| 摘要 | i |
| Abstract | iii |
| 目录 | v |
| 表格 | ix |
| 插图 | xi |
| 第一章 引言 | 1 |
| § 1.1 无导数优化的应用 | 2 |
| § 1.1.1 参数调节 | 2 |
| § 1.1.2 工程设计 | 3 |
| § 1.1.3 分子几何 | 3 |
| § 1.1.4 其他应用 | 4 |
| § 1.2 算法综述 | 4 |
| § 1.2.1 直接搜索方法 | 4 |
| § 1.2.2 基于线搜索的方法 | 6 |
| § 1.2.3 基于信赖域的方法 | 8 |
| § 1.2.4 其他方法 | 13 |
| § 1.2.5 关于术语的注记 | 13 |
| § 1.3 本文主要内容 | 16 |
| 第二章 无导数优化方法的一个评价体系 | 19 |
| § 2.1 引言 | 19 |
| § 2.2 Performance Profile 与 Data Profile | 19 |

| | | |
|------------|--|-----------|
| § 2.3 | 计算机舍入误差的影响: 一个数值例子 | 23 |
| § 2.4 | 一种新的评价体系 | 27 |
| § 2.5 | 总结 | 29 |
| § 2.6 | 附录 | 29 |
| 第三章 | 无导数优化中的最小 Frobenius 范数插值与对称 Broyden 修正 | 33 |
| § 3.1 | 引言 | 33 |
| § 3.2 | 最小 Frobenius 范数插值与对称 Broyden 修正的关系 | 35 |
| § 3.3 | NEWUOA 的重开始技术以及 NEWUOA 源代码的一个改进 | 36 |
| § 3.3.1 | NEWUOA 的重开始技术 | 36 |
| § 3.3.2 | NEWUOA 源代码的一个改进 | 37 |
| § 3.4 | 最小 Frobenius 范数插值与对称 Broyden 修正的比较 | 45 |
| § 3.5 | 总结 | 58 |
| § 3.6 | 关于无导数对称 Broyden 修正的历史注记 | 59 |
| 第四章 | 二次函数的 Sobolev (半) 范数及其在无导数优化中的应用 | 63 |
| § 4.1 | 引言 | 63 |
| § 4.2 | Sobolev (半) 范数简介 | 66 |
| § 4.3 | 二次函数在 ℓ_2 球上的 H^1 半范数 | 67 |
| § 4.4 | H^1 半范数观点下的最小范数插值 | 68 |
| § 4.5 | 扩展的对称 Broyden 修正 | 71 |
| § 4.5.1 | H^1 半范数与扩展的对称 Broyden 修正 | 71 |
| § 4.5.2 | 参量 σ 与 x_0 的选择 | 73 |
| § 4.5.3 | 数值实验 | 74 |
| § 4.6 | 二次函数在 ℓ_p 球上的 H^0 范数与 H^1 半范数 | 83 |
| § 4.6.1 | 关于 Gamma 函数的两个命题 | 84 |
| § 4.6.2 | H^0 范数与 H^1 半范数的表达式 | 86 |
| § 4.6.3 | 定理 4.10 的证明 | 87 |
| § 4.6.4 | 讨论 | 91 |

| | |
|---|------------|
| § 4.6.5 ℓ_p 椭球上的 H^0 范数与 H^1 半范数 | 92 |
| § 4.6.6 带权重的 H^0 范数与 H^1 半范数 | 92 |
| § 4.7 总结 | 93 |
| 第五章 子空间技巧在无导数优化中的应用 | 95 |
| § 5.1 引言 | 95 |
| § 5.2 无导数子空间方法 I | 96 |
| § 5.2.1 算法的提出 | 96 |
| § 5.2.2 数值实验 | 100 |
| § 5.3 无导数子空间方法 II: NEWUOAs | 108 |
| § 5.3.1 一个子空间算法框架 | 109 |
| § 5.3.2 收敛性分析 | 110 |
| § 5.3.3 NEWUOAs: 一个基于 NEWUOA 的无导数子空间算法 | 117 |
| § 5.3.4 NEWUOAs 算法的收敛性 | 120 |
| § 5.3.5 NEWUOAs 算法的实现细节 | 124 |
| § 5.3.6 NEWUOAs 算法的预条件技术 | 125 |
| § 5.3.7 数值实验 | 129 |
| § 5.4 总结 | 148 |
| 第六章 总结与展望 | 151 |
| 参考文献 | 155 |
| 索引 | 177 |
| 完成文章目录 | 179 |
| 简历 | 181 |
| 致谢 | 183 |

表 格

| | | |
|-----|--|-----|
| 2.1 | 20 维 BDQRTIC 问题对 P_1 、 P_2 的实验结果 | 25 |
| 2.2 | 20 维 BDQRTIC 问题 10 次随机实验结果 | 26 |
| 3.1 | 测试问题名称 | 38 |
| 5.1 | NEWUOA 算法与 NEWUOAs 算法对 DQRTIC 问题的表现 | 126 |
| 5.2 | NEWUOA 算法与 NEWUOAs 算法对 POWER 问题的表现 | 127 |
| 5.3 | NEWUOAs 算法求解 2000 维问题的例子 | 147 |

插 图

| | | |
|------|--|----|
| 2.1 | 20 维 BDQRTIC 问题对 P_1 、 P_2 的实验结果 | 25 |
| 3.1 | NEWUOA 与 NEWUOAm 的数值表现 (精度 $\tau = 10^{-2}$) | 39 |
| 3.2 | NEWUOA 与 NEWUOAm 的数值表现 (精度 $\tau = 10^{-4}$) | 40 |
| 3.3 | NEWUOA 与 NEWUOAm 的数值表现 (精度 $\tau = 10^{-6}$) | 41 |
| 3.4 | NEWUOA 与 NEWUOAm 的数值表现 (精度 $\tau = 10^{-8}$) | 42 |
| 3.5 | NEWUOA 与 NEWUOAm 的数值表现 (精度 $\tau = 10^{-10}$) | 43 |
| 3.6 | NEWUOA 与 NEWUOAm 的数值表现 (迭代自然终止) | 44 |
| 3.7 | NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-1}$) | 46 |
| 3.8 | NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-2}$) | 47 |
| 3.9 | NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-3}$) | 48 |
| 3.10 | NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-4}$) | 49 |
| 3.11 | NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-5}$) | 50 |
| 3.12 | NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-6}$) | 51 |
| 3.13 | NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-7}$) | 52 |
| 3.14 | NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-8}$) | 53 |
| 3.15 | NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-9}$) | 54 |
| 3.16 | NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-10}$) | 55 |
| 3.17 | NEWUOA 与 LFN 的数值表现 (迭代自然终止) | 56 |
| 3.18 | NEWUOA 与 LFN 求解 100 维 ARWHEAD 问题与 CHROSEN 问题的函数值下降曲线 | 57 |
| 4.1 | SYMB、ESYMBP 与 ESYMBS 的数值表现 (精度 $\tau = 10^{-2}$) | 77 |
| 4.2 | SYMB、ESYMBP 与 ESYMBS 的数值表现 (精度 $\tau = 10^{-4}$) | 78 |
| 4.3 | SYMB、ESYMBP 与 ESYMBS 的数值表现 (精度 $\tau = 10^{-6}$) | 79 |
| 4.4 | SYMB、ESYMBP 与 ESYMBS 的数值表现 (精度 $\tau = 10^{-8}$) | 80 |

| | | |
|------|--|-----|
| 4.5 | SYMB、ESYMBP 与 ESYMBS 的数值表现 (精度 $\tau = 10^{-10}$) | 81 |
| 4.6 | SYMB、ESYMBP 与 ESYMBS 的数值表现 (迭代自然终止) | 82 |
| 5.1 | NEWUOA 与 SUBSPACE 的数值表现 (精度 $\tau = 10^{-2}$) | 102 |
| 5.2 | NEWUOA 与 SUBSPACE 的数值表现 (精度 $\tau = 10^{-4}$) | 103 |
| 5.3 | NEWUOA 与 SUBSPACE 的数值表现 (精度 $\tau = 10^{-6}$) | 104 |
| 5.4 | NEWUOA 与 SUBSPACE 的数值表现 (精度 $\tau = 10^{-8}$) | 105 |
| 5.5 | NEWUOA 与 SUBSPACE 的数值表现 (精度 $\tau = 10^{-10}$) | 106 |
| 5.6 | NEWUOA 与 SUBSPACE 的数值表现 (迭代自然终止) | 107 |
| 5.7 | NEWUOA 与 SUBSPACE 求解 200 维 ARWHEAD 问题与 CHROSEN 问题的函数值下降曲线 | 108 |
| 5.8 | NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (精度 $\tau = 10^{-2}$) | 132 |
| 5.9 | NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (精度 $\tau = 10^{-4}$) | 133 |
| 5.10 | NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (精度 $\tau = 10^{-6}$) | 134 |
| 5.11 | NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (精度 $\tau = 10^{-8}$) | 135 |
| 5.12 | NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (精度 $\tau = 10^{-10}$) | 136 |
| 5.13 | NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (迭代自然终止) . | 137 |
| 5.14 | NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (精度 $\tau = 10^{-2}$) | 140 |
| 5.15 | NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (精度 $\tau = 10^{-4}$) | 141 |
| 5.16 | NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (精度 $\tau = 10^{-6}$) | 142 |
| 5.17 | NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (精度 $\tau = 10^{-8}$) | 143 |
| 5.18 | NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (精度 $\tau = 10^{-10}$) | 144 |
| 5.19 | NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (迭代自然终止) . | 145 |

第一章 引言

非线性优化是最古老的数学分支之一. 在《几何原本》第三卷的命题 7 与命题 8 中, 欧几里德¹ 讨论了在定圆周上找一点到定点距离最大或最小的问题. 假定此圆为单位圆, 此定点为 $x_0 \in \mathbb{R}^2$, 则该问题用现代的记号可以描述为

$$\max_{x \in \mathbb{R}^2} \|x - x_0\|_2^2 \quad (1.1)$$

$$\text{s.t. } \|x\|_2^2 = 1, \quad (1.2)$$

以及

$$\min_{x \in \mathbb{R}^2} \|x - x_0\|_2^2 \quad (1.3)$$

$$\text{s.t. } \|x\|_2^2 = 1. \quad (1.4)$$

问题 (1.1)–(1.2) 与 (1.3)–(1.4) 就是典型的非线性优化问题.

非线性优化问题的一般形式是

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1.5)$$

$$\text{s.t. } c_i(x) = 0, i = 1, \dots, m_e, \quad (1.6)$$

$$c_i(x) \geq 0, i = m_e + 1, \dots, m. \quad (1.7)$$

其中, n 为正整数, m 与 m_e 为非负整数, 且 $m \geq m_e$; f 与 c_i ($i = 1, \dots, m$) 为 \mathbb{R}^n 上的实函数, 且至少有一个不是线性的. 函数 f 称为目标函数, 诸 c_i 称为约束函数. 当 $m = 0$ 时, 问题 (1.5)–(1.7) 称为无约束优化问题; 否则称为约束优化问题.

非线性优化问题在国防、经济、工程、管理等领域有广泛应用. 正因为如此, 非线性优化计算方法的研究在过去几十年里引起了广泛的兴趣, 取得了丰硕的成果, 比如著名的共轭梯度法 [127, 114, 49, 189, 73] 和拟牛顿法 [50, 64, 21, 124, 56, 22, 52, 125, 126, 53, 24]. 这两种算法的一个共同点是, 它们都利用了问题的导数信息. 事实上, 这也是目前很大一部分优化方法的特点. 现

¹约公元前 325 年 — 公元前 265 年.

有的优化方法大多为迭代法, 并且主要可以分为两类, 即线搜索方法 [115, 190] 和信赖域方法 [179, 35]. 线搜索方法的基本思想是, 每次迭代先确定一个搜索方向, 再沿该方向寻找下一个迭代点. 信赖域方法的基本想法是, 每次迭代先建立问题的一个模型, 然后在某个区域 (信赖域) 内优化该模型, 期望找到更好的点. 传统上, 这两类方法的实现都经常依赖问题的导数信息²: 线搜索方法根据梯度信息构造搜索方向, 如最速下降方向、共轭梯度方向、牛顿方向、拟牛顿方向等; 信赖域方法根据梯度信息构造模型, 如牛顿模型、拟牛顿模型等. 不仅如此, 因为问题 (1.5)–(1.7) 的最优性条件需要由导数来刻画 [82, 90], 所以大部分优化方法的终止准则都依赖问题的导数信息.

然而, 实际应用中, 很多问题的导数信息是不可用的. 比如, 有的问题目标函数没有显式表达式, 其函数值由实际生产、物理实验或计算机模拟给出. 这种问题的导数信息几乎是不可能得到的. 随着计算机模拟成为越来越重要的科研手段, 这种问题变得越来越多. 一般地, 若问题 (1.5)–(1.7) 中 f 和诸 c_i 至少有一个函数的导数信息不可用, 则我们称之为无导数优化问题 (derivative-free optimization problem). 对于无导数优化问题, 依赖导数信息的方法都不能直接应用. 那么, 这类优化问题应该如何求解? 显然, 我们需要不依赖导数信息的方法, 这就是本文要研究的无导数优化方法 (derivative-free optimization methods) [42], 亦称为直接方法 (direct methods) [190].

作为本文的引言, 本章将在 §1.1 中举例介绍无导数优化方法的应用, 在 §1.2 中综述已有的无导数方法, 在 §1.3 中概述本文主要内容. 另外, §1.2.5 是关于本文术语的一个注记.

§1.1 无导数优化的应用

正如上文提到的, 实际应用中很多优化问题的导数信息是不可用的. 本节简要介绍几个例子. 这些例子参考了 Conn, Scheinberg 和 Vicente 的专著 [42] 和丁晓东的博士论文 [188].

§1.1.1 参数调节

大多数计算机程序都含有一些可调节的参数, 它们能够影响程序的性能. 如何设置这些参数以优化程序性能就成为一个重要问题. 传统上, 这些参数或

²必须注意到, 这两类算法框架本身并不依赖问题的导数信息, 这就使得我们可以发展无导数的线搜索方法和信赖域方法. 详见 §1.2.

者根据其数学或物理意义而被预设一个值, 或者由专家或用户根据经验调节. Audet 和 Orban [9] 讨论了另外一种途径, 即用非线性优化的方法自动调节. 假设一段计算机程序含有一组参数 $p \in \mathbb{R}^K$, 其中 K 是这组参数的个数. 每给定 p 的一个值, 程序就有一个性能表现. 假定这个性能可以被量化为 f , 那么 f 就构成了 p 的一个函数. 我们的目的是最大化程序的性能, 所以参数调节问题可以描述为优化问题

$$\max_{p \in \mathbb{R}^K} f(p). \quad (1.8)$$

实际问题中, p 可能要满足一些约束, 比如界约束. 因为 f 没有显式表达式, 它的值必须通过实际测试这段计算机程序得到, 所以这是一个典型的无导数优化问题. 通过这种建模方式, Audet 和 Orban [9] 研究了信赖域算法中含有的四个参数. 文章使用无导数优化算法 MADS [8] 求解问题 (1.8), 得到的参数比经典的参数选取有更好的数值表现. 感兴趣的读者可以参阅 Audet 和 Orban 的文章 [9].

§1.1.2 工程设计

Booker 等 [15, 14, 13] 研究了直升飞机旋翼叶片设计中的一个优化问题. 该问题的目标是找到一种设计使得旋翼轮轴受到的振动最小. 决策变量有叶片的重量、重心位置、各段刚度 (stiffness) 等, 这些变量满足一些界约束和线性约束. 问题的目标函数没有显式表达式, 函数值是由计算机模拟得到的, 每计算一个函数值需要的 CPU 时间从几分钟到几天不等. 这也是一个典型的无导数优化问题. Booker 等使用基于直接搜索的无导数方法求解这一问题, 结果令人满意.

§1.1.3 分子几何

Meza 和 Martinez [103] 以及 Alberto 等 [4] 研究了分子几何 (molecular geometry) 中的一些优化问题. 比如, 考虑一个包含 N 个原子的原子团, 目标是找到一种几何结构使得原子团的总能量最小. 这可以描述为一个含有 $3N - 6$ 个变量的无约束优化问题. 它的目标函数值是由数值模拟得到的. 这种问题的特点是局部极小值点很多. 基于直接搜索的无导数优化方法已被成功应用于与之相关的一类问题.

§1.1.4 其他应用

无导数优化方法在很多其他领域都有应用, 比如自动误差分析 [77]、电路设计 [188]、生物医学成像 [116, 117]、动态定价 [93] 等等. 更多的应用建议参阅 Conn, Scheinberg 和 Vicente 的专著 [42].

§1.2 算法综述

前面提到, 现有的很大一部分优化方法都依赖导数. 然而, 无导数优化方法并不是一类新方法. 恰恰相反, 在优化方法发展的初期, 无导数方法有着广泛的应用. 这一方面是因为当时缺乏深入的优化理论和精细的优化工具, 另一方面是因为早期的无导数方法 (比如单纯形方法 [158, 110]) 简单直观, 易于实现. 发展到今天, 无导数方法大致可分为直接搜索方法 (direct search methods)、基于线搜索的方法 (methods based on line search)、基于信赖域的方法 (methods based trust region) 等. 本节对现有的无导数方法做一个综述. 我们并不试图涵盖所有方法, 而仅仅列举最有代表性的方法和文献. 关于这些方法的细节可以参考列出的文献. 另外, Brent [19]、Kelley [84] 以及 Conn, Scheinberg 和 Vicente [42] 对现有无导数方法有系统的介绍, 丁晓东的博士论文 [188] 也是很好的参考.

§1.2.1 直接搜索方法

学者们 [98, 89, 42] 认为, “直接搜索 (direct search)” 一词是 Hooke 和 Jeeves [79] (1961 年) 提出的, 但最早的直接搜索方法则被归功于 Fermi 和 Metropolis [63] (1952 年). Hooke 和 Jeeves [79] 把直接搜索定义为通过比较试探点与当前最优点的函数值来确定下一个试探点的方法. 更进一步说, 直接搜索方法不使用具体的函数值 (numerical function values), 而仅仅考察函数值之间的相对优劣 [167, 98]. 这意味着直接搜索方法既不显式构造目标函数的模型, 也不显式近似目标函数的梯度 [176]. 显然, 这种搜索方式对函数值信息的利用是不充分的. 这种特性一方面决定了直接搜索方法不会有很快的收敛速度, 另一方面也使得直接搜索方法可以应用于更一般的优化问题 (比如, 目标函数的值域不是实数集, 而是某个定义了偏序的集合 [79]).

模式搜索方法 (pattern search methods) 是一大类直接搜索方法, 其中包括 Fermi-Metropolis 方法 [63]、Box [17] 的 Evolutionary Operation 方法、Hooke-

Jeeves 方法 [79]、Dennis 和 Torczon 的多方向搜索方法 [164, 54, 165]、Lewis 和 Torczon 的 GPS (Generalized Pattern Search) 方法 [94, 166, 95, 96, 7]、Hough, Kolda 和 Torczon 的 APPSPACK (Asynchronous Parallel Pattern Search) 方法 [80, 88]、Abramson, Audet 和 Dennis 的 MADS (Mesh Adeptive Direct Search) 方法 [1] 等等. 这些方法各有特点. 比如, 多方向搜索方法和 APPSPACK 方法是专门为并行计算设计的方法, GPS 方法可以处理界约束和线性约束的问题, 而 MADS 对于非光滑的问题也可以保证全局收敛性. Torczon [165]、Lewis, Torczon 和 Trosset [97] 以及 Audet 和 Dennis [7] 分析了一般模式搜索方法的收敛性.

基于单纯形操作 (simplex-based operations [42]) 的方法是另外一大类直接搜索方法. 其中最著名的是 Nelder 和 Mead [110] 的单纯形方法 (simplex method, 1965 年). 由于简单直观, Nelder-Mead 方法在工业界应用广泛 (截至 2012 年 3 月, 已被引用一万三千多次). Nelder 和 Mead 的方法从 $n + 1$ 个初始点构成的单纯形开始迭代, 根据单纯形顶点处的函数值对单纯形进行反射 (reflection)、扩张 (expansion)、收缩 (contraction) 等操作, 期望单纯形会逐步吻合函数局部的形态 (“The simplex adapts itself to the local landscape” [110]), 并最终收敛于一个局部极小值点. Nelder-Mead 方法没有很好的收敛性理论. Mckinnon [102] 的例子表明, 对严格凸的函数, 该方法可能收敛到非稳定点. 即使对函数 $f(x) = x^T x$ ($n = 2$), Nelder-Mead 方法的收敛性也没有得到证明 [91, 190]. Lagarias 等 [91] 讨论了 Nelder-Mead 方法对低维问题 ($n = 1, 2$) 的收敛性³. Han 和 Neumann [74] 讨论了维数对 Nelder-Mead 方法的影响, 针对函数 $f(x) = x^T x$ 给出了有趣的理论结果. 为了改善 Nelder-Mead 方法的收敛性质, 很多学者对其进行了修正, 比如 Kelley [83]、Nazareth 和 Tseng [109]、Price, Coope 和 Byatt [147] 等等. 需要指出的是, Nelder-Mead 方法受到了 Spendley, Hext 和 Himsworth [158] 的启发, 后者被认为是最早将单纯形操作引入优化的文献 (1962 年). 与 Nelder-Mead 方法同一年, Box [18] 介绍了一个类似于单纯形方法的算法, 可以处理带有界约束的问题.

关于直接搜索方法的更多内容, 请参考 Wright [176]、Lewis, Torczon 和 Trosset [98]、Kolda, Lewis 和 Torczon [89] 等的综述.

³Lagarias 等 [91] 用现代的符号和语言清晰的描述了 Nelder-Mead 方法, MATLAB (R2012a) 的内置函数 `fminsearch` 就是基于该文的描述而实现的. 本文 §2.3 的数值实验是基于 `fminsearch` 函数的.

§1.2.2 基于线搜索的方法

求解无约束优化问题

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1.9)$$

的经典线搜索框架可以简单的描述如下.

算法 1.1. (线搜索框架)

步 1. 给定初始点 x_1 ; $k := 1$.

步 2. 选取搜索方向 d_k .

步 3. 在一定意义下求解

$$\min_{\alpha \geq 0} f(x_k + \alpha d_k), \quad (1.10)$$

得到步长 α_k .

步 4. $x_{k+1} := x_k + \alpha_k d_k$; $k := k + 1$. 转步 2.

可以看到, 该框架并不依赖导数信息. 只要选取搜索方向 d_k 和步长 α_k 时能避免使用导数信息, 线搜索框架就可以应用于无导数优化问题. 经典的坐标轮换 (coordinate descent) 法 [115, 190] 就是一个典型的无导数线搜索方法.

给定搜索方向之后, 步长的选取是一个一维问题, 相对简单, 可以用区间分割法 (如 Fibonacci 法、黄金分割法)、多项式插值法 [190] 等实现. Diniz-Ehrhardt, Martínez, 和 Raydán [55] 讨论了一种非单调的无导数线搜索技巧.

搜索方向的选取是一个相对困难的问题. 我们把已有的方法分为三类, 即交替方向法 (alternating directions method)、共轭方向法 (conjugate directions method) 和基于近似梯度的方法.

坐标轮换法是最原始的交替方向法, 即轮流使用 n 个坐标方向作为搜索方向. 坐标轮换法的一个缺点是, 它和最速下降法一样可能出现锯齿 (zigzag) 现象. 克服这一困难的方法是引入 Hooke 和 Jeeves [79] 提出的模式搜索 (pattern search)⁴. Hooke 和 Jeeves 的做法很直观. 假设初始点为 x_1 , 经过一轮坐标轮换搜索之后得到 x_2 , 那么 $x_2 - x_1$ 很可能是一个下降方向, 所以一个合理的做法是把下一个试探点选为

$$\bar{x} = x_2 + \alpha(x_2 - x_1), \quad (1.11)$$

⁴请参考 §1.2.5 中关于这一术语的注记.

其中 α 为试探步长 (Hooke 和 Jeeves 取 $\alpha = 1$). Hooke 和 Jeeves 把 $x_2 - x_1$ 称为一轮坐标轮换搜索得到的一个模式方向 (pattern). 在一般情况下, 模式步的引入能克服锯齿现象, 使算法收敛更快. 坐标轮换的一个简单推广是把 n 个坐标方向换成任意一组正交基, 而且允许它根据迭代中积累的信息而改变, 这就是转轴法 (rotating coordinate method) 的思想. 这种方法最早由 Rosenbrock [150], 故又称为 Rosenbrock 方法 [190]. 值得指出的是, Rosenbrock 方法在选取正交基时, 新正交基的第一个方向总是上一轮交替方向搜索得到的模式方向. 所以 Rosenbrock 方法天然地融入了模式搜索的思想. 该方法后来经 Davies, Swann 和 Campey [160, 161] 修改. Palmer [119] 和 Powell [122] 讨论了如何比较经济的计算迭代中需要的正交基. 如果每一步都使用精确线搜索, 则坐标轮换法和 Rosenbrock 方法只要收敛就一定收敛到一个稳定点. 袁亚湘 [190] 对交替方向法的收敛性作了介绍. 值得一提的是, 交替方向法的思想在优化领域的应用很广, 这一思想与增广 Lagrange 函数法相结合可以导致一类求解可分约束问题的有效方法. 感兴趣的读者可以参考 Esser [60]、Liu, Yang 和 Zhang [99]、Yang, Zhang 和 Yin [178]、Wen, Golbfarb 和 Yin [171]、He 和 Yuan [75] 等的研究.

共轭方向法是指逐步产生共轭方向作为搜索方向的方法. 这类方法引入的出发点是, 共轭方向能够导致二次终止性. 假设我们极小化凸二次函数

$$f(x) = \frac{1}{2}x^T Hx + g^T x. \quad (1.12)$$

如果在迭代中能逐步产生一组 H -共轭方向 $\{d_1, d_2, \dots, d_n\}$ 作为搜索方向, 并且每次线搜索都是精确的, 则 n 次线搜索完成之后就能得到 f 的全局极小值点. 共轭梯度法就是一种共轭方向法, 但共轭梯度法依赖导数信息. 不用导数信息也可以构造共轭方向. 事实上, 我们有以下定理.

定理 1.2 ([190], 定理 5.3.1). 设 $1 \leq k < n$, d_1, \dots, d_k 线性无关, 且

$$\mathcal{S}_1 = z_1 + \text{span}\{d_1, \dots, d_k\} \quad (1.13)$$

$$\mathcal{S}_2 = z_2 + \text{span}\{d_1, \dots, d_k\} \quad (1.14)$$

是 \mathbb{R}^n 中两个不同的子空间. 若 z_1 与 z_2 分别是二次函数 (1.12) 在 \mathcal{S}_1 与 \mathcal{S}_2 上的极小点, 则 $z_1 - z_2$ 与 $\{d_1, d_2, \dots, d_n\}$ H -共轭.

无导数的共轭方向法就是利用以上定理构造共轭方向的. 最早的共轭方向法由 Smith [157] 提出. Powell [120] 也研究了共轭方向法, Zagwill [184] 对 Powell 的方法做出了改进. 尽管共轭方向法是针对二次函数提出的, 但正如共轭梯度法那样, 它可以应用于一般的非线性问题. 实际计算表明, 对于一般的非线性问题, Powell 的方法优于 Smith 的方法 [190]. 而 Zagwill 的方法对严格凸函数可以保证全局收敛性 [190].

基于近似梯度的线搜索方法的思想很直接, 就是用某种方式得到近似梯度, 然后模拟经典的基于导数的方法 (比如拟牛顿方法). 最直观的近似梯度的方式是有限差分. 差分拟牛顿法就是使用这种近似梯度的拟牛顿法. 最早的差分拟牛顿方法由 Stewart [159] 提出, 他研究了差分 DFP 算法. Gill 和 Murray [65] 研究了基于差分的 Broyden 族方法. 有限差分可以视作插值点特殊选取的插值, 其推广是用更一般的插值或回归建立目标函数的局部模型, 再用模型的梯度作为目标函数的近似梯度. 如果在单纯形顶点上做线性插值则得到单纯形梯度 (simplex gradient/derivative [16, 48, 47, 45]). Gilmore 和 Kelley [66, 85, 86] 提出的 Implicit Filtering 方法本质上是一种基于单纯形梯度的拟牛顿方法. Choi 和 Kelley [28] 分析了 Implicit Filtering 的收敛性. 不论是用差分还是插值近似梯度, 差分步长或插值点距离的控制都是至关重要的问题, Stewart [159] 和 Kelley [85] 讨论了这一问题. 另一种基于近似梯度的无导数方法是无导数拟牛顿 (quasi-Newton without derivatives) 法. 此类方法由 Greenstadt [68, 69] 提出. Greenstadt [68, 69] 推导了基于函数值的拟牛顿条件, 并要求梯度和 Hessian 矩阵满足某些变分极小性质, 从而可以得到近似梯度和近似 Hessian 矩阵, 进一步得到搜索方向.

§1.2.3 基于信赖域的方法

求解无约束优化问题 (1.9) 的经典信赖域算法框架可以表述如下 [35]. 其中 $\|\cdot\|$ 表示 \mathbb{R}^n 上的某种范数.

算法 1.3. (信赖域框架)

步 1. 取初始点 x_1 , 初始信赖域半径 $\Delta_1 > 0$. 设置常数 $\eta_1, \eta_2, \gamma_1, \gamma_2$ 使得

$$0 < \eta_1 \leq \eta_2 < 1, \quad 0 < \gamma_1 \leq \gamma_2 < 1. \quad (1.15)$$

$k := 1$.

步 2. 构造 f 在 x_k 处的局部模型 m_k .

步 3. 近似或精确求解信赖域子问题

$$\min m_k(x_k + d) \quad (1.16)$$

$$\text{s.t. } \|d\| \leq \Delta_k, \quad (1.17)$$

得到试探步 s_k .

步 4. 计算下降比:

$$\rho_k = \frac{f(x_k + s_k) - f(x_k)}{m_k(x_k + s_k) - m_k(x_k)}. \quad (1.18)$$

步 5. 更新信赖域半径:

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{若 } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{若 } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{若 } \rho_k < \eta_1. \end{cases} \quad (1.19)$$

步 6. 若 $f(x_k + s_k) < f(x_k)$, 则 $x_{k+1} = x_k + s_k$, 否则 $x_{k+1} = x_k$. $k := k + 1$.
转步 2.

可以看到, 该框架不依赖导数信息. 只要在建立模型 m_k 时能避免使用导数信息, 该框架就能应用于无导数优化问题. 传统的信赖域方法中, 常用的模型有牛顿模型、拟牛顿模型等, 都依赖导数信息. 那么, 当导数信息不可用时, 模型应该如何建立? 这是基于信赖域的非导数方法的核心问题之一. 目前的方法主要是使用插值, 也就是说, 在第 k 步迭代中, 选择一个插值点集 \mathcal{I}_k , 通过求解插值问题

$$m_k(y) = f(y), \quad y \in \mathcal{I}_k \quad (1.20)$$

来建立模型 m_k .

使用插值建立模型并不是一种新方法. 传统的牛顿模型和拟牛顿模型都可以看做插值模型, 只是它们对导数也有插值要求. 要通过插值问题 (1.20) 确定模型 m_k , 首先要回答的问题是, 插值函数空间如何选取. 目前的方法中, 插值函数空间的选取主要有三种, 即一次多项式函数空间、二次多项式⁵ 函数空间、

⁵准确地说, 应为次数不超过二次的多项式. 为方便起见, 我们以后提到“二次多项式”或“二次函数”都是指次数不超过二次的多项式, 除非另有说明.

和径向基函数 (radial basis functions [130, 132, 23, 141]) 空间. 插值点集 \mathcal{I}_k 的选取也很关键. 因为 \mathcal{I}_k 的几何性质会在很大程度上影响插值问题 (1.20) 解的行为. \mathcal{I}_k 的选取首先要使问题 (1.20) 有解, 即要求 \mathcal{I}_k 是适定 (poised [42]) 的; 然后要保证解对目标函数有比较好的近似, 即要求 \mathcal{I}_k 是良定 (well poised [42]) 的. 关于插值点集适定性 (poisedness) 的讨论可以参考 Sauer 和 Xu [152] 以及 Conn, Scheinberg 和 Vicente [42]. 需要指出的是, 在基于插值的方法中, 第 k 次迭代完成后, \mathcal{I}_k 中的点并不会全部被丢弃, 其中的大部分点会被 \mathcal{I}_{k+1} 继承; 绝大多数情况下, 新得到的迭代点也会进入 \mathcal{I}_{k+1} , 除非这样做会严重影响 \mathcal{I}_{k+1} 的适定性. 这就大大节约了函数值计算次数. 这与基于差分的方法是不同的.

最早使用插值问题 (1.20) 构造模型的是 Winfield [174, 175] (1969 年). Winfield 使用的是二次插值. Powell [128] 通过对目标函数和约束函数进行一次插值构造了 COBYLA 算法, 该算法可以求解带约束的无导数优化问题.

由于 \mathbb{R}^n 中的一个二次函数有 $(n+1)(n+2)/2$ 个自由度, 所以要通过插值问题 (1.20) 唯一确定一个二次模型, 插值点集 \mathcal{I}_k 中至少要有 $(n+1)(n+2)/2$ 个点. 即使 n 不是特别大 (比如几百), 这也是无法接受的. Conn 和 Toint [43] 构造了一个基于二次插值的无导数算法. 该算法的优点在于, 当插值点的个数少于 $(n+1)(n+2)/2$ 时, 仍可以构造插值模型. 其方法是求解二次插值问题

$$\min \|\nabla^2 m_k\|_F^2 + \|\nabla m_k(x_k)\|_2^2 \quad (1.21)$$

$$\text{s.t. } m_k(y) = f(y), \quad y \in \mathcal{I}_k. \quad (1.22)$$

其中, x_k 为第 k 步迭代的信赖域中心. 以 Conn 和 Toint [43] 的算法为雏形, Conn, Scheinberg 和 Toint [37, 38] 通过二次插值构造了 DFO 算法. DFO 算法的模型是通过求解二次插值问题

$$\min \|\nabla^2 m_k\|_F^2 \quad (1.23)$$

$$\text{s.t. } m_k(y) = f(y), \quad y \in \mathcal{I}_k. \quad (1.24)$$

确定的. Newton 基本多项式 (Newton fundamental polynomials) [152] 在 DFO 算法的构造中起了重要作用. DFO 使用 Newton 基本多项式度量插值点集的适定性, 并指导插值点集的选择和更新. Conn, Scheinberg 和 Toint [36] 证明了 DFO 算法的全局收敛性.

Powell [131] 使用二次插值构造了 UOBYQA 算法. 该算法对于维数不是很大 (比如 $n \leq 20$) 的问题表现十分出色. 丁晓东 [188] 对 UOBYQA 作了并行化, 得到

了 PEUOBYQA 算法. 数值实验表明, PEUOBYQA 算法能够有效缩短计算时间, 提高求解精度 [188]. 由于 UOBYQA 使用完全二次插值⁶, 故它需要的函数值计算次数随着问题维数的增长至少是二次的; 另外, Powell [131] 还指出, UOBYQA 每一步迭代的计算量随问题维数的增长是四次的. 所以, UOBYQA 不适用于规模比较大的问题. 为了克服这一困难, Powell [138] 提出了 NEWUOA 算法. 同 DFO 一样, NEWUOA 可以用少于 $(n+1)(n+2)/2$ 个插值点构造二次模型. 其方法是求解二次插值问题

$$\min \|\nabla^2 m_k - \nabla^2 m_{k-1}\|_F^2 \quad (1.25)$$

$$\text{s.t. } m_k(y) = f(y), y \in \mathcal{I}_k. \quad (1.26)$$

Powell 的方法借鉴了拟牛顿方法中最小变化修正 [53] 的思想, 相当于无导数的对称 Broyden 修正. 数值实验表明, 这种构造模型的方法能提取有用的二阶信息, 加快算法的收敛. Powell 建议在迭代中保持插值点的个数为 $2n+1$, 这样除了个别迭代以外, 每步迭代的计算量为 n^2 量级. 实际计算中, NEWUOA 需要的函数值计算次数随问题维数的增长是线性的. 这使得 NEWUOA 可以处理较大规模的问题. 在 Powell [138] 的数值试验中, 对于 160 维的 ARWHEAD [67] 测试函数, NEWUOA 仅需要 8504 次函数值计算就可以求到很高精度的解. 而如果用完全二次插值构造一个 160 个变量的二次函数, 则需要 12800 次函数值计算. 这表明 NEWUOA 修正模型的技巧是十分有效的. Powell [142] 对 NEWUOA 算法做了进一步讨论. 在 UOBYQA 和 NEWUOA 的构造中, Lagrange 插值基函数起到了关键作用. Lagrange 插值基函数不仅用来构造插值模型, 还用来反映插值点集的适定性, 指导插值点集的更新. Powell [134, 133, 135] 详细讨论了 NEWUOA 算法中 Lagrange 插值基函数的使用和更新. 从数值表现来看 [107], NEWUOA 是目前最优秀的无导数算法之一. Powell [143] 将 NEWUOA 算法推广到求解界约束问题, 得到了 BOBYQA 算法. 丁晓东 [188] 针对界约束问题的 DFBOLA 算法和 DFBODA 算法也是 NEWUOA 算法的推广, 它们的数值表现与 BOBYQA 相当 [188].

另一种用少于 $(n+1)(n+2)/2$ 个插值点建立二次插值模型的方法是 Bandeira, Scheinberg 和 Vicente [10] 提出的稀疏二次插值方法. 这种方法通过

⁶也就是, 使用 $(n+1)(n+2)/2$ 个插值条件确定二次插值函数的所有 $(n+1)(n+2)/2$ 个系数.

求解二次插值问题

$$\min \|\text{vec}(\nabla^2 m_k)\|_1 \quad (1.27)$$

$$\text{s.t. } m_k(y) = f(y), \quad y \in \mathcal{I}_k \quad (1.28)$$

得到二次模型, 其中 $\text{vec}(\nabla^2 m_k)$ 是 $\nabla^2 m_k$ 的上三角元素组成的 $(n+1)(n+2)/2$ 维向量. 插值问题 (1.27)–(1.28) 的构造受到了压缩感知 (compressed sensing, compressive sensing, compressive sampling) [25, 59, 26] 理论的启发, 目的是建立具有稀疏 Hessian 矩阵的模型. Bandeira, Scheinberg 和 Vicente [10] 证明, 当目标函数的 Hessian 矩阵满足一定的稀疏性假设且插值点集 \mathcal{I}_k 满足一定条件时, 求解 (1.27)–(1.28) 能够高概率得到一个高精度模型. Bandeira, Scheinberg 和 Vicente [10] 给出了一个基于这种模型的无导数信赖域方法, 并报告了出色的数值结果.

Marazzi 和 Nocedal [101] 构造了另外一个基于二次插值模型的无导数信赖域算法. 该算法的特点是, 每一步的信赖域子问题除了有传统的球约束之外, 还有一个楔形 (wedge) 约束. 其目的是, 在减小目标函数值的同时, 也试图改善插值点集的适定性.

基于二次插值的无导数信赖域方法还有 Vanden Berghen 和 Bersini [168] 的 CONDOR 算法, 以及 Wild [172] 的 MNH 算法. CONDOR 是 UOBYQA 的一个并行变体, 并且可以求解一般的约束优化问题. MNH 使用了极小 Frobenius 范数插值问题 (1.23)–(1.24) 来构造模型, 这一点与 DFO 是类似的. 另外, 针对非线性最小二乘问题, Zhang, Conn 和 Scheinberg [186, 185] 提出了基于二次插值和信赖域的 DFLS 算法, 并且证明了该算法的全局收敛性和对于零余量问题的二次收敛速度. 由于充分利用了问题的特殊结构, DFLS 算法对于最小二乘问题的表现优于 NEWUOA 算法.

利用径向基函数, Oeuvray 和 Bierlaire [116, 117, 118] 构造了 BOOSTERS 算法, 该算法在生物医学成像方面有成功应用. Wild, Regis 和 Shoemaker [173] 的 ORBIT 也是一个使用径向基函数插值的算法.

正如前面提到的, 使用插值建立模型时, 插值点集的选取十分重要. 直观来讲, 当插值点集接近不适定 (比如, 所有插值点近似落入一个超平面) 时, 插值得到的模型不会是真实函数的一个良好近似. 因此, 基于插值模型的信赖域算法在迭代中大都会在必要时引入某种几何步 (geometry step [153], geometry

phase [62]) 改善插值点集的适定性. 比如 NEWUOA 算法在信赖域步效果欠佳 (步长过短或下降比不理想) 时会引入一个模型步 (model step), 以改善插值点集的几何性质, 进而改善插值模型. Conn, Scheinberg 和 Vicente [39, 40, 42] 详细讨论了插值点集适定性的刻画和改善方法, 以及它对插值函数的影响. 近来, Fasano, Morales 和 Nocedal 提出了一种不显式使用任何几何步的方法, 实验表明, 这种方法的数值表现令人满意. Scheinberg 和 Toint [154] 从理论上研究了这一现象, 证明了基于多项式插值模型的信赖域方法具有某种内在的自修正机制 (self-correction mechanism), 使得插值点集的适定性能在迭代中自动得到改善. 但是, Scheinberg 和 Toint [154] 也指出, 要保证算法的全局收敛性, 几何步是不能完全去除的.

除了使用插值构造模型以外, 当函数值信息足够多时, 也可以使用回归模型. Billups, Larson 和 Graf [12] 讨论了如何使用带有权重的回归建立二次模型.

Conn, Scheinberg 和 Vicente [41, 42] 讨论了基于信赖域的无导数算法的全局收敛性.

§ 1.2.4 其他方法

实际应用中, 一些启发式的方法也被用来求解无导数优化问题, 比如模拟退火 (simulated annealing) [87]、遗传算法 (genetic algorithm) [51]、神经网络 (neural networks) [29] 等等. 这些算法大多基于对某些自然过程的模拟. 感兴趣的读者可以参考 Michalewicz 和 Fogel 的专著 [104].

§ 1.2.5 关于术语的注记

由于历史的原因, 无导数优化领域有很多术语. 这些术语在以上综述中已经出现. 为了防止读者在阅读本文或相关文献时出现混乱, 本小节对若干术语作出注记.

§ 1.2.5.1 关于“直接搜索”、“直接方法”与“无导数方法”

一个很有意思的现象是, 学者们对于直接搜索 (direct search) 的定义十分在意. 综述文章 Wright [176]、Lewis, Torczon 和 Trosset [98]、Kolda, Lewis 和 Torczon [89] 都有专门的篇幅讨论直接搜索的定义. 甚至有专门的文献论述这

一问题, 比如 Trosset [167]. 作为直接搜索 (direct search) 一词的滥觞, Hooke 和 Jeeves [79] 特别解释了引入这一概念的动机:

We use the phrase “**direct search**” to describe sequential examination of trial solutions involving comparison of each trial solution with the “best” obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be. The phrase implies our preference, based on experience, for straightforward search strategies which employ no techniques of classical analysis except where there is a demonstrable advantage in doing so.

Trosset [167] 给出的直接搜索的定义是:

A direct search method for numerical optimization is any algorithm that depends on the objective functions only through the ranks of a countable set of function values.

上述定义基本反映了 Hooke 和 Jeeves [79] 的本意, 也与 Wright [176]、Lewis, Torczon 和 Trosset [98]、Kolda, Lewis 和 Torczon [89] 的观点一致. 我们在 § 1.2.1 中综述的直接搜索方法就是指这类方法.

但是, Powell [129] 和 Swann [161] 所综述的直接搜索方法不限于此. 事实上, Powell [129] 和 Swann [161] 讨论了所有不依赖导数信息的方法.

鉴于 Hooke 和 Jeeves [79] 最早提出直接搜索时所作的限制, 本文使用“直接搜索”时仅指 Hooke-Jeeves 意义下的直接搜索. 另外, 本文视“直接方法 (direct methods)”与“无导数方法 (derivative-free methods)”为同义词, 尽管我们只使用后者.

§ 1.2.5.2 关于“模式搜索”

最早提出“模式搜索 (pattern search)”一词的也是 Hooke 和 Jeeves [79]. Torczon [166] (1997 年) 提出了一个模式搜索的抽象定义, 给出了一般的模式搜索的框架并证明了全局收敛性. 在此之后, Audet 和 Dennis [6, 7]、Momma 和 Bennett [105]、Nie 和 Ma [113]、Alberto 等 [4]、Al-Sumait, Al-Othman 和 Sykulskii [3] 等都遵循了这一定义.

但是, 本文的作者认为, Torczon [166] 给出的定义是对 Hooke 和 Jeeves [79] 提出的模式搜索的一种误解, 并未体现 Hooke-Jeeves 模式搜索的思想. 为

了证明这一点, 下面我们来考察相关的原始文献.

Hooke 和 Jeeves [79] 详细描述了模式搜索的基本思想:

The first type of move is an exploratory move designed to acquire knowledge concerning the behavior of the function $S(\varphi)$The rudimentary information of success or failure is utilized by combining it into a “**pattern**” which indicates a probable direction for a successful move. ...The second type of move is a **pattern move** designed to utilize the information acquired in the exploratory moves, and accomplish the actual minimization of the function by moving in the direction of the established “**pattern**”. ...The intuitive basis for this type of move is the presumption that whatever constituted a successful set of moves in the past is likely again to prove successful.

可见, 算法中的搜索步被分为两类, 即探索步 (exploratory move) 和模式步 (pattern move). 而模式步的目的是为了利用探索步发现的模式 (pattern). 这与我们在 § 1.2.2 中的描述是一致的. 为了进一步阐明模式搜索的含义, Hooke 和 Jeeves [79] 指出:

The direct search procedure outlined above has been termed **pattern search** because it is based on the determination of a “**pattern**” of simple moves that will give a successful direction in which to move.

Swann [161] 在综述模式搜索时也指出:

Exploratory moves examine the local behavior of the function and seek to locate the direction of any slope valleys present; **pattern** moves utilize the information yielded by the explorations by progressing along any such valleys.

Lawrence 和 Steiglitz [92] 在改进 Hooke-Jeeves 方法时使用的模式搜索概念也与上述一致.

现在我们来看看 Torczon [166] 给出的定义. 在给出抽象的模式搜索的定义之前, Torczon [166] 描述了其动机:

...it has been apparent to us that the unifying theme that distinguishes these algorithms from other direct search methods is that each of them performs a search using a “**pattern**” of points that is **independent** of the objective function f . This informal insight is the basis for our general definition of pattern search methods.

这里的“pattern”实际是指搜索点所在的网格. 为了描述这种 pattern, Torczon [166] 引入代数中的格 (lattice) 的概念:

Formally, our definition of pattern search methods requires the existence of a lattice T ...

这已经背离了 Hooke-Jeeves 模式搜索的本意. 可以说, Torczon [166] 定义的模式搜索并未体现 Hooke 和 Jeeves [79] 描述的“pattern move”或“pattern search”的思想.

当然, 这并不影响 Torczon [166] 意义下的模式搜索成为一类强有力的算法. 我们在 §1.2.1 中综述的“模式搜索”算法正是指 Torczon [166] 界定的这一类算法. 事实上, 这类算法由于简单直观, 已经成为应用十分广泛的一类无导数方法. 另外, 格已经成为描述和研究一大类直接搜索算法的标准工具, 感兴趣的读者可以参考 Conn, Scheinberg 和 Vicente 的专著 [42].

同时, Hooke-Jeeves 模式搜索的思想是不能被忽视的. 我们将在第五章中借鉴这一思想.

§1.2.5.3 关于“单纯形方法”

Nelder-Mead 单纯形方法与求解线性规划的单纯形方法是两个完全无关的算法, 尽管名称完全相同.

§1.3 本文主要内容

本文将针对无约束优化问题

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1.29)$$

研究无导数方法. 我们假定 f 是 \mathbb{R}^n 上的实值函数, 其导数信息是不可用的. 另外, 我们还假定 f 的函数值计算代价很大 (这在 §1.1 的例子中已经看到). 因此, 在设计算法时我们力求使用尽量少的函数值计算次数. 我们主要研究基于信赖域的算法. 与大多数算法一样, 尽管我们不使用导数信息, 但我们在算法的构造与分析中仍假定目标函数是光滑的.

在第二章中, 我们讨论无导数优化方法的评价方法. 我们观察到, 无导数算法对计算机舍入误差比较敏感, 即使不引入任何人为噪音, 计算结果也有很强的随机性. 现有的评价方法并没很好地反映这一点. 我们将用一个例子清楚地

说明, 传统的评价方法对于无导数算法是不可靠的. 我们介绍一种观察计算机舍入误差影响的方法, 并相应的建立一套评价无导数算法的新体系. 与传统的评价方法相比, 新的评价体系能反映算法对计算机舍入误差的敏感性, 对算法计算开销的刻画也更可靠.

在第三章中, 我们讨论无导数优化中的最小 Frobenius 范数模型与对称 Broyden 修正. 我们首先证明这两种建立模型的策略在某些情况下的等价性. 然后我们研究 NEWUOA 算法中与这两种策略相关的重新开始技术, 并给出一个改进 NEWUOA 源代码的方法. 这种方法只需删除 NEWUOA 源代码的四个字母即可显著改善其计算效果. 最后, 我们在 NEWUOA 算法的框架下比较最小 Frobenius 模型和对称 Broyden 修正的数值表现. 我们指出, 在 NEWUOA 算法的框架下, 当求解精度要求较低时, 最小 Frobenius 范数模型表现更好. 这一事实对应用领域很有意义, 因为应用中很多无导数优化问题对精度的要求并不高.

在第四章中, 我们将偏微分方程理论中经典的 Sobolev 范数理论引入无导数方法的研究中, 从一种新的角度阐述一类最小范数插值模型, 并揭示插值中参数的几何意义. 利用这些结果, 我们研究扩展的对称 Broyden 修正, 给出其中参数的选取方式. 我们的选取方式简单直观, 并且在实验中的表现优于 Powell [144] 的方法. 我们还将用二次函数的系数给出二次函数的在 l_p 球上的 H^0 范数和 H^1 半范数的统一表达式.

在第五章中, 我们讨论子空间方法在无导数优化中的应用, 提出两种无导数子空间方法. 我们的第一种方法借鉴 Hooke-Jeeves 模式搜索的思想, 把信赖域子问题限制在一个低维子空间上求解. 这种策略能够改善 NEWUOA 算法的数值表现. 第二种方法被称为 NEWUOAs 算法 (A NEW Unconstrained Optimization Algorithm with subspace technique based on NEWUOA). 它是本文最成功的算法. 我们首先研究一个十分广泛的算法框架, 建立其全局收敛性和 R-线性收敛速度. 然后, 利用 NEWUOA 算法作为子问题的求解器, 我们不依赖导数地实现该算法框架, 得到 NEWUOAs 算法. 我们将用实验证明, NEWUOAs 算法的数值表现显著优于 NEWUOA 算法, 而后者被认为是目前最好的无导数算法之一. 我们还指出, NEWUOAs 很善于处理初始点质量较差的问题, 从而很适合实际问题的求解. 不仅如此, 我们还将看到, NEWUOAs 算法已经可以十分高效地求解很多维数高达 2000 的问题. 这是令人振奋的, 因为这一规模已经远远超出了目前大部分无

导数算法 (包括 NEWUOA 算法) 的求解范围.

第六章是总结和对未来工作的展望.

第二章 无导数优化方法的一个评价体系

§ 2.1 引言

如何客观地评价和比较不同的算法是优化算法研究中的重要问题. 我们需要可靠的评价体系来比较不同的算法, 以指导应用领域的算法选择和研究领域的算法改进. 在本章中, 我们建立一种评价无导数优化方法的体系, 这种体系将在以后各章中用于算法的比较.

对于一般的优化算法, 目前比较流行的评价方法是使用 Dolan 和 Moré [58] 提出的 Performance Profile. 针对无导数优化的特点, Moré 和 Wild [107] 讨论了如何将 Performance Profile 用于评价无导数算法, 并且提出了 Data Profile 作为另外一种评价标准.

我们发现, 无导数优化算法对计算机舍入误差比较敏感, 即使不引入任何人为噪音, 计算结果也有很大的随机性. 在比较不同的算法时, 应当采取措施减小计算机舍入误差带来的偏差, 并把算法对计算机舍入误差的敏感性也作为一项评价指标. 但这在目前的评价体系中没有得到体现, 这是本章的研究动机. 在本章中, 我们提出一种观察计算机舍入误差影响的方法, 并相应地建立一套评价无导数算法的体系. 新的评价体系能反映算法对计算机舍入误差的敏感性, 对不同算法的比较也更有说服力.

本章的结构如下. 在 § 2.2 中, 我们简要介绍 Performance Profile 和 Data Profile. 在 § 2.3 中, 我们介绍一个数值例子, 指出计算机舍入误差对无导数算法计算结果的影响是不能忽略的. 在 § 2.4 中, 我们建立一套评价无导数算法的新体系. § 2.5 是本章的总结. 本章部分数值实验的源代码在 § 2.6 中以附录形式给出.

§ 2.2 Performance Profile 与 Data Profile

Performance Profile 是为了评价和比较不同优化算法而出现的一个概念, 由 Dolan 和 Moré [58] 针对一般的优化算法 (乃至一般的数值方法) 提出.

假设有一组优化算法 \mathcal{S} 需要比较¹. 我们关心的首要问题是:

¹我们真正能比较的实际是基于算法的解法器 (solver, 算法的特定实现), 因此我们使用字母 \mathcal{S} .

问题 2.1. 对给定问题, \mathcal{S} 中哪一算法的计算开销最小?

这里的计算开销可能是指 CPU 时间、迭代次数、函数值计算次数、梯度计算次数等. 问题 2.1 是很难回答的, 因为没有算法适用于一切问题. 只要 \mathcal{S} 中的算法相差不是太大, 该问题不可能有精确的答案. 所以, 我们只能在概率意义下回答这一问题. Performance Profile 就是为了回答问题 2.1 而提出的概念. 事实上, Performance Profile 在概率意义上回答了以下问题:

问题 2.2. 对给定问题, \mathcal{S} 中哪一算法的计算开销不会超过最小开销的 α 倍 ($\alpha \geq 1$)?

这里的最小开销是指 \mathcal{S} 中算法的求解该问题的最小计算开销. 当 $\alpha = 1$ 时, 问题 2.2 就是问题 2.1.

为了回答问题 2.2, 我们使用 \mathcal{S} 中的算法求解一组测试问题 \mathcal{P} , 统计每一个算法的计算效果 (是否成功) 和计算开销, 然后通过这些统计数据对 \mathcal{S} 中的算法作出比较. 目前受到公认的测试问题集是 Gould, Orban 和 Toint 收集的 CUTEr [67], 其中涵盖了当前文献中大部分的测试问题集, 比如 Hock 和 Schittkowski 的 HS 问题集 [78, 155, 156]、Moré, Garbow 和 Hillstom 的 Argonn 问题集 [106] 等.

令 $t_{p,s}$ 表示算法 $s \in \mathcal{S}$ 求解问题 $p \in \mathcal{P}$ 的计算开销 (若 s 未能成功求解 p , 则令 $t_{p,s} = \infty$). 给定算法 $\bar{s} \in \mathcal{S}$ 和问题 $\bar{p} \in \mathcal{P}$, 定义

$$r_{\bar{p},\bar{s}} = \begin{cases} \frac{t_{\bar{p},\bar{s}}}{\min\{t_{\bar{p},s} : s \in \mathcal{S}\}} & t_{\bar{p},\bar{s}} < \infty, \\ R & t_{\bar{p},\bar{s}} = \infty. \end{cases} \quad (2.1)$$

其中, R 可以是任何满足

$$R > \max\{r_{p,s} : t_{p,s} < \infty, p \in \mathcal{P}, s \in \mathcal{S}\} \quad (2.2)$$

的常数 (可以取无穷), 它的具体取值不会影响下面的讨论. 可以看出, 当 $r_{p,s} < R$ 时, $r_{p,s}$ 的意义是: 在问题 p 上, 算法 s 的计算开销与相应最小开销之比. 另外 $r_{p,s} = R$ 当且仅当 s 求解 p 失败.

为了刻画算法在整个测试问题集上的平均表现, 定义函数

$$\pi_s(\alpha) = \frac{1}{|\mathcal{P}|} |\{p \in \mathcal{P} : r_{p,s} \leq \alpha\}|, \quad 1 \leq \alpha < R. \quad (2.3)$$

其中, $|\cdot|$ 表示集合的基数. 可以看出, π_s 是单调递增的分段常数函数. $\pi_s(\alpha)$ 的意义是: 在求解问题集 \mathcal{P} 时, 算法 s 能在最小开销的 α 倍之内求解的问题比例. 当然, 这必须要求 $\alpha \in [1, R)$. 由于 $r_{p,s}$ 的定义, π_s 在 $[R, \infty)$ 上的值没有意义.

当问题集 \mathcal{P} 足够有代表性时, $\pi_s(\alpha)$ 就估计了 s 能在最小开销的 α 倍之内求解给定问题的概率. 特别的, $\pi_s(1)$ 估计了 s 能胜过 \mathcal{S} 中其他所有算法的概率. 所以, 问题 2.1 的答案是: 给定问题, 算法 s 计算开销最小的概率约为 $\pi_s(1)$; 而问题 2.2 的答案是: 给定问题, 算法 s 计算开销不超过最小开销 α 倍的概率约为 $\pi_s(\alpha)$ ($1 \leq \alpha < R$). 另外, 给定问题, 算法 s 能成功将其求解的概率约为 $\lim_{\alpha \rightarrow R^-} \pi_s(\alpha)$. 粗略地说, π_s 的值越大, 意味着算法 s 相对于 \mathcal{S} 中其他算法的表现越好. Dolan 和 Moré [58] 称 π_s 为算法 s 的 Performance Profile.

在实际应用中, 计算开销的预算往往是有限的, 故人们还关心以下问题:

问题 2.3. 给定计算开销的预算, 算法 s 能成功求解给定问题的概率是多少?

为了回答问题 2.3, Moré 和 Wild [107] 提出了 Data Profile 的概念. 问题 2.3 的回答相对简单. 定义

$$\delta_s(\alpha) = \frac{1}{|\mathcal{P}|} |\{p \in \mathcal{P} : t_{p,s} \leq \alpha\}|, \quad (2.4)$$

则问题 2.3 的答案是: 给定计算开销的预算 α , 算法 s 能成功求解给定问题的概率约为 $\delta_s(\alpha)$. 粗略地说, δ_s 的值越大, 意味着算法 s 越可靠. Moré 和 Wild [107] 称 δ_s 为算法 s 的 Data Profile.

利用 Performance Profile 和 Data Profile, 我们可以将算法的比较可视化: 将 \mathcal{S} 中不同算法的 π_s 图线在同一张图中显示, 图线较高的算法表现较好. 对 δ_s 可以做类似处理. 具体的例子请参考第三至第五章的数值实验.

值得注意的是, 算法的优劣不是绝对的. Performance Profile 意义下表现较好的算法在 Data Profile 意义下未必较好. Moré 和 Wild [107] 对此作了讨论.

有了 Performance Profile 与 Data Profile, 似乎我们已经能够有效的比较不同的算法. 但对于无导数算法来说并非如此. 由 (2.1) 与 (2.4) 可知, Performance Profile 与 Data Profile 完全取决于诸 $t_{p,s}$, 即算法 s 求解问题 p 的计算开销 (若求解失败则视开销为无穷). 可以看出, $t_{p,s}$ 取决于算法的终止准则. 所以, 要客

观地比较不同的算法, 终止准则必须一致. 与基于导数的算法不同, 无导数算法的终止准则很难统一. 另外, 对于无导数优化问题, 人们往往关心算法何时能找到一个具有充分精度的解. 基于这种考虑, Moré 和 Wild [107] 提出, 在测试算法时, 将解的精度作为算法终止准则, 并且在不同精度要求下比较不同算法的表现. 具体的做法是: 给定精度 $\tau \in (0, 1)$, 令算法 $s \in \mathcal{S}$ 求解问题 $p \in \mathcal{P}$; 定义 $t_{p,s}$ 为

$$f_1 - f_k \geq (1 - \tau)(f_1 - f^*), \quad (2.5)$$

第一次被满足时的计算开销, 其中 f_1 为初始点处的函数值, f_k 为算法计算的第 k 个函数值, f^* 为问题最优值; 若直到迭代自然终止², (2.5) 也未被满足, 则 s 没有成功将 p 求解到精度 τ , 定义 $t_{p,s} = \infty$. 实际测试中, 我们未必知道问题的精确最优值. 通常的做法是, 先令 \mathcal{S} 中所有的算法求解 p 直至迭代自然终止, 然后定义 f^* 为得到的最优值中最小的一个. 这样, 给定精度要求, 各算法求解各测试问题的计算开销即可确定, 从而可以使用 Performance Profile 和 Data Profile 比较各算法的表现.

第一章已经提到, 无导数优化问题的函数值计算代价往往很高. 因此, 在评价无导数算法时, 计算开销一般使用函数值计算次数来度量. 另外, 由于函数值计算次数必然受到问题维数的影响, 故在比较无导数算法时, Data Profile 的定义往往修正为

$$\delta_s(\alpha) = \frac{1}{|\mathcal{P}|} |\{p \in \mathcal{P} : \frac{t_{p,s}}{n_p + 1} \leq \alpha\}|, \quad (2.6)$$

其中, $t_{p,s}$ 为函数值计算次数, n_p 为问题 p 的维数. 事实上, 如果使用单纯形梯度 [16, 48, 47, 45] 近似函数的梯度, 每次近似需要使用 $n_p + 1$ 个函数值, 故在比较无导数算法的函数值计算次数时, 把 $n_p + 1$ 个函数值作为一个比较是合理的, 这样既可以使不同维数的问题有可比性, 又可以近似地与基于导数的方法比较.

以上就是 Dolan 和 Moré [58] 以及 Moré 和 Wild [107] 的主要思想. Moré 和 Wild [107] 利用 Performance Profile 和 Data Profile 比较了无导数算法 NEWUOA [138, 142]、NMSMAX [76] 和 APPSPACK [80] 在不同求解精度下的计算开销. 文中报告的结果表明, NEWUOA 的数值表现优于另外两个算法.

作为本节的结束, 我们对 Moré 和 Wild [107] 的方法作一个注记.

²算法内置的终止准则被满足. 比如, 信赖域半径或步长达到下限, 或者计算开销达到上限等等.

条件 (2.5) 是合理的, 但也有其局限性. 由于导数信息的缺失, 无导数方法的终止准则是算法设计中的重要部分. 因此, 算法是否具有适时终止的能力也应当纳入算法的比较当中. 所以, 我们建议, 除了比较算法在给定精度下的开销之外, 还应比较迭代自然终止时的开销. 具体的做法是: 给定问题 $p \in \mathcal{P}$, 记 f^* 为问题的最优值; 给定算法 $s \in \mathcal{S}$, 令算法 s 求解问题 p 直至迭代自然终止, 得到的最优值记为 f_s^* ; 若 f_s^* 与 f^* 足够接近, 则认为 s 成功求解 p , 定义 $t_{p,s}$ 为迭代自然终止时的计算开销; 若 f_s^* 与 f^* 相差很大, 则认为 s 求解 p 失败, 令 $t_{p,s} = \infty$; 最后, 利用 Performance Profile 与 Data Profile 比较各算法. 关于判断 f_s^* 与 f^* 是否足够接近的准则, 我们建议使用

$$f_1 - f_s^* \geq (1 - \varepsilon)(f_1 - f^*), \text{ 且 } f_s^* - f^* \leq \varepsilon \min\{1, |f^*|\}. \quad (2.7)$$

其中, ε 是一个小正数.

§ 2.3 计算机舍入误差的影响: 一个数值例子

前面提到, 在无导数优化问题中, 函数值的计算代价往往很高, 故无导数算法的计算开销一般用函数值计算次数来度量. 从一般的观点来看, 要考察某算法求解某问题的开销, 只要记录终止准则成立时算法花费的函数值计算次数即可. 但是, 实际计算提示我们, 这种做法并不总是可靠. 这是因为, 无导数算法往往对计算机舍入误差比较敏感, 计算结果有很强的随机性, 简单的记录函数值计算次数可能毫无意义. 本节给出一个例子来说明这一点. 该例受到了 Powell [144] 数值试验的启发.

考虑一个测试问题

$$\min_{x \in \mathbb{R}^n} f(x), \quad (2.8)$$

初始点为 x_0 . 我们改变 f 中变量的顺序, 得到一个新的函数和初始点. 准确的说, 取一个 n 阶置换矩阵 P , 令

$$\tilde{f}(x) = f(Px), \quad \tilde{x}_0 = P^{-1}x_0. \quad (2.9)$$

现在, 考虑用算法 s 求解问题

$$\min_{x \in \mathbb{R}^n} \tilde{f}(x), \quad (2.10)$$

初始点为 \tilde{x}_0 . 理论上说, 若 s 的实现不依赖目标函数中的变量顺序³, 则 s 求解问题 (2.10) 的计算结果与 P 无关. 然而, 实际结果并非如此. 下面是一个例子.

我们测试 MATLAB 7.6.0.324 (R2008a) 内置的 `fminsearch` 函数. 此函数是 Nelder-Mead 单纯形算法的一个实现, 该实现基于 Lagarias 等 [91] 对 Nelder-Mead 算法的描述. MATLAB 7.6.0.324 提供了该函数的源代码⁴. 通过检查源代码可以看到, 算法的实现与变量顺序无关. 测试函数为 BDQRTIC 函数 [67]:

$$f(x) = \sum_{i=1}^{n-4} [(x_i^2 + 2x_{i+1}^2 + 3x_{i+2}^2 + 4x_{i+3}^2 + 5x_n^2)^2 + (3 - 4x_i)^2], \quad x \in \mathbb{R}^n, \quad (2.11)$$

初始点为 $x_0 = (1, \dots, 1)$, 维数 n 取 20. 测试平台为 Dell PC (OPTIPLEX 755), 系统为 Ubuntu 9.04 (Linux 2.6.28-15-generic). 取置换矩阵

$$P_1 = (e_{13} \ e_{16} \ e_{15} \ e_{14} \ e_2 \ e_{10} \ e_8 \ e_{12} \ e_1 \ e_{19} \ e_5 \ e_{20} \ e_3 \ e_6 \ e_{18} \ e_7 \ e_{11} \ e_4 \ e_9 \ e_{17})^T \quad (2.12)$$

和

$$P_2 = (e_{10} \ e_8 \ e_{16} \ e_7 \ e_{14} \ e_2 \ e_{19} \ e_{13} \ e_{17} \ e_5 \ e_1 \ e_4 \ e_{18} \ e_3 \ e_{15} \ e_9 \ e_6 \ e_{12} \ e_{11} \ e_{20})^T, \quad (2.13)$$

其中, e_i 为 \mathbb{R}^{20} 中第 i 个坐标向量. 对 P_1 与 P_2 , 分别按照 (2.9) 产生目标函数和初始点, 然后用 `fminsearch` 求解问题 (2.10). 结果是惊人的: 对应于这两个置换矩阵, `fminsearch` 求解问题 (2.10) 的计算开销相差近十倍. 表 2.1 列出了 P_1 与 P_2 对应的计算结果. 其中, 第 2 至第 6 列为不同求解精度对应的函数值计算次数; 第 7 列为迭代自然终止时的函数值计算次数; 最后一列为两次求解得到的最优值. 图 2.1 为两次计算中函数值的下降曲线, 其中 k 表示函数值计算次数, f_k^* 表示 `fminsearch` 使用 k 次函数值计算求得的最优值. 这两次计算的源代码在本章附录 §2.6 中给出.

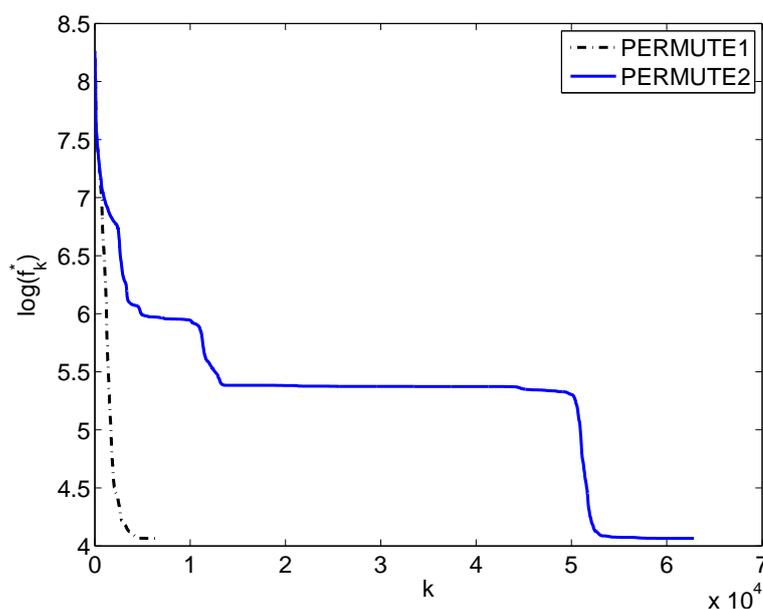
如此巨大的差异是如何产生的? 我们注意, 在数学上, 无论置换矩阵如何选取, `fminsearch` 求解问题 (2.10) 的计算过程都与求解原始问题 (2.8) 等价, 因为 `fminsearch` 的计算过程与变量顺序无关. 因此, 计算结果的任何差异都来自计算机舍入误差的影响. 事实上, 置换矩阵的引入改变了迭代中某些运算 (比

³举例来说, 最速下降法、共轭梯度法的经典实现都不依赖变量顺序, 而坐标轮换法的实现很难不依赖变量顺序.

⁴Unix 版本中, `fminsearch` 的源代码位于 `$MATLAB/toolbox/matlab/funfun/fminsearch.m`, 其中 `$MATLAB` 为 `matlab` 的安装路径.

表 2.1: 20 维 BDQRTIC 问题对 P_1 、 P_2 的实验结果

| | 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} | 10^{-5} | Natural | f_{best} |
|-------|-----------|-----------|-----------|-----------|-----------|---------|-------------------|
| P_1 | 1203 | 1968 | 3590 | 4384 | 4695 | 6617 | 5.832066E+01 |
| P_2 | 4747 | 51511 | 52544 | 56776 | 58081 | 62845 | 5.832041E+01 |

图 2.1: 20 维 BDQRTIC 问题对 P_1 、 P_2 的实验结果

如函数值计算中的求和运算) 的顺序. 由于计算机字长有限, 顺序的改变会影响这些运算的结果, 进一步影响最终计算结果. Dixon 和 Mills [57] 对变尺度算法讨论了类似的现象. 需要注意的是, 置换矩阵乘向量的运算可以通过变量的赋值实现, 不会引入误差.

为了更系统的刻画计算机舍入误差对 `fminsearch` 求解 BDQRTIC 问题 ($n = 20$) 的影响, 我们随机产生了 10 个置换矩阵, 并用 `fminsearch` 求解相应问题. 表 2.2 列出了这 10 次计算的结果. 其中, 第 2 至第 6 列为不同计算精度对应的函数值计算次数; 第 7 列为迭代自然终止时的函数值计算次数; 最后一列为各次计算求到的最优值; 最后一行是第 2 至第 11 行数据的相对标

表 2.2: 20 维 BDQRTIC 问题 10 次随机实验结果

| | 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} | 10^{-5} | Natural | f_{best} |
|------|-----------|-----------|-----------|-----------|-----------|---------|-------------------|
| 1 | 2719 | 8948 | 10251 | 11141 | 13455 | 16053 | 5.832041E+01 |
| 2 | 2428 | 4851 | 8701 | 15306 | 17588 | 21585 | 5.832042E+01 |
| 3 | 3541 | 29261 | 29631 | 30976 | 36221 | 39567 | 5.832042E+01 |
| 4 | 1654 | 4381 | 14720 | 15900 | 16753 | 22554 | 5.832041E+01 |
| 5 | 9268 | 29452 | 31151 | 31923 | 32521 | 37716 | 5.832047E+01 |
| 6 | 1828 | 18565 | 19652 | 22327 | 22746 | 26977 | 5.832041E+01 |
| 7 | 3004 | 6549 | 17014 | 24780 | 25400 | 28626 | 5.832041E+01 |
| 8 | 4353 | 6425 | 8288 | 9084 | 10066 | 16469 | 5.832041E+01 |
| 9 | 3267 | 18659 | 20606 | 28658 | 31326 | 39946 | 5.832041E+01 |
| 10 | 3190 | 5651 | 16207 | 17624 | 18976 | 21887 | 5.832041E+01 |
| rstd | 0.61 | 0.75 | 0.45 | 0.39 | 0.38 | 0.33 | 0.00 |

准差 (rstd)⁵. 我们看到, 不论终止条件如何要求, 函数值计算次数都有很强的随机性, 相对标准差都在 0.30 以上. 这说明计算机舍入误差对 `fminsearch` 的影响不能忽略.

需要注意的是, BDQRTIC 并不是一个十分困难的问题, 因为它的目标函数是凸的. $n = 20$ 也不是一个特别高的维数.

结束本节之前, 我们指出, 对计算机舍入误差的敏感性并非 `fminsearch` 独有. 对于基于信赖域的算法, Zhang [187] 给出了一个戏剧性的例子. Zhang [187] 将两个无导数信赖域算法 `SYMB` 与 `ESYMBS` 用于求解以下问题

$$f(x) = \sum_{i=1}^{n-4} [(x_i^2 + 2x_{i+1}^2 + 3x_{i+2}^2 + 4x_{i+3}^2 + 5x_n^2)^2 - 4x_i + 3], \quad x \in \mathbb{R}^n, \quad (2.14)$$

初始点为 $x_0 = (1, \dots, 1)$, 维数 n 取 20. 该问题是 Powell [136] 给出的另一版本的 BDQRTIC, 不妨称之为 BDQRTICP. Zhang [187] 使用 `SYMB` 与 `ESYMBS` 求解 20 维的 BDQRTICP 问题, 两个算法自然终止时花费的函数值计算次数分别为 760 (`SYMB`) 与 4124 (`ESYMBS`); 然后, 改变 BDQRTICP 中变量的顺序, 得到一个新的测试问题, 并使用 `SYMB` 与 `ESYMBS` 求解新的问题, 结果是有意思

⁵一组数据的相对标准差是指其标准差与均值之比, 刻画了这组数据的集中程度.

的: 两个算法自然终止时花费的函数值计算次数分别为 4267 (SYMB) 与 833 (ESYMBS). 因为算法 SYMB 和 ESYMBS 的实现与变量顺序无关, 所以两次计算在理论上完全等价, 而结果却正好相反. 上述戏剧性的结果也完全是由计算机舍入误差导致的. 这一结果也提示我们, 通过简单的记录算法的函数值计算次数来比较不同的无导数算法是不可靠的. 我们需要更可靠的算法评价体系, 这就是 §2.4 要讨论的问题.

§2.4 一种新的评价体系

§2.3 的例子表明, 要想可靠地比较两个无导数优化算法, 必须将计算机舍入误差的影响考虑在内; 另外, 算法对计算机舍入误差的敏感性也应当作为评价算法的一项指标. 基于这两点考虑, 我们给出一个新的评价体系. 根据 §2.3, 有理由把无导数算法求解给定问题的计算开销视为一个随机变量, 因此我们引入统计的方法来观测这一随机变量.

假定算法 s 与变量顺序无关, 我们要考察算法 s 求解问题 p 的函数值计算次数. 设 p 的目标函数为 f , 初始点为 \hat{x} . 随机产生 N 个置换矩阵 P_i ($i = 1, 2, \dots, N$), 令

$$f_i(x) = f(P_i x), \quad \hat{x}_i = P_i^{-1} \hat{x}, \quad i = 1, 2, \dots, N. \quad (2.15)$$

使用算法 s 求解

$$\min_{x \in \mathbb{R}^n} f_i(x), \quad (2.16)$$

记录其函数值计算次数 $\#f_i$. 若算法 s 求解 p 失败则令 $\#f_i = \infty$. 由此, 我们得到一个向量 $\#f = (\#f_1, \dots, \#f_N)$. 计算该向量的均值与标准差:

$$\text{mean}(\#f) = \frac{1}{N} \sum_i^N \#f_i, \quad (2.17)$$

$$\text{std}(\#f) = \sqrt{\frac{1}{N} \sum_i^N [\#f_i - \text{mean}(\#f)]^2}. \quad (2.18)$$

若某 $\#f_i = \infty$ 则置上述统计量为无穷⁶. 当 N 足够大时, $\text{mean}(\#f)$ 就估计了 s 求解 p 的平均函数值计算开销, 而 $\text{std}(\#f)$ 反映了 s 求解 p 时对计算机舍入

⁶这是一种比较严厉的做法. 另一种选择是对 $\text{mean}(\#f)$ 和 $\text{std}(\#f)$ 作出适当惩罚. 我们使用前者.

误差的敏感性. 我们还可以使用相对标准差

$$\text{rstd}(\#f) = \frac{\text{std}(\#f)}{\text{mean}(\#f)} \quad (2.19)$$

作为标准化的敏感性度量⁷. 利用这些统计量, 我们就能相对可靠地反映 s 在 p 上的表现.

现在假定有一组算法 \mathcal{S} 需要评价. 我们考察这组算法的两个指标: 计算开销与对计算机舍入误差的敏感性.

首先考虑计算开销的比较. 如 §2.2 介绍的一样, 我们可以取测试问题集 \mathcal{P} , 然后利用 Performance Profile 与 Data Profile 对算法进行比较. 与 §2.2 不同的是, 我们使用上面定义的 $\text{mean}(\#f)$ 作为算法计算开销的度量 (即 §2.2 中的 $t_{p,s}$). 另外, 给定问题 p , 有两点需要注意:

a.) 为了公平, \mathcal{S} 中所有算法在计算 $\text{mean}(\#f)$ 时应使用相同的一组置换矩阵 P_i ($i = 1, \dots, N$);

b.) 在条件 (2.5) 与条件 (2.7) 中, f^* 应取

$$\min\{\text{算法 } s \text{ 对函数 } f_i \text{ 求到的最优值}, s \in \mathcal{S}, i = 1, \dots, N\}. \quad (2.20)$$

现在考虑算法对计算机舍入误差的敏感性. 类比问题 2.1 和 2.2, 我们提出以下两个问题:

问题 2.4. \mathcal{S} 中哪一算法对计算机舍入误差的敏感性最低 (稳定性最高)?

问题 2.5. \mathcal{S} 中哪一算法对计算机舍入误差的敏感性不会超过最稳定算法的 α 倍 ($\alpha \geq 1$)?

问题 2.5 提示我们, Performance Profile 可以用于敏感性的比较. 我们只要把 (2.1) 中的 $t_{p,s}$ 定义为算法 s 求解问题 p 时的 $\text{std}(\#f)$ 或 $\text{rstd}(\#f)$ 即可; 我们称 $\text{std}(\#f)$ 的 Performance Profile 为算法 s 的 Sensitivity Profile, 记为 σ_s ; 而称 $\text{rstd}(\#f)$ 的 Performance Profile 为算法 s 的 R-Sensitivity Profile, 记为 ρ_s . 它们的值越大意味着算法敏感性越低, 稳定性越高.

⁷需要注意的是, 相对标准差对比较慢的算法更有利.

这样, 通过引入统计的方法, 我们建立了一种评价和比较无导数优化算法的新体系. 新的体系通过 Sensitivity Profile 和 R-Sensitivity Profile 反映算法对计算机舍入误差的敏感性, 对算法计算开销的比较也更可靠. 需要强调的是, 只有参与评价的所有算法均与变量顺序无关时, 上述评价体系才是合理的.

§ 2.5 总结

算法的评价是算法设计中的一个重要问题. 合理的评价方法应该能够指导算法设计和应用领域对算法的选择. Performance Profile 和 Data Profile 提供了算法评价的有效方法. 然而, 现有的评价方法也存在着不足, 即没有将计算机舍入误差对算法的影响考虑在内. § 2.3 的例子很清楚地表明, 这种影响对于无导数优化方法是不可忽略的. 为了更客观地评价无导数优化算法, 我们引入了统计的方法, 建立了一套新的评价体系. 新的体系不但引入 Sensitivity Profile 和 R-Sensitivity Profile 比较不同算法对计算机舍入误差的敏感性, 而且能够更可靠地比较不同算法的计算开销. 在以后各章中, 我们将使用这一体系比较不同的算法.

值得注意的是, 这里比较的敏感性是对于计算机舍入误差而言的. 我们并未引入任何人为噪音. § 2.4 介绍的统计方法也可以应用于比较算法对噪音的稳定性, 这里不再赘述. 需要强调的是, 新的评价体系只有在算法与变量顺序无关时才适用. 另外, 新的评价体系也可以用于一般优化算法的比较.

§ 2.6 附录

本节给出 § 2.3 中对应于表 2.1 与图 2.1 的程序源代码. 给出这些源代码的目的有两个: 第一, 证明表 2.1 与图 2.1 中展示的不稳定性的确是由计算机舍入误差引起的; 第二, 演示如何在测试中引入置换矩阵. 我们的程序使用 MATLAB 编写.

测试中, 我们使用 P_1 与 P_2 改变 BDQRTIC 问题的变量顺序, 然后调用 `fminsearch` 函数求解产生的问题. 下面是源代码. 其中, 数组 `perm1` 与 `perm2` 是 $1, \dots, 20$ 的重排列, 用于表示置换矩阵 P_1 与 P_2 .

下面是在我们的测试环境中运行 `testpermute.m` 输出的结果.

```
fmin1 =  
    58.320655411509634  
output1 =  
    iterations: 5218  
    funcCount: 6617  
    algorithm: 'Nelder-Mead simplex direct search'  
    message: [1x194 char]  
fmin2 =  
    58.320413830188400  
output2 =  
    iterations: 50269  
    funcCount: 62845  
    algorithm: 'Nelder-Mead simplex direct search'  
    message: [1x194 char]
```

要得到表 2.1 与图 2.1 展示的数据, 还需要在 `testpermute.m` 与 `bdqrtic.m` 中引入全局变量以记录迭代过程中计算的函数值, 我们不再赘述其中的细节.

第三章 无导数优化中的最小 Frobenius 范数插值与对称 Broyden 修正

§ 3.1 引言

第一章提到, 二次插值模型在基于信赖域的无导数算法中有广泛应用. 比如 Winfield [174, 175] 的算法、DFO [37, 38, 36]、UOBYQA [131]、NEWUOA [138, 142] 和 MNH [172] 等方法都使用以下插值问题

$$Q_k(y) = f(y), \quad y \in \mathcal{I}_k \quad (3.1)$$

确定第 k 步迭代的模型 Q_k , 其中 Q_k 为二次函数, \mathcal{I}_k 为第 k 步迭代的插值点集.

问题 (3.1) 是关于 Q_k 系数的一个线性方程组. 因为 \mathbb{R}^n 中的一个二次函数有 $(n+1)(n+2)/2$ 个自由度, 所以要想通过问题 (3.1) 确定唯一的一个 Q_k , 至少需要 $(n+1)(n+2)/2$ 个函数值信息. 即使 n 不是特别大, 这也是无法接受的. 所以必须考虑欠定 (underdetermined) 的二次插值, 即 $|\mathcal{I}_k| < (n+1)(n+2)/2$ 的情形. 当 $|\mathcal{I}_k| < (n+1)(n+2)/2$ 时, 问题 (3.1) 是一个欠定的线性方程组, 故一般有无穷多个二次函数满足该插值系统. 如何从中确定唯一的模型? 一种有效的做法是在插值条件 (3.1) 下极小化一个泛函, 从而得到唯一解. 也就是说, 通过求解

$$\min_{Q_k \in \mathcal{Q}} \mathcal{F}_k(Q_k) \quad (3.2)$$

$$\text{s.t. } Q_k(y) = f(y), \quad y \in \mathcal{I}_k \quad (3.3)$$

来确定模型. 其中, \mathcal{Q} 是 \mathbb{R}^n 中所有二次函数构成的空间, \mathcal{F}_k 是 \mathcal{Q} 上的一个泛函.

DFO 算法与 NEWUOA 算法就是通过上述方法建立模型的. 具体地说, DFO 使用最小 Frobenius 范数插值建立模型, 即定义 Q_k 为插值问题

$$\min_{Q_k \in \mathcal{Q}} \|\nabla^2 Q_k\|_F^2 \quad (3.4)$$

$$\text{s.t. } Q_k(y) = f(y), \quad y \in \mathcal{I}_k \quad (3.5)$$

的解, 这类似于寻求欠定线性系统的最小 2-范数解. 我们把 (3.4)–(3.5) 定义的模型称为最小 Frobenius 范数模型. 与 DFO 不同, NEWUOA 的模型由第 $k-1$ 步迭代的模型通过最小 Frobenius 范数修正得到, 即 Q_k 被定义为插值问题

$$\min_{Q_k \in \mathcal{Q}} \|\nabla^2 Q_k - \nabla^2 Q_{k-1}\|_F^2 \quad (3.6)$$

$$\text{s.t. } Q_k(y) = f(y), \quad y \in \mathcal{I}_k \quad (3.7)$$

的解. 其中, Q_{k-1} 是第 $k-1$ 步迭代的模型, $Q_0 = 0$. NEWUOA 构造模型的方法 (3.6)–(3.7) 受到了拟牛顿法中最小变化修正 [53] 的启发, 是不使用导数信息的对称 Broyden 修正¹.

另外, NEWUOA 还包含了一种重新开始技术: 当检测到对称 Broyden 修正 (3.6)–(3.7) 决定的二次函数不适合作为 f 的模型时, 算法放弃对 Q_{k-1} 的修正, 在第 k 步使用最小 Frobenius 范数插值 (3.4)–(3.5) 定义模型 Q_k . 这种重新开始 NEWUOA 的迭代中发生的次数比较少, 却能很大程度上提高算法效率.

Conn, Scheinberg 和 Vicente [42] 认为, (3.4)–(3.5) 与 (3.6)–(3.7) 定义的模型² 已经被证明是目前基于插值模型和信赖域的无导数方法中最成功的二阶模型³.

本章研究最小 Frobenius 范数插值 (3.4)–(3.5) 与对称 Broyden 修正 (3.6)–(3.7). §3.2 证明这两种策略在某些情况下的等价性, 指出在 DFO 算法框架下, 两种方法定义的模型有时是相同的. §3.3 对 NEWUOA 的重开始技术做一个讨论, 并提出一种与之相关的改进 NEWUOA 源代码的方法, 这种方法只要修改现有版本的 NEWUOA 代码中的一行⁴ 即可明显提高算法的性能. §3.4 在 NEWUOA 的算法框架下对最小 Frobenius 范数模型和对称 Broyden 修正得到的模型做一个比较; 我们发现, 当求解精度不高时, 最小 Frobenius 范数模型更有优势, 而越到迭代后期对称 Broyden 修正得到的模型越有优势. §3.5 是本章讨论的一个总结. §3.6 是关于对称 Broyden 修正的历史注记.

¹请参考 §3.6 中的注记.

²与我们的术语不同, Conn, Scheinberg 和 Vicente [42] 将两种策略定义的模型统称为最小 Frobenius 范数模型 (minimum Frobenius norm models). 为了强调两种策略的区别以及第二种策略的历史渊源, 我们对两种模型使用不同的术语. 请参考 §3.6 中的注记.

³“Minimum Frobenius norm models have so far proven to be the most successful second-order models in interpolation-based trust-region-methods” [42] (5.4 节).

⁴事实上, 只需删除四个字母. 详情请参考 §3.3.2.

§ 3.2 最小 Frobenius 范数插值与对称 Broyden 修正的关系

可以看到, 对称 Broyden 修正是关于 $Q_k - Q_{k-1}$ 的最小 Frobenius 范数插值. 事实上, 二者的关系不止于此. 本节要证明, 这两种建立模型的策略在某些情况下是等价的.

在 NEWUOA 与 DFO 中, 插值点集的更新策略是不同的. 一个突出的表现是, NEWUOA 始终保持插值点数目不变, 而 DFO 则使用尽量多的插值点⁵. 在每次迭代完成后, NEWUOA 或者保持插值点集不变, 或者用新的迭代点取代插值点集中的一个点. 而在 DFO 中, 只要不破坏插值点集的适定性, 算法会把新的迭代点加入插值点集而不去掉任何已有的点⁶, 这导致新的插值点集包含旧的插值点集. 现在我们指出, 在这种情况下, 最小 Frobenius 范数插值 (3.4)–(3.5) 与对称 Broyden 修正 (3.6)–(3.7) 定义的模型是相同的. 也就是说, 我们有以下定理.

定理 3.1. 设 Q_{k-1} 为最小 Frobenius 范数模型, 且 $\mathcal{I}_k \supset \mathcal{I}_{k-1}$, 则插值问题 (3.4)–(3.5) 与 (3.6)–(3.7) 等价.

定理 3.1 在几何上是直观的. 我们定义 \mathcal{Q} 上的半内积 (semi-inner product [44])

$$\langle P, Q \rangle = \text{Tr} [(\nabla^2 P)(\nabla^2 Q)], \quad P, Q \in \mathcal{Q}, \quad (3.8)$$

则插值问题 (3.4)–(3.5) 与 (3.6)–(3.7) 为该半内积下的投影问题. 记

$$\mathcal{H}_k = \{Q \in \mathcal{Q} : Q(y) = 0, y \in \mathcal{I}_k\}, \quad k \geq 1. \quad (3.9)$$

因 $\mathcal{I}_k \supset \mathcal{I}_{k-1}$, 故 $\mathcal{H}_k \subset \mathcal{H}_{k-1}$. Q_{k-1} 为最小 Frobenius 范数模型意味着 $Q_{k-1} \perp \mathcal{H}_{k-1}$, 从而 $Q_{k-1} \perp \mathcal{H}_k$. 因此, 对任何 $Q \in \mathcal{Q}$, $Q \perp \mathcal{H}_k$ 当且仅当 $Q - Q_{k-1} \perp \mathcal{H}_k$, 从而插值问题 (3.4)–(3.5) 与 (3.6)–(3.7) 等价.

由此可见, 虽然 DFO 算法基于最小 Frobenius 范数模型, 但在某些情形下, 它本质上也采用了对称 Broyden 修正. 有意思的是, 虽然 NEWUOA 算法基于对称 Broyden 修正, 但它有时也采用最小 Frobenius 范数模型, 这就是本章引言中提到的重新开始技术. 下一节我们对这种技术做一个研究.

⁵ “Naturally, given a set of points with known function values, we would like to interpolate as many of them as possible, since we would like to use as much of the provided information as possible” [38].

⁶ “If we can add the new point to the interpolation set without dropping any other point from the set, we do so” [38].

§ 3.3 NEWUOA 的重开始技术以及 NEWUOA 源代码的一个改进

§ 3.3.1 NEWUOA 的重开始技术

本章引言中提到, 当探测到对称 Broyden 修正定义的模型不适合作为第 k 步迭代的模型时, NEWUOA 会放弃对 Q_{k-1} 的修正, 将 Q_k 定义为最小 Frobenius 范数模型. 这是一种重开始技术, 因为 Q_1 就是最小 Frobenius 范数模型. 在计算中, 这种重开始发生的次数不多, 却会大大提高算法的效率. 重开始的时机十分重要, 盲目重开始会丧失模型在之前迭代中积累的信息, 与最小变化修正的思想相悖. 那么, 为什么要重开始? 何时重开始?

在 NEWUOA 的计算中, 当模型系数过大时, 算法效率会降低. 故 NEWUOA 试图自动识别系数过大的模型, 并代之以最小 Frobenius 范数模型. 为此, 在每步信赖域迭代并更新插值点集后, Powell [138] 检查

$$\text{RATIO} \leq 0.01, \text{ 且 } \|\nabla Q_{\text{int}}(x_0)\|_2 \leq 0.1\|\nabla Q(x_0)\|_2 \quad (3.10)$$

是否成立, 其中 RATIO 为信赖域迭代的下降比 (即算法 1.3 中的 ρ_k), Q_{int} 为新插值点集上的最小 Frobenius 范数模型, Q 为对称 Broyden 修正定义的模型, x_0 为插值基准点 (请参考 Powell [138]); 若连续三次检查 (3.10) 都成立, 则定义下一个模型为 Q_{int} , 也就是重开始.

条件 (3.10) 中, $\text{RATIO} \leq 0.01$ 意味着上一个模型下降比不佳. 一方面, 这使我们对该模型经最小变化修正得到的模型有负面预期; 另一方面, 如果上一个模型由对称 Broyden 修正得到, 这也说明上一次修正并不成功. $\|\nabla Q_{\text{int}}(x_0)\|_2 \leq 0.1\|\nabla Q(x_0)\|_2$ 意味着 Q 的一阶项系数远大于 Q_{int} . 不仅如此, 根据 Q_{int} 的定义, Q 的二阶项系数也会大于 Q_{int} . 这说明, 如果当前仍选择对称 Broyden 修正, 将会导致一个系数比较大的模型. 基于这两点, 可能用最小范数模型取代对称 Broyden 修正是合理的. 尽管如此, Powell [138] 仍倾向于尽量使用对称 Broyden 修正, 故只有在上述情况连续出现三次时才放弃修正, 选择重开始.

上述重开始技术的引入大大提高了 NEWUOA 的效率, 这方面的例子可以参考 Powell [138] 的数值结果. 关于重开始的更多细节, 也请参考 Powell [138].

§3.3.2 NEWUOA 源代码的一个改进

本小节的讨论基于 Powell 教授 2010 年 11 月 1 日提供给本文作者的 NEWUOA 算法的 Fortran 77 源代码. 我们发现, 在 NEWUOA 的源代码中, 重开始的准则与 Powell [138] 的叙述有所不同. 其中的一个差异在于对下降比的要求. 源代码的重开始准则对下降比的要求是

$$|\text{RATIO}| \leq 0.01. \quad (3.11)$$

上述条件体现在 NEWUOA 源代码的第 446 行. 我们的问题是, 条件 $|\text{RATIO}| \leq 0.01$ 与条件 $\text{RATIO} \leq 0.01$ 哪一个更优? 这看似是一个微不足道的问题. 然而, 我们将用实验证明, 二者的带来的差异是十分显著的.

我们把 NEWUOA 源代码的第 446 行

```
IF (DABS(RATIO) .GT. 1.0D-2) THEN
```

修改为

```
IF ((RATIO) .GT. 1.0D-2) THEN
```

得到一个新版本的 NEWUOA 代码, 称之为 NEWUOAm (NEWUOA modified). 然后我们采用第二章 §2.4 建立的评价体系来比较 NEWUOAm 与原始的 NEWUOA 代码. 我们的测试问题集是 Toint [162]、CUTEr [67]、Powell [138] 以及 Lukšan, Matonoha 和 Vlček [100] 中的 50 个可变维数的无约束优化问题, 问题名称在表 3.1 中给出. 其中, TOINTRIC 取自 Toint [162], ARWHEAD、CHROSEN、PENALTY1、PENALTY2、SPHRPTS 和 VARDIM 取自 Powell [138], DIXMAANM、DIXMAANN、DIXMAANO、DIXMAANP 和 FLETGBV3 取自 Lukšan, Matonoha 和 Vlček [100], 其余问题取自 CUTEr [67]. ARGLINA、ARGLINB 和 ARGLINC 问题中参数 m 设为 $2n$. 问题目标函数与初始点请参阅相应文献. 对于每一个测试问题, 我们对 20、24、28、32、36、40 等 6 个维数进行了求解⁷. 每一次求解都按照 §2.4 做了 10 次随机置换. 我们统计了求解精度 (即条件 (2.5) 中的 τ) 为 10^{-i} ($i = 2, 4, 6, 8, 10$) 时的函数值计算次数, 也统计了迭代自然终止时的函数值计算次数 (条件 (2.7) 中取⁸ $\varepsilon = 10^{-10}$).

⁷测试中, 我们实际进行了 $2 \times 10 \times 6 \times 50 = 6000$ 次求解. 由于无导数优化方法效率较低, 所以实验过程十分耗时. 正因为如此, 我们没有测试较高维数的问题. 第五章将专门讨论高维问题的求解.

⁸我们对条件 (2.7) 中参数 ε 的设置十分严苛. 这是因为, 原始版本的 NEWUOA 自然终止时求得的解精度大都很高, 100 维以内的问题一般在 10^{-10} 以上; 如果修改后的版本能减少函数值计算次数, 但代价是求解精度的降低, 我们认为这种修改是不可接受的, 应当视为失败. 这与比较两个完全不同的算法时所采用的标准是不同的.

表 3.1: 测试问题名称

| | | | | |
|----------|----------|----------|-----------|----------|
| ARGLINA | ARGLINB | ARGLINC | ARWHEAD | BDQRTIC |
| BROWNAL | BROYDN3D | BRYBND | CHROSEN | COSINE |
| CRAGGLVY | CURLY10 | CURLY20 | CURLY30 | DIXMAANE |
| DIXMAANF | DIXMAANG | DIXMAANH | DIXMAANI | DIXMAANJ |
| DIXMAANK | DIXMAANL | DIXMAANM | DIXMAANN | DIXMAANO |
| DIXMAANP | DQRTIC | EDENSCH | EG2 | ENGVAL1 |
| FLETGBV2 | FLETGBV3 | FREUROTH | GENBROWN | GENHUMPS |
| INTEGREQ | LIARWHD | NONCVXUN | PENALTY1 | PENALTY2 |
| POWER | SBRYBND | SCHMVETT | SCOSINE | SPARSINE |
| SPARSQUR | SPHRPTS | TOINTGSS | TOINTTRIG | VARDIM |

NEWUOA 的代码中包含三个用户可选参数, 即每个模型所用插值点的个数 NPT、初始信赖域半径 RHOBEQ 以及终止信赖域半径 RHOEND. 在实验中, NPT 按 Powell [138] 的推荐取 $2n + 1$, n 为问题维数; RHOEND 取 NEWUOA 代码预设的 10^{-6} ; RHOBEQ 的取法分两种情况: Powell [138] 的测试问题按文中给出的方式选取, 其他问题取 1. NEWUOA 使用了两个信赖域半径, 当较小的信赖域半径达到 RHOEND 时迭代终止. 详情请参考 Powell [138]. NEWUOAm 的代码仅仅修改了 NEWUOA 代码的第 446 行, 其余部分与 NEWUOA 完全相同. 实验中, 我们设置最大函数值计算次数为 50000.

我们的实验平台是 Dell PC (OPTIPLEX 755), 系统是 Ubuntu 9.04 (Linux 2.6.38-15-generic), 编译器是 GCC 4.3.3 中的 gfortran.

图 3.1 至图 3.6 给出了 NEWUOA 与 NEWUOAm 在满足不同精度要求以及迭代自然终止时的 Performance Profile、Data Profile、Sensitivity Profile 与 R-Sensitivity Profile. 图线越高说明算法表现越好. 四种 Profile 的定义请参考第二章.

可以看到, 无论终止条件如何, NEWUOAm 的 Performance Profile、Data Profile、Sensitivity Profile 和 R-Sensitivity Profile 总是高于 NEWUOA. 这说明, 在我们的实验中, NEWUOAm 无论在计算开销还是对计算机舍入误差的敏感性方面都优于 NEWUOA. 特别值得注意的是, 每一种终止条件下, NEWUOAm 的 Performance Profile 都是几乎水平的. 这说明, 对实验中的几乎所有问题, 只要

NEWUOAm 求解成功, 它所花费的函数值计算次数就不比 NEWUOA 多.

结论是显而易见的: 在我们的测试中, NEWUOAm 的表现优于原始版本的 NEWUOA 代码. 也就是说, 我们仅仅改动了 NEWUOA 源代码的一行 (删掉了 DABS 四个字母), 就得到了一个改进版本的代码. 这从一个侧面反映了在 NEWUOA 中选择合适的时机重新开始的重要性.

基于以上, 我们建议用 NEWUOAm 代码代替原始版本的 NEWUOA 代码. 今后, 本文中提到的 NEWUOA 代码均指 NEWUOAm 版本的代码, 原始版本的代码不再使用. 我们希望在我们的算法与 NEWUOA 进行比较时, NEWUOA 能发挥最大的潜力. 这样的比较才是有意义的.

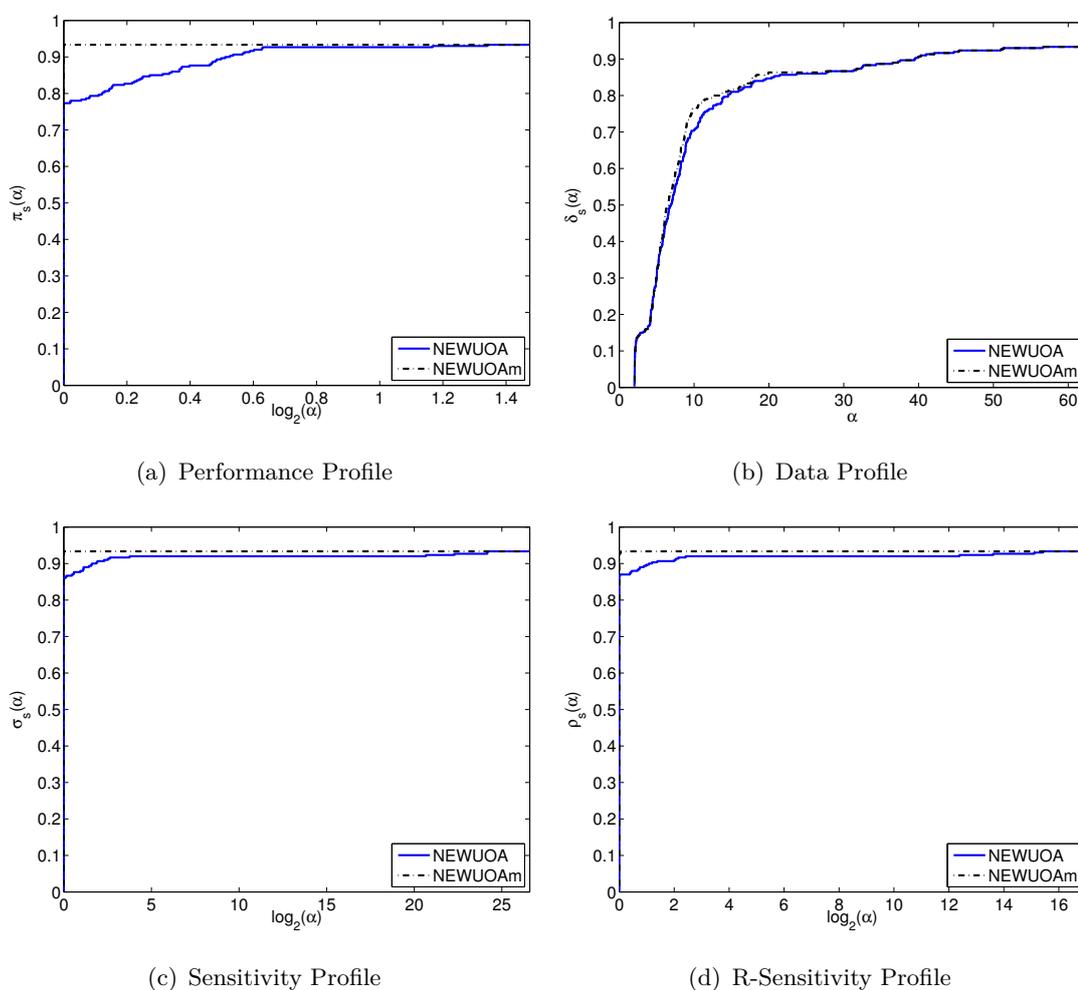
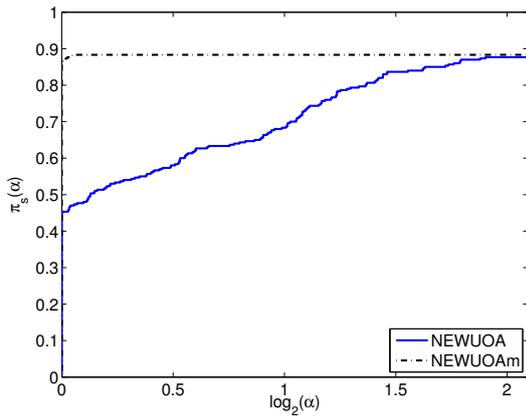
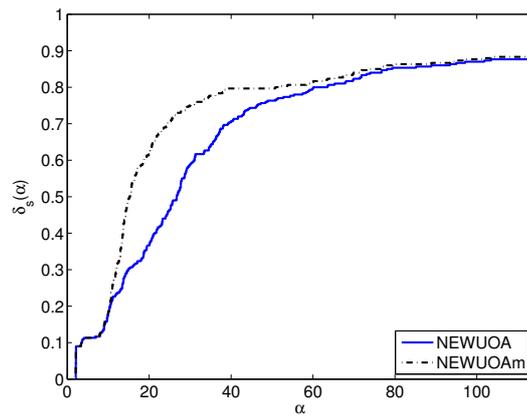


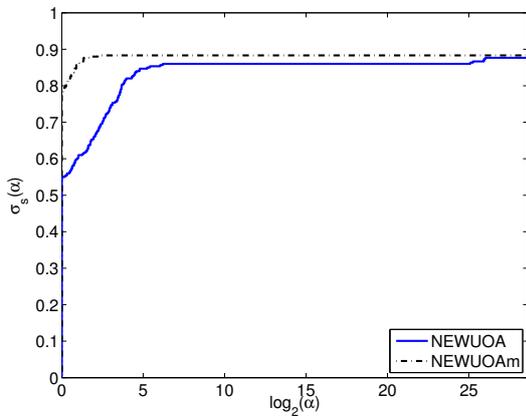
图 3.1: NEWUOA 与 NEWUOAm 的数值表现 (精度 $\tau = 10^{-2}$)



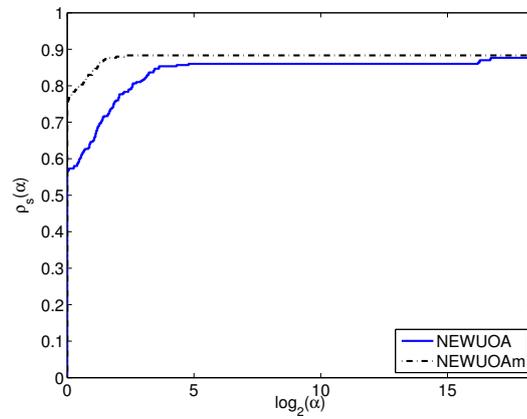
(a) Performance Profile



(b) Data Profile



(c) Sensitivity Profile



(d) R-Sensitivity Profile

图 3.2: NEWUOA 与 NEWUOA_m 的数值表现 (精度 $\tau = 10^{-4}$)

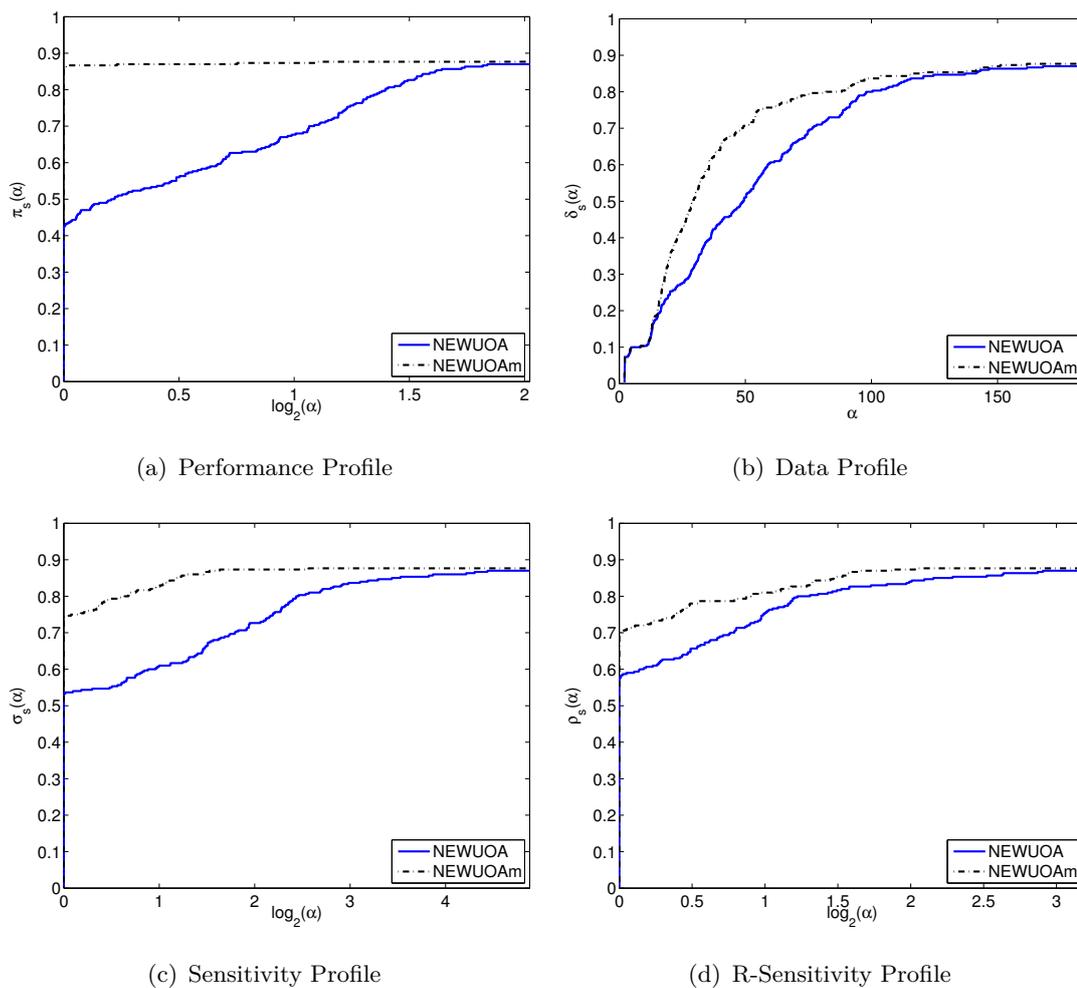


图 3.3: NEWUOA 与 NEWUOA_m 的数值表现 (精度 $\tau = 10^{-6}$)

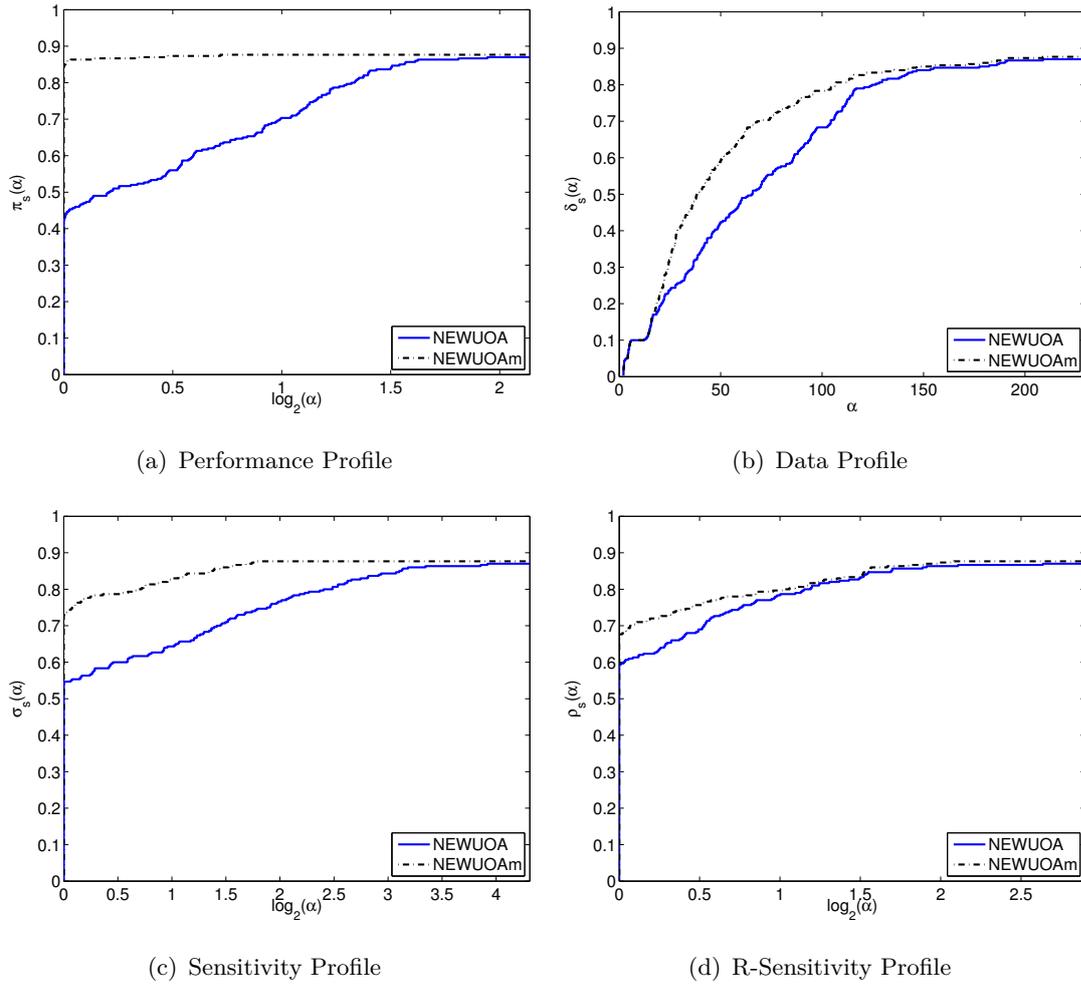


图 3.4: NEWUOA 与 NEWUOA_m 的数值表现 (精度 $\tau = 10^{-8}$)

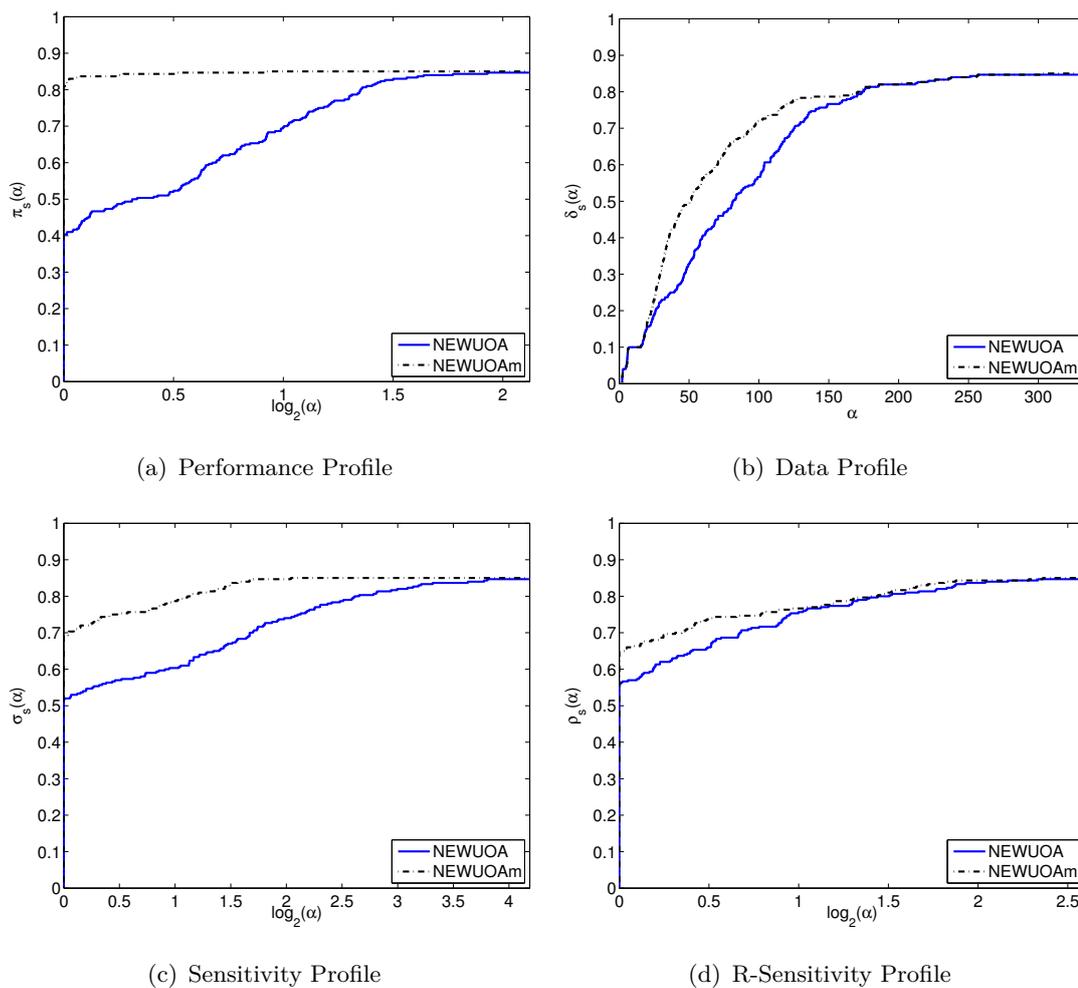
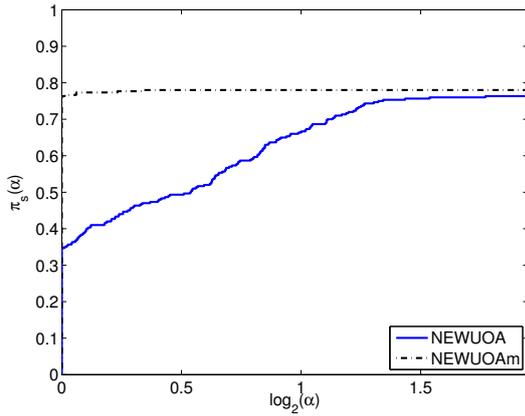
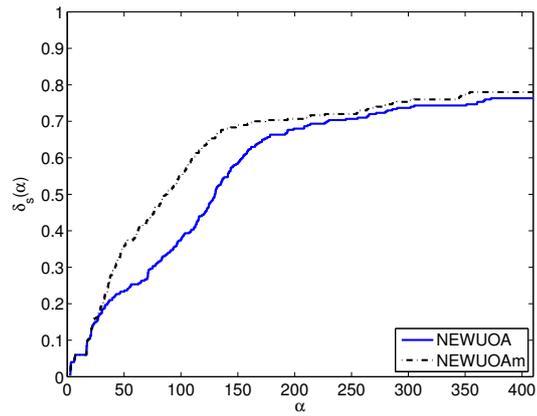


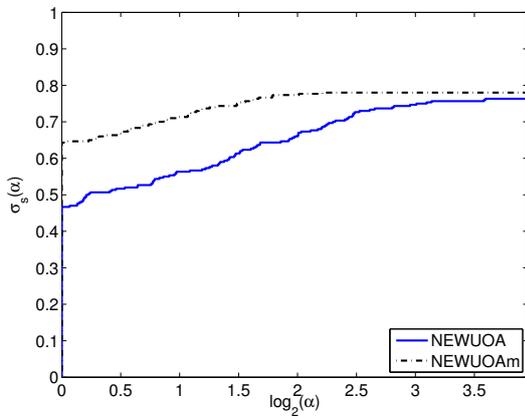
图 3.5: NEWUOA 与 NEWUOA_m 的数值表现 (精度 $\tau = 10^{-10}$)



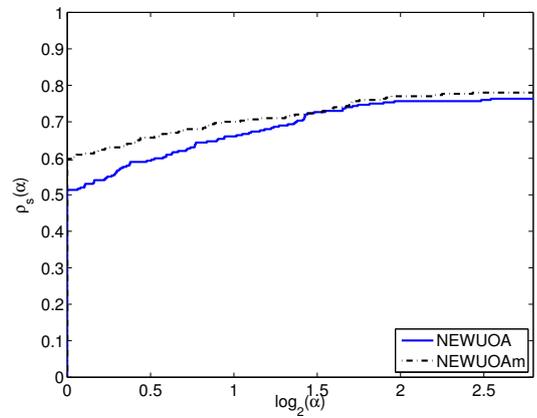
(a) Performance Profile



(b) Data Profile



(c) Sensitivity Profile



(d) R-Sensitivity Profile

图 3.6: NEWUOA 与 NEWUOA_m 的数值表现 (迭代自然终止)

§ 3.4 最小 Frobenius 范数插值与对称 Broyden 修正的比较

既然最小 Frobenius 范数模型与对称 Broyden 修正得到的模型“已经被证明是目前基于插值模型和信赖域的无导数方法中最成功的二阶模型”，那么一个自然的问题是：这两种模型哪个更好？本节通过数值实验对这两种模型的实际计算效果做一个比较。

我们的实验在 NEWUOA 的算法框架下进行。我们定义一个算法 LFN，该算法每一步都使用最小 Frobenius 范数模型，除此之外与 NEWUOA 没有任何区别。因为 NEWUOA 包含了 § 3.1 中介绍的重开始技术，所以 LFN 算法等价于每步都重新开始的新 UOA 算法。故我们可以很容易的由 NEWUOA 的代码得到 LFN 的代码：只要修改 NEWUOA 代码中的重开始准则，令算法每一步都重新开始即可。

我们采用第二章 § 2.4 建立的评价体系来比较这两种算法。我们的测试问题集是 § 3.3.2 中使用的 50 个可变维数的无约束优化问题。对于每一个测试问题，我们对 20、24、28、32、36、40 等 6 个维数进行了求解。每一次求解都按照 § 2.4 做了 10 次随机置换。我们统计了求解精度（即条件 (2.5) 中的 τ ）为 10^{-i} ($i = 1, \dots, 10$) 时的函数值计算次数，也统计了迭代自然终止时的函数值计算次数（条件 (2.7) 中取 $\varepsilon = 10^{-10}$ ）。

NEWUOA 的代码中用户可选参数 NPT、RHOBEQ 和 RHOEND 的选取与 § 3.3.2 相同。LFN 算法的代码仅仅修改了 NEWUOA 代码的重开始准则，其余部分与 NEWUOA 完全相同。实验中，我们设置最大函数值计算次数为 50000。

我们的实验环境与 § 3.3.2 相同。

图 3.7 至图 3.17 给出了 NEWUOA 算法与 LFN 算法在满足不同精度要求以及迭代自然终止时的 Performance Profile、Data Profile、Sensitivity Profile 与 R-Sensitivity Profile。图线越高说明算法表现越好。四种 Profile 的定义请参考第二章。

实验结果表明，NEWUOA 算法与 LFN 算法的对比很大程度上依赖于求解精度的要求。我们先来比较算法使用的函数值计算次数，这体现在 Performance Profile 和 Data Profile 中。可以看出，当精度为 10^{-1} 时，两种算法的表现相当；精度为 10^{-2} 与 10^{-3} 时，LFN 算法的表现明显优于 NEWUOA 算法；精度为 10^{-4} 时，LFN 算法略优于 NEWUOA 算法，但区别不大；当求解精度不低于 10^{-5} 以及算法自然终止时，NEWUOA 算法的表现明显优于 LFN 算法。

我们再来比较算法对计算机舍入误差的敏感性，这体现在 Sensitivity

Profile 和 R-Sensitivity Profile 中. 可以看出, 当精度不高于 10^{-5} 时, LFN 算法的表现优于 NEWUOA 算法; 当精度不低于 10^{-6} 以及算法自然终止时, 在 Sensitivity Profile 的意义下 NEWUOA 算法的表现优于 LFN 算法, 而在 R-Sensitivity Profile 的意义下, LFN 算法的表现并不逊于 NEWUOA 算法 (LFN 算法的 R-Sensitivity Profile 最终低于 NEWUOA 算法, 是因为后者求解成功率更高). Sensitivity Profile 和 R-Sensitivity Profile 的不同是因为 R-Sensitivity Profile 是一个相对于函数值计算次数的比值, 因而它对较慢的算法更有利.

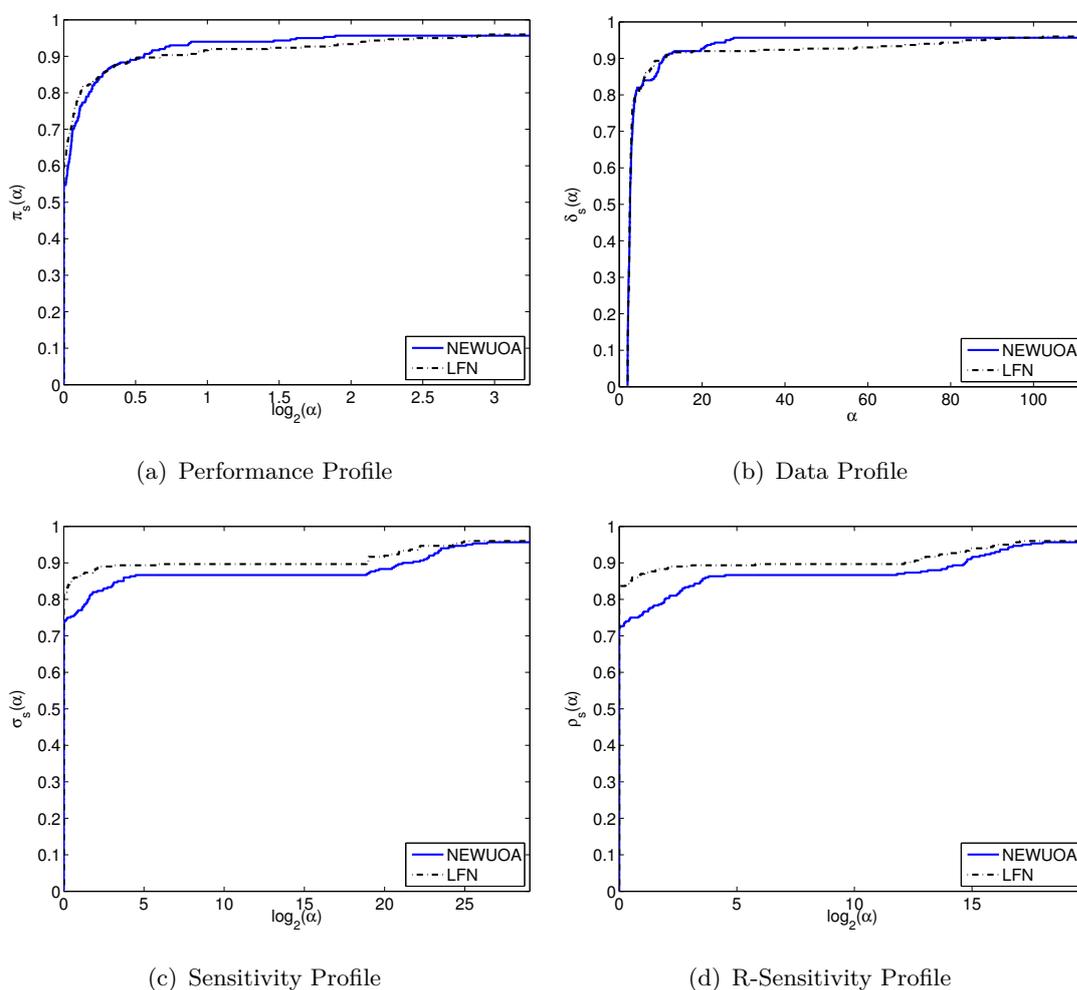


图 3.7: NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-1}$)

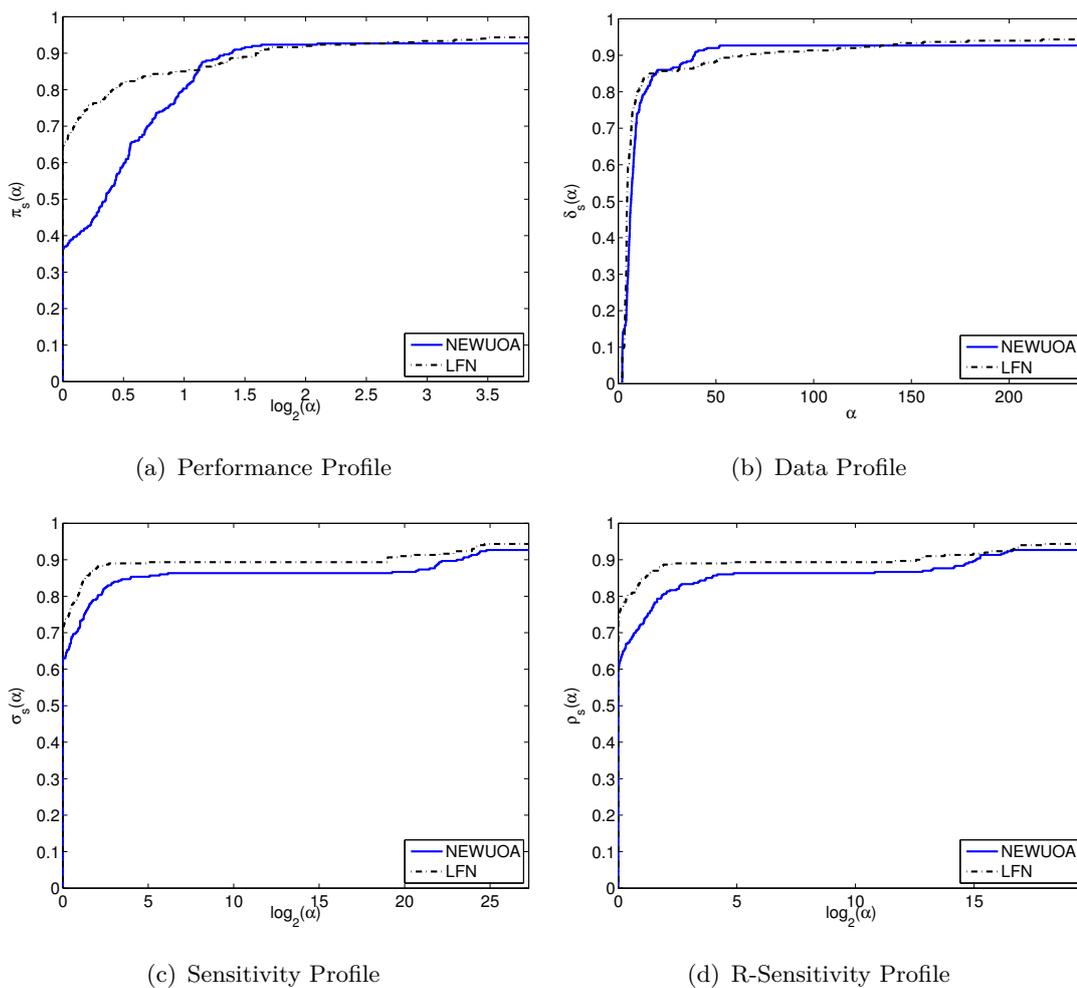
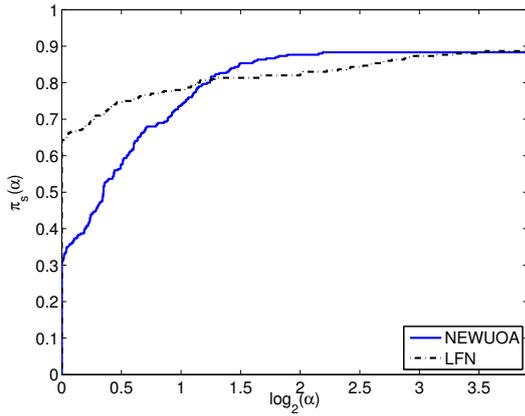
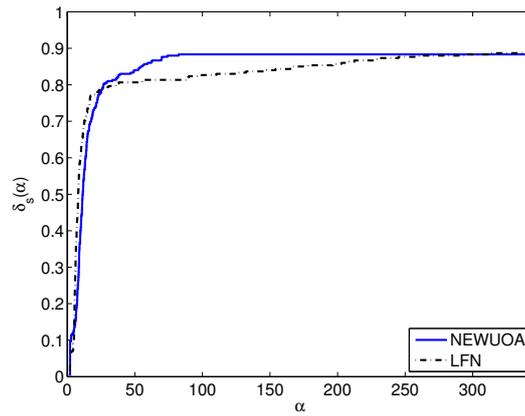


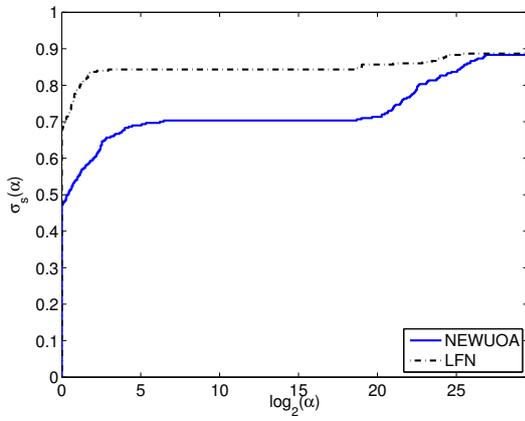
图 3.8: NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-2}$)



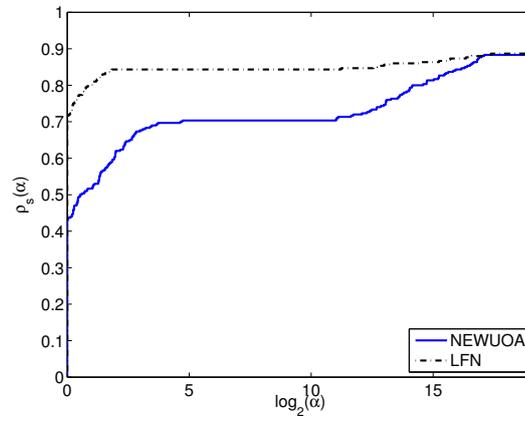
(a) Performance Profile



(b) Data Profile



(c) Sensitivity Profile



(d) R-Sensitivity Profile

图 3.9: NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-3}$)

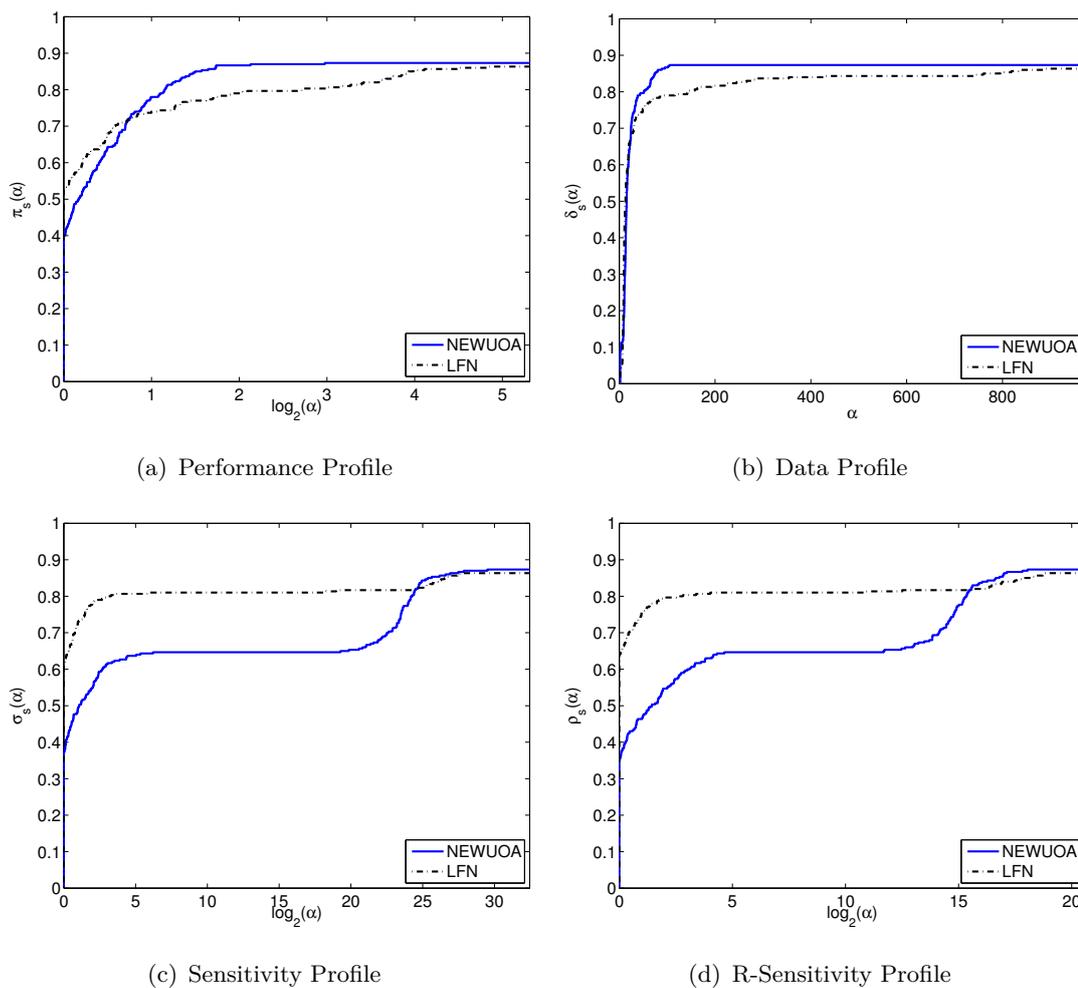
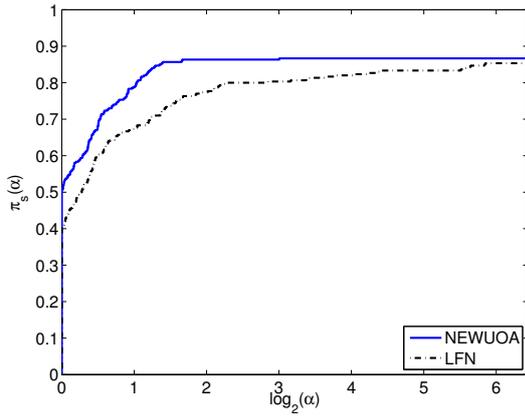
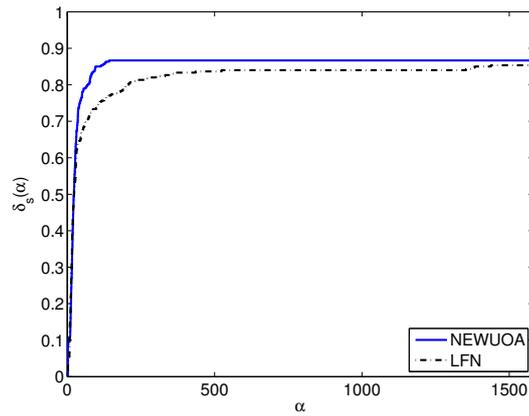


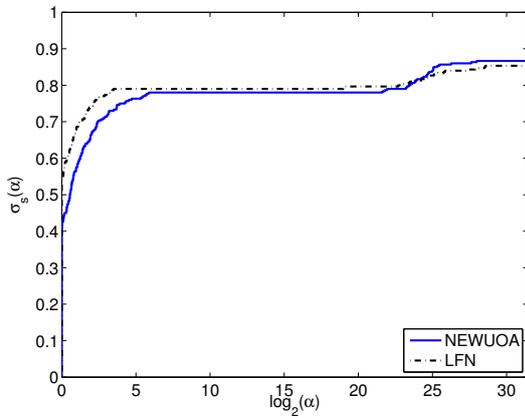
图 3.10: NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-4}$)



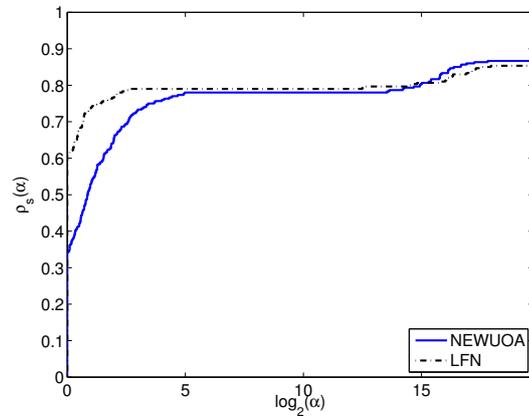
(a) Performance Profile



(b) Data Profile



(c) Sensitivity Profile



(d) R-Sensitivity Profile

图 3.11: NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-5}$)

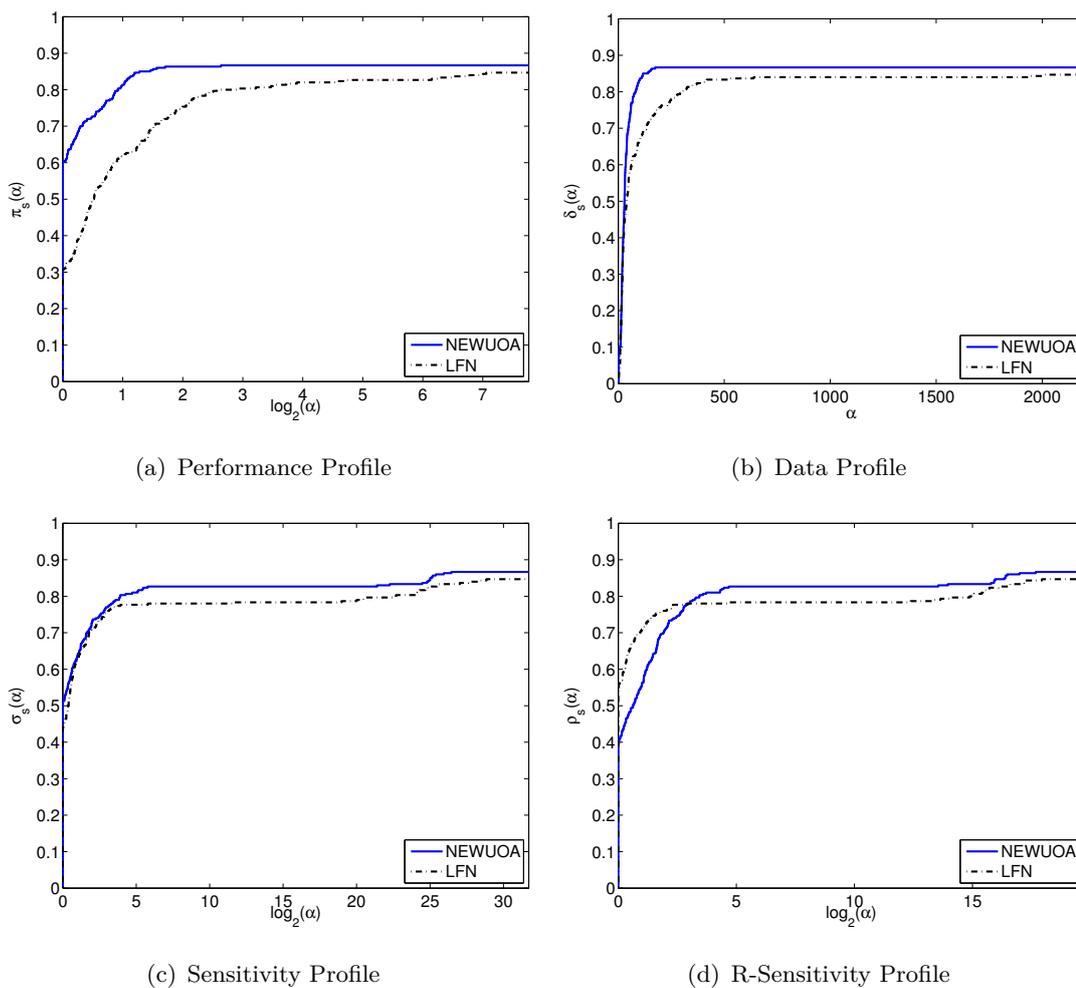
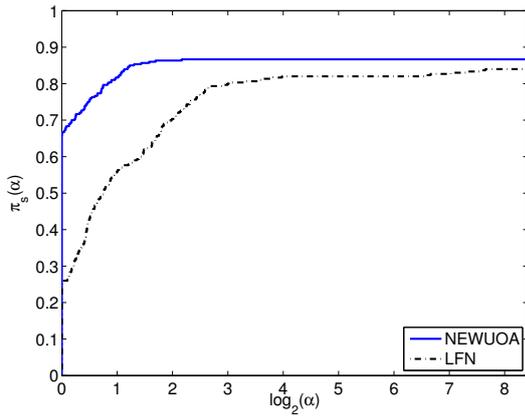
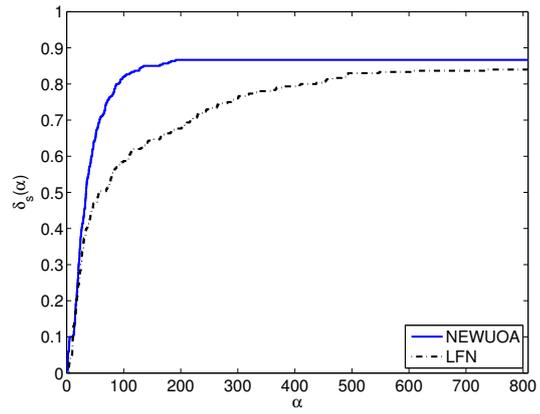


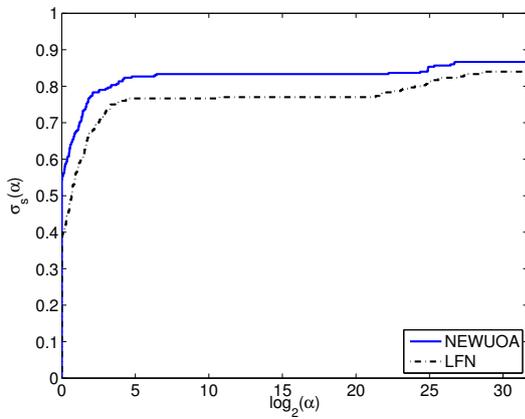
图 3.12: NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-6}$)



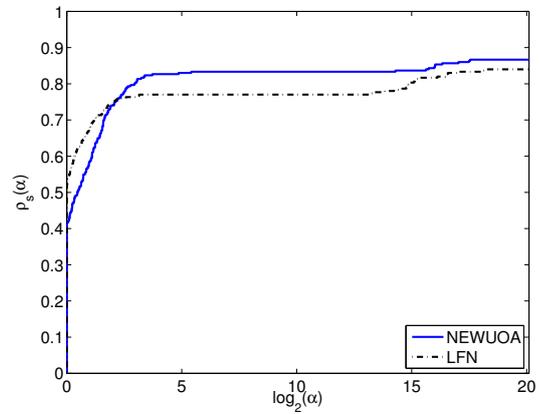
(a) Performance Profile



(b) Data Profile



(c) Sensitivity Profile



(d) R-Sensitivity Profile

图 3.13: NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-7}$)

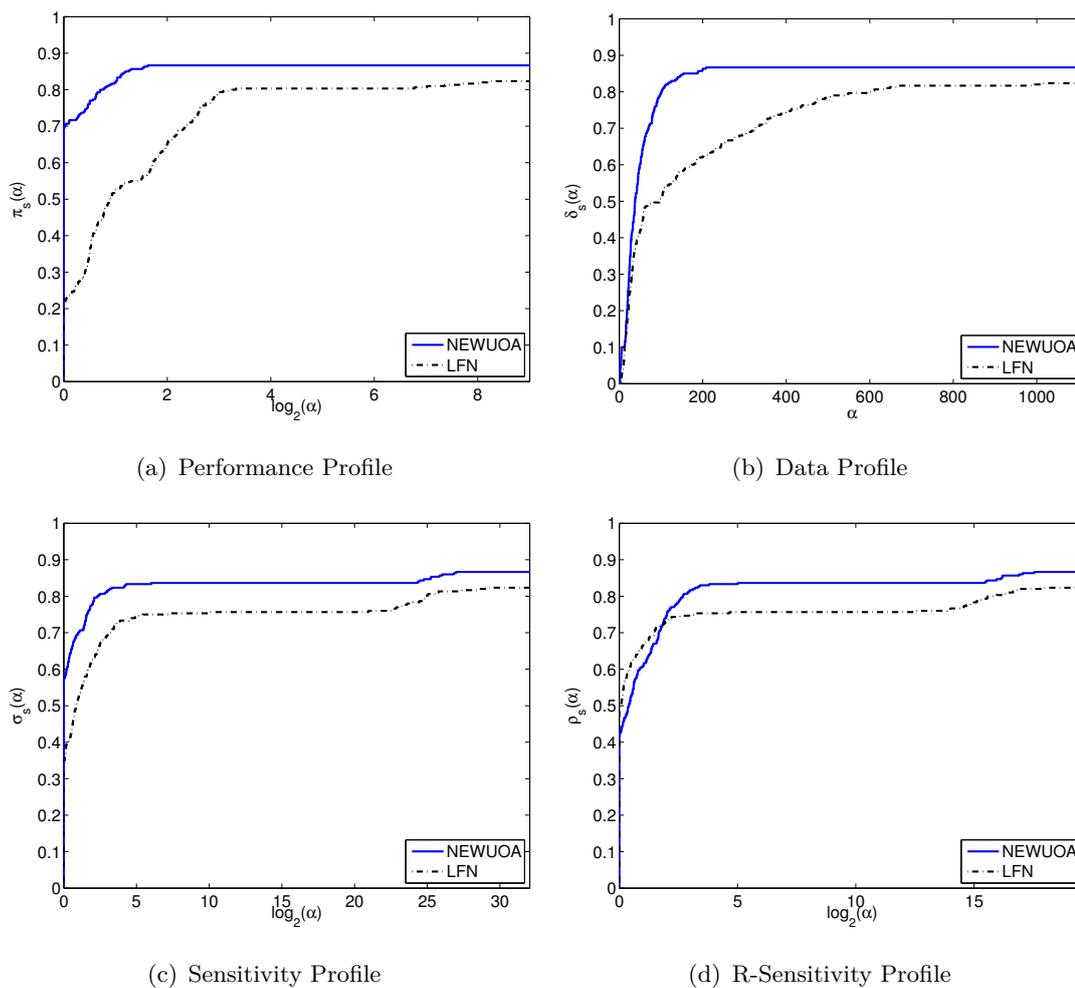


图 3.14: NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-8}$)

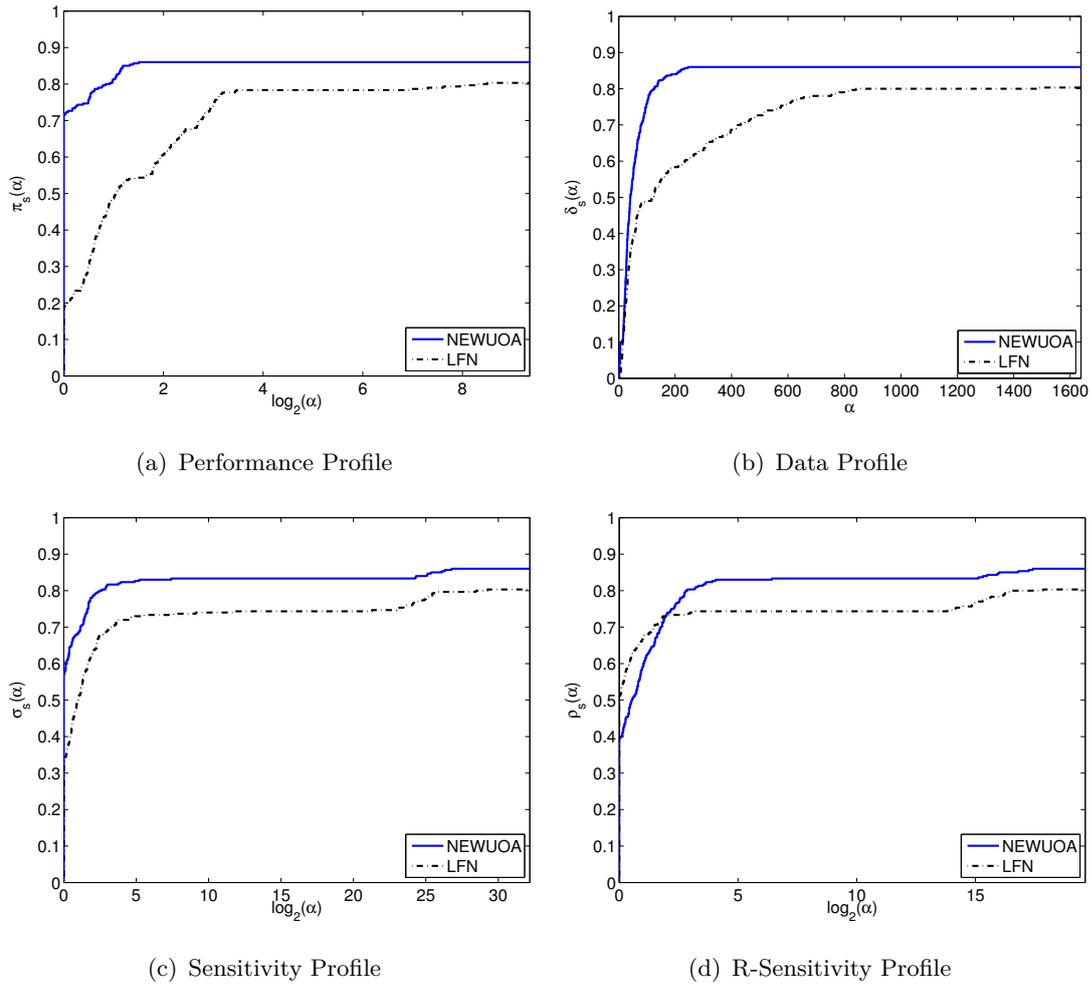


图 3.15: NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-9}$)

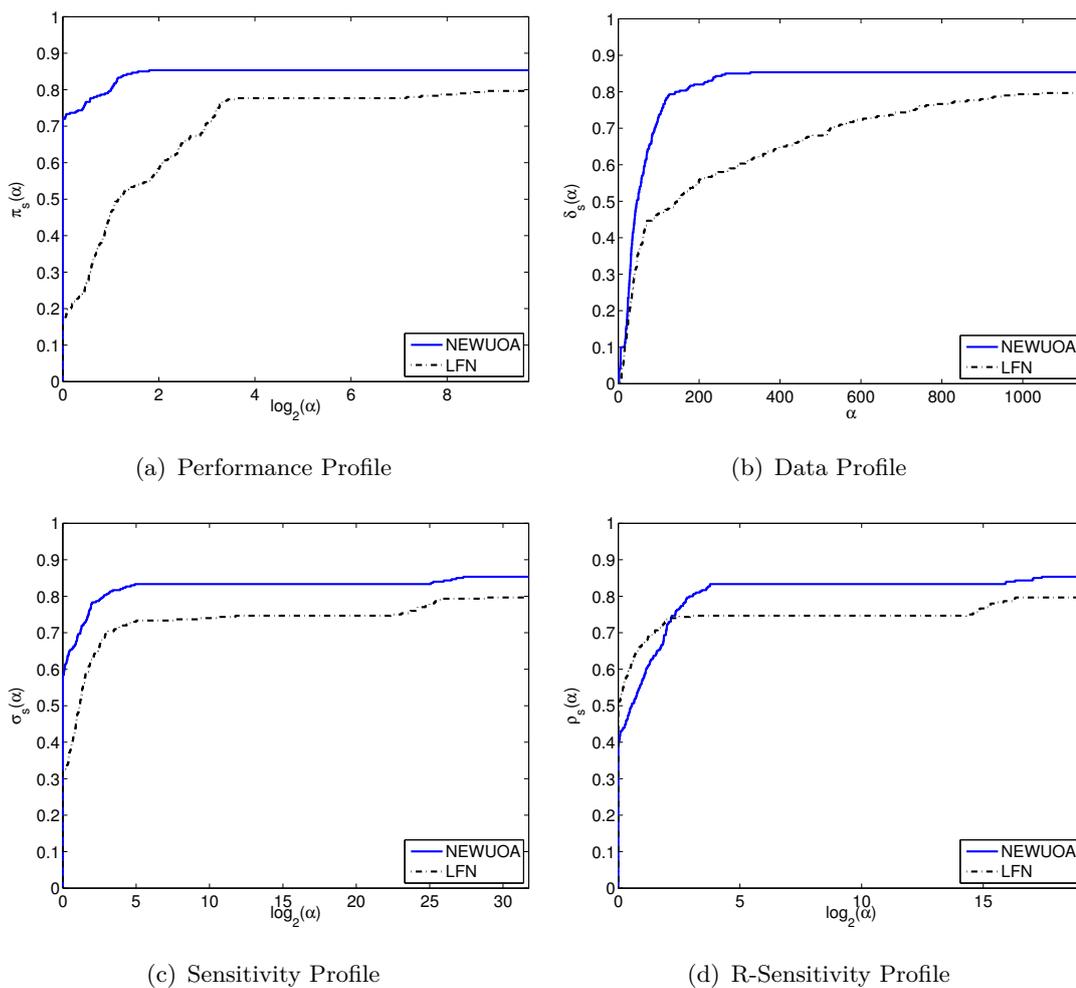


图 3.16: NEWUOA 与 LFN 的数值表现 (精度 $\tau = 10^{-10}$)

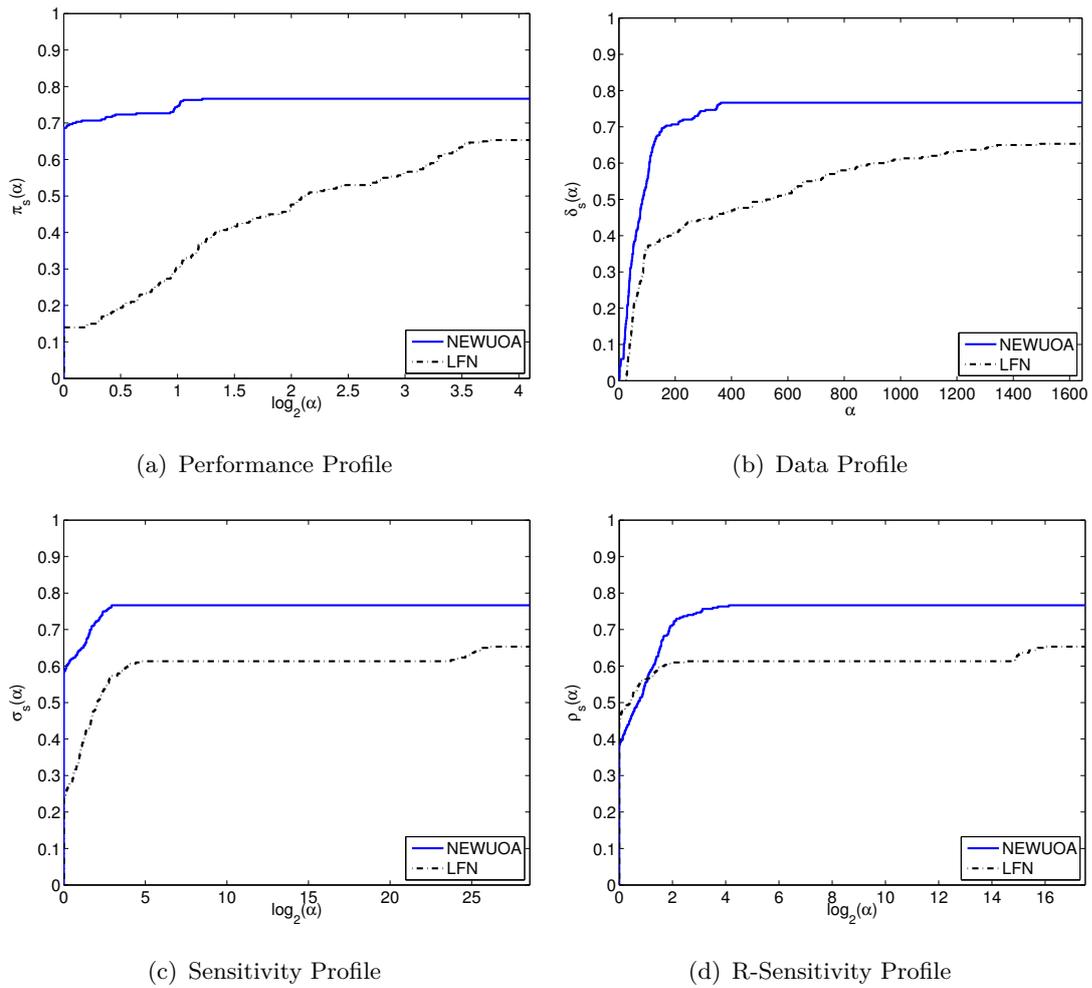


图 3.17: NEWUOA 与 LFN 的数值表现 (迭代自然终止)

我们的结论是: 在我们的实验中, 当求解精度不高 (比如不高于 10^{-4}) 时, 综合计算开销与稳定性来看 LFN 算法优于 NEWUOA 算法; 当求解精度很高 (比如不低于 10^{-5}) 时, NEWUOA 算法则优于 LFN 算法. 由此可见, NEWUOA 越到迭代后期表现越好, 这是因为, 离解越近, 目标函数的曲率信息越重要, 而对称 Broyden 修正比最小 Frobenius 范数模型更善于收集曲率信息. LFN 算法在低精度要求下的良好表现十分值得注意, 因为在无导数优化的实际应用中, 10^{-4} 可能已经是一个足够高的精度了.

上述实验中我们仅仅求解了 20 至 40 维的问题. 图 3.18 展示了 NEWUOA 算法与 LFN 算法分别求解 100 维 ARWHEAD 问题与 CHROSEN 问题得到的函数值下降曲线. 从中, 我们很清晰地看到, 在这两次求解中, 迭代初期 LFN 算法表现更好, 而迭代后期 NEWUOA 算法表现更好.

必须指出的是, 我们的实验不能作为 NEWUOA 算法与 DFO 算法的比较. 这是因为, 我们的比较是在 NEWUOA 的算法框架下进行的. DFO 算法的核心除了最小 Frobenius 范数模型之外还有很多重要技术, 比如 Newton 基本多项式的使用, 插值点集的选择与更新等等. 所有这些技术有机结合在一起, 不能进行简单的切割.

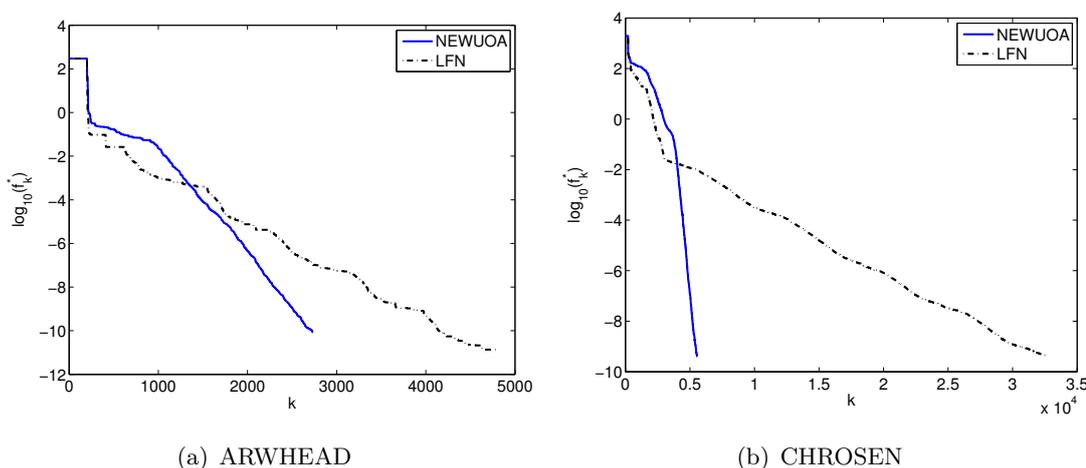


图 3.18: NEWUOA 与 LFN 求解 100 维 ARWHEAD 问题与 CHROSEN 问题的函数值下降曲线

§3.5 总结

本章研究了无导数优化中的最小 Frobenius 范数插值与对称 Broyden 修正. 二者目前被认为是基于插值模型和信赖域的无导数方法中最好的模型构造策略. 我们首先证明了这两种策略在某些情形下的等价性, 指出基于最小 Frobenius 范数模型的 DFO 算法有时也采用了对称 Broyden 修正. 另一方面, 基于对称 Broyden 修正的 NEWUOA 算法有时也采用最小 Frobenius 范数模型, 这就是 NEWUOA 算法的重开始技术. 我们发现 NEWUOA 源代码中的重开始准则与 Powell [138] 中叙述的有所不同. 为了比较它们对下降比的要求哪一个更优, 我们修改 NEWUOA 源代码中的一行, 得到了一个新版的代码 NEWUOAm. 在我们的实验中 NEWUOAm 代码明显优于原始版本的 NEWUOA 代码. 因此, 我们建议用 NEWUOAm 代码代替原始版本的 NEWUOA 代码. 我们还比较了最小 Frobenius 范数模型与对称 Broyden 修正在 NEWUOA 算法框架下的表现. 在我们的实验中, 求解精度不高时前者较优, 而越到迭代后期对称 Broyden 修正表现越好, 这是因为对称 Broyden 修正在迭代过程中逐步积累的曲率信息有助于提高局部收敛速度. 因为很多实际问题对于求解精度要求并不高, 所以最小 Frobenius 范数模型在低精度要求下的良好表现应该引起足够的重视.

在 §3.3 中, 我们并没有把 NEWUOAm 称为对 NEWUOA 算法的改进, 而称为对 NEWUOA 源代码的改进. 因为这一改进已经包含在 Powell [138] 对 NEWUOA 的描述中. 这仅仅是编程实现的不同, 算法并没有区别. 我们指出这一改进的目的有两个, 一是展示重开始技术对 NEWUOA 的重要性, 二是为后面的数值比较提供一个更合理的 NEWUOA 的版本. 我们希望在我们的算法与 NEWUOA 比较时, NEWUOA 能发挥最大的潜力, 这样的比较才有意义.

通过 §3.3 与 §3.4 的实验我们看出, 最小 Frobenius 范数模型与对称 Broyden 修正各有优势, 两者合理的结合可以有效提高算法效率. 因此有必要研究两种策略杂交的算法. 一种显而易见的做法是, 迭代初期使用最小 Frobenius 范数模型, 迭代后期转向对称 Broyden 修正. 如何选择转换的准则是其中的关键问题. 更一般的做法是, 在第 k 步使用凸组合

$$(1 - \sigma_k)Q_k^F + \sigma_k Q_k^B \quad (3.12)$$

作为模型, 其中 Q_k^F 为最小 Frobenius 范数模型, Q_k^B 为对称 Broyden 修正得到的模型, σ_k 是凸组合参数. 显然该模型仍满足插值条件 (3.1). 其中的关键问题

是 σ_k 的调节. 借鉴 NEWUOA 的重开始策略, 我们可以根据 Q_{k-1} 的表现 (如信赖域子问题的下降比) 以及 Q_k^B 与 Q_k^F 的系数对比来确定 σ_k . 还有一种选择是, 每一步求解两个信赖域子问题, 一个使用 Q_k^F 作为模型, 另一个使用 Q_k^B 作为模型; 第 $k+1$ 步迭代中, 通过前面迭代中两种模型的表现决定 Q_{k+1}^B 应该由 Q_k^F 还是 Q_k^B 修正得到. 这种策略的优点在于它适合并行计算. 以上这些都是值得研究的问题.

§ 3.6 关于无导数对称 Broyden 修正的历史注记

本节是一个历史注记. 我们给出无导数对称 Broyden 修正与经典方法的渊源, 并简述该修正方法的研究历史.

对称 Broyden 修正最早出现在拟牛顿方法中. 假定我们求解无约束优化问题

$$\min_{x \in \mathbb{R}^n} f(x). \quad (3.13)$$

其中 f 的导数信息可用. 设在第 k 步迭代我们有 $\nabla^2 f(x_k)$ 的一个近似 B_k , 且

$$s_k = x_{k+1} - x_k \neq 0, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k), \quad (3.14)$$

则拟牛顿方法的对称 Broyden 修正公式是

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^T + s_k (y_k - B_k s_k)^T}{s_k^T s_k} - \frac{s_k^T (y_k - B_k s_k) s_k s_k^T}{(s_k^T s_k)^2}. \quad (3.15)$$

该修正可以理解为 Broyden 非对称秩 1 修正 [190] 的对称化. 它由 Powell [123] (1970 年) 提出, 所以常被称为 Powell 的对称 Broyden 公式, 简称 PSB (Powell's symmetric Broyden) 公式. Dennis 和 Schnabel [53] 指出, PSB 公式是一个最小 Frobenius 范数修正公式, 也就是有以下定理.

定理 3.2 ([53], 定理 4.2). 设 B_k 对称, 则 PSB 公式 (3.15) 给出的 B_{k+1} 是问题

$$\min \|B - B_k\|_F \quad (3.16)$$

$$\text{s.t. } B s_k = y_k, \quad B_k^T = B_k \quad (3.17)$$

的唯一解.

用插值的语言, 定理 3.2 可描述为:

定理 3.3. 设 Q_k 是一个二次函数. 插值问题

$$\min_{Q \in \mathcal{Q}} \|\nabla^2 Q - \nabla^2 Q_k\|_F \quad (3.18)$$

$$\text{s.t. } \nabla Q(x_{k+1}) = \nabla f(x_{k+1}) \quad (3.19)$$

$$\nabla Q(x_k) = \nabla f(x_k) \quad (3.20)$$

$$Q(x_{k+1}) = f(x_{k+1}) \quad (3.21)$$

的唯一解是

$$Q(x) = \frac{1}{2}(x - x_{k+1})^T B_{k+1}(x - x_{k+1}) + [\nabla f(x_{k+1})]^T(x - x_{k+1}) + f(x_{k+1}). \quad (3.22)$$

其中, B_{k+1} 由 PSB 公式 (3.15) 给出, $B_k = \nabla^2 Q_k$.

由此可见, 修正 (3.6)–(3.7) 是 PSB 公式的无导数版本.

Powell 最早将修正 (3.6)–(3.7) 引入无导数优化方法中. 这一修正最早见于文献是在 Powell [134] (2002 年 2 月), 最早发表是在 Powell [136] (2003 年). Powell [138] 给出了一段有趣的历史注记. 那里提到, 关于该技巧的研究始于 2002 年 1 月. 当发现这一修正的良好表现之后, Powell 着手研究基于该技巧的算法, 即后来的 NEWUOA 算法. 但这一修正的 Fortran 实现直到 2003 年 12 月才完成, 因为 “throughout the first 18 months of the development, computer rounding errors caused unacceptable loss of accuracy in a few difficult test problems”. NEWUOA 算法的研究是漫长和充满挫折的⁹. 关于 NEWUOA 的研究进展, Powell 在 2002 年 2 月 [134]、2002 年 10 月 [133]、2003 年 3 月 [135] 和 2004 年 1 月 [140] 先后作了报告. 直到 2004 年 11 月, 也就是修正 (3.6)–(3.7) 提出近 3 年之后, Powell [138] 才第一次详细介绍了基于该修正的 NEWUOA 算法. 后来, Powell [144] (2010 年 3 月) 将修正 (3.6)–(3.7) 做了扩展 (我们将在第四章中详细讨论这一扩展).

Powell [134, 133, 135, 140, 138] 最初将修正 (3.6)–(3.7) 称为 “最小 Frobenius 范数修正 (least Frobenius norm updating)”；而在 Powell 后来的文章 [143, 144,

⁹ “The development of NEWUOA has taken nearly three years. The work was very frustrating ...” [138].

145] 以及 Powell 2011 年 9 月 21 日发给本文作者的 email 中, 该修正被称为“对称 Broyden (symmetric Broyden)”. 我们倾向于使用第二个名称, 因为它揭示了该方法的历史渊源.

第四章 二次函数的 Sobolev (半) 范数及其在无导数优化中的应用

§ 4.1 引言

第三章提到, 插值问题

$$\min_{Q_k \in \mathcal{Q}} \mathcal{F}_k(Q_k) \quad (4.1)$$

$$\text{s.t. } Q_k(y) = f(y), \quad y \in \mathcal{I}_k \quad (4.2)$$

经常用来构造无导数算法中的信赖域子问题的模型. 其中, \mathcal{Q} 为 \mathbb{R}^n 中所有二次函数构成的空间, \mathcal{F}_k 为 \mathcal{Q} 上的一个泛函, \mathcal{I}_k 为一个插值点集. 根据本文作者涉猎的文献, 无导数方法的研究中, 最早使用 (4.1)–(4.2) 构造模型的是 Conn 和 Toint [43]. 那里

$$\mathcal{F}_k(Q) = \|\nabla^2 Q\|_F^2 + \|\nabla Q(x_k)\|_2^2. \quad (4.3)$$

其中, x_k 为第 k 步迭代的信赖域中心. Conn 和 Toint [43] 报告, 使用上述模型构造的一个算法比一个基于有限差分和 LANCELOT [34] 子程序的算法更高效和稳定. 但是, 作者们并未针对性地解释目标泛函 (4.3) 的引入动机. 以 Conn 和 Toint [43] 的算法为雏形, Conn, Scheinberg 和 Toint [36, 37, 38] 构造了 DFO 算法. DFO 算法仍使用插值问题 (4.1)–(4.2) 构造模型, 不过 DFO 的模型是第三章介绍的最小 Frobenius 范数模型. 换句话说, DFO 使用的目标泛函为

$$\mathcal{F}_k(Q) = \|\nabla^2 Q\|_F^2. \quad (4.4)$$

Wild [172] 提出的 MNH 算法也使用了最小 Frobenius 范数模型. Conn, Scheinberg 和 Vicente [42] 通过下面的定理为目标泛函 (4.4) 提供了一个解释.

定理 4.1 ([42], 定理 5.4). 设插值点集 $\mathcal{I}_k = \{y_0, y_1, \dots, y_m\}$ ($m \geq n$) 包含在球

$$\mathcal{B} = \{x \in \mathbb{R}^n : \|x - y_0\|_2 \leq r\} \quad (4.5)$$

中, 且矩阵

$$L = \frac{1}{r}(y_1 - y_0 \quad \cdots \quad y_m - y_0) \quad (4.6)$$

秩为 n . 若 f 于球 \mathcal{B} 上可微, ∇f 于球 \mathcal{B} 上 Lipschitz 连续, 且 Lipschitz 常数为 $\nu > 0$, 那么对任何满足插值条件

$$Q(y) = f(y), \quad y \in \mathcal{I}_k \quad (4.7)$$

的二次函数 Q 与任何 $x \in \mathcal{B}$, 有

$$\|\nabla Q(x) - \nabla f(x)\|_2 \leq \frac{5\sqrt{m}}{2} \|L^+\|_2 (\nu + \|\nabla^2 Q\|_2) r, \quad (4.8)$$

以及

$$|Q(x) - f(x)| \leq \left(\frac{5\sqrt{m}}{2} \|L^+\|_2 + \frac{1}{2} \right) (\nu + \|\nabla^2 Q\|_2) r^2. \quad (4.9)$$

其中, L^+ 为 L 之广义逆.

在定理 4.1 的意义下, 极小化二次模型 Hessian 矩阵的某种范数有助于模型对目标函数的逼近. 需要注意的是, 定理 4.1 中 Hessian 矩阵的范数是 2-范数, 而 DFO 算法的目标泛函 (4.4) 中 Hessian 矩阵的范数是 Frobenius 范数. 换句话说, 定理 4.1 并不能为目标泛函 (4.4) 提供精确的解释.

Powell [134, 133, 135, 140, 138, 142, 143] 将对称 Broyden 修正引入无导数方法的研究中. 第三章已经介绍, 对称 Broyden 修正是使用目标泛函

$$\mathcal{F}_k(Q) = \|\nabla^2 Q - \nabla^2 Q_{k-1}\|_F^2 \quad (4.10)$$

来确定第 k 步迭代的模型. 对称 Broyden 修正的一个优点是, 如果目标函数 f 为二次函数, 那么由该修正得到的模型 Q_k 满足

$$\begin{aligned} & \|\nabla^2 Q_k - \nabla^2 f\|_F^2 \\ &= \|\nabla^2 Q_{k-1} - \nabla^2 f\|_F^2 - \|\nabla^2 Q_k - \nabla^2 Q_{k-1}\|_F^2 \\ &\leq \|\nabla^2 Q_{k-1} - \nabla^2 f\|_F^2. \end{aligned} \quad (4.11)$$

因此, $\nabla^2 Q_k$ 比 $\nabla^2 Q_{k-1}$ 更好地逼近 $\nabla^2 f$, 除非 $\nabla^2 Q_k = \nabla^2 Q_{k-1}$.

Powell [144, 146] 推广了对称 Broyden 修正. 其做法是在目标泛函 (4.10) 中引入一个一阶项, 即考虑

$$\mathcal{F}_k(Q) = \|\nabla^2 Q - \nabla^2 Q_{k-1}\|_F^2 + \sigma \|\nabla Q(x_0) - \nabla Q_{k-1}(x_0)\|_2^2, \quad (4.12)$$

其中非负常数 σ 与点 $x_0 \in \mathbb{R}^n$ 为待选择的参量. 我们称使用上述目标泛函的修正方法为扩展的对称 Broyden 修正. 该修正的一个核心问题是参量 σ 与 x_0 的选取. 利用 Lagrange 插值基函数, Powell [144, 146] 给出了这些参量的选取方式.

Custódio, Rocha 和 Vicente [46] 将插值问题 (4.1)–(4.2) 确定的模型用于改进直接搜索方法. 作者考虑了最小 Frobenius 范数模型和对称 Broyden 修正确定的模型, 也就是在插值问题 (4.1)–(4.2) 中使用目标泛函 (4.4) 和 (4.10) 得到的模型. 作者报告, 在他们的方法中目标泛函 (4.10) 表现更优.

不论使用目标泛函 (4.3)、(4.4)、(4.10) 还是 (4.12), 插值问题 (4.1)–(4.2) 都可以归结为以下问题:

$$\min_{Q \in \mathcal{Q}} \|\nabla^2 Q\|_F^2 + \sigma \|\nabla Q(x_0)\|_2^2 \quad (4.13)$$

$$\text{s.t. } Q(y) = F(y), \quad y \in \mathcal{I}, \quad (4.14)$$

其中, σ 为非负常数, $x_0 \in \mathbb{R}^n$ 为固定点, \mathcal{I} 为一个插值点集, F 为定义在 \mathcal{I} 上的一个函数.

需要注意的是, 函数 F 未必等于插值条件 (4.2) 中的函数 f . 比如, 要把对称 Broyden 修正与扩展的对称 Broyden 修正转化为插值问题 (4.13)–(4.14) 的形式, 我们需要令

$$Q = Q_k - Q_{k-1}, \quad (4.15)$$

因而

$$F(y) = f(y) - Q_{k-1}(y), \quad y \in \mathcal{I}_k. \quad (4.16)$$

在 Powell [134, 133, 135, 140, 138, 142, 143, 144, 146] 的方法里, \mathcal{I}_k 中仅有一点不含于 \mathcal{I}_{k-1} , 故除了此点之外, F 在 \mathcal{I}_k 上每一点处都为零.

我们称插值问题 (4.13)–(4.14) 为最小范数插值 (least-norm interpolation) 问题. 该插值问题的目标泛函是有意思的. 实践已经证明了其有效性. 而且, 由于它是一个二次泛函, 所以十分容易处理. 本章就是要研究这一简单而又实用的目标泛函. 我们的主要目的是为该泛函提供一些新的理解, 使用的工具是 PDE 理论中经典的 H^1 半范数. 我们将回答一些关于该泛函的基本问题. 比如说, 该泛函的分析和几何意义是什么? 为什么要组合 Hessian 矩阵 Frobenius 范数的平方与梯度 2-范数的平方? 参量 σ 与 x_0 的几何意义是什么, 应该如何选取?

本章的结构如下. §4.2 简要介绍 Sobolev 空间中的范数与半范数的定义. §4.3 研究二次函数在 ℓ_2 球上的 H^1 半范数; 我们用二次函数的系数显式给出这种半范数的表达式. 在 §4.4 中, 我们证明最小范数插值实际上是在一个 ℓ_2 球上极小化插值函数的 H^1 半范数, 而参数 σ 与 x_0 分别对应球的半径与球心; 这为理解最小范数插值提供了一种新视角. 利用这些新的观察, §4.5 研究扩展的对称 Broyden 修正, 利用 σ 与 x_0 的几何意义, 我们给出这些参量的选取方式. §4.6 研究二次函数在 ℓ_p ($1 \leq p \leq \infty$) 球上的 H^0 范数与 H^1 半范数, 对这些范数与半范数做一个统一的讨论, 用二次函数的系数显式给出它们的表达式; 这些结果对于使用 ℓ_p 做信赖域的方法可能是有用的. §4.7 是本章的一个总结.

§4.2 Sobolev (半) 范数简介

本节简要介绍 Sobolev 空间中的范数与半范数的定义. 为简单计, 我们不引入广义函数, 而仅仅在普通函数的意义下介绍. 广义函数意义下的 Sobolev 空间理论请参考 Adams [2] 和 Evans [61].

为了简洁地描述高阶偏导数, 我们需要使用重指标 (multi-index). 关于重指标的概念, 读者可参考任何一本标准的多元微积分教材. 下面我们给出 Sobolev 范数与半范数的定义.

定义 4.2. 设 Ω 为 \mathbb{R}^n 的一个区域, f 为 Ω 上的一个函数, k 为非负整数, $1 \leq p < \infty$. 若 f 在 Ω 上 m 次可微, 且对任何 n 重指标 α 满足 $|\alpha| \leq k$, 有

$$\frac{\partial^\alpha f}{\partial x^\alpha} \in L^p(\Omega), \quad (4.17)$$

则定义

$$\|f\|_{H^{k,p}(\Omega)} = \left(\sum_{|\alpha| \leq k} \left\| \frac{\partial^\alpha f}{\partial x^\alpha} \right\|_{L^p(\Omega)}^p \right)^{\frac{1}{p}} \quad (4.18)$$

为 f 在 Ω 上的 $H^{k,p}$ 范数, 定义

$$|f|_{H^{k,p}(\Omega)} = \left(\sum_{|\alpha|=k} \left\| \frac{\partial^\alpha f}{\partial x^\alpha} \right\|_{L^p(\Omega)}^p \right)^{\frac{1}{p}} \quad (4.19)$$

为 f 在 Ω 上的 $H^{k,p}$ 半范数.

$H^{k,2}$ 常简写为 H^k . 本章中我们只考虑函数的 H^0 范数与 H^1 半范数. 按照定义 4.2, 它们分别是

$$\|f\|_{H^0(\Omega)} = \left[\int_{\Omega} |f(x)|^2 dx \right]^{\frac{1}{2}} \quad (4.20)$$

和

$$\|f\|_{H^1(\Omega)} = \left[\int_{\Omega} \|\nabla f(x)\|_2^2 dx \right]^{\frac{1}{2}}. \quad (4.21)$$

可见, 它们分别是 f 与 ∇f 的 $L^2(\Omega)$ 范数.

§ 4.3 二次函数在 ℓ_2 球上的 H^1 半范数

本节讨论二次函数在 \mathbb{R}^n 中的 ℓ_2 球上的 H^1 半范数. 我们用二次函数的系数给出这种半范数的显式表达式.

定理 4.3. 设 x_0 为 \mathbb{R}^n 中一点, r 为一正数, 且

$$\mathcal{B}_2^r(x_0) = \{x \in \mathbb{R}^n : \|x - x_0\|_2 \leq r\}. \quad (4.22)$$

那么对任何 $Q \in \mathcal{Q}$,

$$|Q|_{H^1(\mathcal{B}_2^r(x_0))}^2 = V_2 r^n \left[\frac{r^2}{n+2} \|\nabla^2 Q\|_F^2 + \|\nabla Q(x_0)\|_2^2 \right], \quad (4.23)$$

其中 V_2 是 \mathbb{R}^n 中 ℓ_2 单位球的体积.

证明. 令

$$g = \nabla Q(x_0), \quad B = \nabla^2 Q. \quad (4.24)$$

那么

$$\begin{aligned} & |Q|_{H^1(\mathcal{B}_2^r(x_0))}^2 \\ &= \int_{\|x-x_0\|_2 \leq r} \|B(x-x_0) + g\|_2^2 dx \\ &= \int_{\|x\|_2 \leq r} \|Bx + g\|_2^2 dx \\ &= \int_{\|x\|_2 \leq r} (x^T B^2 x + 2x^T Bg + \|g\|_2^2) dx. \end{aligned} \quad (4.25)$$

由于对称性, $x^T B g$ 之积分为零. 另外,

$$\int_{\|x\|_2 \leq r} \|g\|_2^2 dx = V_2 r^n \|g\|_2^2. \quad (4.26)$$

因此我们只需积分 $x^T B^2 x$. 不失一般性, 假设 B 为对角矩阵, 否则做一个旋转即可. 这时, 由对称性可以得到

$$\int_{\|x\|_2 \leq r} x^T B^2 x dx = \frac{1}{n} \|B\|_F^2 \int_{\|x\|_2 \leq r} \|x\|_2^2 dx. \quad (4.27)$$

令 $V_2(t)$ 和 $S_2(t)$ 分别表示 \mathbb{R}^n 中半径为 t 的 ℓ_2 球的体积和表面积. 那么

$$S_2(t) = \frac{d}{dt} V_2(t) = \frac{d}{dt} V_2 t^n = n V_2 t^{n-1}. \quad (4.28)$$

因此

$$\int_{\|x\|_2 \leq r} \|x\|_2^2 dx = \int_0^r dt \int_{\|x\|_2=t} \|x\|_2^2 dS = \int_0^r t^2 S_2(t) dt = \frac{n r^{n+2}}{n+2} V_2. \quad (4.29)$$

故所论成立. ■

定理 4.3 说明, 二次函数 Q 在 ℓ_2 球上的 H^1 半范数的平方是 $\|\nabla^2 Q\|_F^2$ 和 $\|\nabla Q(x_0)\|_2^2$ 的一个组合, 其中 x_0 为球心, 组合系数由球的半径决定. 这一结果有两个意义. 第一, 它使我们可以从整体上度量二次函数的梯度在一个 ℓ_2 球上的大小. 这对一个二次函数来说并不是平凡的, 因为其梯度不是常量. 第二, 它揭示了将 Hessian 矩阵 Frobenius 范数的平方和梯度 2-范数的平方做组合的分析和几何意义, 从而我们可以根据几何意义选择组合中的参数.

§ 4.4 H^1 半范数观点下的最小范数插值

二次函数的 H^1 半范数提供了理解最小范数插值的一种新视角. 本节我们详细的讨论如何利用 H^1 半范数描述和理解最小范数插值.

为方便记, 我们把最小范数插值重新写在这里, 并且称之为问题 $P_1(\sigma)$.

$$\begin{aligned} \min_{Q \in \mathcal{Q}} \quad & \|\nabla^2 Q\|_F^2 + \sigma \|\nabla Q(x_0)\|_2^2 \\ \text{s.t.} \quad & Q(x) = F(x), \quad x \in \mathcal{I}. \end{aligned} \quad (P_1(\sigma))$$

我们假设插值系统

$$Q(x) = F(x), \quad x \in \mathcal{I} \quad (4.30)$$

在 \mathcal{Q} 上是相容的, 也就是说

$$\{Q \in \mathcal{Q} : Q(x) = F(x), \quad x \in \mathcal{I}\} \neq \emptyset. \quad (4.31)$$

由定理 4.3, 可以看到问题 $P_1(\sigma)$ 与问题 $P_2(r)$:

$$\begin{aligned} \min_{Q \in \mathcal{Q}} |Q|_{H^1(\mathcal{B}_2^r(x_0))} \\ \text{s.t. } Q(x) = F(x), \quad x \in \mathcal{I} \end{aligned} \quad (P_2(r))$$

在某种意义上是等价的. 本节的目的是要严格阐述这种等价性.

当 $\sigma > 0$ 时, 这种等价性是简单的. 我们将它描述如下.

定理 4.4. 若 $\sigma > 0$, 则最小范数插值问题 $P_1(\sigma)$ 等价于问题 $P_2(r)$, 其中 $r = \sqrt{(n+2)/\sigma}$.

可见, 当 $\sigma > 0$ 时, 最小范数插值问题本质上是在一个 l_2 球上极小化插值函数的 H^1 半范数. 参数 σ 与 x_0 的几何意义已经很清楚: x_0 为该 l_2 球的球心, $\sqrt{(n+2)/\sigma}$ 为球的半径.

现在我们考虑 $\sigma = 0$ 时的最小范数插值. 由于此时 $P_1(\sigma)$ 可能有多解, 我们重新定义 $P_1(0)$ 为如下的双层最小范数插值问题

$$\begin{aligned} \min_{Q \in \mathcal{Q}} \|\nabla Q(x_0)\|_2 \\ \text{s.t. } \min_{Q \in \mathcal{Q}} \|\nabla^2 Q\|_{\mathbb{F}} \\ \text{s.t. } Q(x) = F(x), \quad x \in \mathcal{I}. \end{aligned} \quad (P_1(0))$$

这种重定义是合理的, 因为我们有以下命题.

命题 4.5. 任给 $\sigma \geq 0$, 问题 $P_1(\sigma)$ 有唯一解 Q_σ . 并且, 当 σ 趋于 0 时 Q_σ 趋于¹ Q_0 .

¹我们定义 \mathcal{Q} 上的收敛为按系数收敛.

证明. 首先, 我们证明 $\sigma > 0$ 时问题 $P_1(\sigma)$ 解的唯一性. 记

$$|Q|_\sigma = [\|\nabla^2 Q\|_F^2 + \sigma \|\nabla Q(x_0)\|_2^2]^{\frac{1}{2}}. \quad (4.32)$$

设 q_1 与 q_2 都是问题 $P_1(\sigma)$ 的解, 那么它们的平均也是解. 因而由等式

$$\frac{1}{2}(|q_1|_\sigma^2 + |q_2|_\sigma^2) - \left| \frac{1}{2}(q_1 + q_2) \right|_\sigma^2 = \frac{1}{4}|q_1 - q_2|_\sigma^2, \quad (4.33)$$

可知 $|q_1 - q_2|_\sigma = 0$. 因此 $\nabla^2 q_1 = \nabla^2 q_2$, 且 $\nabla q_1(x_0) = \nabla q_2(x_0)$. 这时借助插值条件

$$q_1(x) = F(x) = q_2(x), \quad x \in \mathcal{I} \quad (4.34)$$

即知 $q_1 = q_2$.

我们可以用相同的方法证明问题 $P_1(0)$ 解的唯一性, 只要注意

$$\frac{1}{2}(\|\nabla^2 q_1\|_F^2 + \|\nabla^2 q_2\|_F^2) - \left\| \frac{1}{2}\nabla^2(q_1 + q_2) \right\|_F^2 = \frac{1}{4}\|\nabla^2 q_1 - \nabla^2 q_2\|_F^2 \quad (4.35)$$

和一个关于梯度的类似等式即可.

现在我们证明 σ 趋于 0 时 Q_σ 收敛到 Q_0 . 假设此收敛性不成立, 则存在正数序列 $\{\sigma_k\}$ 使得 $\{\sigma_k\}$ 趋于 0 但 $\{Q_{\sigma_k}\}$ 不收敛到 Q_0 . 注意到

$$\|\nabla^2 Q_{\sigma_k}\|_F \geq \|\nabla^2 Q_0\|_F \quad (4.36)$$

与

$$\|\nabla^2 Q_{\sigma_k}\|_F^2 + \sigma_k \|\nabla Q_{\sigma_k}(x_0)\|_2^2 \leq \|\nabla^2 Q_0\|_F^2 + \sigma_k \|\nabla Q_0(x_0)\|_2^2, \quad (4.37)$$

我们有

$$\|\nabla Q_{\sigma_k}(x_0)\|_2 \leq \|\nabla Q_0(x_0)\|_2. \quad (4.38)$$

根据 (4.37)–(4.38) 和插值条件, $\{Q_{\sigma_k}\}$ 有界, 故我们可以假设 $\{Q_{\sigma_k}\}$ 收敛于 $\bar{Q} \neq Q_0$. 不等式 (4.36) 以及不等式 (4.37) 表明 $\|\nabla^2 \bar{Q}\|_F = \|\nabla^2 Q_0\|_F$, 不等式 (4.38) 表明 $\|\nabla \bar{Q}(x_0)\|_2 \leq \|\nabla Q_0(x_0)\|_2$. 因此 \bar{Q} 为问题 $P_1(0)$ 的另一个解, 与上面证明的唯一性矛盾. ■

命题 4.5 说明问题 $P_2(r)$ 对任何正数 r 有唯一解. 现在我们可以给出问题 $P_1(0)$ 与 $P_2(r)$ 的关系.

定理 4.6. 当 r 趋于无穷时, 问题 $P_2(r)$ 的解收敛到问题 $P_1(0)$ 的解.

定理 4.6 表明, 当 $\sigma = 0$ 时, 最小范数插值问题在极限的意义下求一个在 \mathbb{R}^n 上具有最小 H^1 半范数的插值函数. 因此定理 4.6 将定理 4.4 推广到了 $\sigma = 0$ 的情形.

现在我们已经能够回答本章引言列出的所有问题. 最小范数插值问题目标泛函的意义是极小化插值函数在某个 ℓ_2 球上的 H^1 半范数. 组合 Hessian 矩阵 Frobenius 范数的平方与梯度 2-范数的平方是为了度量插值函数在该球上的 H^1 半范数. 参数 $\sqrt{(n+2)/\sigma}$ 与 x_0 分别是该球的半径和球心. 当 $\sigma = 0$ 时, 我们可以在极限的意义下理解最小范数插值.

§ 4.5 扩展的对称 Broyden 修正

作为 § 4.3 和 § 4.4 理论的一个应用, 我们讨论 Powell [144, 146] 提出的扩展的对称 Broyden 修正. 正如 § 4.1 介绍的, 该修正通过求解

$$\min_{Q_k \in \mathcal{Q}} \|\nabla^2 Q_k - \nabla^2 Q_{k-1}\|_F^2 + \sigma \|\nabla Q_k(x_0) - \nabla Q_{k-1}(x_0)\|_2^2 \quad (4.39)$$

$$\text{s.t. } Q_k(x) = f(x), \quad x \in \mathcal{I}_k, \quad (4.40)$$

得到第 k 步迭代的模型 Q_k , 其中 σ 与 x_0 为待决定的参量. 当 $\sigma = 0$ 时, 它就是 Powell [134, 133, 135, 140, 138, 142, 143] 所研究的对称 Broyden 修正. 我们只考虑 $\sigma > 0$ 的情况. 本节利用 § 4.3 和 § 4.4 的理论, 根据 σ 和 x_0 的几何意义给出这两个参量的选取方式, 并用数值实验与 Powell [144, 146] 的选取方式进行比较. 在我们的实验中, 我们的参数选取方式优于 Powell 的选取方式. 这从一个侧面说明我们的理论的确为理解最小范数插值提供了一种有效的工具.

§ 4.5.1 H^1 半范数与扩展的对称 Broyden 修正

根据定理 4.4, 插值问题 (4.39)–(4.40) 等价于

$$\min_{Q_k \in \mathcal{Q}} |Q_k - Q_{k-1}|_{H^1(\mathcal{B})} \quad (4.41)$$

$$\text{s.t. } Q_k(x) = f(x), \quad x \in \mathcal{I}_k, \quad (4.42)$$

其中

$$\mathcal{B} = \left\{ x \in \mathbb{R}^n : \|x - x_0\|_2 \leq \sqrt{(n+2)/\sigma} \right\}. \quad (4.43)$$

可见, 扩展的对称 Broyden 修正在 H^1 半范数的意义下寻求离上一个模型最近的模型.

与对称 Broyden 修正类似, 扩展的对称 Broyden 修正对二次目标函数具有如下良好的逼近性质.

命题 4.7. 若 f 为二次函数, 则问题 (4.39)–(4.40) 定义的模型 Q_k 满足

$$|Q_k - f|_{H^1(\mathcal{B})}^2 = |Q_{k-1} - f|_{H^1(\mathcal{B})}^2 - |Q_k - Q_{k-1}|_{H^1(\mathcal{B})}^2, \quad (4.44)$$

其中 \mathcal{B} 如 (4.43) 定义.

证明. 令 $Q_t = Q_k + t(Q_k - f)$, t 是任何实数. 那么 Q_t 为一个二次函数且满足插值条件 (4.42). 故由 Q_k 的最优性知二次函数

$$\varphi(t) = |Q_t - Q_{k-1}|_{H^1(\mathcal{B})}^2 \quad (4.45)$$

在 $t = 0$ 时达到最小值. 将 $\varphi(t)$ 展开, 我们有

$$\begin{aligned} \varphi(t) &= t^2 |Q_k - f|_{H^1(\mathcal{B})}^2 \\ &\quad + 2t \int_{\mathcal{B}} [\nabla(Q_k - Q_{k-1})(x)]^T [\nabla(f - Q_{k-1})(x)] dx \\ &\quad + |Q_k - Q_{k-1}|_{H^1(\mathcal{B})}^2. \end{aligned} \quad (4.46)$$

因而

$$\int_{\mathcal{B}} [\nabla(Q_k - Q_{k-1})(x)]^T [\nabla(f - Q_{k-1})(x)] dx = 0. \quad (4.47)$$

考虑 $\varphi(-1)$, 我们就得到了 (4.44). ■

根据定理 4.3, 等式 (4.44) 等价于 Powell [144] 的等式 (1.9). 等式 (4.44) 表明

$$\int_{\mathcal{B}} \|\nabla Q_k(x) - \nabla f(x)\|_2^2 dx \leq \int_{\mathcal{B}} \|\nabla Q_{k-1}(x) - \nabla f(x)\|_2^2 dx. \quad (4.48)$$

换句话说, 在 \mathcal{B} 上, ∇Q_k 比 ∇Q_{k-1} 更好地逼近 ∇f , 除非 $Q_k = Q_{k-1}$.

§4.5.2 参量 σ 与 x_0 的选择

现在我们把注意力转向扩展的对称 Broyden 修正 (4.39)–(4.40) 中参量 σ 与 x_0 的选择. 这是该修正的核心问题之一. 我们注意, 该修正的目的是为信赖域子问题提供一个模型. 在进行修正之前, 信赖域已经确定. 我们设第 k 步迭代的信赖域为

$$\{x : \|x - x_k\|_2 \leq \Delta_k\}, \quad (4.49)$$

点 x_k 为信赖域中心, Δ_k 为信赖域半径.

在叙述我们的选取方式之前, 我们先来看 Powell [144, 146] 是如何选取 σ 与 x_0 的. Powell [144, 146] 的选取是通过考察插值问题 (4.39)–(4.40) 的 Lagrange 插值基函数完成的. 假定第 k 步迭代的插值点集为

$$\mathcal{I}_k = \{y_0, y_1, \dots, y_m\}, \quad (4.50)$$

那么插值问题 (4.39)–(4.40) 的第 i 个插值基函数 l_i 定义为问题

$$\min_{l_i \in \mathcal{Q}} \|\nabla^2 l_i\|_F^2 + \sigma \|\nabla l_i\|_2^2 \quad (4.51)$$

$$\text{s.t. } l_i(y_j) = \delta_{ij}, \quad j = 0, 1, \dots, m, \quad (4.52)$$

的解, 其中 δ_{ij} 为 Kronecker 符号, $i = 0, 1, \dots, m$. 本章引言中已经提到, 在 Powell [144, 146] 的算法中, \mathcal{I}_k 仅有一个点不在 \mathcal{I}_{k-1} 中. 我们不妨假设这一点为 y_0 . 由于 Q_{k-1} 在 \mathcal{I}_{k-1} 上插值 f , 故

$$Q_{k-1}(y_j) = f(y_j), \quad j = 1, 2, \dots, m. \quad (4.53)$$

因而插值条件 (4.40) 可以重写为

$$Q(y_j) - Q_{k-1}(y_j) = [f(y_0) - Q_{k-1}(y_0)] \delta_{0j}, \quad j = 0, 1, \dots, m. \quad (4.54)$$

因此问题 (4.39)–(4.40) 的解为

$$Q_k = Q_{k-1} + [f(y_0) - Q_{k-1}(y_0)] l_0. \quad (4.55)$$

可见, 基函数 l_0 在该修正中扮演了核心角色. Powell [144, 146] 的基本思想是选择 σ 与 x_0 使得 l_0 有良好的性态. Powell 指出, 为了使 l_0 行为良好, $\|x_0 - \bar{x}\|_2 / \Delta$ 不可太大. 因此 Powell 选择 x_0 为 x_k , 即当前的信赖域中心. σ 的选取比较复

杂, 但基本想法很直观, 即试图平衡 $\|\nabla^2 l_0\|_F^2$ 与 $\sigma\|\nabla l_0(x_0)\|_2^2$ 的大小. 因此 σ 被设置为 η/ξ , 其中 η 与 ξ 分别估计了 $\|\nabla^2 l_0\|_F^2$ 与 $\|\nabla l_0(x_0)\|_2^2$ 的尺度. 详细的讨论请参考 Powell [144, 146].

现在, 我们来介绍如何运用 §4.4 与 §4.5.1 的理论选择参量 σ 与 x_0 . §4.5.1 中, 我们使用 H^1 半范数重新阐释了扩展的对称 Broyden 修正, 这使得我们可以根据 σ 与 x_0 的几何意义选取这两个参量. 根据问题 (4.41)–(4.42), 选取 σ 与 x_0 等价于选择球 \mathcal{B} . 那么, \mathcal{B} 在修正过程中扮演了怎样的角色? 首先, 该修正极小化 Q_k 与 Q_{k-1} 在 $H^1(\mathcal{B})$ 半范数下的距离, 试图在 \mathcal{B} 上保持 Q_{k-1} 中的信息; 其次, 二次目标函数情形下的 (4.44) 与 (4.48) 暗示我们, 该修正具有在 \mathcal{B} 上改善模型逼近性质的趋势. 因此, 我们应该合理地选择 \mathcal{B} , 使得新模型 Q_k 在 \mathcal{B} 上的行为恰好是我们所关心的. 把球 \mathcal{B} 设置为信赖域球 (4.49) 是一种选择. 但是, 我们倾向于选择一个更大的球. 这是因为, 模型 Q_k 不仅会影响第 k 步迭代, 而且也会影响以后的迭代, 因为 Q_k 的性质一定程度上会通过修正过程遗传给后面的模型². 所以, 仅仅考虑当前的信赖域是短视的. 更合理的选择是令 \mathcal{B} 为球 $\{x : \|x - x_k\|_2 \leq M\Delta_k\}$, M 是一个大于 1 的数. 另外, 在实际计算中我们发现, 令 \mathcal{B} 包含插值点集 \mathcal{I}_k 有利于算法的表现. 因此我们选择

$$\mathcal{B} = \{x : \|x - \bar{x}\|_2 \leq r\}, \quad (4.56)$$

其中

$$r = \max \left\{ M\Delta_k, \max_{y \in \mathcal{I}_k} \|y - x_k\|_2 \right\}. \quad (4.57)$$

进一步,

$$\sigma = \frac{n+2}{r^2}, \quad x_0 = x_k. \quad (4.58)$$

所以我们对 x_0 的选择与 Powell [144, 146] 相同, 但 σ 的选取是新的.

§4.5.3 数值实验

借助二次函数的 H^1 半范数, 我们提出了一种为扩展的对称 Broyden 修正选择参数 σ 的新方法. 本小节通过数值实验将新的方法与 Powell [144, 146] 的方法做一个对比. 我们也与原始的对称 Broyden 修正进行对比, 观察一阶项的引入是否会提高算法的表现.

我们的比较基于 Powell [144] 的 Fortran 77 源代码³. 该代码实现了一个基

²需要注意, 扩展的对称 Broyden 修正是一种最小变化修正, 因此这种遗传的影响是很大的.

³该代码尚未公开. 本文使用的版本是 Powell 教授于 2011 年 6 月 2 日提供给本文作者的.

于扩展的对称 Broyden 修正的无导数信赖域算法. 该代码是 BOBYQA [143] 软件的一个扩展版本, 可以用于求解界约束问题. 由于目前我们只考虑无约束问题, 我们将代码中表示界约束的变量设置为无穷. 我们比较了该代码的三个版本:

- a.) SYMB: σ 总设置为 0. 这相当于使用原始版本的对称 Broyden 修正.
- b.) ESYMBP: σ 按照 Powell [144, 146] 的方法选取. 请参考 § 4.5.2 第二段.
- c.) ESYMBS: σ 按照 (4.57)–(4.58) 选取. 我们令 (4.57) 中 $M = 10$.

需要指出的是, SYMB 相当于一个新版本的 NEWUOA. 但是, 为了保证诸算法的差别完全来自于 σ 的选取, 我们不能用 NEWUOA 代替 SYMB 进行比较.

我们采用第二章 § 2.4 建立的评价体系比较 SYMB、ESYMBP 和 ESYMBS. 我们的测试问题集是 § 3.3.2 中使用的 50 个可变维数的无约束优化问题. 对于每一个测试问题, 我们对 10、12、14、16、18、20 等 6 个维数进行了求解⁴. 每一次求解都按照 § 2.4 做了 10 次随机置换. 我们统计了求解精度 (即条件 (2.5) 中的 τ) 为 10^{-i} ($i = 2, 4, 6, 8, 10$) 时的函数值计算次数, 也统计了迭代自然终止时的函数值计算次数 (条件 (2.7) 中取 $\varepsilon = 10^{-10}$).

与 NEWUOA 相同, Powell [144] 的代码中也包含用户可选参数 NPT、RHOBEG 和 RHOEND. 在 SYMB、ESYMBP 与 ESYMBS 中, 这些参数的选取与 § 3.3.2 相同. 类似于 NEWUOA, Powell [144] 的代码使用了两个信赖域半径, 当较小的信赖域半径达到 RHOEND 时迭代终止. 详情请参考 Powell [138]. 实验中, 我们设置最大函数值计算次数为 50000.

我们的实验环境与 § 3.3.2 相同.

图 4.1 至图 4.6 给出了 SYMB、ESYMBP 与 ESYMBS 在满足不同精度要求以及迭代自然终止时的 Performance Profile、Data Profile、Sensitivity Profile 与 R-Sensitivity Profile. 图线越高说明算法表现越好. 四种 Profile 的定义请参考第二章.

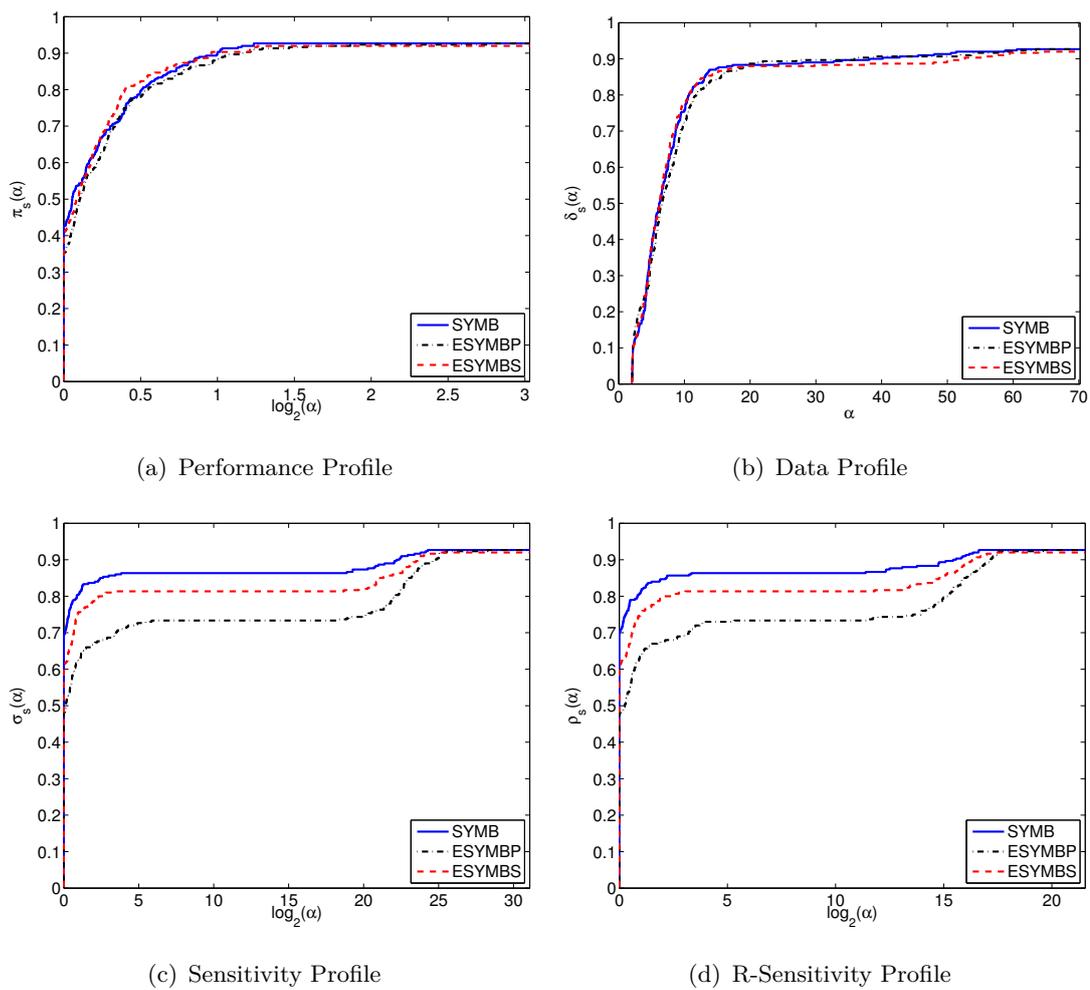
我们先来比较函数值计算次数, 这体现在 Performance Profile 与 Data Profile 中. 可以看到, 不论在哪一种终止条件下, SYMB 的表现都是最好的. 这说明, 在我们的实验中, 在修正中考虑一阶项并未带来函数值计算次数的减少.

⁴与 NEWUOA 相比, 这里使用的代码效率比较低, 所以我们仅仅求解了维数较低的问题.

在 ESYMBP 与 ESYMBS 之间, 后者表现略优. 这说明, 相较于 Powell [144, 146] 的 σ 选取方式, 我们的选取方式减少了实验中的函数值计算次数.

我们再来考察算法对于计算机舍入误差的敏感性, 这体现在 Sensitivity Profile 与 R-Sensitivity Profile 中. 结论与上面是类似的: 在我们的实验中, 在修正中考虑一阶项并未提高算法对计算机舍入误差的稳定性; 但我们的 σ 选取方式在实验中的表现优于 Powell [144, 146] 的选取方式.

我们的结论是: 在我们的实验中, 在对称 Broyden 修正的目标泛函中引入一阶项并未带来算法的改善; 对于扩展的对称 Broyden 修正中的参数 σ , 我们的选取方式表现优于 Powell [144, 146] 的选取方式.

图 4.1: SYMB、ESYMBP 与 ESYMBS 的数值表现 (精度 $\tau = 10^{-2}$)

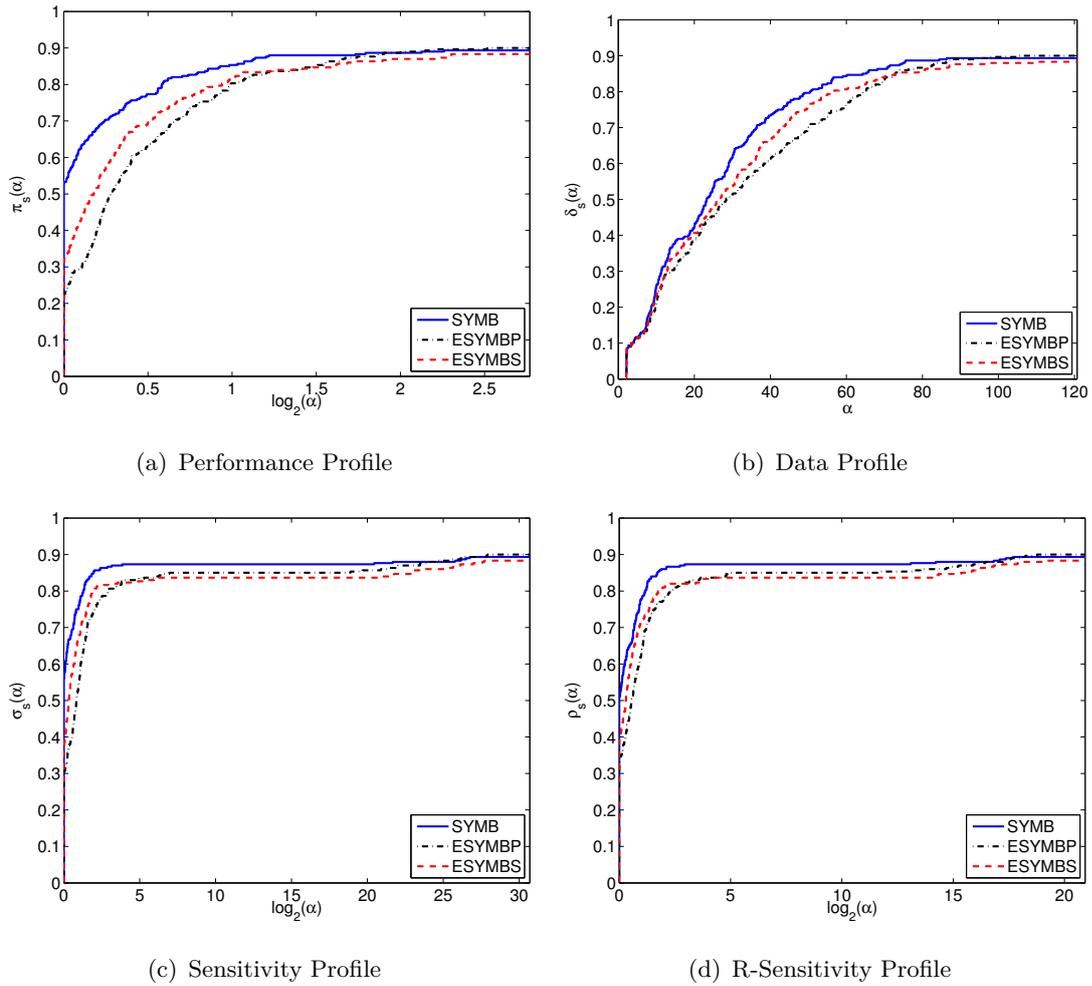


图 4.2: SYMB、ESYMBP 与 ESYMBS 的数值表现 (精度 $\tau = 10^{-4}$)

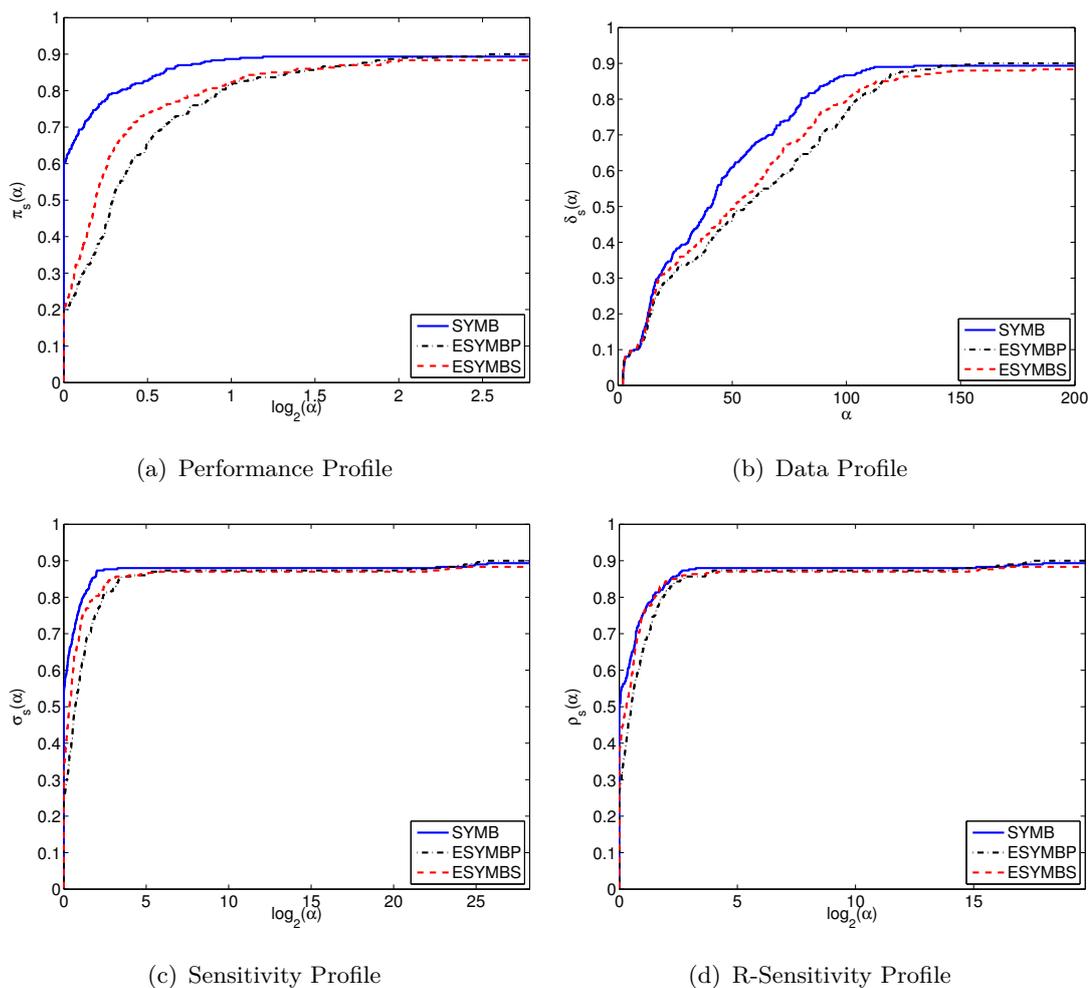
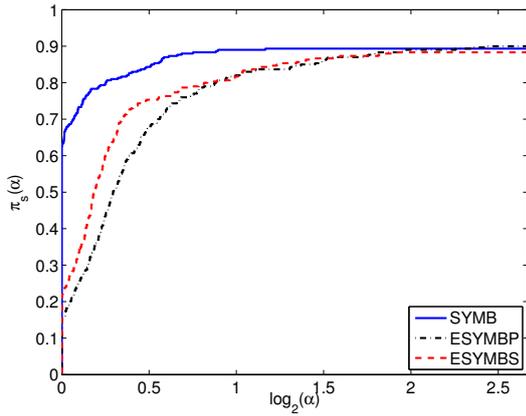
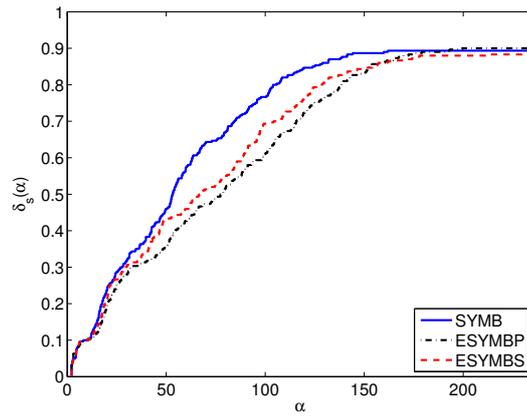


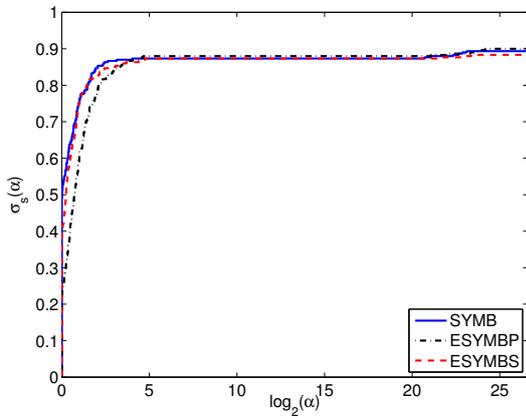
图 4.3: SYMB、ESYMBP 与 ESYMBS 的数值表现 (精度 $\tau = 10^{-6}$)



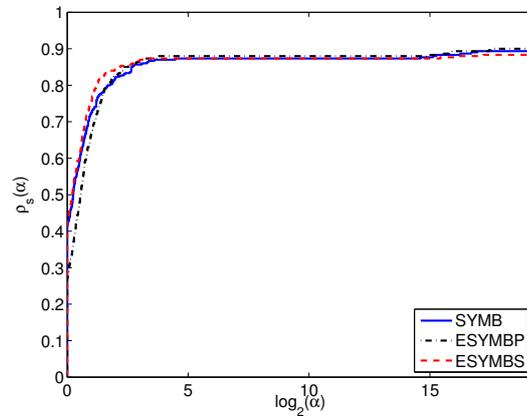
(a) Performance Profile



(b) Data Profile

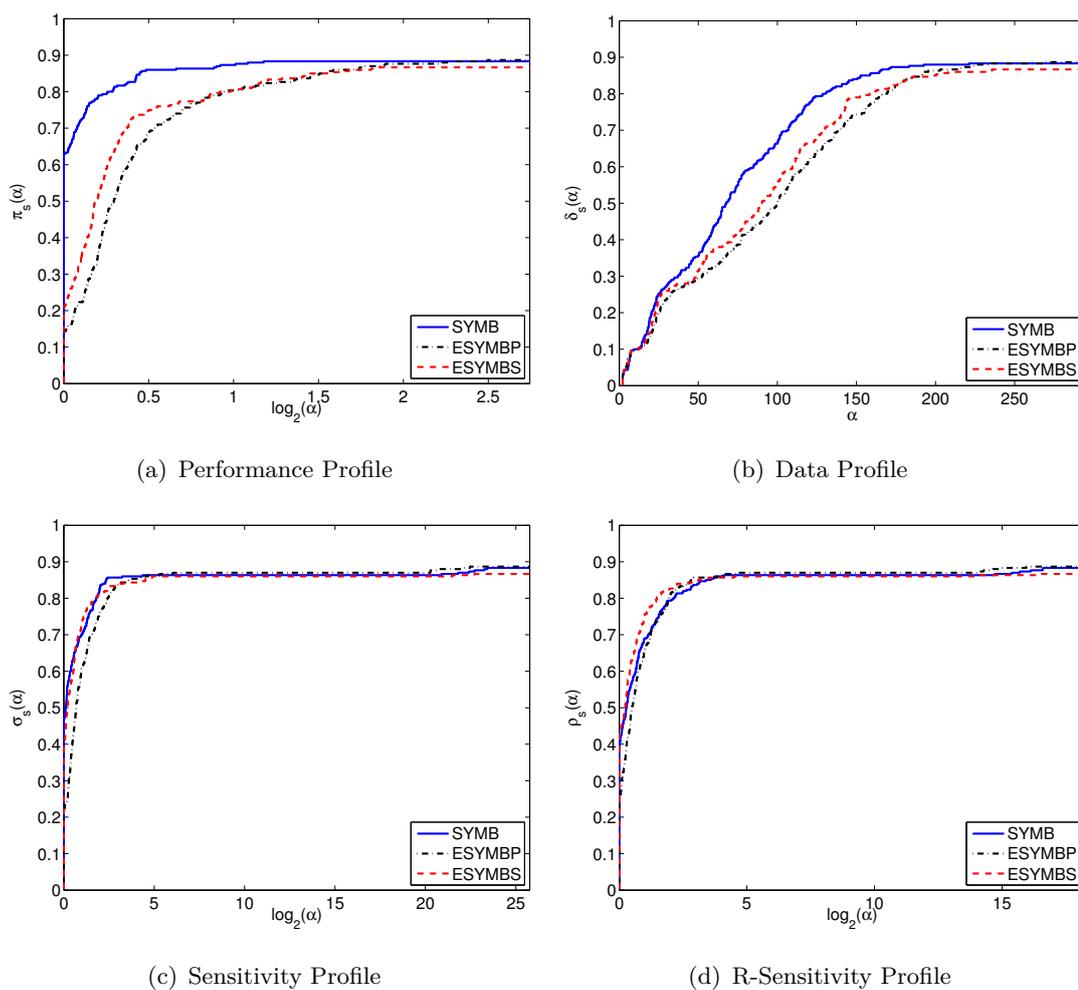


(c) Sensitivity Profile



(d) R-Sensitivity Profile

图 4.4: SYMB、ESYMBP 与 ESYMBS 的数值表现 (精度 $\tau = 10^{-8}$)

图 4.5: SYMB、ESYMBP 与 ESYMBS 的数值表现 (精度 $\tau = 10^{-10}$)

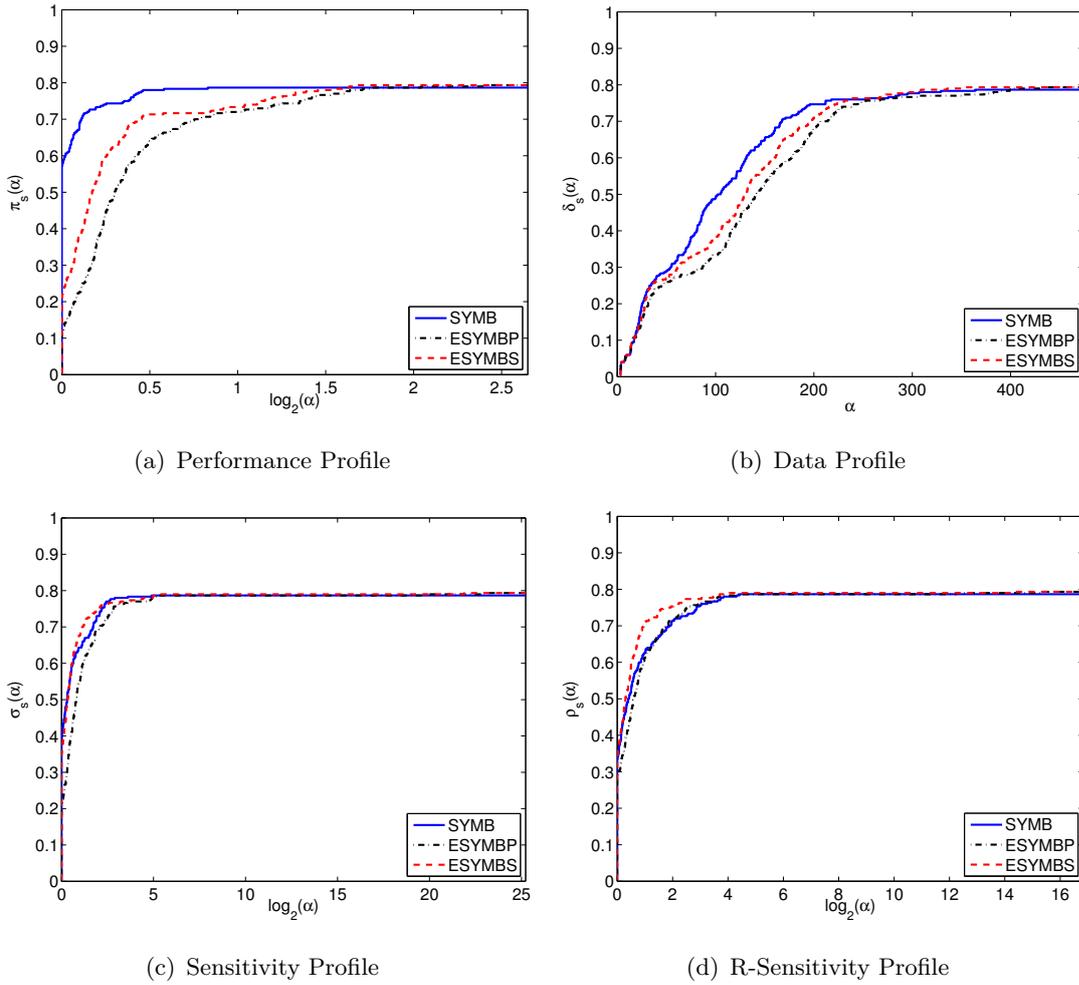


图 4.6: SYMB、ESYMBP 与 ESYMBS 的数值表现 (迭代自然终止)

§ 4.6 二次函数在 ℓ_p 球上的 H^0 范数与 H^1 半范数

本节对二次函数在 ℓ_p ($1 \leq p \leq \infty$) 球上的 H^0 范数与 H^1 半范数做一个统一的讨论. 我们将用二次函数的系数给出这些范数与半范数的显式表达式. 这些结果对于使用 ℓ_p 球做信赖域的方法可能是有用的.

本节使用的记号较多, 我们在这里给出统一的说明. 这些符号仅在本节适用. 其他章节请参考章节内部说明或遵照惯例.

- Q 为如下的二次函数

$$Q(x) = \frac{1}{2}x^T Bx + g^T x + c, \quad x \in \mathbb{R}^n, \quad (4.59)$$

其中 $B \in \mathbb{R}^{n \times n}$ 为对称矩阵, $g \in \mathbb{R}^n$ 为向量, c 为一个实数.

- D 为 B 的对角线. 换句话说, $D \in \mathbb{R}^{n \times n}$ 为一个对角矩阵, 其元素为矩阵 B 的对角元.
- p 为一个不小于 1 的实数, 或 $p = \infty$.
- r 为一个正数.
- \mathcal{B}_p^r 为 \mathbb{R}^n 中以 0 为球心, 以 r 为半径的 ℓ_p 球:

$$\mathcal{B}_p^r = \{x \in \mathbb{R}^n; \|x\|_p \leq r\}. \quad (4.60)$$

- V_p 为 \mathcal{B}_p^1 的体积.
- 给定 $x \in \mathbb{R}^n$ 与 $i \in \{1, 2, \dots, n\}$, x_i 为 x 的第 i 个坐标分量⁵.
- $\Gamma(\cdot)$ 为 Gamma 函数, $\beta(\cdot, \cdot)$ 为 Beta 函数⁶ [5, 151], 即

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt, \quad z \in \mathbb{C}, \operatorname{Re}(z) > 0, \quad (4.61)$$

$$\beta(u, v) = \int_0^1 t^{u-1} (1-t)^{v-1} dt, \quad u, v \in \mathbb{C}, \operatorname{Re}(u) > 0, \operatorname{Re}(v) > 0. \quad (4.62)$$

- γ 为欧拉常数 (Euler's constant) [5, 151].

⁵这与我们在其他章节中的记号不同. 其他章节中我们用 x_k 表示第 k 个迭代点, 用 x_0 表示一个定点.

⁶Beta 函数的标准记号是 B , 但这里我们用 B 表示二次函数 Q 的 Hessian 矩阵.

§4.6.1 关于 Gamma 函数的两个命题

本小节给出关于 Gamma 函数的两个命题. 它们将在后面的讨论中用到.

命题 4.8. 任给实数 $\lambda > 0$, 有

$$\lim_{t \rightarrow 0^+} \frac{\Gamma(t)}{\Gamma(\lambda t)} = \lambda. \quad (4.63)$$

证明. 根据 Gamma 函数的 Weierstrass 乘积 [5] 表示, 我们有

$$\Gamma(t) = \frac{e^{-\gamma t}}{t} \prod_{k=1}^{\infty} \left[\left(1 + \frac{t}{k}\right)^{-1} e^{\frac{t}{k}} \right]. \quad (4.64)$$

因此, 只要证明

$$\lim_{t \rightarrow 0^+} \frac{\prod_{k=1}^{\infty} (1 + \frac{t}{k})^{-1} e^{\frac{t}{k}}}{\prod_{k=1}^{\infty} (1 + \frac{\lambda t}{k})^{-1} e^{\frac{\lambda t}{k}}} = 1 \quad (4.65)$$

即可. 该式可由下面的估计得到.

任给 $t \in (0, \frac{1}{2(1+|\lambda|)})$,

$$\begin{aligned} & \left| \log \frac{\prod_{k=1}^{\infty} (1 + \frac{t}{k})^{-1} e^{\frac{t}{k}}}{\prod_{k=1}^{\infty} (1 + \frac{\lambda t}{k})^{-1} e^{\frac{\lambda t}{k}}} \right| \\ &= \left| \sum_{k=1}^{\infty} \left[\frac{t}{k} - \log \left(1 + \frac{t}{k}\right) \right] - \sum_{k=1}^{\infty} \left[\frac{\lambda t}{k} - \log \left(1 + \frac{\lambda t}{k}\right) \right] \right| \\ &\leq \sum_{k=1}^{\infty} \left| \frac{(1-\lambda)t}{k} - \log \left[1 + \frac{(1-\lambda)t}{k + \lambda t} \right] \right| \\ &= \sum_{k=1}^{\infty} \left| \frac{(1-\lambda)t}{k} - \frac{(1-\lambda)t}{k + \lambda t} + \frac{1}{2(1 + \xi_k)^2} \cdot \frac{(1-\lambda)^2 t^2}{(k + \lambda t)^2} \right| \\ &\leq \sum_{k=1}^{\infty} \left[\frac{|1-\lambda|\lambda t^2}{k^2} + \frac{(1-\lambda)^2 t^2}{2(1/2)^2 k^2} \right] \\ &= \frac{\pi^2}{6} (\lambda + 2|1-\lambda|) |1-\lambda| t^2, \end{aligned} \quad (4.66)$$

其中 ξ_k 来自 Taylor 展式的 Lagrange 余项, 且 $t \in (0, \frac{1}{2(1+|\lambda|)})$ 蕴含 $\xi_k \geq -\frac{1}{2}$. ■

命题 4.9. 设 λ 与 μ 为正实数, 且 $\lambda + \mu \geq 2$. 那么函数

$$\phi(t) = \frac{\Gamma(\lambda t)\Gamma(\mu t)}{[\Gamma(t)]^{\lambda+\mu}} \quad (4.67)$$

于 $(0, \infty)$ 单调递增; 该单调性是严格的, 除非 $\lambda = 1 = \mu$.

证明. 我们只考虑 λ 与 μ 至少有一个不为 1 的情况, 并通过考察导数证明 $\log \phi(t)$ 的严格单调性. 为此, 考虑 *digamma* 函数

$$\psi(t) = \frac{d}{dt} \log \Gamma(t) \quad (4.68)$$

以及它的部分分式展式 [5]

$$\psi(t) = -\gamma - \frac{1}{t} + \sum_{k=1}^{\infty} \frac{t}{k(t+k)}. \quad (4.69)$$

对任何 $t > 0$,

$$\begin{aligned} & \frac{d}{dt} \log \phi(t) \\ &= \lambda \psi(\lambda t) + \mu \psi(\mu t) - (\lambda + \mu) \psi(t) \\ &= \lambda \left[-\gamma - \frac{1}{\lambda t} + \sum_{k=1}^{\infty} \frac{\lambda t}{k(\lambda t + k)} \right] + \mu \left[-\gamma - \frac{1}{\mu t} + \sum_{k=1}^{\infty} \frac{\mu t}{k(\mu t + k)} \right] \\ & \quad - (\lambda + \mu) \left[-\gamma - \frac{1}{t} + \sum_{k=1}^{\infty} \frac{t}{k(t+k)} \right] \\ &= \frac{1}{t}(\lambda + \mu - 2) + \sum_{k=1}^{\infty} \frac{t}{k} \left(\frac{\lambda^2}{\lambda t + k} + \frac{\mu^2}{\mu t + k} - \frac{\lambda + \mu}{t + k} \right) \\ &= \frac{1}{t}(\lambda + \mu - 2) + \sum_{k=1}^{\infty} \frac{kt[(\lambda^2 + \mu^2) - (\lambda + \mu)] + t^2\lambda\mu(\lambda + \mu - 2)}{(\lambda t + k)(\mu t + k)(t + k)} \\ &> 0, \end{aligned} \quad (4.70)$$

最后一个不等式是由于我们对 λ 与 μ 的假定蕴含

$$\lambda^2 + \mu^2 > \lambda + \mu. \quad (4.71)$$

■

§4.6.2 H^0 范数与 H^1 半范数的表达式

本小节给出 Q 在 \mathcal{B}_p^r 上的 H^0 范数与 H^1 半范数的表达式. 这些范数与半范数可以由 B 、 g 和 c 显式给出. 我们把这些表达式列在定理 4.10 中. 需要注意的是, $p = \infty$ 的情形也涵盖在定理 4.10 给出的表达式中, 这一点我们将在注 4.11 中做出说明. 定理 4.10 的证明将在 §4.6.3 给出.

定理 4.10.

$$\|Q\|_{H^0(\mathcal{B}_p^r)}^2 = \frac{I}{2}\|B\|_F^2 + \frac{I}{4}\text{Tr}^2 B + \frac{J-3I}{4}\|D\|_F^2 + K(c\text{Tr}B + \|g\|_2^2) + Vc^2; \quad (4.72)$$

$$|Q|_{H^1(\mathcal{B}_p^r)}^2 = K\|B\|_F^2 + V\|g\|_2^2. \quad (4.73)$$

其中,

$$I = \frac{[\Gamma(\frac{3}{p})]^2\Gamma(\frac{n}{p})}{[\Gamma(\frac{1}{p})]^2\Gamma(\frac{n+4}{p})} \cdot \frac{n}{n+4} \cdot V_p r^{n+4}, \quad (4.74)$$

$$J = \frac{\Gamma(\frac{5}{p})\Gamma(\frac{n}{p})}{\Gamma(\frac{1}{p})\Gamma(\frac{n+4}{p})} \cdot \frac{n}{n+4} \cdot V_p r^{n+4}, \quad (4.75)$$

$$K = \frac{\Gamma(\frac{3}{p})\Gamma(\frac{n}{p})}{\Gamma(\frac{1}{p})\Gamma(\frac{n+2}{p})} \cdot \frac{n}{n+2} \cdot V_p r^{n+2}, \quad (4.76)$$

$$V = V_p r^n. \quad (4.77)$$

注 4.11. 定理 4.10 涵盖了 $p = \infty$ 的情形. 当 $p = \infty$ 时, 我们在极限的意义下理解 (4.74)–(4.76), 也就是说

$$I = \lim_{p \rightarrow \infty} \frac{[\Gamma(\frac{3}{p})]^2\Gamma(\frac{n}{p})}{[\Gamma(\frac{1}{p})]^2\Gamma(\frac{n+4}{p})} \cdot \frac{n}{n+4} \cdot V_\infty r^{n+4}, \quad (4.74')$$

$$J = \lim_{p \rightarrow \infty} \frac{\Gamma(\frac{5}{p})\Gamma(\frac{n}{p})}{\Gamma(\frac{1}{p})\Gamma(\frac{n+4}{p})} \cdot \frac{n}{n+4} \cdot V_\infty r^{n+4}, \quad (4.75')$$

$$K = \lim_{p \rightarrow \infty} \frac{\Gamma(\frac{3}{p})\Gamma(\frac{n}{p})}{\Gamma(\frac{1}{p})\Gamma(\frac{n+2}{p})} \cdot \frac{n}{n+2} \cdot V_\infty r^{n+2}. \quad (4.76')$$

以上极限的具体值可以由命题 4.8 容易地得到.

作为定理 4.10 的特例, 我们给出 $p = 2$ 的情况如下.

推论 4.12.

$$\|Q\|_{H^0(\mathcal{B}_2^r)}^2 = V_2 r^n \left[\frac{r^4 (2\|B\|_F^2 + \text{Tr}^2 B)}{4(n+2)(n+4)} + \frac{r^2(c\text{Tr}B + \|g\|_2^2)}{n+2} + c^2 \right]; \quad (4.77)$$

$$|Q|_{H^1(\mathcal{B}_2^r)}^2 = V_2 r^n \left[\frac{r^2}{n+2} \|B\|_F^2 + \|g\|_2^2 \right]. \quad (4.78)$$

公式 (4.78) 已经在 §4.3 中给出.

§4.6.3 定理 4.10 的证明

本小节给出定理 4.10 的证明.

下面我们假设 $n \geq 2$, 因为 $n = 1$ 的情形是平凡的.

为方便计, 我们先给出几个引理. 我们首先考察几个单项式的积分, 相关结果列在以下两个引理中.

引理 4.13. 设 $1 \leq i, j, k, l \leq n$.

- a.) x_i 与 $x_i x_j x_k$ 在 \mathcal{B}_p^r 上的积分为 0.
- b.) $x_i x_j$ 与 $x_i x_j x_k^2$ 在 \mathcal{B}_p^r 上的积分为 0, 只要 $i \neq j$.
- c.) $x_i x_j x_k x_l$ 在 \mathcal{B}_p^r 上的积分为 0, 只要 i, j, k 与 l 两两不同.

引理 4.13 是平凡的, 我们略去证明.

引理 4.14. 设 $1 \leq i < j \leq n$, 且 k_1, k_2 为非负偶数, 则

$$\int_{\mathcal{B}_p^r} x_i^{k_1} x_j^{k_2} dx = \frac{\Gamma(\frac{k_1+1}{p})\Gamma(\frac{k_2+1}{p})\Gamma(\frac{n}{p})}{\Gamma(\frac{1}{p})\Gamma(\frac{1}{p})\Gamma(\frac{n+k_1+k_2}{p})} \cdot \frac{n}{n+k_1+k_2} \cdot V_p r^{n+k_1+k_2}. \quad (4.79)$$

注 4.15. 引理 4.14 涵盖了 $p = \infty$ 的情形. 当 $p = \infty$, 时, 我们在极限的意义下理解 (4.79), 也就是说,

$$\int_{\mathcal{B}_\infty^r} x_i^{k_1} x_j^{k_2} dx = \lim_{p \rightarrow \infty} \frac{\Gamma(\frac{k_1+1}{p})\Gamma(\frac{k_2+1}{p})\Gamma(\frac{n}{p})}{\Gamma(\frac{1}{p})\Gamma(\frac{1}{p})\Gamma(\frac{n+k_1+k_2}{p})} \cdot \frac{n}{n+k_1+k_2} \cdot V_\infty r^{n+k_1+k_2}. \quad (4.79')$$

证明. 不失一般性, 假设 $r = 1$. 另外, 我们还假设 $n \geq 3$, 因为 $n = 2$ 的情形是平凡的. 记 \mathbb{R}^{n-2} 中 ℓ_p 单位球的体积为 $V_{p,n-2}$.

我们首先证明 (4.79) 对 $p \in [1, \infty)$ 成立, 然后证明它在极限的意义下对 $p = \infty$ 也成立.

当 $p \in [1, \infty)$ 时,

$$\begin{aligned}
 & \int_{B_p^1} x_i^{k_1} x_j^{k_2} dx \\
 &= \int_{|u|^p + |v|^p \leq 1} u^{k_1} v^{k_2} du dv \int_{w \in \mathbb{R}^{n-2}, \|w\|_p \leq (1 - |u|^p - |v|^p)^{\frac{1}{p}}} dw \\
 &= V_{p,n-2} \int_{|u|^p + |v|^p \leq 1} u^{k_1} v^{k_2} (1 - |u|^p - |v|^p)^{\frac{n-2}{p}} du dv \\
 &= 4V_{p,n-2} \int_{u^p + v^p \leq 1, u, v \geq 0} u^{k_1} v^{k_2} (1 - u^p - v^p)^{\frac{n-2}{p}} du dv.
 \end{aligned} \tag{4.80}$$

考虑变换

$$u^{\frac{p}{2}} = \rho \cos \theta, \tag{4.81}$$

$$v^{\frac{p}{2}} = \rho \sin \theta, \tag{4.82}$$

我们有

$$\begin{aligned}
 & \int_{B_p^1} x_i^{k_1} x_j^{k_2} dx \\
 &= \frac{16}{p^2} V_{p,n-2} \int_0^{\frac{\pi}{2}} (\cos \theta)^{\frac{2k_1+2}{p}-1} (\sin \theta)^{\frac{2k_2+2}{p}-1} d\theta \int_0^1 \rho^{\frac{2k_1+2k_2+4}{p}-1} (1 - \rho^2)^{\frac{n-2}{p}} d\rho \\
 &= \frac{4}{p^2} V_{p,n-2} \beta\left(\frac{k_1+1}{p}, \frac{k_2+1}{p}\right) \beta\left(\frac{k_1+k_2+2}{p}, \frac{n-2}{p} + 1\right).
 \end{aligned} \tag{4.83}$$

令 k_1 与 k_2 为 0, 我们得到

$$V_p = \frac{4}{p^2} V_{p,n-2} \beta\left(\frac{1}{p}, \frac{1}{p}\right) \beta\left(\frac{2}{p}, \frac{n-2}{p} + 1\right). \tag{4.84}$$

因而

$$\begin{aligned}
 & \int_{B_p^1} x_i^{k_1} x_j^{k_2} dx \\
 &= \frac{\beta\left(\frac{k_1+1}{p}, \frac{k_2+1}{p}\right) \beta\left(\frac{k_1+k_2+2}{p}, \frac{n-2}{p} + 1\right)}{\beta\left(\frac{1}{p}, \frac{1}{p}\right) \beta\left(\frac{2}{p}, \frac{n-2}{p} + 1\right)} V_p \\
 &= \frac{\Gamma\left(\frac{k_1+1}{p}\right) \Gamma\left(\frac{k_2+1}{p}\right) \Gamma\left(\frac{k_1+k_2+2}{p}\right) \Gamma\left(\frac{n-2}{p} + 1\right) \Gamma\left(\frac{2}{p}\right) \Gamma\left(\frac{n}{p} + 1\right)}{\Gamma\left(\frac{1}{p}\right) \Gamma\left(\frac{1}{p}\right) \Gamma\left(\frac{2}{p}\right) \Gamma\left(\frac{n-2}{p} + 1\right) \Gamma\left(\frac{k_1+k_2+2}{p}\right) \Gamma\left(\frac{n+k_1+k_2}{p} + 1\right)} V_p \\
 &= \frac{\Gamma\left(\frac{k_1+1}{p}\right) \Gamma\left(\frac{k_2+1}{p}\right) \Gamma\left(\frac{n}{p}\right)}{\Gamma\left(\frac{1}{p}\right) \Gamma\left(\frac{1}{p}\right) \Gamma\left(\frac{n+k_1+k_2}{p}\right)} \cdot \frac{n}{n+k_1+k_2} V_p.
 \end{aligned} \tag{4.85}$$

故 (4.79) 对有限的 p 成立.

现在我们考虑 $p = \infty$ 之情形. 根据 Lebesgue 控制收敛定理,

$$\int_{B_p^1} x_i^{k_1} x_j^{k_2} dx \rightarrow \int_{B_\infty^1} x_i^{k_1} x_j^{k_2} dx, \quad V_p \rightarrow V_\infty \quad (p \rightarrow \infty). \tag{4.86}$$

(4.86) 与 (4.79) 说明 $p \rightarrow \infty$ 时 $\frac{\Gamma\left(\frac{k_1+1}{p}\right)\Gamma\left(\frac{k_2+1}{p}\right)\Gamma\left(\frac{n}{p}\right)}{\Gamma\left(\frac{1}{p}\right)\Gamma\left(\frac{1}{p}\right)\Gamma\left(\frac{n+k_1+k_2}{p}\right)}$ 是收敛的, 且 (4.79') 成立. 也就是说, (4.79) 在极限的意义下对 $p = \infty$ 成立. \blacksquare

注 4.16. 直接积分, 我们可以得到

$$\int_{B_\infty^r} x_i^{k_1} x_j^{k_2} dx = \frac{V_\infty r^{n+k_1+k_2}}{(k_1+1)(k_2+1)}. \tag{4.87}$$

根据命题 4.8, 这与 (4.79') 是相同的.

现在我们再考虑几个积分.

引理 4.17. 分别记 $\int_{B_p^r} x_1^2 x_2^2 dx$ 、 $\int_{B_p^r} x_1^4 dx$ 与 $\int_{B_p^r} x_1^2 dx$ 为 I 、 J 与 K . 我们有如下等式.

a.)

$$\int_{B_p^r} g^T x dx = 0; \tag{4.88}$$

$$\int_{B_p^r} g^T B x dx = 0; \tag{4.89}$$

$$\int_{B_p^r} (g^T x)(x^T B x) dx = 0. \tag{4.90}$$

b.)

$$\int_{\mathcal{B}_p^r} x^T B x \, dx = K \operatorname{Tr} B; \quad (4.91)$$

$$\int_{\mathcal{B}_p^r} x^T B^2 x \, dx = K \|B\|_F^2; \quad (4.92)$$

$$\int_{\mathcal{B}_p^r} (x^T B x)^2 \, dx = I(2\|B\|_F^2 + \operatorname{Tr}^2 B) + (J - 3I)\|D\|_F^2; \quad (4.93)$$

$$\int_{\mathcal{B}_p^r} (g^T x)^2 \, dx = K \|g\|_2^2. \quad (4.94)$$

证明. (4.88)–(4.90) 可由引理 4.13 直接得到. 至于 (4.91)–(4.94), 我们仅证明 (4.93) 作为示例. 记 B 的 (i, j) 元素为 B_{ij} . 由引理 4.13,

$$\begin{aligned} & \int_{\mathcal{B}_p^r} (x^T B x)^2 \, dx \\ &= \int_{\mathcal{B}_p^r} \sum_{i,j,k,l} (x_i B_{ij} x_j)(x_k B_{kl} x_l) \, dx \\ &= \int_{\mathcal{B}_p^r} \left(\sum_i B_{ii}^2 x_i^4 + 2 \sum_{i \neq j} B_{ij}^2 x_i^2 x_j^2 + \sum_{i \neq j} B_{ii} B_{jj} x_i^2 x_j^2 \right) \, dx \quad (4.95) \\ &= J \|D\|_F^2 + 2I(\|B\|_F^2 - \|D\|_F^2) + I(\operatorname{Tr}^2 B - \|D\|_F^2) \\ &= I(2\|B\|_F^2 + \operatorname{Tr}^2 B) + (J - 3I)\|D\|_F^2. \end{aligned}$$

■

有了引理 4.14 与引理 4.17, 定理 4.10 的证明已是水到渠成. 我们现在给出该证明.

证明. 根据引理 4.17, 我们有

$$\begin{aligned} & \|Q\|_{H^0(\mathcal{B}_p^r)}^2 \\ &= \int_{\mathcal{B}_p^r} \left[\frac{1}{2} x^T B x + g^T x + c \right]^2 \, dx \\ &= \int_{\mathcal{B}_p^r} \left[\frac{1}{4} (x^T B x)^2 + (g^T x)^2 + c x^T B x + c^2 \right] \, dx \quad (4.96) \\ &= \frac{1}{4} I(2\|B\|_F^2 + \operatorname{Tr}^2 B) + \frac{1}{4} (J - 3I)\|D\|_F^2 \\ &\quad + K(\|g\|_2^2 + c \operatorname{Tr} B) + V_p r^n c^2, \end{aligned}$$

以及

$$\begin{aligned}
 & |Q|_{H^1(\mathcal{B}_p^r)}^2 \\
 &= \int_{\mathcal{B}_p^r} \|Bx + g\|_2^2 dx \\
 &= \int_{\mathcal{B}_p^r} [x^T B^2 x + \|g\|_2^2] dx \\
 &= K \|B\|_F^2 + V_p r^n \|g\|_2^2.
 \end{aligned} \tag{4.97}$$

现在使用引理 4.14 即可得到定理 4.10 的结论. 因引理 4.14 在极限的意义下对 $p = \infty$ 成立, 故定理 4.10 亦然. ■

§ 4.6.4 讨论

§ 4.6.4.1 正交不变性

我们称 \mathcal{Q} 上的泛函 \mathcal{F} 是正交不变的, 若

$$\mathcal{F}(q \circ T) = \mathcal{F}(q). \tag{4.98}$$

对任何 $q \in \mathcal{Q}$ 与 \mathbb{R}^n 上的任何正交变换 T 成立.

命题 4.18.

a.) $\|\cdot\|_{H^0(\mathcal{B}_p^r)}$ 是正交不变的当且仅当 $p = 2$.

b.) 对任何 $p \geq 1$, $|\cdot|_{H^1(\mathcal{B}_p^r)}$ 是正交不变的.

证明. 根据定理 4.10, $\|\cdot\|_{H^0(\mathcal{B}_p^r)}$ 正交不变当且仅当

$$J = 3I. \tag{4.99}$$

这是因为 $\|B\|_F$ 、 $\text{Tr}B$ 与 $\|g\|_2$ 皆正交不变, 而 $\|D\|_F$ 则不然. 等式 (4.99) 等价于

$$\frac{\Gamma\left(\frac{5}{p}\right) \Gamma\left(\frac{1}{p}\right)}{\left[\Gamma\left(\frac{3}{p}\right)\right]^2} = 3. \tag{4.100}$$

由命题 4.9, (4.100) 左边对 $p \in (0, \infty)$ 严格单调递减, 因而 (4.100) 至多有一解. 显然 $p = 2$ 是一解. 故所论成立.

根据定理 4.10, $|\cdot|_{H^1(\mathcal{B}_p^r)}$ 的正交不变性显然. ■

§4.6.5 ℓ_p 椭球上的 H^0 范数与 H^1 半范数

当考虑以 ℓ_p 椭球为信赖域的方法时, 二次函数在 ℓ_p 椭球上的范数与半范数可能是有用的. \mathbb{R}^n 中的 ℓ_p 椭球是指

$$\{x \in \mathbb{R}^n; \|A(x - x_0)\|_p \leq r\}, \quad (4.101)$$

其中 $A \in \mathbb{R}^{n \times n}$ 为一个非奇异矩阵. 由定理 4.10, 二次函数在 ℓ_p 椭球上的 H^0 范数与 H^1 半范数可以很容易地通过变量替换得到.

§4.6.6 带权重的 H^0 范数与 H^1 半范数

如果我们对二次函数在不同位置处的性质关注度不同, 考虑带权重的 H^0 范数与 H^1 半范数是有意义的. 在 \mathcal{B}_p^r 上, 带权重的 H^0 范数与 H^1 半范数是指

$$\|Q\|_{H^0(\mathcal{B}_p^r, w)} = \left[\int_{\mathcal{B}_p^r} w(x) |f(x)|^2 dx \right]^{1/2}, \quad (4.102)$$

$$|Q|_{H^1(\mathcal{B}_p^r, w)} = \left[\int_{\mathcal{B}_p^r} w(x) \|\nabla f(x)\|_2^2 dx \right]^{1/2}, \quad (4.103)$$

其中 w 为 \mathcal{B}_p^r 上的非负可积函数, 表示权重. 这样的范数或半范数可以用 §4.6.3 中的方法计算, 主要工作量是计算相关单项式的加权积分.

作为一个例子, 对于权函数

$$w(x) = r^2 - \|x\|_2^2, \quad (4.104)$$

我们不加证明地给出 $\|Q\|_{H^0(\mathcal{B}_2^r, w)}$ 与 $|Q|_{H^1(\mathcal{B}_2^r, w)}$ 如下.

命题 4.19.

$$\|Q\|_{H^0(\mathcal{B}_2^r, w)}^2 = \frac{2V_2 r^{n+2}}{n+2} \left[\frac{r^4 (2\|B\|_F^2 + \text{Tr}^2 B)}{4(n+4)(n+6)} + \frac{r^2 (c \text{Tr} B + \|g\|_2^2)}{n+4} + c^2 \right]; \quad (4.105)$$

$$|Q|_{H^1(\mathcal{B}_2^r, w)}^2 = \frac{2V_2 r^{n+2}}{n+2} \left[\frac{r^2}{n+4} \|B\|_F^2 + \|g\|_2^2 \right]. \quad (4.106)$$

§ 4.7 总结

最小范数插值在无导数优化中有广泛的应用. 受其目标泛函的启发, 我们研究了二次函数在 l_2 球上的 H^1 半范数. 我们用二次函数的系数给出了这种半范数的显式表达式. 我们发现, 最小范数插值原来是在一个球上极小化插值函数的 H^1 半范数; 而且, 插值中的参量 σ 与 x_0 分别决定了球的半径与球心. 这些观察提供了一种理解最小范数插值的新视角.

借助于这些新的观察, 我们研究了 Powell [144, 146] 提出的扩展的对称 Broyden 修正. 我们讨论了如何根据 σ 与 x_0 的几何意义为修正选取合适的参量. 我们的讨论导致了与 Powell [144, 146] 相同的 x_0 选取方式, 但 σ 的选取则是新的. 我们对 σ 的选择十分直观和简单. 在我们的数值实验中, 新的 σ 表现优于 Powell [144, 146] 选取的 σ . 这从一个侧面说明我们的理论的确为理解最小范数插值提供了一种有效的工具.

一个有意思的问题是, 作为无导数优化算法中建立模型的方法, 扩展的对称 Broyden 修正是否可以改善原始的对称 Broyden 修正. 到目前为止, 我们无法给出一个肯定的回答. 然而, Powell [144, 146] 的数值实验表明扩展版本的修正有助于提高算法的局部收敛速度. 如果能在理论上证明这种提高将是十分有意思的. 我们认为, 在得到任何理论性的证据之前, 对扩展的对称 Broyden 修正下任何结论都还为时尚早.

除了在 l_2 球上的 H^1 半范数之外, 我们还对二次函数在 l_p ($1 \leq p \leq \infty$) 球上的 H^0 范数和 H^1 半范数做了统一的讨论. 我们用二次函数的系数给出了这些范数和半范数的显式表达式. 这些结果对于使用 l_p 球作为信赖域的方法可能是有用的. 值得注意的是, 对所有 $p \in [1, \infty]$, 二次函数 Q 在 l_p 球

$$\mathcal{B}_p^r(x_0) = \{x \in \mathbb{R}^n : \|x - x_0\|_p \leq r\} \quad (4.107)$$

上的 H^1 半范数的平方都是 $\|\nabla^2 Q\|_F^2$ 与 $\|\nabla Q(x_0)\|_2^2$ 的组合. 但我们在 § 4.5.1 中仅以 $p = 2$ 的情况为依据进行了讨论, 这是因为我们考虑的信赖域是 l_2 球.

第五章 子空间技巧在无导数优化中的应用

§ 5.1 引言

到目前为止, 对于无约束优化问题, 无导数优化方法能处理的问题规模尚相对有限. 在前面的章节中, 我们也主要考虑了小规模问题. 有研究者建议, 使用无导数算法的问题规模最好比较适中, 比如不超过 100 维 [38, 42]. 这其中的一个原因是, \mathbb{R}^n 中的二次模型有 $(n+1)(n+2)/2$ 个自由度, 即使 n 只有几百, 要建立一个二次模型也是很困难的. 随着无导数方法研究的进展, 这一状况得到了一定改善. 比如, 第三章介绍的欠定二次插值使得人们能够用远少于 $(n+1)(n+2)/2$ 个函数值信息建立二次模型, 并且依然可以提取有效的二阶信息 [10, 37, 38, 137, 139, 138, 142, 146]. NEWUOA 是利用欠定二次插值模型的一个代表性方法, 它能够求解的问题规模可以达到 320 维 [138, 142], 已经属于规模很大的问题. 另一种解决规模较大问题的途径是针对特殊结构的问题设计算法, 比如部分可分问题¹ [30, 31, 32, 148, 163], 最小二乘问题 [121, 20, 149, 27, 186, 185] 等, 但这往往要求对目标函数的结构有足够的了解.

本章研究另一类解决大规模问题的途径: 子空间方法. 子空间方法的基本想法是, 每一步在全空间 \mathbb{R}^n 的一个低维子空间上极小化目标函数, 得到下一个迭代点. 在传统的非线性优化领域, 子空间方法已经成为处理大规模问题的重要手段. 随着人们感兴趣的问题规模越来越大, 子空间方法日益受到研究者的重视 [183, 33, 112, 111, 169, 108, 170, 182, 81]. 这方面的研究可以参考 Yuan 的综述 [180, 181]. 在无导数优化方法的研究领域, 子空间方法并非一种新的技巧. 恰恰相反, 它是一种十分古典的技巧: 经典的坐标轮换法就是一种子空间方法, 只不过子空间的选取相当原始. 在最近的一些无导数方法中, 子空间技巧也有体现. 比如在 Colson 和 Toint [32] 以及 Price 和 Toint [148] 针对部分可分问题的方法中, 问题的结构导致方法自然具有某些子空间特性. 再比如, Wu 等 [177] 提出了一种基于模式搜索的子空间方法. 每一次迭代中, 该方法在一个近似梯度和上一迭代步张开的一个二维子空间上做模式搜索, 其中近似梯度由差

¹部分可分 (partially separable) 这一概念由 Griewank 和 Toint [71, 72, 70] 提出. Griewank 和 Toint [71] 证明, 部分可分函数类涵盖了所有具有稀疏 Hessian 矩阵的足够光滑函数.

分得到.

本章提出两个无导数子空间方法. 在 §5.2 中, 利用 Hooke-Jeeves 模式搜索的思想, 针对无导数信赖域方法, 我们提出在子空间上求解信赖域子问题的技巧. 数值实验表明, 该技巧能够改善 NEWUOA 算法的表现. 在 §5.3 中, 我们讨论一个很一般的子空间算法框架, 分析其全局收敛性和局部收敛速度, 并且基于 NEWUOA 算法给出该算法框架的一个无导数实现, 我们称之为 NEWUOAs 算法. NEWUOAs 算法是本文中最成功的算法. 在我们的实验中, NEWUOAs 算法无论在计算开销还是稳定性上都明显优于 NEWUOA 算法, 后者被认为是目前最优秀的无导数算法之一. 我们的实验还表明, NEWUOAs 算法很适合处理初始点质量较差的问题. 这对实际应用领域很有意义, 因为很多实际问题很难给出一个好的初始点. 不仅如此, 一个令人振奋的事实是, NEWUOAs 算法已经能高效求解某些维数高达 2000 的问题. 这是一个突破, 因为目前大多数无导数优化算法 (包括 NEWUOA 算法) 至多可以求解几百维的问题; 对于它们, 2000 维的问题几乎是不可求解的.

§5.2 无导数子空间方法 I

§5.2.1 算法的提出

考虑基于二次插值模型的无导数信赖域方法. 像传统信赖域方法一样, 这种方法每一步建立目标函数的一个模型, 然后在一个信赖域上极小化该模型, 得到一个新的试探点. 由于导数信息不可知, 这里的模型通过二次插值来建立. 第三章和第四章已经详细介绍了如何建立这种模型. 下面是一个基于二次插值模型的无导数信赖域算法框架.

算法 5.1.

步 1. 给定初始点 x_1 和初始插值点集 \mathcal{I}_1 满足 $x_1 \in \mathcal{I}_1$, 且

$$f(x_1) = \min_{y \in \mathcal{I}_1} f(y). \quad (5.1)$$

选择初始信赖域半径 Δ_1 . $k := 1$.

步 2. 构造二次模型 Q_k 满足插值条件

$$Q_k(y) = f(y), \quad y \in \mathcal{I}_k. \quad (5.2)$$

步 3. 求解信赖域子问题

$$\min_{d \in \mathbb{R}^n} Q_k(x_k + d) \quad (5.3)$$

$$\text{s.t. } \|d\| \leq \Delta_k, \quad (5.4)$$

得到试探步 s_k .

步 4. 若 $f(x_k + s_k) < f(x_k)$, 则 $x_{k+1} := x_k + s_k$; 否则 $x_{k+1} := x_k$.

步 5. 判断插值点集的适定性是否良好. 如有必要, 执行几何步以提高插值点集的适定性. 更新插值点集得到 \mathcal{I}_{k+1} , 使得 \mathcal{I}_{k+1} 包含 x_{k+1} .

步 6. 根据 s_k 的表现以及插值点集的适定性更新信赖域半径得到 Δ_{k+1} .
 $k := k + 1$. 转步 2.

我们认为, 一个适合大规模问题的算法, 它的函数值计算次数应该至多是 $O(n)$ 量级的. 因此, 在每一步, 我们至多有 $O(n)$ 的函数值信息可用, 也就是说

$$|\mathcal{I}_k| = O(n). \quad (5.5)$$

然而, \mathbb{R}^n 中一个二次模型却有 $(n+1)(n+2)/2$ 个自由度. 因此, 我们可用于确定模型的信息严重不足. 这就是为什么人们要研究欠定二次插值. 问题规模越大, 上述不足越严重, 从而导致由此建立的模型可信度越低.

除此之外, 基于插值模型的算法所面对的一个共同问题是, 插值点集的适定性会随着迭代而逐步丧失, 从而也导致模型可信度降低. 事实上, 之所以在算法 5.1 中引入几何步, 就是为了在必要时改善插值点集的适定性, 期望得到一个有效的模型.

也就是说, 由于可用信息的严重不足和插值点集适定性的丧失, 无导数信赖域方法中的模型经常可信度较低. 正因为如此, 实际计算中, 这种信赖域方法的很多信赖步都是失败的. 换句话说, 很多时候模型并没有引导我们找到更好的点, 而是误导我们走向了错误的方向. 这与基于导数的方法差别很大.

那么, 如何改善这一状况? 如何在模型可信度较低时提高目标函数获得下降的可能性?

我们采用子空间的技术. 基本思路是, 当认为模型可信度很低时, 我们将信赖域子问题限制在一个子空间上求解. 对该子空间, 我们有两个要求: 第一, 这

一子空间应该很有潜力,也就是说,目标函数很有可能会在该子空间上获得下降;第二,这一子空间的维数应该很低(比如2维或3维).换句话说,对于可信度较低的模型,我们只允许它在一个“有潜力”的低维子空间上工作;因为目标函数在该子空间上取得下降的概率比较大,而子空间维数又很低,所以我们期望模型会得到更多立功的机会.再换句话说,当认为模型可信度较低时,我们转而信任子空间.

以上策略可以视为经典的 Hooke-Jeeves 模式搜索思想的推广. Hooke-Jeeves 模式搜索的思想是,通过当前的信息,确定一个很有可能使函数获得下降的方向,然后集中精力在这一方向上求解.我们这里把方向推广成了子空间.

基于以上的想法,我们把算法 5.1 修改为如下的子空间算法.

算法 5.2.

步 1. 给定初始点 x_1 和初始插值点集 \mathcal{I}_1 满足 $x_1 \in \mathcal{I}_1$, 且

$$f(x_1) = \min_{y \in \mathcal{I}_1} f(y). \quad (5.6)$$

选择初始信赖域半径 Δ_1 . $k := 1$.

步 2. 构造模型 Q_k 满足插值条件

$$Q_k(y) = f(y), \quad y \in \mathcal{I}_k. \quad (5.7)$$

步 3. 根据 Q_k 蕴含的信息以及前面的迭代, 决定一个子空间 \mathcal{S}_k , 求解子空间信赖域子问题

$$\min_{d \in \mathcal{S}_k} Q_k(d) \quad (5.8)$$

$$\text{s.t. } \|d\| \leq \Delta_k, \quad (5.9)$$

得到试探步 s_k .

步 4. 若 $f(x_k + s_k) < f(x_k)$, 则 $x_{k+1} := x_k + s_k$; 否则 $x_{k+1} := x_k$.

步 5. 判断插值点集的适定性是否良好. 如有必要, 执行几何步以提高插值点集的适定性. 更新插值点集得到 \mathcal{I}_{k+1} , 使得 \mathcal{I}_{k+1} 包含 x_{k+1} .

步 6. 根据 s_k 的表现以及插值点集的适定性更新信赖域半径得到 Δ_{k+1} .
 $k := k + 1$. 转步 2.

算法 5.2 的关键问题在于步 3 中子空间 \mathcal{S}_k 的选取. 下面我们就来讨论这一问题.

首先, 当有充分理由在全空间求解信赖域子问题时, 我们应该取 $\mathcal{S}_k = \mathbb{R}^n$. 这包括以下三种情形:

- 情形 1. 若初始插值点集 \mathcal{I}_1 的适定性很好, 则只要插值点集适定性尚未变坏, 模型就都是可以信任的, 此时应当在全空间求解信赖域子问题. 那么, 如何判断插值点集适定性是否已经变坏? 若当前步之前所有信赖域迭代的下降比都不低于某常数 $\alpha_1 > 0$, 则我们认为插值点集适定性尚未变坏. 实际计算中我们取 $\alpha_1 = 0.1$.
- 情形 2. 几何步之后的第一个模型可以信任, 此时应当在全空间求解信赖域子问题. 这是因为, 几何步的目的是改善插值点集的适定性, 从而得到比较有效的模型.
- 情形 3. 若 \mathcal{S}_{k-1} 是全空间, 且第 $k-1$ 步信赖域迭代很成功, 则应当在全空间求解信赖域子问题. 这是因为, 既然上一步在全空间上求解信赖域子问题导致了很大的成功, 那么我们应该在这一步采取相同的策略以望延续这种成功. 实际计算中, 如果第 $k-1$ 步信赖域下降比大于 1, 那么我们认为它是很成功的.

除以上三种情形之外, 我们认为模型可信度较低, 从而取 \mathcal{S}_k 为一个有潜力的低维子空间. 这时, \mathcal{S}_k 应该包含子空间 $\text{span}\{\nabla Q_k(x_k), d_{k-1}\}$. 原因是:

第一, \mathcal{S}_k 应包含 $\nabla Q_k(x_k)$, 否则信赖域子问题 (5.8-5.9) 可能只有零解, 无法得到有效的试探步;

第二, 目标函数很有可能在方向 d_{k-1} 上取得下降, 这受到了 Hooke-Jeeves 模式搜索 [79] 的启发.

事实上, 当目标函数梯度信息可用时, $\text{span}\{\nabla f(x_k), d_{k-1}\}$ 是一个经典的子空间, 它与共轭梯度法密切相关, Yuan 和 Stoer [183] 从这一子空间出发研究了共轭梯度法. 子空间 $\text{span}\{\nabla Q_k(x_k), d_{k-1}\}$ 是该子空间的一个近似. 除此之外,

我们在计算中发现, 方向

$$\bar{d}_k = \sum_{y \in \mathcal{I}_k} \frac{f(y) - f(x_k)}{\|y - x_k\|_2} \cdot \frac{y - x_k}{\|y - x_k\|_2} \quad (5.10)$$

加入 \mathcal{S}_k 能够改善算法的表现. 该方向是诸 $y - x_k$ 所在单位方向的一个加权组合, 权重为目标函数在各方向上的平均下降速度. 该方向可视为 $\nabla f(x_k)$ 的一个近似.

综合以上, 我们取

$$\mathcal{S}_k = \text{span}\{\nabla Q_k(x_k), d_{k-1}, \bar{d}_k\}. \quad (5.11)$$

以上, 我们给出了子空间 \mathcal{S}_k 的定义方式, 从而得到了一种子空间策略. 把这一策略应用于 NEWUOA 算法, 我们得到一个子空间算法, 不妨称之为 SUBSPACE 算法. 下面, 我们通过数值实验来比较 SUBSPACE 算法与 NEWUOA 算法. 我们将看到, 上述子空间策略确实能够改善 NEWUOA 算法的表现.

§5.2.2 数值实验

本小节通过数值实验比较 SUBSPACE 算法与 NEWUOA 算法, 以证明我们提出的子空间策略是成功的.

我们采用第二章 §2.4 建立的评价体系来比较这两个算法. 我们的测试问题集是 §3.3.2 中使用的 50 个可变维数的无约束优化问题. 对于每一个测试问题, 我们对 20、40、60、80、100 等 5 个维数进行了求解. 每一次求解都按照 §2.4 做了 10 次随机置换. 我们统计了求解精度 (即条件 (2.5) 中的 τ) 为 10^{-i} ($i = 2, 4, 6, 8, 10$) 时的函数值计算次数, 也统计了迭代自然终止时的函数值计算次数 (条件 (2.7) 中取 $\varepsilon = 10^{-10}$).

NEWUOA 的代码中用户可选参数 NPT、RHOBEQ 和 RHOEND 的选取与 §3.3.2 相同, SUBSPACE 算法的代码除了在子空间上求解信赖域子问题外, 其余部分与 NEWUOA 完全相同.

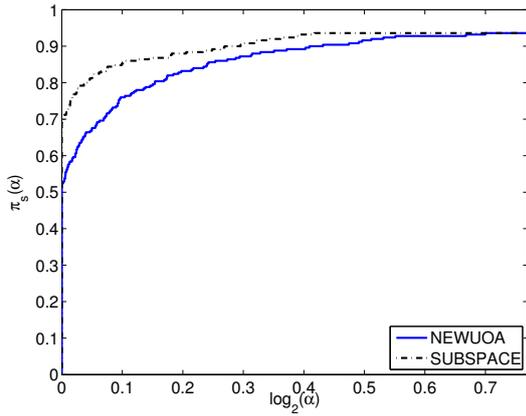
我们的实验环境与 §3.3.2 相同.

图 5.1 至图 5.6 给出了 NEWUOA 算法与 SUBSPACE 算法在满足不同精度要求以及迭代自然终止时的 Performance Profile、Data Profile、Sensitivity Profile 与 R-Sensitivity Profile. 图线越高说明算法表现越好. 四种 Profile 的定义请参考第二章.

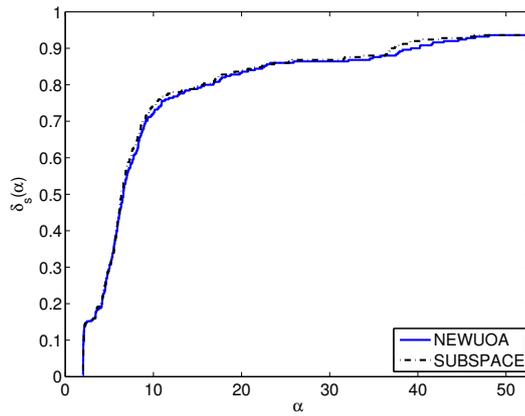
我们先来比较算法的函数值计算次数, 这体现在 Performance Profile 和 Data Profile 上. 可以看到, 两个算法的 Data Profile 区别不大; 但从 Performance Profile 来看, SUBSPACE 算法的表现要好于 NEWUOA 算法. 也就是说, 我们引入的子空间策略在平均上减少了函数值计算次数.

我们再来比较算法对计算机舍入误差的敏感性, 这体现在 Sensitivity Profile 和 R-Sensitivity Profile 上. 可以看到, 两个算法的在这方面相差不大. 求解精度为 10^{-2} 、 10^{-4} 和 10^{-8} 时, SUBSPACE 算法好于 NEWUOA 算法, 迭代自然终止时, NEWUOA 算法好于 SUBSPACE 算法, 但差距都不大.

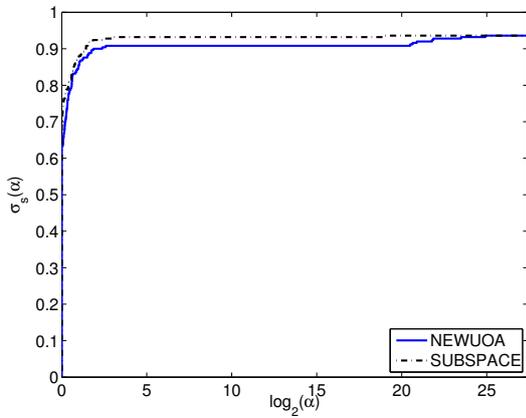
总体来说, SUBSPACE 算法的表现优于 NEWUOA 算法. 这说明我们引入的子空间策略的确提高了算法的性能. 我们要强调的是, 这种提高是不平凡的, 因为: 这两个算法的唯一区别是, NEWUOA 每步迭代都在 n 维 (n 可能高达 100) 全空间求解信赖域子问题, 而 SUBSPACE 是把信赖域子问题限制在一个低维 (至多 3 维) 子空间上求解, 除非有充分理由不这么做; 也就是说, NEWUOA 工作总是很“努力”, 而 SUBSPACE 则经常“偷懒”; 然而最后的结果却是 SUBSPACE 表现更好. 这种看似反常的现象不能不引起我们的注意, 它充分说明 SUBSPACE 算法“偷懒”的方式 (即子空间策略) 是成功的.



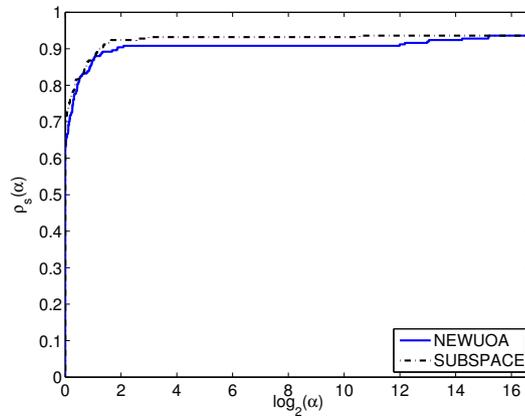
(a) Performance Profile



(b) Data Profile

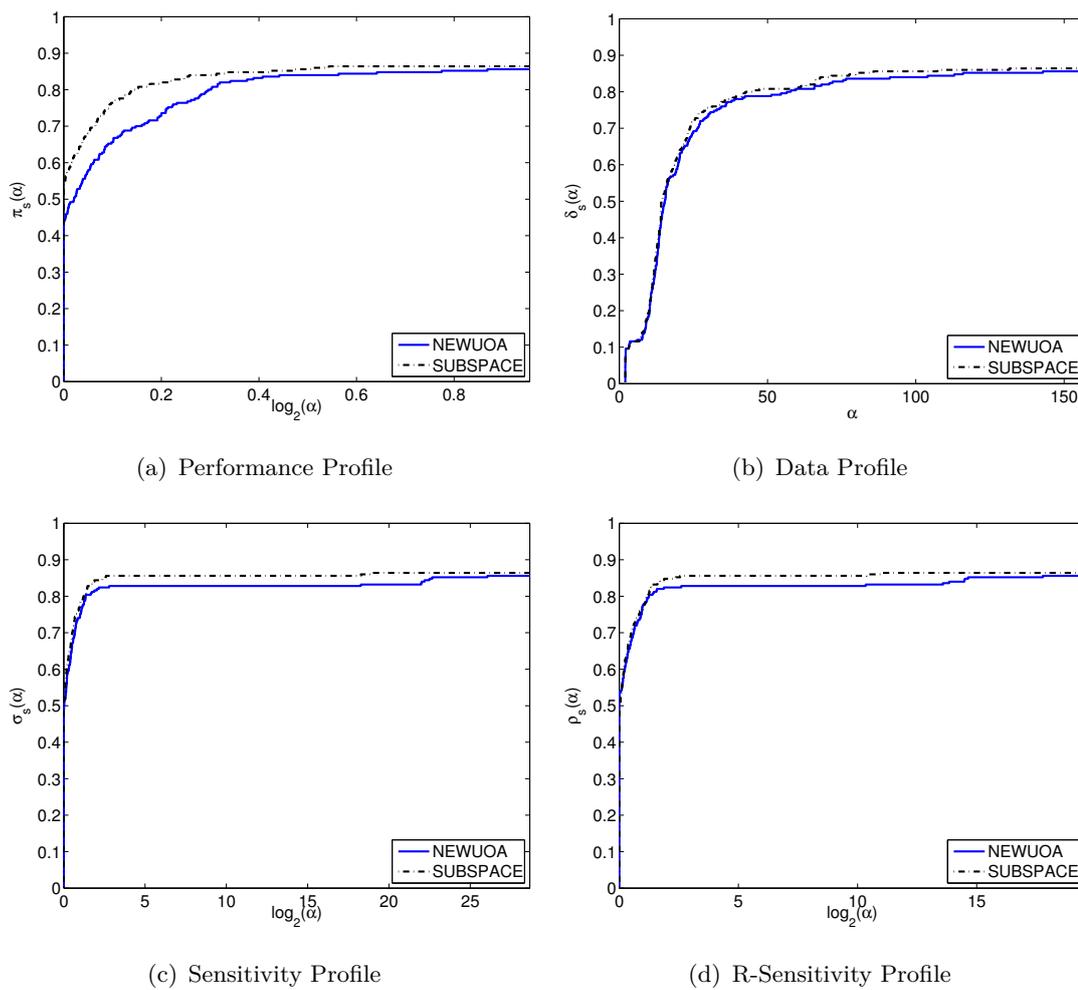


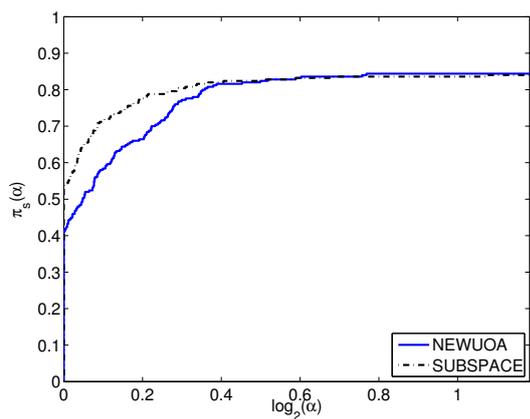
(c) Sensitivity Profile



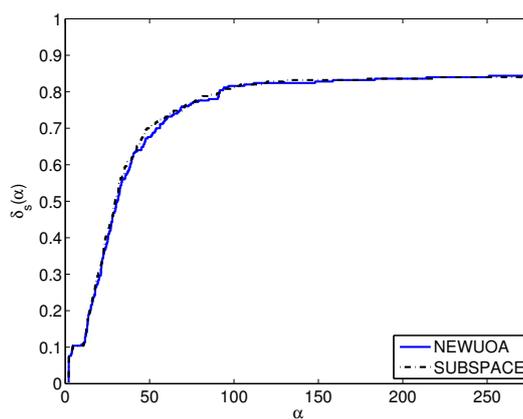
(d) R-Sensitivity Profile

图 5.1: NEWUOA 与 SUBSPACE 的数值表现 (精度 $\tau = 10^{-2}$)

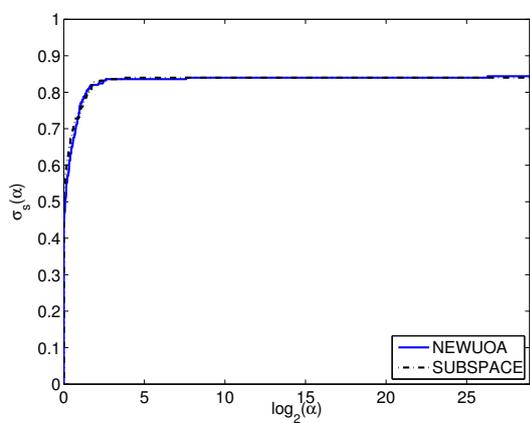
图 5.2: NEWUOA 与 SUBSPACE 的数值表现 (精度 $\tau = 10^{-4}$)



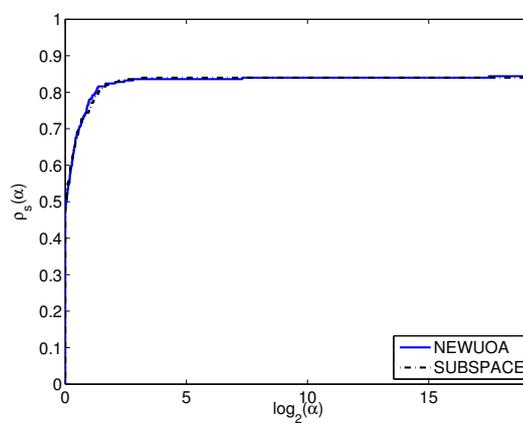
(a) Performance Profile



(b) Data Profile

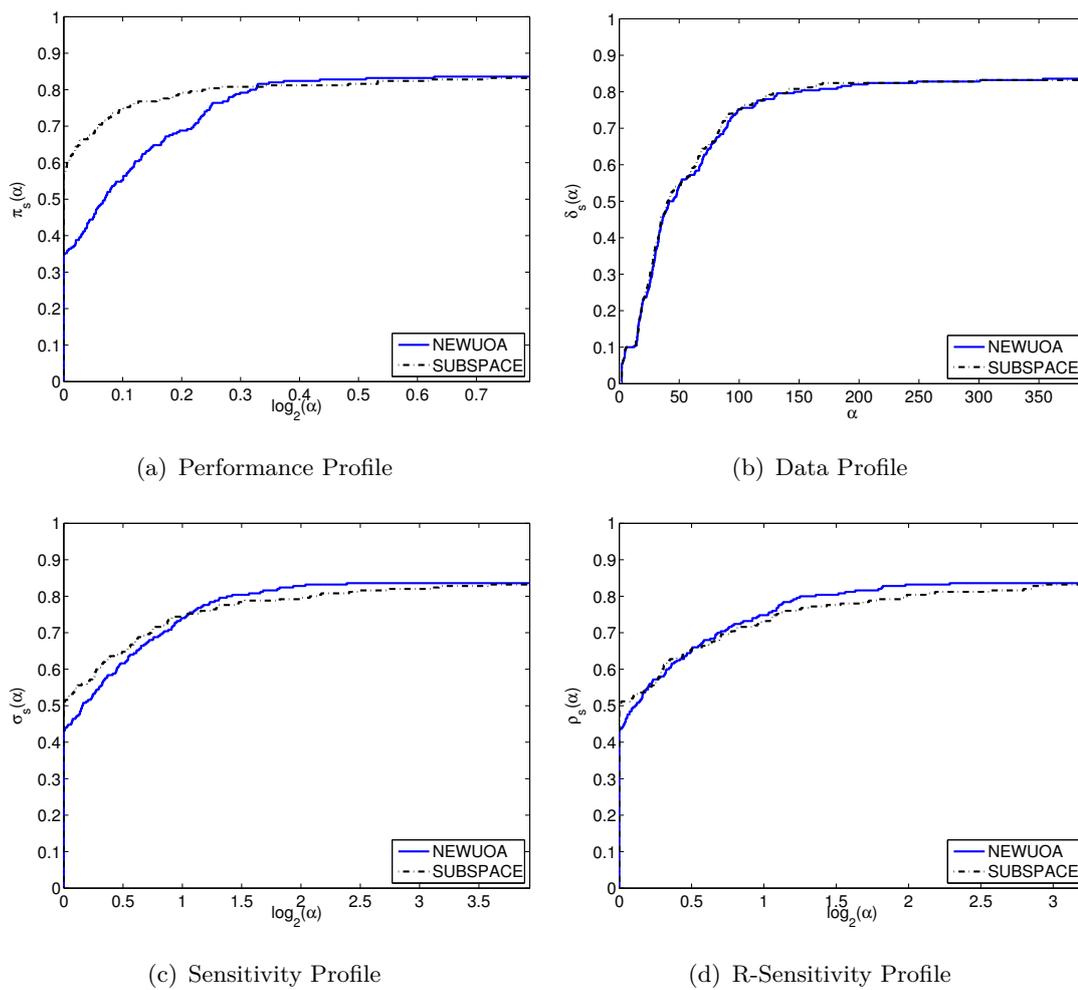


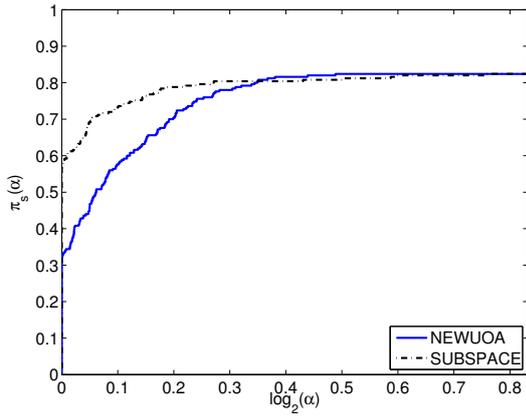
(c) Sensitivity Profile



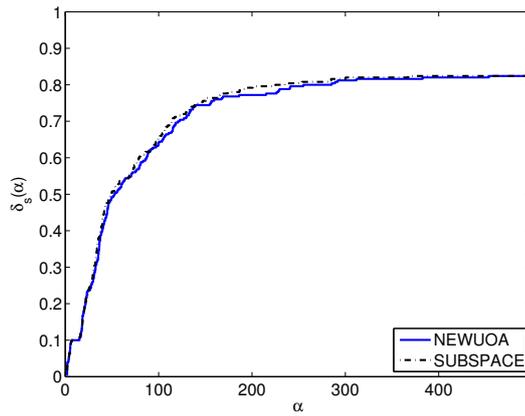
(d) R-Sensitivity Profile

图 5.3: NEWUOA 与 SUBSPACE 的数值表现 (精度 $\tau = 10^{-6}$)

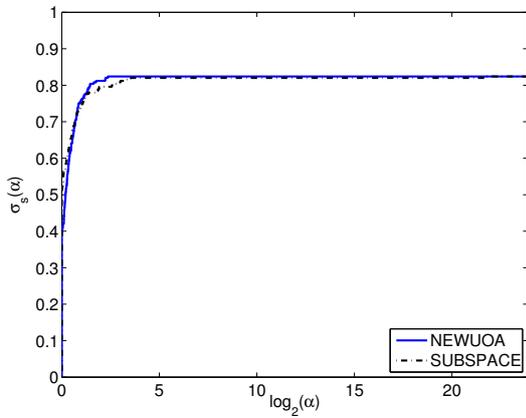
图 5.4: NEWUOA 与 SUBSPACE 的数值表现 (精度 $\tau = 10^{-8}$)



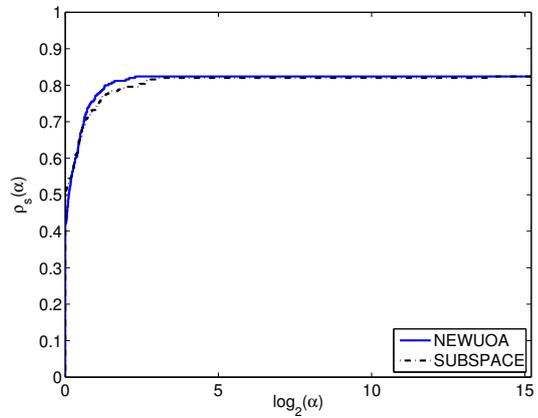
(a) Performance Profile



(b) Data Profile



(c) Sensitivity Profile



(d) R-Sensitivity Profile

图 5.5: NEWUOA 与 SUBSPACE 的数值表现 (精度 $\tau = 10^{-10}$)

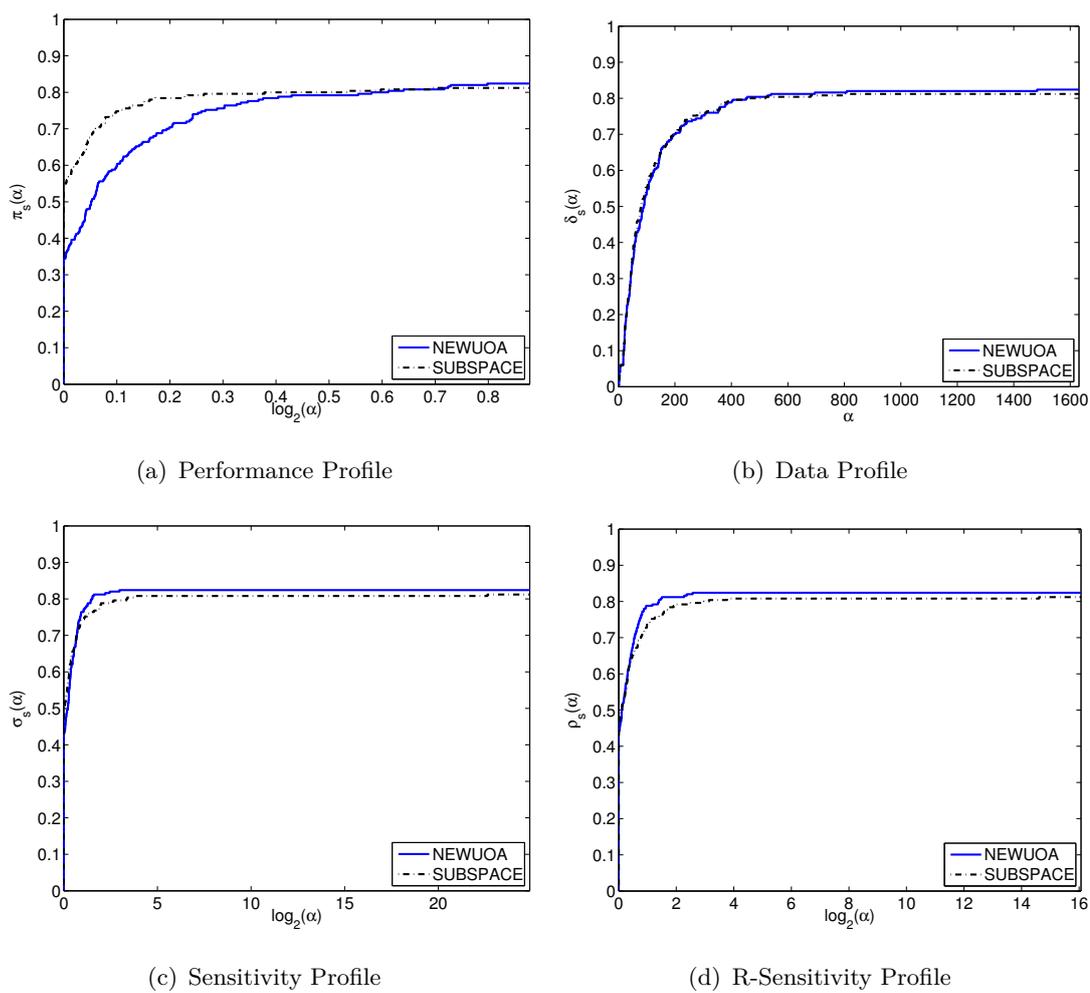


图 5.6: NEWUOA 与 SUBSPACE 的数值表现 (迭代自然终止)

上述实验中, 我们仅仅求解了规模不超过 100 维的问题. 图 5.7 展示了 NEWUOA 算法与 SUBSPACE 算法分别求解 200 维 ARWHEAD 问题与 CHROSEN 问题得到的函数值下降曲线. 可以清楚地看到, 我们的子空间策略加快了函数值的下降. 那么, 求解两个问题的过程中, 信赖域子问题在子空间和全空间求解的比例分别是多少呢? ARWHEAD 问题为 1078 : 2146, CHROSEN 问题为 2856 : 3058. 也就是说, NEWUOA 算法每一步都在 200 维的全空间求解信赖域子问题, 而 SUBSPACE 算法有三分之一以上的迭代把信赖域子问题限制在一个至多 3 维的子空间上求解, 结果却是后者表现更好. 这一不同寻常的现象清楚地表明, 我们的子空间策略是成功的.

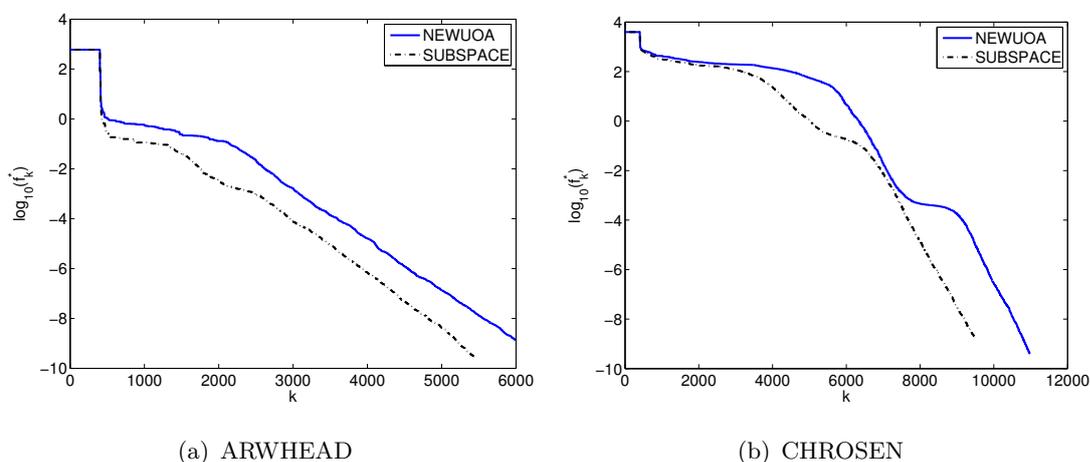


图 5.7: NEWUOA 与 SUBSPACE 求解 200 维 ARWHEAD 问题与 CHROSEN 问题的函数值下降曲线

§ 5.3 无导数子空间方法 II: NEWUOAs

本节介绍另一个无导数子空间算法 — NEWUOAs 算法. 这是本文最成功的算法. NEWUOAs 算法的基本想法十分朴素: 把困难问题分解为一系列容易的问题求解. 具体地说, NEWUOAs 算法把大规模无导数优化问题分解为一系列的小规模无导数优化问题求解. 实验证明, 这一策略是非常成功的.

本节的结构如下. 我们先在 § 5.3.1 讨论一个十分一般性的子空间算法框架. 然后, 在 § 5.3.2 中, 我们分析该框架的全局收敛性和局部收敛速度. § 5.3.3 至 § 5.3.6 将详细介绍基于这一框架的 NEWUOAs 算法. 在 § 5.3.7 中, 我们用数值实

验证明 NEWUOAs 算法是十分成功的, 它在大规模问题和坏初始点问题上的表现显著优于 NEWUOA 算法, 而且可已解决很多 NEWUOA 算法几乎不可求解的大规模问题.

为方便计, 本节使用 $\|\cdot\|$ 表示 \mathbb{R}^n 上的 2-范数和 $\mathbb{R}^{n \times n}$ 上的谱范数.

§5.3.1 一个子空间算法框架

本小节研究一个十分一般性的子空间算法框架. 借助已有的无导数优化算法, 我们可以从该框架导出一类无导数子空间算法. 这一算法框架如下.

算法 5.3. (一个子空间算法框架)

步 1. 取初始点 x_1 . $k := 1$.

步 2. 取 \mathbb{R}^n 的一个子空间 S_k .

步 3. 精确或近似求解子空间子问题:

$$\min_{d \in S_k} f(x_k + d), \quad (5.12)$$

得到 d_k .

步 4. 若 $f(x_k + d_k) < f(x_k)$, 则 $x_{k+1} := x_k + d_k$; 否则 $x_{k+1} := x_k$. $k := k + 1$.
转步 2.

算法 5.3 本质上与 Conn, Gould, 和 Toint [33] 提出的 ISM 算法框架相同. 该算法 5.3 十分具有一般性: 太一般以至于几乎没有给出任何细节. 然而, 在 §5.3.2 中我们将看到, 即使是对于这个十分一般的算法, 我们也可以做很多有意思且有用的理论分析.

算法 5.3 的一个特点是, 它没有提及任何导数信息. 这对我们是至关重要的. 这一特点使得我们在实现该算法时避免使用导数. 该算法的核心在步 2 和步 3. 只要在这两步不使用导数, 我们就得到了一个无导数算法. 例如, 经典的坐标轮换法就是算法 5.3 一个无导数实现. Wu 等 [177] 的算法也可以纳入到该框架下. 在 §5.3.3 中, 我们将给出算法 5.3 的一个基于 NEWUOA 算法的实现, 我们称之为 NEWUOAs 算法 (A NEW Unconstrained Optimization Algorithm with subspace technique based on NEWUOA).

需要注意的是, 算法 5.3 没有提及终止准则. 我们将在 §5.3.3 给出实用的终止准则.

§5.3.2 收敛性分析

在讨论算法 5.3 的具体实现之前, 本小节先建立其的全局收敛性和 R-线性收敛速度. 这些理论结果将指导我们对算法的实现.

在以下分析中, 我们用 P_k 表示 \mathbb{R}^n 到 \mathcal{S}_k 的正交投影, 记号 $\lambda_1(\cdot)$ 与 $\lambda_n(\cdot)$ 分别表示一个 n 阶对称矩阵的最小和最大特征值.

§5.3.2.1 全局收敛性

下面我们建立算法 5.3 的全局收敛性. 我们提出一个能够保证其全局收敛性的条件 (条件 5.6). 对于强凸 [11] 的目标函数, 我们还证明这一条件对保证算法 5.3 的全局收敛性是充分必要的.

在整个 §5.3.2.1 中, 我们对目标函数做出以下假设.

假设 5.4. 目标函数 f 有下界, 二次连续可微, 且二阶导数有界.

我们令 M 表示 $\|\nabla^2 f\|$ 的一个上界. 在假设 5.4 下, 我们有以下引理.

引理 5.5. 设 $x \in \mathbb{R}^n$, \mathcal{S} 为 \mathbb{R}^n 的一个子空间, 则

$$f(x) - \inf_{d \in \mathcal{S}} f(x+d) \geq \frac{1}{2M} \|P\nabla f(x)\|^2, \quad (5.13)$$

其中 P 为 \mathbb{R}^n 到 \mathcal{S} 的正交投影.

证明. 令 $d_0 = P\nabla f(x)$. 定义

$$q(t) = f(x) + td_0^T \nabla f(x) + \frac{1}{2} M \|d_0\|^2 t^2. \quad (5.14)$$

则对任意 t , 有

$$f(x + td_0) \leq q(t). \quad (5.15)$$

故

$$f(x) - \inf_{t \geq 0} f(x + td_0) \geq f(x) - \inf_{t \geq 0} q(t) = \frac{[d_0^T \nabla f(x)]^2}{2M \|d_0\|^2} = \frac{1}{2M} \|P\nabla f(x)\|^2. \quad (5.16)$$

因 $d_0 \in \mathcal{S}$, 故 (5.16) 蕴含 (5.13). ■

要保证算法 5.3 的全局收敛性, 很自然的需要子空间 \mathcal{S}_k 能反映目标函数 f 的足够多的信息, 且 x_{k+1} 足够接近 f 在 $x_k + \mathcal{S}_k$ 上的极小点. 因此, 我们提出以下条件:

条件 5.6. 存在一个无穷整数集合 \mathcal{K} 满足

$$\text{a.) } (I - P_k)\nabla f(x_k) \rightarrow 0, \text{ 当 } k \in \mathcal{K} \text{ 且 } k \rightarrow \infty;$$

$$\text{b.) } f(x_{k+1}) - \inf_{d \in \mathcal{S}_k} f(x_k + d) \rightarrow 0, \text{ 当 } k \in \mathcal{K} \text{ 且 } k \rightarrow \infty.$$

条件 5.6 的 a.) 等价于: 存在序列 $\{\tilde{g}_k\}$ 使得 $\tilde{g}_k \in \mathcal{S}_k$ 且

$$\tilde{g}_k - \nabla f(x_k) \rightarrow 0, \text{ 当 } k \in \mathcal{K} \text{ 且 } k \rightarrow \infty. \quad (5.17)$$

粗略地讲, 这就是要求当 $k \in \mathcal{K}$ 时, \mathcal{S}_k 中存在向量 \tilde{g}_k 能良好地逼近 $\nabla f(x_k)$.

现在我们证明条件 5.6 能保证算法 5.3 的全局收敛性.

定理 5.7. 若条件 5.6 对算法 5.3 成立, 则

$$\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0. \quad (5.18)$$

证明. 我们证明

$$\|\nabla f(x_k)\| \rightarrow 0, \text{ 当 } k \in \mathcal{K} \text{ 且 } k \rightarrow \infty. \quad (5.19)$$

假若不然, 存在 $\varepsilon_0 > 0$ 以及 \mathcal{K} 的无穷子集 $\bar{\mathcal{K}}$ 使得

$$\|\nabla f(x_k)\| \geq \varepsilon_0, \quad k \in \bar{\mathcal{K}}. \quad (5.20)$$

根据条件 5.6, 不妨假设对 $k \in \bar{\mathcal{K}}$ 有

$$\|(I - P_k)\nabla f(x_k)\| \leq \frac{\varepsilon_0}{2}, \quad (5.21)$$

$$f(x_{k+1}) - \inf_{x \in \mathcal{S}_k} f(x_k + d) \leq \frac{\varepsilon_0^2}{16M}. \quad (5.22)$$

对任意 $k \in \bar{\mathcal{K}}$, 由引理 5.5 与 (5.20)、(5.21) 以及 (5.22) 知

$$\begin{aligned}
 & f(x_k) - f(x_{k+1}) \\
 &= \left[f(x_k) - \inf_{d \in \mathcal{S}_k} f(x_k + d) \right] - \left[f(x_{k+1}) - \inf_{d \in \mathcal{S}_k} f(x_k + d) \right] \\
 &\geq \frac{1}{2M} \|P_k \nabla f(x_k)\|^2 - \frac{\varepsilon_0^2}{16M} \\
 &\geq \frac{\varepsilon_0^2}{8M} - \frac{\varepsilon_0^2}{16M} \\
 &= \frac{\varepsilon_0^2}{16M}.
 \end{aligned} \tag{5.23}$$

这与 $\bar{\mathcal{K}}$ 为无穷集以及 f 有下界矛盾. ■

定理 5.7 表明, 条件 5.6 对保证算法 5.3 的全局收敛性是充分的. 容易看到, 它在下面的意义下也是必要的.

命题 5.8. 若算法 5.3 产生的点列 $\{x_k\}$ 满足

$$\lim_{k \rightarrow \infty} f(x_k) = \inf_{x \in \mathbb{R}^n} f(x), \tag{5.24}$$

则条件 5.6 成立.

证明. 我们将证明条件 5.6 对 $\mathcal{K} = \{k : k \geq 1\}$ 成立. 据引理 5.5,

$$f(x_k) - \inf_{x \in \mathbb{R}^n} f(x) \geq \frac{1}{2M} \|\nabla f(x_k)\|^2. \tag{5.25}$$

故 $\nabla f(x_k) \rightarrow 0$, 因此条件 5.6 的 a.) 成立. 条件 5.6 的 b.) 可由

$$f(x_{k+1}) - \inf_{x \in \mathbb{R}^n} f(x) \geq f(x_{k+1}) - \inf_{d \in \mathcal{S}_k} f(x_k + d) \geq 0 \tag{5.26}$$

得到. ■

精确地讲, 当目标函数强凸时, 条件 5.6 对保证算法 5.3 的全局收敛是充分必要的.

定理 5.9. 假设 f 强凸, 算法 5.3 产生点列 $\{x_k\}$. 则 $\{x_k\}$ 收敛于 f 的唯一极小点当且仅当条件 5.6 成立.

根据定理 5.7 与命题 5.8, 定理 5.9 是显然的.

粗略地讲, 我们的分析表明, 为了保证算法 5.3 全局收敛, 必须且只需要求 \mathcal{S}_k 包含 $\nabla f(x_k)$ 的一个足够好的近似, 且 d_k 为子问题 (5.12) 的一个足够好的解. 这些观察将引导我们对算法 5.3 的实现 (详情见 §5.3.3).

§5.3.2.2 R-线性收敛速度

下面, 我们建立算法 5.3 的 R-线性收敛速度. 我们提出一个能够保证算法 R-线性收敛速度的条件 (条件 5.13). 对于凸目标函数, 我们还证明该条件对算法的 R-线性收敛速度是充分必要的.

在整个 §5.3.2.2 中, 我们做出以下假设. 该假设在收敛速度的分析中是常见的.

假设 5.10. 算法 5.3 产生的点列 $\{x_k\}$ 满足

a.) $x_k \rightarrow x^*$;

b.) f 在 x^* 的一个邻域内二次连续可微, $\nabla f(x^*) = 0$, 且 $\nabla^2 f(x^*)$ 正定.

根据假设 5.10 的 b.), 存在正数 λ 、 μ 和 δ 使得当 $\|x - x^*\| \leq \delta$ 时

$$\lambda \leq \lambda_1(\nabla^2 f(x)) \leq \lambda_n(\nabla^2 f(x)) \leq \mu. \quad (5.27)$$

容易看到, 当 $\|x - x^*\| \leq \delta$ 时,

$$\frac{\lambda}{2} \|x - x^*\|^2 \leq f(x) - f(x^*) \leq \frac{\mu}{2} \|x - x^*\|^2, \quad (5.28)$$

以及

$$\lambda \|x - x^*\| \leq \|\nabla f(x)\| \leq \mu \|x - x^*\|. \quad (5.29)$$

另外, 可以证明, 存在正数 δ' 使得当 $\|x - x^*\| \leq \delta'$ 时

$$f(x) - f(x^*) \leq \frac{1}{\lambda} \|\nabla f(x)\|^2 \quad (5.30)$$

以上结论的证明请参考袁亚湘 [190] (46 页).

以下引理本质上与引理 5.5 是相同的, 不过这里我们需要更细致的分析.

引理 5.11. 若 x 满足 $\|x - x^*\| \leq \delta/2$, \mathcal{S} 为 \mathbb{R}^n 的子空间, 则

$$f(x) - \inf_{d \in \mathcal{S}} f(x+d) \geq \frac{1}{2\mu} \|P\nabla f(x)\|^2, \quad (5.31)$$

其中 P 为 \mathbb{R}^n 到 \mathcal{S} 的正交投影.

证明. 根据 (5.27), 当 $\|tP\nabla f(x)\| \leq \delta/2$ 时,

$$f(x - tP\nabla f(x)) \leq f(x) - \|P\nabla f(x)\|^2 t + \frac{\mu}{2} \|P\nabla f(x)\|^2 t^2, \quad (5.32)$$

考虑 $t^* = \mu^{-1}$. 因为

$$\|t^*P\nabla f(x)\| \leq \mu^{-1} \|\nabla f(x)\| \leq \|x - x^*\| \leq \frac{\delta}{2}, \quad (5.33)$$

不等式 (5.32) 对 t^* 成立. 故

$$\inf_{d \in \mathcal{S}} f(x+d) \leq f(x - t^*P\nabla f(x)) \leq f(x) - \frac{1}{2\mu} \|P\nabla f(x)\|^2. \quad (5.34)$$

因而 (5.31) 为真. ■

下面的引理很简单, 但对我们的分析很关键.

引理 5.12. 设 $\{\alpha_k\}$ 为一个单调递减的非负数列, 且存在常数 $\alpha > 0$ 与 $\rho_1, \rho_2 \in (0, 1)$ 使得对任意 $k \geq 1$,

$$\alpha_k \leq \alpha\rho_1^k, \text{ 或 } \alpha_{k+1} \leq \rho_2\alpha_k. \quad (5.35)$$

那么

$$\alpha_k \leq \bar{\alpha}\rho^k, \quad (5.36)$$

其中,

$$\rho = \max\{\rho_1, \rho_2\}, \quad \bar{\alpha} = \rho^{-1} \max\{\alpha_1, \alpha\}. \quad (5.37)$$

证明. 我们用归纳法证明, 当 $k=1$ 时, (5.36) 成立. 假设 (5.36) 对 k 成立. 若 $\alpha_k \leq \alpha\rho_1^k$, 则 $\alpha_{k+1} \leq \alpha_k \leq \alpha\rho_1^k \leq \bar{\alpha}\rho^{k+1}$. 否则, $\alpha_{k+1} \leq \rho_2\alpha_k \leq \rho\alpha_k \leq \bar{\alpha}\rho^{k+1}$. ■

根据我们在 §5.3.2.1 的经验, 我们提出以下条件.

条件 5.13.

a.) $(I - P_k)\nabla f(x_k)$ R -线性收敛于零;

b.) $f(x_{k+1}) - \inf_{d \in \mathcal{S}_k} f(x_k + d)$ R -线性收敛于零.

现在我们证明条件 5.13 能保证算法 5.3 的 R -线性收敛性.

定理 5.14. 若条件 5.13 对算法 5.3 成立, 则 x_k R -线性收敛于 x^* .

证明. 记

$$\bar{f}_k = \inf_{d \in \mathcal{S}_k} f(x_k + d), \quad (5.38)$$

以及

$$\alpha_k = \frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)}. \quad (5.39)$$

根据条件 5.13, 存在常数 $\alpha > 0$ 与 $\rho \in (0, 1)$ 使得

$$\|(I - P_k)\nabla f(x_k)\| \leq \mu\alpha\rho^k, \quad (5.40)$$

$$f(x_{k+1}) - \bar{f}_k \leq \frac{1}{2}\alpha\rho^k. \quad (5.41)$$

由 (5.30) 与 (5.31) 知, 当 k 足够大时

$$f(x_k) - f(x^*) \leq \frac{1}{\lambda}\|\nabla f(x_k)\|^2, \quad (5.42)$$

$$f(x_k) - \bar{f}_k \geq \frac{1}{2\mu}\|P_k\nabla f(x_k)\|^2. \quad (5.43)$$

因此, 对足够大的 k ,

$$\begin{aligned} \alpha_k &= 1 - \frac{f(x_k) - f(x_{k+1})}{f(x_k) - f(x^*)} \\ &= 1 - \frac{f(x_k) - \bar{f}_k}{f(x_k) - f(x^*)} + \frac{f(x_{k+1}) - \bar{f}_k}{f(x_k) - f(x^*)} \\ &\leq 1 - \frac{\|P_k\nabla f(x_k)\|^2}{2\mu[f(x_k) - f(x^*)]} + \frac{\alpha\rho^k}{2[f(x_k) - f(x^*)]} \\ &\leq 1 - \frac{\|\nabla f(x_k)\|^2 - \mu\alpha\rho^k}{2\mu[f(x_k) - f(x^*)]} + \frac{\alpha\rho^k}{2[f(x_k) - f(x^*)]} \\ &\leq 1 - \frac{\lambda}{2\mu} + \frac{\alpha\rho^k}{2[f(x_k) - f(x^*)]} + \frac{\alpha\rho^k}{2[f(x_k) - f(x^*)]} \\ &= 1 - \frac{\lambda}{2\mu} + \frac{\alpha\rho^k}{f(x_k) - f(x^*)}, \end{aligned} \quad (5.44)$$

从而

$$f(x_k) - f(x^*) \leq \frac{4\mu\alpha}{\lambda} \rho^k, \text{ 或 } \alpha_k \leq 1 - \frac{\lambda}{4\mu}. \quad (5.45)$$

根据引理 5.12, (5.45) 蕴含 $\{f(x_k) - f(x^*)\}$ 的 R-线性收敛速度. 同时, 不等式 (5.28) 说明 k 足够大时 $\|x_k - x^*\|$ 被 $2[f(x_k) - f(x^*)]/\lambda$ 控制. 因此 x_k R-线性收敛于 x^* . ■

定理 5.14 说明条件 5.13 对保证算法 5.3 的 R-线性收敛速度是充分的. 容易看到, 该条件在下述意义下也是必要的.

命题 5.15. 若 x^* 为 f 的全局极小点, 且 x_k R-线性收敛于 x^* , 则条件 5.13 对算法 5.3 成立.

证明. 当 k 足够大时,

$$\|\nabla f(x_k)\| \leq \mu \|x_k - x^*\|. \quad (5.46)$$

因此 $\nabla f(x_k)$ R-线性收敛于零, 这蕴含条件 5.13 的 a). 条件 5.13 的 b.) 可由

$$0 \leq f(x_{k+1}) - \inf_{d \in \mathcal{S}_k} f(x_k + d) \leq f(x_{k+1}) - f(x^*) \leq \frac{\mu}{2} \|x_{k+1} - x^*\|^2 \quad (5.47)$$

得到. ■

精确地讲, 对于凸目标函数, 条件 5.13 对算法 5.3 的 R-线性收敛速度是充分必要的,

定理 5.16. 若 f 是凸的, 则 x_k R-线性收敛于 x^* 当且仅当条件 5.13 对算法 5.3 成立.

根据定理 5.14 与命题 5.15, 定理 5.16 是显然的.

粗略地讲, 我们的分析表明, 为了使算法 5.3 R-线性收敛, 必须且只需 \mathcal{S}_k 包含一个向量 R-线性地近似 $\nabla f(x_k)$, 且 d_k R-线性地近似子问题 (5.12) 的解. (5.45) 提示我们, 算法 5.3 的收敛速度受 $\nabla^2 f$ 的条件数的影响. 这些观察将引导我们对算法 5.3 的实现 (详情见 §5.3.3 与 §5.3.6).

§5.3.3 NEWUOAs: 一个基于 NEWUOA 的无导数子空间算法

使用 NEWUOA 算法作为子问题 (5.12) 的求解器, 我们能够得到算法 5.3 的一个无导数实现, 即 NEWUOAs 算法. 本小节介绍 NEWUOAs 算法的实现过程.

为了实现算法 5.3, 我们只需指明步 2 中如何定义子空间 \mathcal{S}_k , 以及步 3 中如何求解子问题 (5.12). 为了使我们的算法不依赖导数, 我们必须不依赖导数地完成这些工作. 幸运的是, 这是可能的. §5.3.2 中的理论分析已经为我们做好了充分的铺垫.

根据 §5.3.2 中的理论, 要使算法 5.3 全局收敛, \mathcal{S}_k 不需要包含 $\nabla f(x_k)$, 而只要包含 $\nabla f(x_k)$ 的一个足够好的近似. 假定我们用某种方式, 比如有限差分、单纯形梯度 [16, 48, 47, 45] 或更一般的插值得到了一个近似梯度 \tilde{g}_k , 则只需令 \mathcal{S}_k 包含 \tilde{g}_k 即可. \mathcal{S}_k 的具体定义可以借鉴基于导数的子空间方法 [183, 33, 112, 111, 169, 108, 170, 180, 181, 182, 81]. §5.3.2 中的理论还告诉我们, 子问题 (5.12) 也不需要精确求解. 我们只要调用已有的无导数优化算法求解子问题 (5.12) 到一定精度即可. 由于这是一个低维问题, 现有算法可以很高效地处理.

我们用插值的方法构造近似梯度, 用 NEWUOA 算法求解子问题 (5.12), 得到了算法 5.3 的一个无导数实现. 我们称该算法为 NEWUOAs (A NEW Unconstrained Optimization Algorithm with subspace technique based on NEWUOA).

具体地讲, 在第 k 步迭代, 我们在 x_k 周围取插值点, 通过最小 Frobenius 范数插值得到 f 的一个模型 Q_k , 然后取 $\tilde{g}_k = \nabla Q_k(x_k)$ 作为 $\nabla f(x_k)$ 的一个近似. 我们把 NEWUOA 初始化时建立模型的过程分离出来, 作为一个子程序用于实现上述插值. 我们把该子程序命名为 MODEL. 它含有三个参量: 向量 $x_0 \in \mathbb{R}^n$ 、正数 h 和正整数 m . 在我们这里 $x_0 = x_k$. 计算中, MODEL 按一定规则选取 m 个形如

$$y = x_0 + \alpha h e_i + \beta h e_j \quad (5.48)$$

的点对 f 进行插值, 其中 e_i 与 e_j 为两个坐标向量, α 与 β 的可能取值为 0、1 和 -1 . 为了使插值问题有解, MODEL 要求 $m \leq (n+1)(n+2)/2$. 另外, NEWUOA 中还要求 $m \geq n+2$, 我们放宽为 $m \geq n+1$, 即我们允许线性模型. 实

际计算中, 我们建议取 $m = 2n + 1$. 此时插值点的取法为

$$y_1 = x_0, \quad (5.49)$$

$$y_{i+1} = x_0 + he_i, \quad (5.50)$$

$$y_{i+n+1} = x_0 - he_i, \quad (5.51)$$

其中 $i = 1, \dots, n$. 可以看出, 该插值点集确定的最小 Frobenius 范数插值模型 Q 的系数为

$$[\nabla Q(x_0)]_i = \frac{1}{2h} [f(x_0 + he_i) - f(x_0 - he_i)], \quad (5.52)$$

$$[\nabla^2 Q(x_0)]_{ii} = \frac{1}{h^2} [f(x_0 + he_i) + f(x_0 - he_i) - 2f(x_0)], \quad (5.53)$$

$$[\nabla^2 Q(x_0)]_{ij} = 0, \quad i \neq j, \quad (5.54)$$

其中 $i, j = 1, \dots, n$. 可见, 此时的最小 Frobenius 范数插值等价于中心差分. 关于 m 取其他值时插值点集的具体定义请参考 Powell [138].

调用 $\text{MODEL}(x_k, h, m)$, 我们得到 x_k 处的最小 Frobenius 范数模型 Q_k ; 令 $\tilde{g}_k = \nabla Q_k(x_k)$, 就得到近似梯度 \tilde{g}_k . \tilde{g}_k 的近似精度由 h 决定, 为引用方便, 我们将 $m = 2n + 1$ 时的精度详述如下.

命题 5.17. 设 Q_k 由 $\text{MODEL}(x_k, h, 2n + 1)$ 得到.

a.) 若 f 在 \mathbb{R}^n 上二次连续可微, 且二阶导数有界, 则

$$\|\nabla f(x_k) - \nabla Q_k(x_k)\| \leq \frac{1}{2} \sup_{x \in \mathbb{R}^n} \left\{ \sum_{i=1}^n \left[\frac{\partial^2 f}{\partial x_i^2}(x) \right]^2 \right\}^{\frac{1}{2}} h; \quad (5.55)$$

b.) 若 f 在 \mathbb{R}^n 上三次连续可微, 且三阶导数有界, 则

$$\|\nabla f(x_k) - \nabla Q_k(x_k)\| \leq \frac{1}{6} \sup_{x \in \mathbb{R}^n} \left\{ \sum_{i=1}^n \left[\frac{\partial^3 f}{\partial x_i^3}(x) \right]^2 \right\}^{\frac{1}{2}} h^2. \quad (5.56)$$

证明. 这实际上就是中心差分的误差估计, 可以利用 Taylor 展开很容易地得到. 以 (5.55) 为例: 对任何 $i \in \{1, 2, \dots, n\}$, 由 (5.52) 知

$$|[\nabla f(x_0)]_i - [\nabla Q(x_0)]_i| \leq \frac{1}{2} \sup_{x \in \mathbb{R}^n} \left| \frac{\partial^2 f}{\partial x_i^2}(x) \right| h. \quad (5.57)$$

故 (5.55) 成立. ■

得到近似梯度之后, 我们来定义子空间 \mathcal{S}_k . 我们提出两种子空间的定义方式, 这里先介绍第一种, 另外一种在 §5.3.6 中介绍. 第一种子空间是简单和直接的: 参照共轭梯度步所在的子空间 [183], 我们取

$$\mathcal{S}_k = \text{span}\{\tilde{g}_k, s_{k-1}\}, \quad (5.58)$$

其中 $s_{k-1} = x_k - \bar{x}_{k-1}$, 且 \bar{x}_{k-1} 为最近一个与 x_k 不同的迭代点. 这样, 算法 5.3 的步 2 就实现了, 并且这一过程没有用到导数信息.

步 3 的实现更加简单. 我们只要调用 NEWUOA 算法求解子问题 (5.12) 到合适的精度即可. 那么, 如何控制 NEWUOA 算法的求解精度? 在 §3.3.2 中我们提到, NEWUOA 代码中含有一个用户可选参数 RHOEND, 该参数就反映了算法的求解精度². 所以, 我们可以在调用 NEWUOA 时通过设置 RHOEND 的值来控制精度.

用上述方法, 我们不依赖导数地实现了算法 5.3 的步 2 与步 3, 从而得到了一个无导数优化算法, 即 NEWUOAs 算法. 我们将 NEWUOAs 算法的计算过程总结如下.

算法 5.18. (NEWUOAs)

步 1. 取正数序列 $\{h_k\}$ 、 $\{p_k\}$ 和常数 $\varepsilon \geq 0$. 确定初始点 x_1 ; $s_0 := 0$; $k := 1$.

步 2. 选取整数 $m_k \in [n+1, (n+1)(n+2)/2]$, 调用 $\text{MODEL}(x_k, h_k, m_k)$, 获得 x_k 处的近似梯度 \tilde{g}_k . 若 $h_k < \varepsilon$ 且 $\|\tilde{g}_k\| < \varepsilon$, 终止. 令

$$\mathcal{S}_k = \text{span}\{\tilde{g}_k, s_{k-1}\}. \quad (5.59)$$

步 3. 设置 $\text{RHOEND} = p_k$, 调用 NEWUOA 求解子问题

$$\min_{d \in \mathcal{S}_k} f(x_k + d) \quad (5.60)$$

得到 d_k .

² “The parameter ρ_{end} , which has to satisfy $\rho_{\text{end}} \leq \rho_{\text{beg}}$, should have the magnitude of the required accuracy in the final values of the variables” [138]. 参数 ρ_{end} 在 NEWUOA 源代码中的变量名即为 RHOEND. 实际计算中, 通常可以观察到 $\|x_{\text{fin}} - x^*\|_\infty$ 为 ρ_{end} 量级的, 其中 x_{fin} 为 NEWUOA 算法求得的解, x^* 为最优解. 请参考 Powell [138] 的数值结果.

步 4. 若 $f(x_k + d_k) < f(x_k)$, 则 $x_{k+1} := x_k + d_k$, $s_k := d_k$; 否则 $x_{k+1} := x_k$, $s_k := s_{k-1}$. $k := k + 1$. 转步 2.

事实上, NEWUOAs 算法的迭代分内外两层. 上面主要描述了外层迭代过程. 内层迭代是步 3 中求解子问题 (5.60) 的过程; 这部分迭代由 NEWUOA 算法完成, 不需要我们管理. 因此我们所有的讨论总是把注意力集中在外层迭代上.

为了避免一开始就陷于细节, 我们把 NEWUOAs 算法的一些具体实现 (比如 $\{h_k\}$ 与 $\{p_k\}$ 的选取) 放在 §5.3.5 中介绍. 在此之前, §5.3.4 将证明 NEWUOAs 算法在理论上的收敛性和计算上的有限终止性³.

§5.3.4 NEWUOAs 算法的收敛性

本小节的讨论以 $m_k \equiv 2n + 1$ 为例. 对于 m_k 取 $[n + 1, (n + 1)(n + 2)/2]$ 中其他整数的情况, 讨论是类似的. 另外, 本小节的分析不要求子空间 \mathcal{S}_k 严格按照 (5.59) 定义; 只要 $\tilde{g}_k \in \mathcal{S}_k$, 这里的分析就成立.

NEWUOAs 算法的表现依赖于 NEWUOA 求解子问题 (5.60) 的效果. 为了保证 NEWUOAs 算法的收敛性, 我们需要以下假设.

假设 5.19. 存在常数 c 与无穷正整数集合 \mathcal{K} 使得对任何 $k \in \mathcal{K}$, 步 3 得到的 d_k 满足

$$\text{dist}(d_k, \arg \min_{d \in \mathcal{S}_k} f(x_k + d)) \leq cp_k. \quad (5.61)$$

事实上, 以上假设正符合 NEWUOA 算法中参数 RHOEND 的含义.

我们先在 $\varepsilon = 0$ 的理想情况下讨论 NEWUOAs 的全局收敛性和局部收敛速度. 与 §5.3.2 一样, 我们用 P_k 表示 \mathbb{R}^n 到 \mathcal{S}_k 的正交投影.

下面的定理建立了 NEWUOAs 算法的全局收敛性.

定理 5.20. 设 $\varepsilon = 0$, 目标函数 f 满足假设 5.4, 且 NEWUOA 算法满足假设 5.19. 若

$$\liminf_{k \in \mathcal{K}, k \rightarrow \infty} h_k = 0 = \liminf_{k \in \mathcal{K}, k \rightarrow \infty} p_k, \quad (5.62)$$

则

$$\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0. \quad (5.63)$$

³这里的“有限终止”并非是说算法在有限步之内能求到精确解, 而是说有限步之内会满足终止准则正常退出.

证明. 不妨假设 $\{h_k\}$ 与 $\{p_k\}$ 皆在 \mathcal{K} 上趋于零 (否则, 取 \mathcal{K} 的一个满足条件的无穷子集代替 \mathcal{K} 即可).

由假设 5.4 及 (5.55), $\|\nabla f(x_k) - \tilde{g}_k\|$ 被 h_k 控制, 故 $\nabla f(x_k) - \tilde{g}_k$ 在 \mathcal{K} 上收敛于零. 从而由

$$\|(I - P_k)\nabla f(x_k)\| = \|\nabla f(x_k) - P_k\nabla f(x_k)\| \leq \|\nabla f(x_k) - \tilde{g}_k\| \quad (5.64)$$

知 $(I - P_k)\nabla f(x_k)$ 在 \mathcal{K} 上收敛于零.

另一方面, 由假设 5.19 知

$$f(x_k + d_k) - \inf_{d \in \mathcal{S}_k} f(x_k + d) \rightarrow 0, \text{ 当 } k \in \mathcal{K} \text{ 且 } k \rightarrow \infty. \quad (5.65)$$

故由

$$f(x_{k+1}) \leq f(x_k + d_k) \quad (5.66)$$

知

$$f(x_{k+1}) - \inf_{d \in \mathcal{S}_k} f(x_k + d) \rightarrow 0, \text{ 当 } k \in \mathcal{K} \text{ 且 } k \rightarrow \infty. \quad (5.67)$$

根据定理 5.7, 全局收敛性 (5.63) 成立. ■

用相同的方法, 我们可以由定理 5.14 导出 NEWUOAs 算法的 R-线性收敛速度.

定理 5.21. 设 $\varepsilon = 0$, 目标函数 f 与迭代点列 $\{x_k\}$ 满足假设 5.10, 且 (5.61) 对足够大的 k 都成立. 若序列 $\{h_k\}$ 与 $\{p_k\}$ R-线性收敛于零, 则 $\{x_k\}$ R-线性收敛于 x^* .

证明. 与定理 5.20 的证明类似, 我们可以看到 $\|(I - P_k)\nabla f(x_k)\|$ 以及 $f(x_{k+1}) - \inf_{d \in \mathcal{S}_k} f(x_k + d)$ 皆 R-线性收敛于零. 故由定理 5.14 知所论成立. ■

实际计算中, 我们取 $\varepsilon > 0$. 这时, 如果令 $\{h_k\}$ 与 $\{p_k\}$ 趋于零, 那么 NEWUOAs 将在有限步内终止. 但是, 计算中令 $\{h_k\}$ 与 $\{p_k\}$ 趋于零可能是不合理的. 过小的 h_k 会使得插值不稳定, 而过小的 p_k 会增加子问题 (5.60) 的求解代价. 因此, 我们建议在实际计算中给这两个序列规定合理的下界. 那么, 此时 NEWUOAs 算法是否仍有限终止? 下面的两个定理回答了这一问题.

定理 5.22. 设目标函数 f 满足假设 5.4, NEWUOA 算法满足假设 5.19. 若 $\varepsilon > 0$, 且

$$\liminf_{k \in \mathcal{K}, k \rightarrow \infty} M\sqrt{n}\varepsilon h_k + M^2 c^2 p_k^2 < \varepsilon^2, \quad (5.68)$$

则 NEWUOA_s 算法有限终止. 其中, M 为 $\|\nabla^2 f\|$ 的一个上界且 $M \geq 1$.

证明. 不妨假设对所有 $k \in \mathcal{K}$, 有

$$M\sqrt{n}\varepsilon h_k + M^2 c^2 p_k^2 \leq \delta^2. \quad (5.69)$$

其中, δ 为一个常数, 且 $\delta < \varepsilon$.

给定 $k \in \mathcal{K}$, 我们来估计第 k 步迭代的下降量. 令 P_k 表示 \mathbb{R}^n 到 \mathcal{S}_k 的正交投影. 由引理 5.5 以及假设 5.19 知

$$\begin{aligned} & f(x_k) - f(x_{k+1}) \\ & \geq f(x_k) - f(x_k + d_k) \\ & \geq f(x_k) - \inf_{d \in \mathcal{S}_k} f(x_k + d) - \frac{1}{2} M c^2 p_k^2 \\ & \geq \frac{1}{2M} \|P_k \nabla f(x_k)\|^2 - \frac{1}{2} M c^2 p_k^2. \end{aligned} \quad (5.70)$$

另一方面,

$$\begin{aligned} & \|P_k \nabla f(x_k)\|^2 \\ & = \|\nabla f(x_k)\|^2 - \|\nabla f(x_k) - P_k \nabla f(x_k)\|^2 \\ & \geq [\|\tilde{g}_k\| - \|\nabla f(x_k) - \tilde{g}_k\|]^2 - \|\nabla f(x_k) - \tilde{g}_k\|^2 \\ & = \|\tilde{g}_k\|^2 - 2\|\tilde{g}_k\| \|\nabla f(x_k) - \tilde{g}_k\| \\ & \geq \|\tilde{g}_k\|^2 - M\sqrt{n} \|\tilde{g}_k\| h_k. \end{aligned} \quad (5.71)$$

故

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2M} \|\tilde{g}_k\|^2 - \frac{1}{2} \sqrt{n} \|\tilde{g}_k\| h_k - \frac{1}{2} M c^2 p_k^2. \quad (5.72)$$

由 (5.69) 以及 $M \geq 1$ 知

$$h_k < \frac{\varepsilon}{M\sqrt{n}} \leq \varepsilon. \quad (5.73)$$

假若 NEWUOA_s 算法不有限终止, 则

$$\|\tilde{g}_k\| \geq \varepsilon > M\sqrt{n} h_k. \quad (5.74)$$

从而由 (5.72) 与 (5.69) 知,

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2M}\varepsilon^2 - \frac{1}{2}\sqrt{n}\varepsilon h_k - \frac{1}{2}M c^2 p_k^2 \geq \frac{1}{2M}(\varepsilon^2 - \delta^2). \quad (5.75)$$

以上估计对任何 $k \in \mathcal{K}$ 皆成立. 这与 \mathcal{K} 为无穷集且 f 有下界矛盾. ■

用同样的方法可以证明如下定理.

定理 5.23. 设目标函数 f 满足假设 5.4, 三次连续可微且三阶导数有界, NEWUOA 算法满足假设 5.19. 若 $\varepsilon > 0$, 且

$$\liminf_{k \in \mathcal{K}, k \rightarrow \infty} \frac{1}{3}\bar{M}\varepsilon h_k^2 + M^2 c^2 p_k^2 < \varepsilon^2, \quad (5.76)$$

则 NEWUOAs 算法有限终止. 其中, M 为 $\|\nabla^2 f\|$ 的一个上界, \bar{M} 为

$$\left\{ \sum_{i=1}^n \left[\frac{\partial^3 f}{\partial x_i^3} \right]^2 \right\}^{\frac{1}{2}} \quad (5.77)$$

的一个上界且 $\bar{M} \geq 3$.

定理 5.22 与 5.23 说明, 只要序列 $\{h_k\}$ 与 $\{p_k\}$ 的下界足够小, NEWUOAs 算法就会有限终止. 此外, (5.68) 与 (5.76) 为我们估计合理的下界提供了依据.

算法 NEWUOAs 的一个优点在于, 我们能够估计解的质量. 下面的定理给出了迭代点处梯度大小的估计. 这些估计是命题 5.17 的直接推论.

定理 5.24. 考虑 NEWUOAs 算法的迭代点 x_k .

a.) 若 f 在 \mathbb{R}^n 上二次连续可微且二阶导数有界, 则

$$\|\nabla f(x_k)\| \leq \|\tilde{g}_k\| + \frac{1}{2} \sup_{x \in \mathbb{R}^n} \left\{ \sum_{i=1}^n \left[\frac{\partial^2 f}{\partial x_i^2}(x) \right]^2 \right\}^{\frac{1}{2}} h_k; \quad (5.78)$$

b.) 若 f 在 \mathbb{R}^n 上三次连续可微且三阶导数有界, 则

$$\|\nabla f(x_k)\| \leq \|\tilde{g}_k\| + \frac{1}{6} \sup_{x \in \mathbb{R}^n} \left\{ \sum_{i=1}^n \left[\frac{\partial^3 f}{\partial x_i^3}(x) \right]^2 \right\}^{\frac{1}{2}} h_k^2. \quad (5.79)$$

§5.3.5 NEWUOAs 算法的实现细节

下面我们介绍算法 5.18 中未提到的实现细节.

首先是序列 $\{h_k\}$ 的选取. h_k 控制了第 k 步插值点集的集中程度. 当 $m_k = 2n + 1$ 时, 插值等价于中心差分, h_k 就是差分步长. 我们认为 h_1 不宜过小: 过早采用过于集中的插值点集会丧失捕捉到目标函数的大尺度信息的机会, 使迭代陷于低质量的局部极小值点. h_1 在尺度上应大致反映初始点离最优点的距离. 在确定 h_1 之后, 我们令

$$h_k = \rho_1^{k-1} h_1, \quad (5.80)$$

其中 $\rho_1 \in (0, 1)$ 是一个常数.

其次是序列 $\{p_k\}$ 的选取. p_k 控制了第 k 步迭代中子问题 (5.60) 的求解精度. 实际计算中, 我们发现, 最初几次迭代的子空间往往包含非常好的点 (即使这时近似梯度 \tilde{g}_k 还很不精确), 所以我们希望一开始就充分求解子问题 (5.60). 故我们取

$$p_k = \min\{\varepsilon, \rho_2^k\}, \quad (5.81)$$

其中 $\rho_2 \in (0, 1)$ 是一个常数.

§5.3.4 提到, 在实际计算中, 令序列 $\{h_k\}$ 与 $\{p_k\}$ 趋于零可能是不合理的. 因此我们建议给它们规定正的下界. 要保证 NEWUOAs 算法有限终止, 合理的下界可以根据 (5.68) 或 (5.76) 得到. 在实际计算中, 我们观察到, c 一般为 \sqrt{n} 量级的⁴. 假设 $c \leq \sqrt{n}$, 则由 (5.68) 知, $\{h_k\}$ 与 $\{p_k\}$ 的一个比较合理的下界是 $\varepsilon/(2M\sqrt{n})$, 其中 M 为 $\|\nabla^2 f\|$ 不小于 1 的一个上界估计. 结合 (5.80) 与 (5.81), 我们可以取

$$h_k = \max\left\{\rho_1^{k-1} h_1, \frac{\varepsilon}{2M\sqrt{n}}\right\}, \quad (5.82)$$

$$p_k = \max\left\{\min\{\varepsilon, \rho_2^k\}, \frac{\varepsilon}{2M\sqrt{n}}\right\}. \quad (5.83)$$

在实际计算中我们用 (5.82) 定义 $\{h_k\}$, 用 (5.83) 定义 $\{p_k\}$. 如果假设 f 三次连续可微且三阶导数有界, 则序列 $\{h_k\}$ 还允许更大的下界. 当然, 因为我们无法得到 (5.68) 与 (5.76) 中的常数 c 、 M 和 \bar{M} 的准确值, 所以只能估计合理的

⁴这是因为, 一般情况下, NEWUOA 得到的解每一个分量都能达到 RHOEND 量级的精度.

下界应该具有什么量级. 采用 (5.82)–(5.83) 定义 $\{h_k\}$ 和 $\{p_k\}$, NEWUOAs 算法的有限终止性是没有理论保证的, 因而需要增加其他的终止准则, 比如规定最大函数值计算次数.

在调用 NEWUOA 求解子问题 (5.60) 之前, 我们先构造矩阵 $B \in \mathbb{R}^{n \times \bar{n}}$, 使得 B 的列构成子空间 \mathcal{S}_k 的一组标准正交基, 其中 \bar{n} 为 \mathcal{S}_k 之维数. 按算法 5.18 的描述, \bar{n} 为 1 或 2. 利用 B , 子问题 (5.60) 可转化为

$$\min_{\bar{d} \in \mathbb{R}^{\bar{n}}} f(x_k + B\bar{d}). \quad (5.84)$$

问题 (5.84) 可以直接调用 NEWUOA 求解. 求解的初始点为 0. 在得到问题 (5.84) 的解 \bar{d}_k 之后, 令 $d_k = B\bar{d}_k$ 即得到子问题 (5.60) 的解.

调用 NEWUOA 时, 我们除了需要设置 RHOEND 之外, 还需要设置其初始信赖域半径 RHOBEG. 实际计算中, 我们发现, 比较大的初始信赖域半径更有优势. 我们更新 RHOBEG 的策略是:

$$\text{RHOBEG}_{k+1} = \max\{p_{k+1}, h_{k+1}, \|d_k\|, \rho_3 \text{RHOBEG}_k\}, \quad (5.85)$$

其中 $\rho_3 \in (0, 1)$ 是一个常数.

调用 NEWUOA 时, 另外一个需要设置的参数是每一个模型使用的插值点个数 NPT. 数值实验表明, 求解小规模问题时, 完全二次插值优于欠定二次插值和对称 Broyden 修正. 所以我们建议取 $\text{NPT} = (\bar{n} + 1)(\bar{n} + 2)/2$, 其中 \bar{n} 为子问题维数. 按照算法 5.18 的描述, $\bar{n} = 2$, 故 $\text{NPT} = 6$.

在调用 MODEL 构造 \tilde{g}_k 时, 我们选取的插值点中可能会出现比 x_k 函数值更优的点. 我们不妨把插值点中函数值最优的点记为 \hat{x}_k . 如果 $f(\hat{x}_k) < f(x_k)$, 那么比较合理的做法是把子问题 (5.60) 中的 x_k 更改为 \hat{x}_k . 在实际计算中我们采用这种策略. 但是这种做法带来的改变并不大, 原因是插值点集的直径在迭代中会较快的趋于零. 用 \hat{x}_k 替代 x_k 之后, 需要将 \tilde{g}_k 重新定义为 $\nabla Q_k(\hat{x}_k)$. 当 $k \geq 2$ 时, 我们还重新定义 s_{k-1} 为 $\hat{x}_k - x_k + s_{k-1}$.

最后, 如果累计三次出现 $\|d_k\|$ 很小, 比如 $\|d_k\| < \alpha_1 \varepsilon$ (α_1 是一个小正数), 则很有可能继续迭代并不会带来本质的改进, 故我们建议此时终止迭代.

§5.3.6 NEWUOAs 算法的预条件技术

算法 5.18 与 §5.3.5 介绍了 NEWUOAs 算法的全部实现过程. 那么, 这样得到的 NEWUOAs 算法计算效果如何? 我们先来看一个例子. 考虑测试问题 DQRTIC

表 5.1: NEWUOA 算法与 NEWUOAs 算法对 DQRTIC 问题的表现

| | NEWUOA | | NEWUOAs | |
|-----|--------|--------------|---------|--------------|
| 100 | 10000 | 1.248559E-18 | 4906 | 1.772290E-17 |
| 150 | 10000 | 4.926753E-01 | 7112 | 4.344702E-17 |
| 200 | 10000 | 1.263032E+03 | 9233 | 4.256090E-12 |
| 250 | 10000 | 1.980351E+05 | 10000 | 5.472021E-10 |

[67]. 其目标函数为

$$f(x) = \sum_{i=1}^n (x_i - i)^4 \quad (5.86)$$

初始点为 $(2, 2, \dots, 2)$, 最优值为 0. 表 5.1 给出了 NEWUOA 算法和 NEWUOAs 算法求解 DQRTIC 问题 ($n = 100, 150, 200, 250$) 的函数值计算次数以及得到的最优函数值. 计算中 NEWUOA 的参数 NPT、RHOBEQ 和 RHOEND 分别取 $2n + 1$ 、1 和 10^{-6} ; NEWUOAs 的参数为 $m_k = 2n + 1$, $\varepsilon = 10^{-6}$, $h_1 = 1$, $\rho_1 = \rho_2 = \rho_3 = 0.5$, $M = 50$, $\alpha_1 = 0.1$. 另外, 我们限制最大函数值计算次数为 10000. 与以往不同, 对每一个问题我们仅按原始变量顺序求解一次, 不进行随机置换. 在这个问题上, NEWUOAs 算法的效率明显高于 NEWUOA, 充分显示了子空间技术对较大规模问题的威力.

现在, 让我们来看另外一个例子. 考虑测试问题 POWER [67]. 其目标函数为

$$f(x) = \sum_{i=1}^n i^2 x_i^2, \quad (5.87)$$

初始点为 $(1, 1, \dots, 1)$, 最优值为 0. 表 5.2 给出了 NEWUOA 算法和 NEWUOAs 算法求解 POWER 问题 ($n = 100, 150, 200, 250$) 的函数值计算次数以及得到的最优函数值. 计算中 NEWUOA 与 NEWUOAs 的参数选取同上; 最大函数值计算次数仍置为 10000; 我们仍然只求解原始变量顺序下的问题. 在这个问题上, NEWUOAs 算法遇到了困难.

现在我们来分析 NEWUOAs 算法在 POWER 问题上表现不佳的原因. 因为 POWER 是一个二次函数, 且 Hessian 矩阵为对角矩阵, 所以我们每次插值得到的二次模型是精确的. 故对于该问题 NEWUOAs 算法大致相当于共轭梯度法⁵.

⁵但不完全相同. 这是因为, 如果插值构造模型时出现函数值更优的点, 迭代点会相应移动.

表 5.2: NEWUOA 算法与 NEWUOAs 算法对 POWER 问题的表现

| | NEWUOA | | NEWUOAs | |
|-----|--------|--------------|---------|--------------|
| 100 | 244 | 5.940111E-15 | 10000 | 6.734304E+00 |
| 150 | 350 | 6.366113E-13 | 10000 | 6.541123E+01 |
| 200 | 455 | 7.894867E-13 | 10000 | 3.301559E+02 |
| 250 | 565 | 2.627950E-12 | 10000 | 1.110297E+03 |

而 POWER 问题的条件数很大, 故 NEWUOAs 算法表现不佳是很正常的.

对于一般问题, 我们在建立算法 5.3 的 R-线性收敛速度 (定理 5.14) 时已经看到, 问题的条件数很大时算法的收敛速度可以很慢. 这是所有基于算法 5.3 框架的方法都无法回避的问题, 包括 NEWUOAs 算法. 所以, 有必要在 NEWUOAs 算法中引入预条件技术.

但问题是, 有效的预条件需要用到问题的二阶信息, 这对于我们来说是难以实现的. 幸运的是, 在第 k 步迭代中, 我们不仅构造了 x_k 处的近似梯度 \tilde{g}_k , 而且还通过插值构造了一个二次模型 Q_k . 我们可以把 $\nabla^2 Q_k$ 作为 x_k 处的近似 Hessian 矩阵, 用它进行预条件.

假设 $\nabla^2 Q_k$ 是一个正定矩阵, 那么在子空间 \mathcal{S}_k 的定义 (5.59) 中用 $[\nabla^2 Q_k]^{-1} \tilde{g}_k$ 代替 \tilde{g}_k , 就完成了预条件. 如果 $\nabla^2 Q_k$ 不是正定矩阵, 应该如何处理? 这时我们仍试图由 $\nabla^2 Q_k$ 构造一个矩阵 A_k , 然后令 $A_k \tilde{g}_k$ 代替 \tilde{g}_k 用于子空间 \mathcal{S}_k 的定义. 那么 A_k 应如何选取?

为了叙述方便, 我们把 $\nabla^2 Q_k$ 当做 f 的精确 Hessian 矩阵, 并且假设 $\nabla^2 Q_k$ 是一个对角矩阵

$$\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n), \quad (5.88)$$

否则做一个旋转即可. 一个最直接的方法是取 A_k 为 $\nabla^2 Q_k$ 的广义逆, 也就是

$$\text{diag}(\lambda_1^+, \lambda_2^+, \dots, \lambda_n^+), \quad (5.89)$$

其中

$$\lambda^+ = \begin{cases} \lambda^{-1} & \text{若 } \lambda \neq 0, \\ 0 & \text{若 } \lambda = 0. \end{cases} \quad (5.90)$$

这种做法有两个缺点. 第一, 广义逆是不连续的, 数值上不稳定. 第二, 这种做法对负曲率方向的处理不合理. 我们来解释为什么这么说. 假定 $\lambda_i < 0$, 则 f 在第 i 个坐标方向 e_i 有负曲率; 并且, $|\lambda_i|$ 越大负曲率越大, 从而 f 越容易在子空间 $\text{span}\{e_i\}$ 上获得大的下降. 但是, 如果取 A_k 为 $\nabla^2 Q_k$ 的广义逆, 则 $|\lambda_i|$ 越大, $A_k \tilde{g}_k$ 离子空间 $\text{span}\{e_i\}$ 越远, $|\lambda_i|$ 很大时二者几近垂直. 这时取子空间 $\mathcal{S}_k = \text{span}\{A_k \tilde{g}_k, s_{k-1}\}$ 就会使第 k 步迭代浪费在 $\text{span}\{e_i\}$ 上获得下降的机会, 这正与我们想要的相反. 当出现负曲率方向时, 我们希望 $A_k \tilde{g}_k$ 靠近负曲率方向, 并且负曲率越大越靠近. 基于这种考虑, 我们定义

$$\varphi(\lambda) = \begin{cases} \lambda^{-1} & \text{若 } \lambda > \varepsilon_0, \\ -\frac{\lambda}{\varepsilon_0^2} + \frac{2}{\varepsilon_0} & \text{若 } \lambda \leq \varepsilon_0, \end{cases} \quad (5.91)$$

其中 ε_0 是一个小正数; 进一步, 定义 A_k 为

$$\text{diag}(\varphi(\lambda_1), \varphi(\lambda_2), \dots, \varphi(\lambda_n)). \quad (5.92)$$

对于一般的 $\nabla^2 Q_k$, 若其特征值分解为

$$\nabla^2 Q_k = U \Lambda U^T, \quad (5.93)$$

U 为正交矩阵, Λ 为对角矩阵, 则令

$$A_k = \varphi(\nabla^2 Q_k) \equiv U \varphi(\Lambda) U^T. \quad (5.94)$$

其中, $\varphi(\Lambda)$ 的定义依照 (5.92). 在计算中, 若 $m_k = 2n + 1$, 则 $\nabla^2 Q_k$ 为对角矩阵, 因而 A_k 是容易得到的. 在 NEWUOAs 算法中我们取 $\varepsilon_0 = \alpha_2 \max_{1 \leq i \leq n} |\lambda_i|$, α_2 为一个小正数.

定义好 A_k 之后, 我们取 \mathcal{S}_k 为 $\text{span}\{A_k \tilde{g}_k, s_{k-1}\}$ 就完成了预条件. 但一个新的问题是, 这样选择的 \mathcal{S}_k 不能保证算法收敛, 因为我们需要 \mathcal{S}_k 中包含一个近似梯度. 因此, 我们定义

$$\mathcal{S}_k = \text{span}\{\tilde{g}_k, A_k \tilde{g}_k, s_{k-1}\}. \quad (5.95)$$

这就是我们在 §5.3.3 提到的另外一种子空间的定义方式.

预条件对 NEWUOAs 算法带来的改进是不言而喻的. 比如, 对于 100、150、200 和 250 维的 POWER 问题, 预条件之后的 NEWUOAs 算法分别在函数值

计算次数为 232、332、432 和 533 时求到了精度高于 10^{-20} 的解. 事实上, 因为 POWER 是正定二次函数, 且 Hessian 矩阵为对角阵, 所以预条件后的 NEWUOAs 算法在理论上能一次求得问题最优解.

从现在开始, 我们所指的 NEWUOAs 算法都是指使用了预条件技术的算法, 也就是说, S_k 由 (5.95) 定义.

§5.3.7 数值实验

本小节通过数值实验证明 NEWUOAs 算法是成功的. 我们进行三组实验. 前两组实验, 我们将 NEWUOAs 算法与目前最优秀的无导数算法之一 NEWUOA 算法进行比较; 我们分别比较了较大规模问题和坏初始点问题. 第三组实验, 我们给出 NEWUOAs 求解部分 2000 维问题的例子, 以此证明 NEWUOAs 求解大规模问题的能力. 因为 2000 维问题对 NEWUOA 算法来说几乎是不可解的, 所以第三组实验没有进行对比.

§5.3.7.1 NEWUOAs 算法与 NEWUOA 算法的比较 — 较大规模问题

现在我们采用第二章 §2.4 建立的评价体系比较 NEWUOA 算法与 NEWUOAs 算法对较大规模问题的表现. 我们的测试问题集是 §3.3.2 中使用的 50 个可变维数的无约束优化问题. 对于每一个测试问题, 我们对 50、100、150、200 等 4 个维数进行了求解⁶. 每一次求解都按照 §2.4 做了 5 次随机置换⁷. 我们统计了求解精度 (即条件 (2.5) 中的 τ) 为 10^{-i} ($i = 2, 4, 6, 8, 10$) 时的函数值计算次数, 也统计了迭代自然终止时的函数值计算次数 (条件 (2.7) 中取⁸ $\varepsilon = 10^{-6}$).

NEWUOA 算法的代码中用户可选参数 NPT、RHOBEG 和 RHOEND 的选取与 §3.3.2 相同. NEWUOAs 算法用 Fortran 77 实现. 算法中的参数设置为: $m_k = 2n + 1$, $\varepsilon = 10^{-6}$, h_1 与 NEWUOA 中 RHOBEG 相等, $\rho_1 = \rho_2 = \rho_3 = 0.5$, $M = 50$, $\alpha_1 = 0.1$, $\alpha_2 = 10^{-6}$. 另外, 实验中, 我们设置最大函数值计算次数为 $100n$, n 为问题维数.

⁶为了在可接受的时间内得到实验结果, 这一组实验没有求解更大规模的问题. 关于更大规模问题的例子, 请参考 §5.3.7.3.

⁷由于问题维数较大, 计算比较耗时, 我们只做了 5 次随机置换, 这与前面测试小规模问题是不同的.

⁸因为 NEWUOA 算法与 NEWUOAs 算法是类型完全不同的算法, 所以我们比较时放宽了判定是否求到同一最优解的标准. 请参考第三章注 8. 宽松的判定标准于 NEWUOAs 算法并不有利, 因为该算法倾向于得到精度很高的解.

需要注意的是, 实际计算中, NEWUOA 算法的参数 RHOEND 往往相当于解点与最优解的 l_∞ 距离, 而 NEWUOAs 算法的参数 ε 大致相当于解点与最优解的 l_2 距离 (请参考定理 5.24). 因而, 在比较这两个算法时, 合理的参数选择应考虑问题维数的影响. 比如, 假设解点与最优点在各分量上的距离大致相当, 则设置 $\varepsilon = \sqrt{n}\text{RHOEND}$ 是合理的. 但是, 由于这里测试的问题维数至多 200, 所以我们忽略这一因素, 在比较时总是设置 $\varepsilon = \text{RHOEND}$. 这导致 NEWUOAs 得到的解精度往往很高, 从而花费更多的函数值计算次数.

我们的实验环境与 §3.3.2 相同.

图 5.8 至图 5.13 给出了 NEWUOA 算法与 NEWUOAs 算法在满足不同精度要求以及迭代自然终止时的 Performance Profile、Data Profile、Sensitivity Profile 和 R-Sensitivity Profile. 图线越高说明算法表现越好. 四种 Profile 的定义请参考第二章.

从图中可以看出, 两种算法的对比十分明显. 在不同终止条件下, 不论是算法的函数值计算次数 (体现在 Performance Profile 和 Data Profile), 还是算法对计算机舍入误差的敏感性 (体现在 Sensitivity Profile 和 R-Sensitivity Profile), NEWUOAs 算法的表现都明显好于 NEWUOA 算法.

为了更清楚地看到 NEWUOAs 算法的优势, 下面我们以求解精度 $\tau = 10^{-10}$ (图 5.12) 为例, 分析这两个算法的 Performance Profile 和 Data Profile 数据. Sensitivity Profile 和 R-Sensitivity Profile 的分析类似于 Performance Profile.

在图 5.12 的 Performance Profile 中, NEWUOA 图线的起点为 (0, 0.125), 第一个最高点为 (4.828, 0.520); NEWUOAs 图线的起点为 (0, 0.650), 第一个最高点为 (1.565, 0.755). 这说明, 当精度要求为 10^{-10} 时, 我们有:

- a.) NEWUOA 算法成功求解⁹了 52.0% 的问题; NEWUOAs 算法成功求解了 75.5% 的问题.
- b.) 有 12.5% 的问题, NEWUOA 算法花费的函数值计算次数少于或等于 NEWUOAs 算法; 有 65.0% 的问题, NEWUOAs 算法花费的函数值计算次数少于或等于 NEWUOA 算法.
- c.) 最好情形下, NEWUOAs 算法花费的函数值计算次数约为 NEWUOA 算法的

⁹关于成功求解的判定条件, 请参考第二章的 (2.5) 和 (2.20).

0.035 倍 ($2^{-4.828} \approx 0.035$); 最坏情形下, NEWUOA_s 算法花费的函数值计算次数约为 NEWUOA 算法的 2.96 倍 ($2^{1.565} \approx 2.96$).

在图 5.12 的 Data Profile 中, NEWUOA 图线对应于横坐标 10、20、30 和 40 的纵坐标分别为 0.100、0.165、0.225 和 0.335; NEWUOA_s 图线对应于横坐标 10、20、30 和 40 的纵坐标分别为 0.440、0.565、0.670 和 0.720. 这说明, 当精度要求为 10^{-10} 时, 我们有:

- a.) 给定 10 个单纯形梯度的预算, NEWUOA 算法可成功求解 10.0% 的问题, 而 NEWUOA_s 算法可成功求解 44.0% 的问题.
- b.) 给定 20 个单纯形梯度的预算, NEWUOA 算法可成功求解 16.5% 的问题, 而 NEWUOA_s 算法可成功求解 56.5% 的问题.
- c.) 给定 30 个单纯形梯度的预算, NEWUOA 算法可成功求解 22.5% 的问题, 而 NEWUOA_s 算法可成功求解 67.0% 的问题.
- d.) 给定 40 个单纯形梯度的预算, NEWUOA 算法可成功求解 33.5% 的问题, 而 NEWUOA_s 算法可成功求解 72.0% 的问题.

实验结果充分说明, 对于较大规模的问题, 与目前最优秀的无导数算法之一 NEWUOA 相比, NEWUOA_s 算法的优势是十分显著的.

为了在可接受的时间内得到实验结果, 本次试验没有求解 200 维以上的问题. 事实上, 问题规模越大 NEWUOA_s 算法的优势越明显. 关于更大规模的问题, 请参考 §5.3.7.3 的实验. 那里我们将用 NEWUOA_s 求解 2000 维的问题, 这一问题对 NEWUOA 来说几乎是不可求解的.

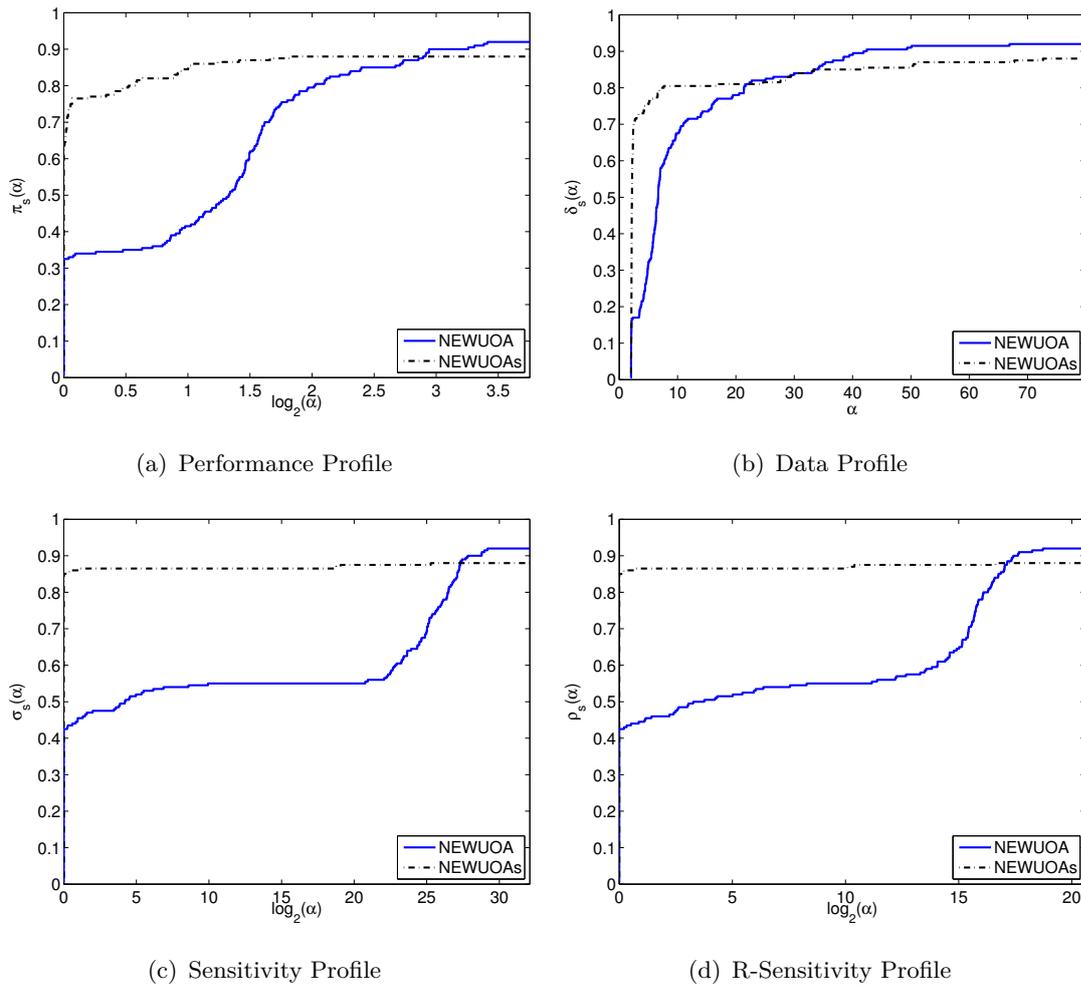
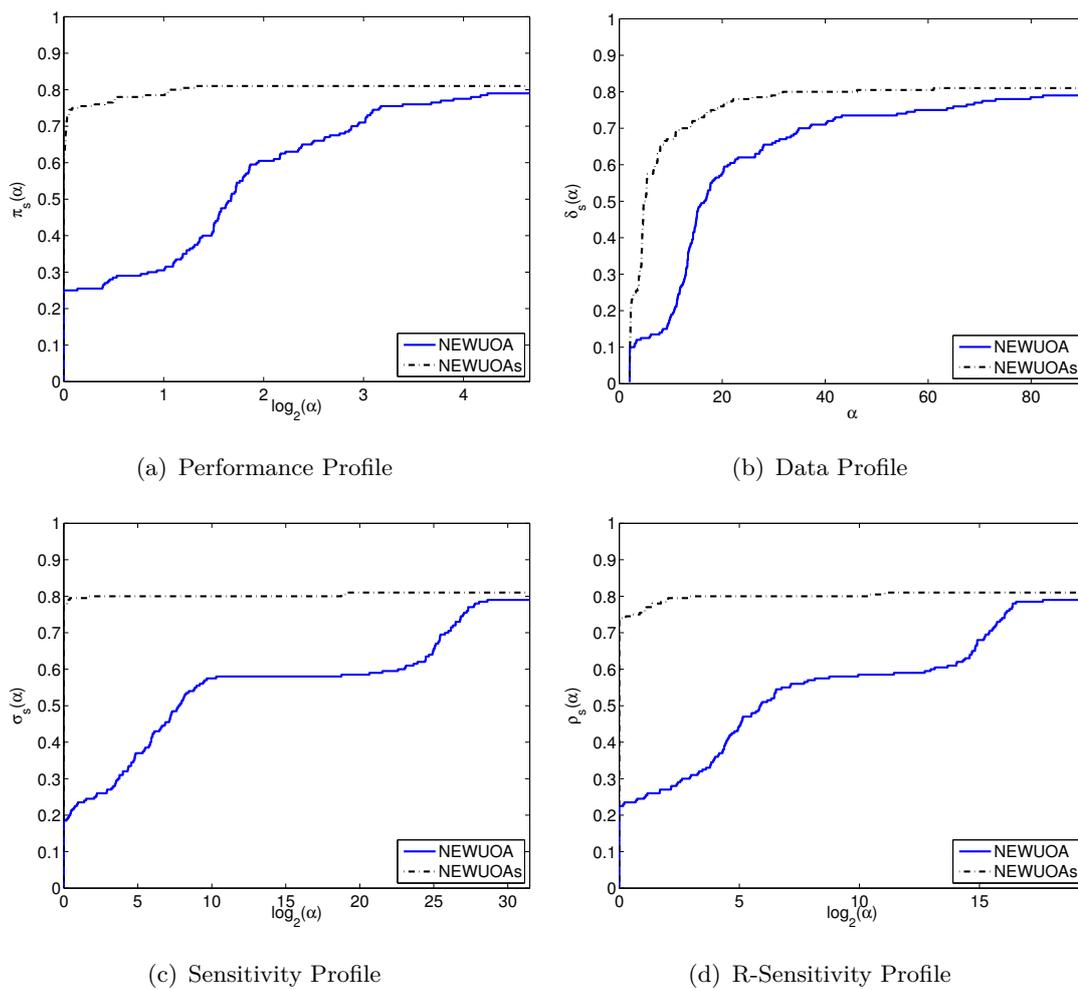


图 5.8: NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (精度 $\tau = 10^{-2}$)

图 5.9: NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (精度 $\tau = 10^{-4}$)

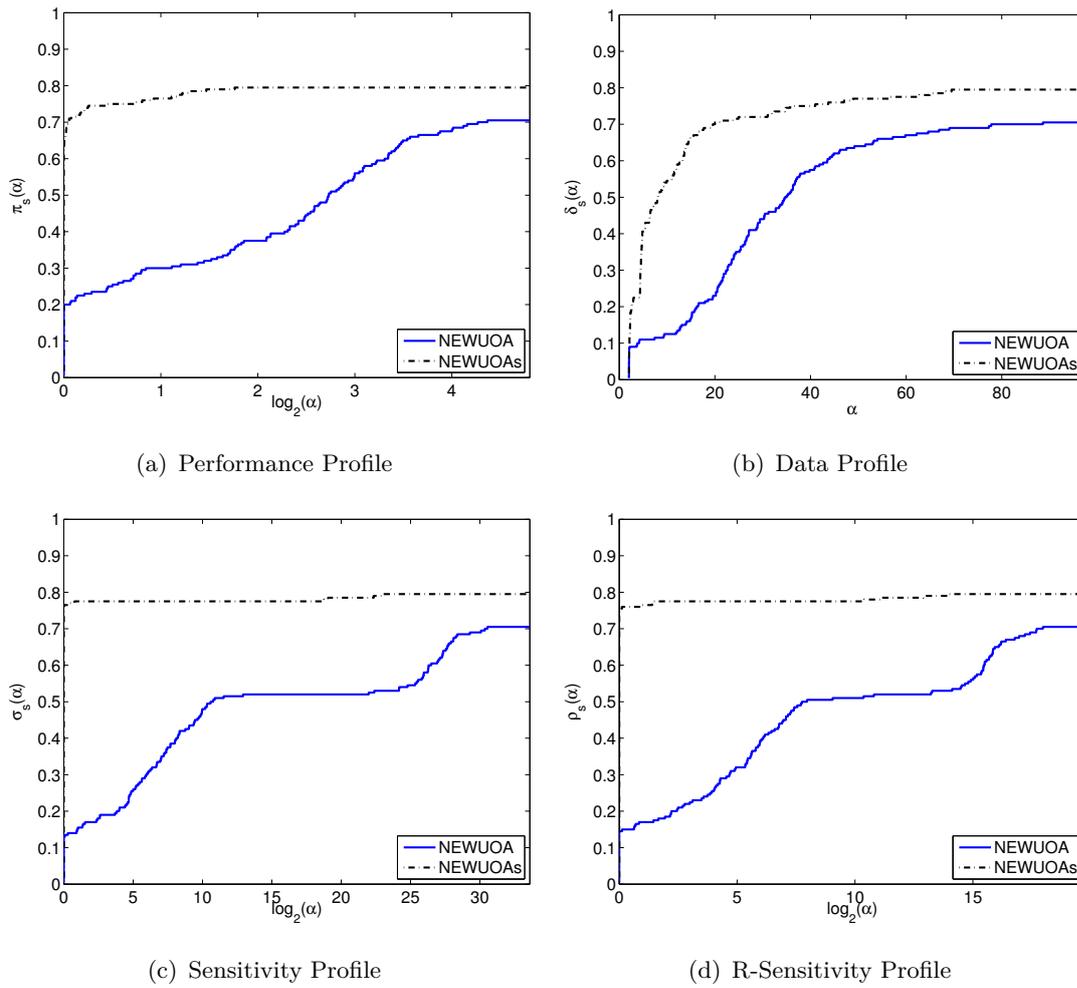


图 5.10: NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (精度 $\tau = 10^{-6}$)

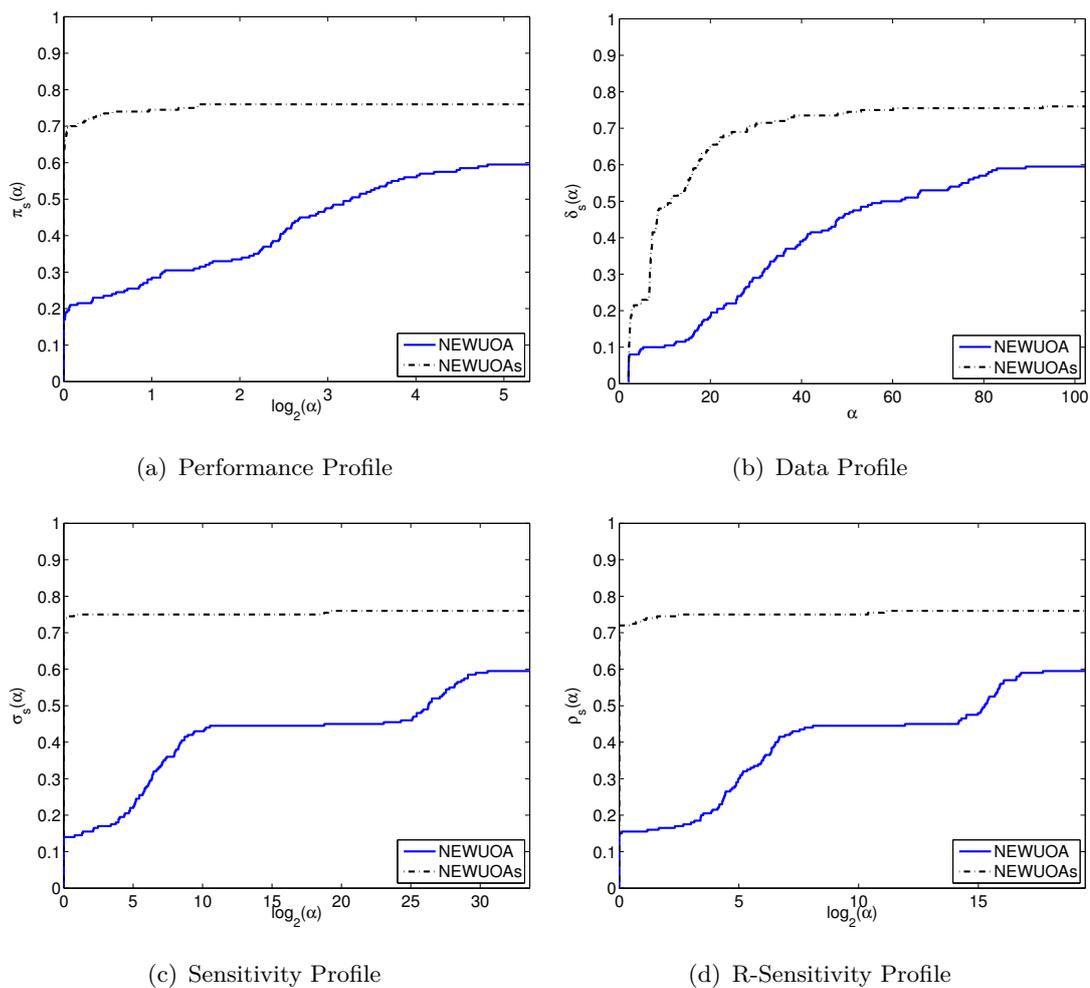


图 5.11: NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (精度 $\tau = 10^{-8}$)

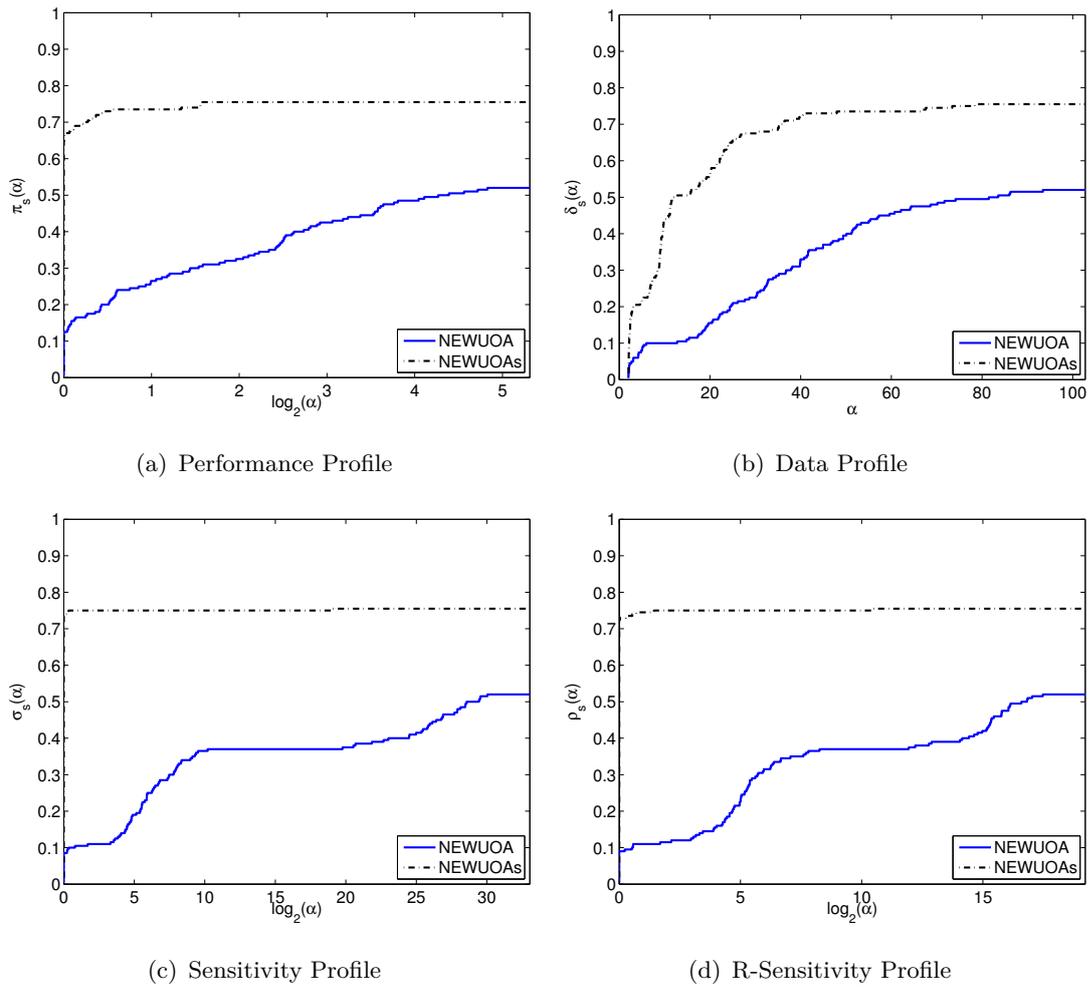


图 5.12: NEWUOA 与 NEWUOAs 对较大规模问题的数值表现 (精度 $\tau = 10^{-10}$)

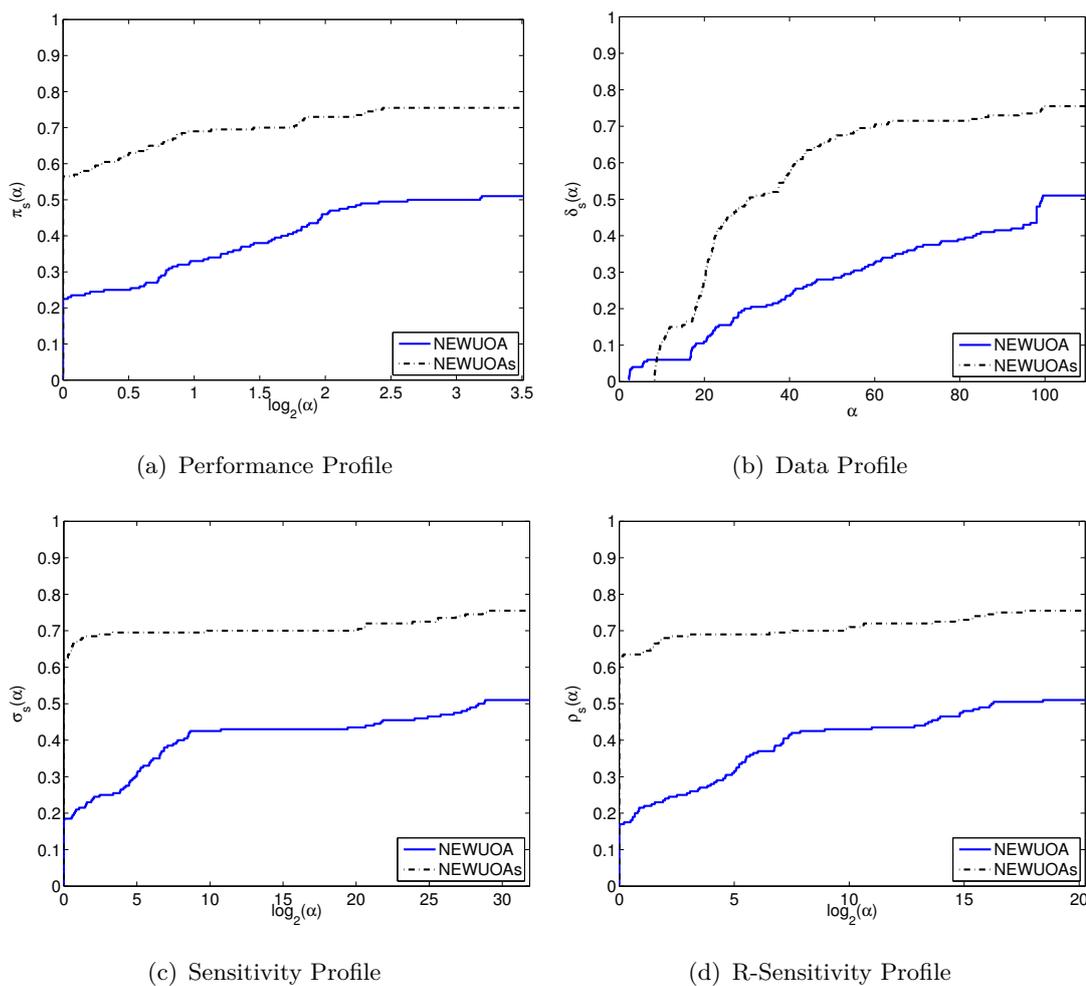


图 5.13: NEWUOA 与 NEWUOA_s 对较大规模问题的数值表现 (迭代自然终止)

§5.3.7.2 NEWUOAs 算法与 NEWUOA 算法的比较 — 坏初始点问题

现在我们采用第二章 §2.4 建立的评价体系比较 NEWUOA 算法与 NEWUOAs 算法对坏初始点问题的表现. 我们的测试问题集是 §3.3.2 中使用的 50 个可变维数的无约束优化问题. 为了得到坏初始点问题, 我们对测试问题的初始点作了修改: 设 \hat{x} 为标准的初始点, 则我们的实验中将初始点取为

$$(\hat{x}_1 + 100, \hat{x}_2 + 100, \dots, \hat{x}_n + 100). \quad (5.96)$$

对于每一个测试问题, 我们对 10、20、30、40 等 4 个维数进行了求解¹⁰. 每一次求解都按照 §2.4 做了 10 次随机置换. 我们统计了求解精度 (即条件 (2.5) 中的 τ) 为 10^{-i} ($i = 2, 4, 6, 8, 10$) 时的函数值计算次数, 也统计了迭代自然终止时的函数值计算次数 (条件 (2.7) 中取 $\varepsilon = 10^{-6}$).

两个算法中参数的设置以及实验环境与 §5.3.7.1 相同.

图 5.14 至图 5.19 给出了 NEWUOA 算法与 NEWUOAs 算法在满足不同精度要求以及迭代自然终止时的 Performance Profile、Data Profile、Sensitivity Profile 和 R-Sensitivity Profile. 图线越高说明算法表现越好. 四种 Profile 的定义请参考第二章.

从图中可以看出, 两种算法的对比是非常明显. 在不同终止条件下, 不论是算法的函数值计算次数 (体现在 Performance Profile 和 Data Profile), 还是算法对计算机舍入误差的敏感性 (体现在 Sensitivity Profile 和 R-Sensitivity Profile), NEWUOAs 算法的表现都明显好于 NEWUOA 算法.

为了更清楚地看到 NEWUOAs 算法的优势, 下面我们以求解精度 $\tau = 10^{-10}$ (图 5.18) 为例, 分析这两个算法的 Performance Profile 和 Data Profile 数据. Sensitivity Profile 和 R-Sensitivity Profile 的分析类似于 Performance Profile.

在图 5.18 的 Performance Profile 中, NEWUOA 图线的起点为 (0, 0.110), 第一个最高点为 (4.060, 0.445); NEWUOAs 图线的起点为 (0, 0.800), 第一个最高点为 (1.954, 0.855). 这说明, 当精度要求为 10^{-10} 时, 我们有:

- a.) NEWUOA 算法成功求解¹¹了 44.5% 的问题; NEWUOAs 算法成功求解了 85.5% 的问题.

¹⁰为了在可接受的时间内得到实验结果, 这一组实验只求解了小规模的问题. 事实上, 问题规模越大 NEWUOAs 算法越有优势.

¹¹关于成功求解的判定条件, 请参考第二章的 (2.5) 和 (2.20).

- b.) 有 11.0% 的问题, NEWUOA 算法花费的函数值计算次数少于或等于 NEWUOA_s 算法; 有 80.0% 的问题, NEWUOA_s 算法花费的函数值计算次数少于或等于 NEWUOA 算法.
- c.) 最好情形下, NEWUOA_s 算法花费的函数值计算次数约为 NEWUOA 算法的 0.060 倍 ($2^{-4.060} \approx 0.060$); 最坏情形下, NEWUOA_s 算法花费的函数值计算次数约为 NEWUOA 算法的 3.87 倍 ($2^{1.954} \approx 3.87$).

在图 5.18 的 Data Profile 中, NEWUOA 图线对应于横坐标 10、20、30 和 40 的纵坐标分别为 0.080、0.125、0.155 和 0.190; NEWUOA_s 图线对应于横坐标 10、20、30 和 40 的纵坐标分别为 0.330、0.660、0.765 和 0.815. 这说明, 当精度要求为 10^{-10} 时, 我们有:

- a.) 给定 10 个单纯形梯度的预算, NEWUOA 算法可成功求解 8.0% 的问题, 而 NEWUOA_s 算法可成功求解 33.0% 的问题.
- b.) 给定 20 个单纯形梯度的预算, NEWUOA 算法可成功求解 12.5% 的问题, 而 NEWUOA_s 算法可成功求解 66.0% 的问题.
- c.) 给定 30 个单纯形梯度的预算, NEWUOA 算法可成功求解 15.5% 的问题, 而 NEWUOA_s 算法可成功求解 76.5% 的问题.
- d.) 给定 40 个单纯形梯度的预算, NEWUOA 算法可成功求解 19.0% 的问题, 而 NEWUOA_s 算法可成功求解 81.5% 的问题.

与 §5.3.7.1 的结果相比, 初始点变坏使得 NEWUOA_s 算法的优势变得更加明显. 事实上, 子空间的技巧使得 NEWUOA_s 能够充分利用下降方向, 在迭代最初几步迅速获得下降. 因而 NEWUOA_s 算法对初始点不敏感, 很适合求解坏初始点问题. 这对实际应用领域的意义是显而易见的, 因为很多实际问题很难给出一个好的初始点.

实验结果充分说明, 对于坏初始点问题, 与目前最优秀的无导数算法之一 NEWUOA 相比, NEWUOA_s 算法的优势是非常显著的; NEWUOA_s 算法很适合求解坏初始点问题.

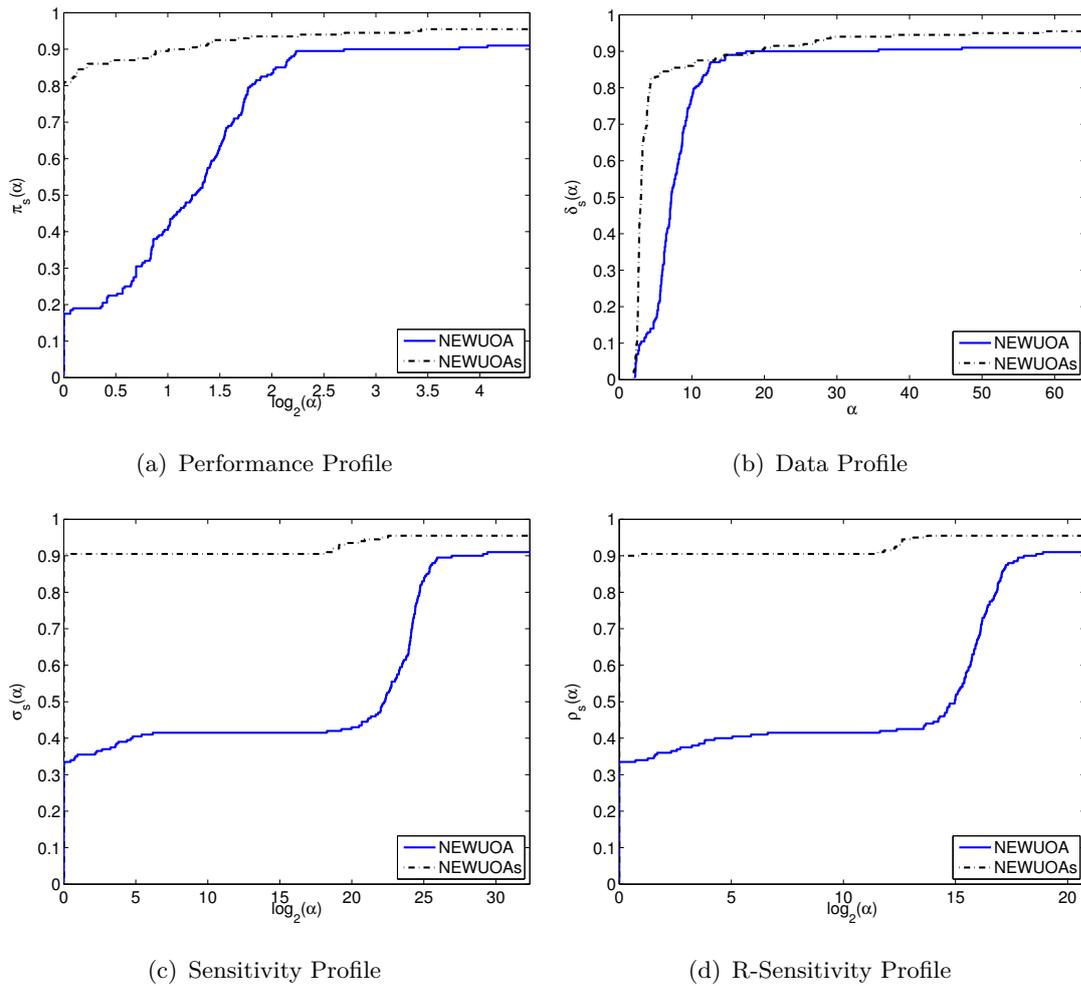
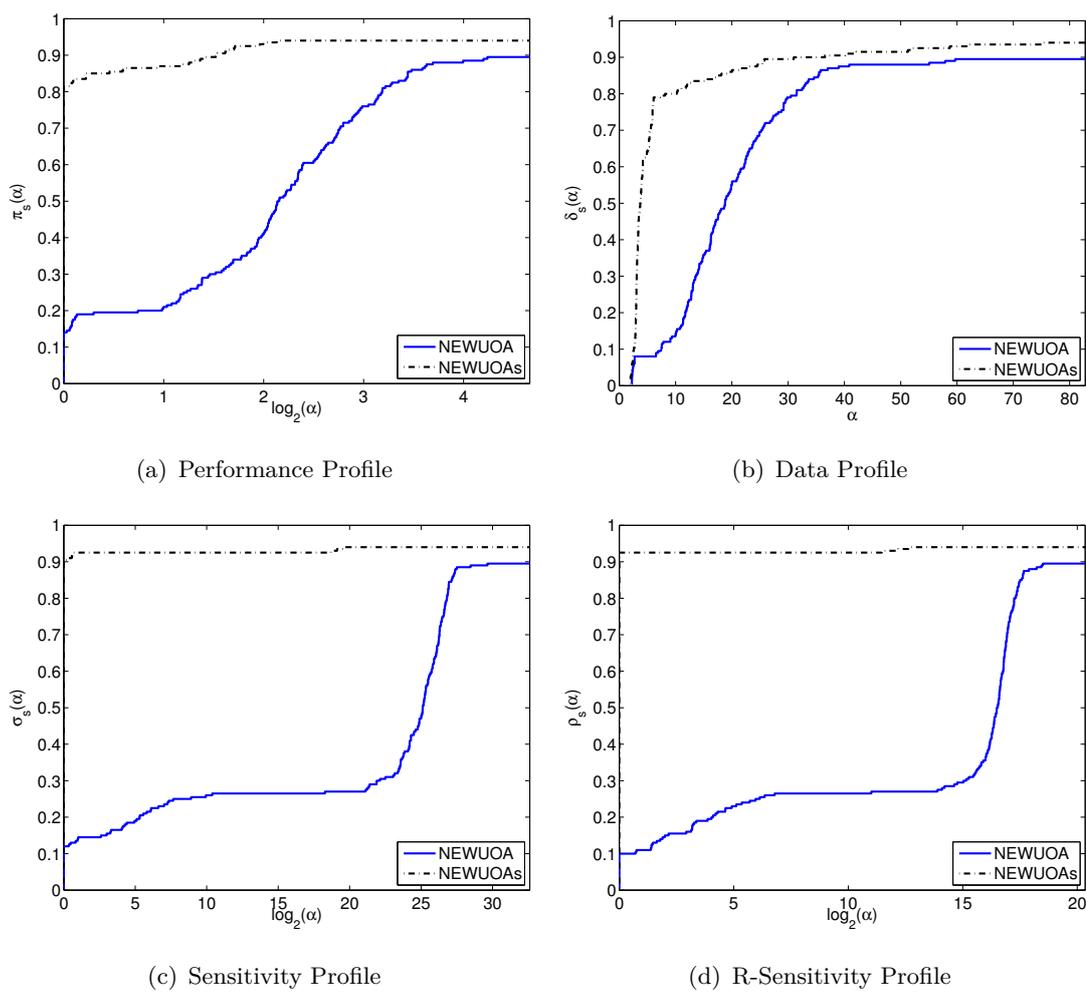


图 5.14: NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (精度 $\tau = 10^{-2}$)

图 5.15: NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (精度 $\tau = 10^{-4}$)

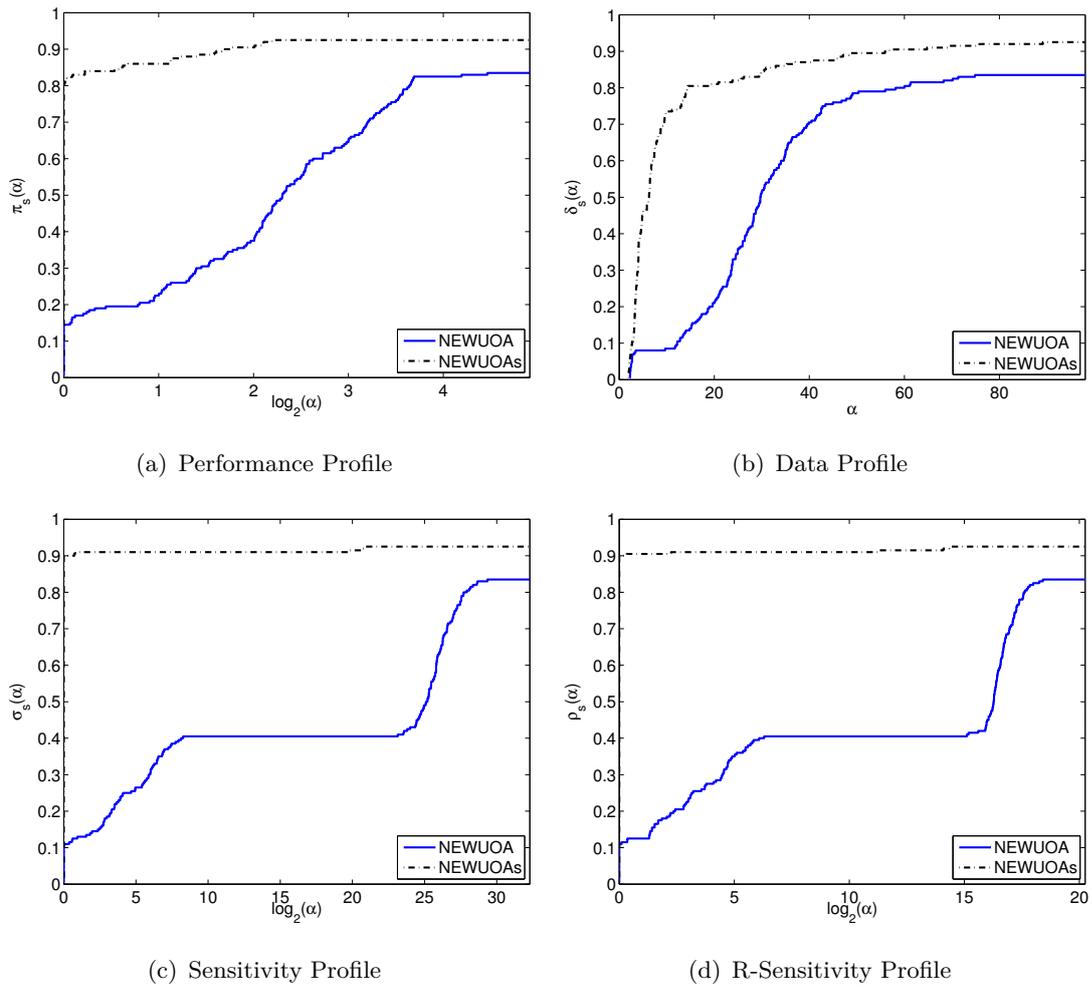
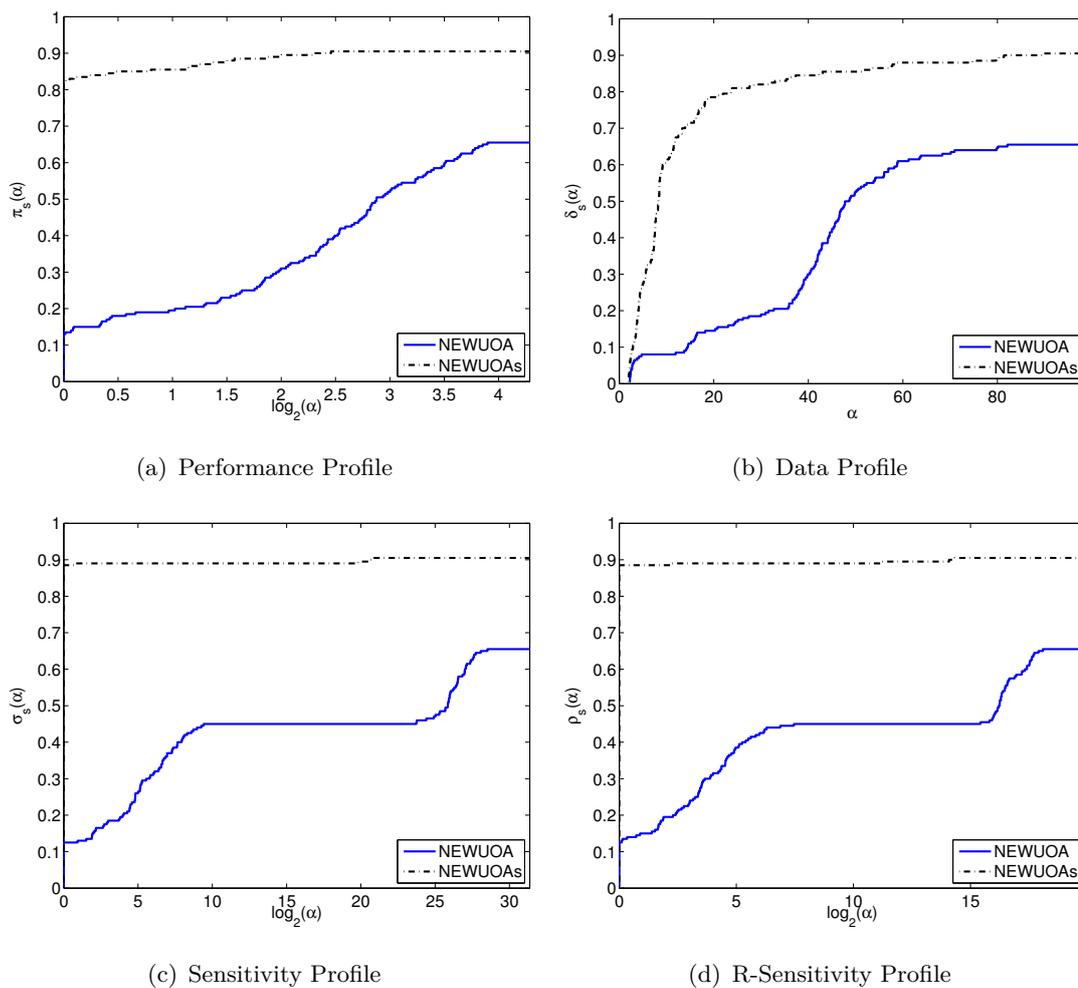


图 5.16: NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (精度 $\tau = 10^{-6}$)

图 5.17: NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (精度 $\tau = 10^{-8}$)

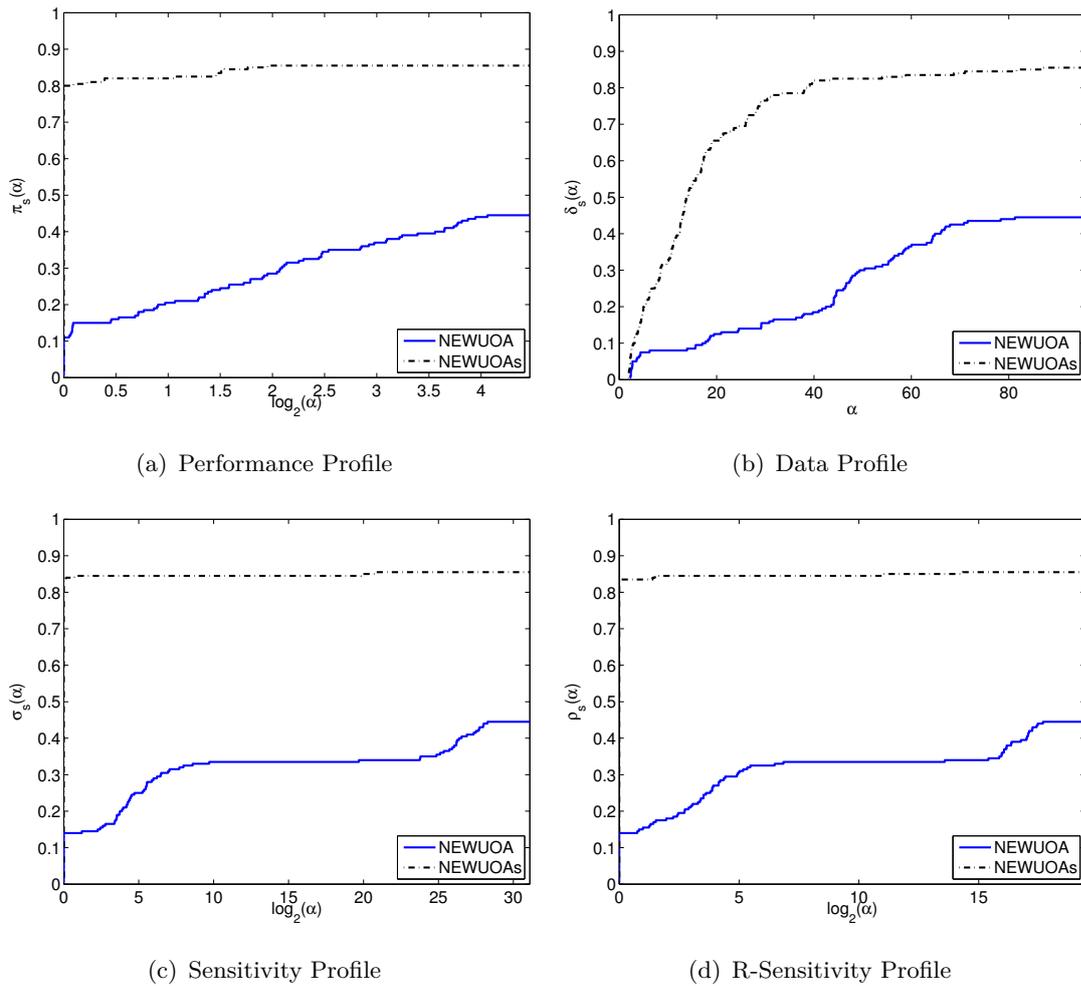
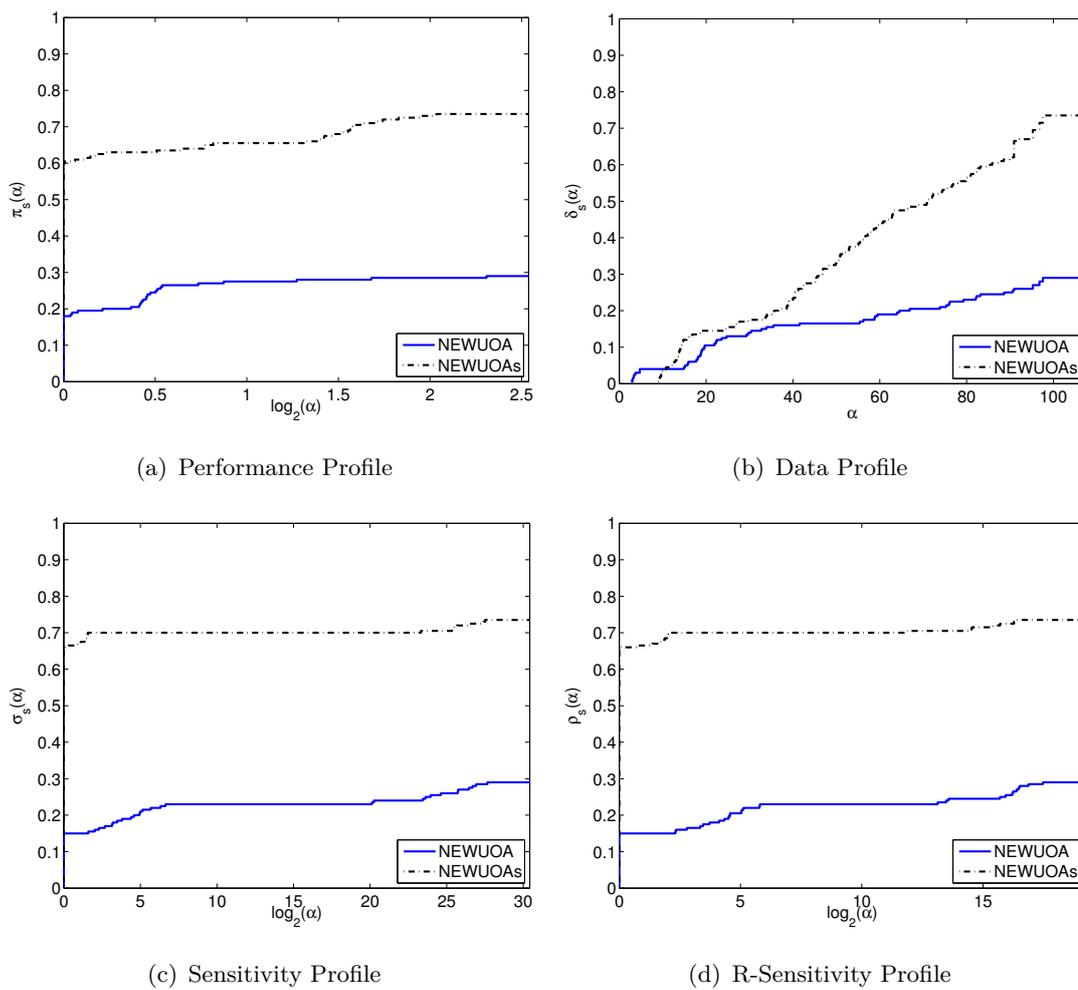


图 5.18: NEWUOA 与 NEWUOAs 对坏初始点问题的数值表现 (精度 $\tau = 10^{-10}$)

图 5.19: NEWUOA 与 NEWUOA_s 对坏初始点问题的数值表现 (迭代自然终止)

§5.3.7.3 NEWUOAs 算法求解大规模问题

NEWUOAs 算法能够求解规模很大的问题. 这里我们以 2000 维为例, 给出 NEWUOAs 算法求解若干测试问题的计算结果. 这些问题取自 §3.3.2 列出的 50 个测试问题. 与以往不同, 对每一个问题我们仅按原始变量顺序求解一次, 不进行随机置换. 实验中, NEWUOAs 的参数设置与 §5.3.7.1 相同. 我们限制最大函数值计算次数为 50000. 也就是说, 仅分配给 NEWUOAs 算法不到 25 个单纯形梯度的预算. 这是一个很低的预算, 因为, 对于 2000 维的问题, 即使是用基于导数的算法, 25 个梯度的预算也是比较低的.

我们的实验平台是 Thinkpad T400 (2765-MU4), 系统是 Ubuntu 9.04 (Linux 2.6.33-020633-generic), 编译器是 GCC 4.3.3 中的 gfortran.

表 5.3 列出了这部分问题迭代自然终止时的计算结果. 其中, f_{start} 为初始函数值, f_{best} 为 NEWUOAs 得到的最优函数值, $\#f$ 为函数值计算次数, CPU 为求解花费的 CPU 时间, 单位为秒.

对于这部分问题, NEWUOAs 都用很少的函数值计算次数求到了精度很高的解¹², 所用 CPU 时间也都很短. 这一结果是令人振奋的, 因为对于目前大部分的无导数算法, 2000 维的问题几乎是不可求解的.

因为 NEWUOA 算法几乎不能求解这一规模的问题, 我们无法给出 NEWUOA 算法的计算结果. 为了对比, 我们在相同环境下用 NEWUOA 求解了 500 维的 ARWHEAD 问题, 所用的函数值计算次数是 19374 次, CPU 时间是 2507.66 秒, 得到的最优函数值为 $4.880701\text{E}-09$. 由于 NEWUOA 的函数值计算次数和迭代次数大致与问题维数成正比, 每步计算量大致与维数平方成正比, 所以可以估算 NEWUOA 算法求解 2000 维 ARWHEAD 问题大约需要 70000–80000 次函数值计算和 40–50 个小时的 CPU 时间. 而 NEWUOAs 算法求解 2000 维 ARWHEAD 问题 (表 5.3 中第 4 行) 的代价是 16095 次函数值计算, 6.42 秒的 CPU 时间, 且得到的最优值达到了非常高的精度 (机器显示为 0.0000000000000000). 这一例子又一次显示了 NEWUOAs 算法的威力.

¹²ARGLINA 问题精确最优值为 2000, ARGLINB 问题精确最优值为 999.625047, ARGLINC 问题精确最优值为 1001.12505, DIXMAAN* 问题精确最优值为 1, 其余问题精确最优值为 0.

表 5.3: NEWUOAs 算法求解 2000 维问题的例子

| | f_{start} | f_{best} | $\#f$ | CPU (s) |
|----------|--------------------|-------------------|-------|---------|
| ARGLINA | 1.000000E+04 | 2.000000E+03 | 20136 | 8.59 |
| ARGLINB | 8.545072E+22 | 9.996250E+02 | 16155 | 14.35 |
| ARGLINC | 8.515207E+22 | 1.001125E+03 | 16096 | 13.17 |
| ARWHEAD | 5.997000E+03 | 0.000000E+00 | 16095 | 6.42 |
| BROYDN3D | 2.011000E+03 | 4.259692E-05 | 50000 | 18.55 |
| BRYBND | 7.200000E+04 | 6.486038E-09 | 50000 | 26.09 |
| CHROSEN | 3.998000E+04 | 2.018482E-04 | 50000 | 18.66 |
| DIXMAANE | 1.471453E+04 | 1.000000E+00 | 36264 | 21.12 |
| DIXMAANF | 2.734976E+04 | 1.000000E+00 | 36384 | 31.07 |
| DIXMAANG | 5.069653E+04 | 1.000000E+00 | 36393 | 22.72 |
| DIXMAANH | 1.011255E+05 | 1.000000E+00 | 40481 | 28.38 |
| DIXMAANI | 1.333800E+04 | 1.000000E+00 | 40363 | 25.88 |
| DIXMAANJ | 2.599484E+04 | 1.000000E+00 | 44527 | 151.42 |
| DIXMAANK | 4.932000E+04 | 1.000000E+00 | 40497 | 184.36 |
| DIXMAANL | 9.970237E+04 | 1.000000E+00 | 40516 | 84.70 |
| DIXMAANM | 6.233115E+03 | 1.000000E+00 | 40375 | 30.12 |
| DIXMAANN | 1.344689E+04 | 1.000000E+00 | 40439 | 26.45 |
| DIXMAANO | 2.422412E+04 | 1.000000E+00 | 40475 | 25.28 |
| DIXMAANP | 4.750292E+04 | 1.000000E+00 | 50000 | 47.05 |
| DQRTIC | 6.376035E+15 | 1.214880E-38 | 40854 | 14.70 |
| GENHUMPS | 5.122260E+07 | 1.624799E-26 | 36467 | 23.54 |
| LIARWHD | 1.170000E+06 | 2.428807E-24 | 16208 | 6.73 |
| POWER | 2.668667E+09 | 1.423292E-11 | 20130 | 19.19 |
| SPARSQUR | 5.627812E+05 | 6.381755E-30 | 16209 | 9.87 |

§5.4 总结

到目前为止, 无导数优化方法可处理的问题规模还比较有限. 本章研究无导数的子空间方法, 试图为解决大规模无导数优化问题提供一条途径. 其中, 我们提出的 NEWUOAs 算法已经可以高效求解很多高达 2000 维的问题. 这是一个突破, 因为目前大多数无导数优化算法 (比如最优秀的无导数算法之一 NEWUOA 算法) 至多可以求解几百维的问题; 对于它们, 2000 维的问题几乎是不可解的.

我们提出了两种子空间方法.

§5.2 研究的子空间方法, 其出发点是对无导数信赖域方法中使用的插值模型的不信任. 无导数算法中, 由于可用信息的匮乏和插值点集适应性的逐步丧失, 插值得到的模型经常与目标函数相去甚远, 甚至在一个很小的信赖域上也无法模拟目标函数. 这一点与基于导数的方法差别很大. 当模型质量较低时, 过分依赖模型 (比如, 精确求解信赖域子问题) 可能会受其误导. 为了避免这一现象, 当我们认为模型可信度较低时, 我们提出将信赖域子问题限制在一个子空间上求解的策略. 我们一方面要求这一子空间很有潜力, 即目标函数很有可能会在该子空间上获得下降; 另一方面要求它维数很低 (比如 2 维或 3 维). 我们期望通过这一方式提高目标函数获得下降的可能性. 换句话说, 对可信度较低的模型, 我们只给它在一个有潜力的低维子空间上工作的机会; 再换句话说, 当我们认为模型可信度较低时, 我们转而信任子空间. 当然, 如果我们有充分的理由信任当前模型, 我们仍在全空间上求解信赖域子问题.

以上策略可以视为经典的 Hooke-Jeeves 模式搜索思想的推广. Hooke-Jeeves 模式搜索中, 算法通过当前的信息, 推测函数很有可能在何方向上取得下降, 然后集中精力在这一方向上求解. 我们的策略把方向推广成了子空间.

我们把上述策略应用于 NEWUOA 算法, 得到了一种无导数的子空间算法 SUBSPACE. §5.2.2 的数值结果表明, 我们提出的策略是成功的. 我们的子空间策略尽管是一种“偷懒”的策略, 但在平均上却减少了函数值计算次数, 这一点十分值得注意.

为了更有效地解决大规模问题, 我们在 §5.3 提出了另外两种子空间算法 — NEWUOAs 算法. 这种方法的基本想法很朴素, 就是把困难问题转化为一系列容易问题来求解, 把大规模无导数优化问题分解为一系列小规模无导数优化问题来求解. 我们首先给出了一个十分广泛的子空间算法框架, 并建立了该算

法框架的全局收敛性和 R-线性收敛速度. 然后我们借助 NEWUOA 算法不依赖导数地实现了这一算法, 得到了 NEWUOAs 算法. 我们证明了 NEWUOAs 算法在理论上的全局收敛性和 R-线性收敛速度, 也证明了它在计算上的有限终止性. NEWUOAs 算法的一个优点是, 只要算法正常终止, 我们就能得到一个质量有保证的解, 并且我们给出了解的质量的估计. 我们还提出了一种预条件技术, 这种技术显著改善了 NEWUOAs 算法对坏条件问题的数值表现.

§5.3.7 的数值结果表明, NEWUOAs 算法是十分成功的. 与目前最好的无导数算法 NEWUOA 算法相比, NEWUOAs 算法明显减少了函数值计算次数, 同时提高了求解过程对计算机舍入误差的稳定性. 我们还发现, NEWUOAs 算法很善于处理初始点质量很坏的问题. 这对于实际应用领域的意义是显而易见的, 因为很多实际问题很难给出一个好的初始点. 使用 NEWUOAs 算法, 我们已经可以高效的求解很多高达 2000 维的问题, 这一规模已经超出了目前大部分无导数算法的求解范围.

值得一提的是, 由于假定函数值计算是无导数优化的主要计算开销, 故我们并没有强调 NEWUOAs 算法与 NEWUOA 算法在 CPU 时间上的差异. 实际上 NEWUOAs 算法在这一方面的优势是很大的. 由于 NEWUOAs 算法大部分时间只在一个低维子空间上工作, 它的代数计算量比 NEWUOA 算法少得多. 因此, 当函数值计算与代数运算难度相当时, NEWUOAs 算法的 CPU 时间比 NEWUOAs 少得多. §5.3.7.3 给出的例子充分说明了这一点.

尽管 NEWUOAs 算法也使用插值模型, 用于得到近似梯度和预条件矩阵, 但是它不会在插值点集的适定性方面遭遇困难, 这是 NEWUOAs 的又一大优点. 这是因为, 算法外层迭代中插值点集的几何形状是不变的, 因此它的适定性不会随着迭代而变坏; 而在内层迭代中, 由于问题维数很低, 插值点集的适定性不是很大的问题.

当 NEWUOAs 算法中参数 $m_k \equiv 2n + 1$ 时, 它所用的最小 Frobenius 范数插值模型等价于中心差分得到的模型. 此时 NEWUOAs 可视为一种差分和插值杂交的算法: 外层迭代用差分, 内层迭代用插值. 值得注意的是, 由于外层迭代次数很少 (一般在 20 步以内, 很多问题只需几步), 所以差分出现的次数很少.

预条件技术的使用大大改善了 NEWUOAs 算法的表现. 据本文作者所知, 其他无导数算法中还没有引入预条件技术的先例. 如何在一般的无导数方法中使用预条件是一个值得研究的问题. 在基于模型的方法中, 模型提供的近似

Hessian 矩阵应当作为预条件的重要依据.

NEWUOAs 算法仍然有很大的改进余地. 比如, 在子程序 MODEL 中, 插值点的选取 (5.48) 依赖于坐标方向. 一种可以考虑的改进是, 每次外迭代结束后, 选取一组新的正交基 $\{e_i\}$ 用于插值点的选取; 新正交基的取法可以借鉴 Rosenbrock 转轴法 [150] 及其改进 [160, 161, 119, 122]. 另外, 每次外迭代结束后, NEWUOAs 几乎完全抛弃了前面迭代中积累的信息. 倘若能把这些信息加以利用, 可能会给算法带来改进. 比如, 在第 k 步外迭代结束后, 目标函数在超平面 $x_{k+1} + \mathcal{S}_k$ 上已经近似达到极小, 因而

$$P_k \nabla f(x_{k+1}) \approx 0, \quad (5.97)$$

其中 P_k 为 \mathbb{R}^n 到 \mathcal{S}_k 的正交投影. 式 (5.97) 对构造 x_{k+1} 处的近似梯度是一个参考.

最后, 我们指出, NEWUOAs 算法仅仅是算法 5.3 的一种实现. 使用其他方法实现算法 5.3 中的步 2 和步 3, 我们还可以得到其他的子空间算法.

第六章 总结与展望

大多数优化方法都依赖问题的导数信息. 但是, 在实际应用中, 大量问题的导数信息都是不可用的. 这就要求我们研究不使用导数信息的方法, 这就是本文研究的无导数优化算法.

算法的评价和比较是算法的研究过程中的一个重要问题. 当没有严格的理论作为算法的支撑时, 我们尤其需要客观的评价体系来比较不同的算法, 以指导应用领域的算法选择和研究领域的算法改进. 无导数优化算法恰恰就是这样一类理论相对缺乏的方法. 因此, 在第二章里, 我们研究了如何客观可信地评价和比较不同的无导数优化算法. 我们用一个例子清楚地说明, 传统的评价方法对于无导数算法是不可靠的. 通过统计方法的引入, 我们建立了评价无导数方法的一套新体系. 与传统的评价方法相比, 新方法的优点是: 第一, 它能够反映算法对计算机舍入误差的稳定性; 第二, 它能更可靠的度量算法的计算开销. 一个值得考虑的问题是, 如何将这套体系与现有的测试环境比如 CUTEr 进行有机的对接, 形成一个针对无导数算法的测试环境. 我们在这方面已经做了一部分尝试.

本文的第三章到第五章针对无约束的无导数优化问题研究了基于信赖域的无导数算法.

第三章研究了无导数优化算法中表现十分出色的两种模型, 即最小 Frobenius 范数模型和通过对称 Broyden 修正建立的模型. 我们第一次指出了这两种模型在一些情况下的等价性. 与这两种模型紧密相关的一个问题是 NEWUOA 算法的重开始技术. 通过修改 NEWUOA 源代码中的重开始条件, 我们给出了一个改进版本的 NEWUOA 代码. 新版代码仅仅删除了原始代码中的四个字母, 就显著降低了代码的计算开销, 并且明显提高了代码对计算机舍入误差的稳定性. 我们系统地比较了两种模型在 NEWUOA 算法框架下的表现, 并且指出了一个十分值得注意的事实: 即使是在 NEWUOA 的算法框架下, NEWUOA 使用的对称 Broyden 修正也并不总占优势; 当求解精度不太高时, 最小 Frobenius 范数模型表现更好. 这一事实对于实际应用领域很有意义, 因为实际的无导数优化问题对解的精度要求往往不高. 一个很有意思的问题是, 如何将这两种出色的模型杂交以得到更好的算法. 我们将在以后的研究中考虑这一问题.

在第四章, 我们率先将 PDE 理论中的 Sobolev 范数和半范数引入无导数算法的研究中. 我们用二次函数的系数给出了二次函数在 l_p 球上的 H^0 范数和 H^1 半范数的显式表达式. 使用 H^1 半范数, 我们研究了无导数优化中有广泛应用的最小范数插值. 我们证明, 最小范数插值实际上是在一个 l_2 球上极小化插值函数的 H^1 半范数. 这一观察为理解最小范数插值提供了有力的工具. 通过这一观察, 我们首次指出了最小范数插值中两个参量的几何意义. 我们将这些理论用于研究扩展的对称 Broyden 修正, 得到了简单并且有效的参数选取方式. 当 $p \neq 2$ 时, 二次函数在 l_p 球上的 H^0 范数和 H^1 半范数可用于研究使用非欧球做信赖域的方法. 比如, 在考虑界约束问题时, 我们可以使用 l_∞ 信赖域, 此时二次函数在 l_∞ 球上的 H^0 范数和 H^1 半范数可以作为量化模型性态的指标. 这是我们进一步要研究的问题之一. 与本章相关的一个理论问题是, 扩展的对称 Broyden 修正与原始的对称 Broyden 修正哪一个更好. 这一问题目前还很难作出回答, 我们将在以后的研究中继续予以关注.

第五章研究了子空间技巧在无导数优化中的应用. 我们的目的是利用子空间的方法求解大规模的无导数优化问题. 我们提出了两种无导数的子空间算法. 在第一种子空间方法中, 利用 Hooke-Jeeves 模式搜索的思想, 针对无导数信赖域方法, 我们提出在子空间上求解信赖域子问题的技巧. 这种子空间技术改善了 NEWUOA 算法的数值表现. 第二种子空间方法, 即 NEWUOAs 算法, 是本文最成功的算法. 我们首先研究了一个很一般的子空间算法框架, 然后借助 NEWUOA 算法实现了该算法框架, 得到了 NEWUOAs 算法. 我们证明了 NEWUOAs 算法在理论上的全局收敛性、R-线性收敛速度和计算上的有限终止性. 我们还提出了一种预条件技术, 显著改善了 NEWUOAs 算法对坏条件问题的表现. 据本文作者所知, 这是无导数算法中第一次引入预条件技术. 实验证明, NEWUOAs 算法在计算开销和稳定性上都明显优于 NEWUOA 算法, 后者是目前最好的无导数算法之一. 我们还发现, NEWUOAs 算法很善于处理初始点质量较差的问题. 这对实际应用领域很有意义, 因为很多实际问题很难给出好的初始点. 此外, NEWUOAs 算法可以十分高效地求解很多维数高达 2000 的问题. 这是令人振奋的, 因为这一规模远远超出了目前大部分无导数算法 (包括 NEWUOA 算法) 的求解范围.

鉴于 NEWUOAs 算法的出色表现, 我们下一步将会把它推广到带有约束的问题. 另外, 如果能将预条件技巧应用到更多的无导数优化方法, 那将是十分有意思和有意义的. 这将是未来关注的一个研究课题.

本文所讨论的内容只是无导数优化领域的一个分支. 无导数优化领域中还有很多有意思的问题我们没有涉及. 比如有特殊结构的问题、带约束的问题和非光滑问题等等. 这些都是我们将来要考虑的问题.

参考文献

- [1] M. A. Abramson, C. Audet, and J. E. Dennis Jr., Nonlinear programming by mesh adaptive direct searches, *SIAG/OPT Views-and-News*, 17:2–11, 2006.
- [2] R. A. Adams, *Sobolev spaces*, Pure and Applied Mathematics, Academic Press, New York, 1975.
- [3] J. S. Al-Sumait, A. K. Al-Othman, and J. K. Sykulski, Application of pattern search method to power system valve-point economic load dispatch, *International Journal of Electrical Power and Energy Systems*, 29(10):720–730, 2007.
- [4] P. Alberto, F. Nogueira, H. Rocha, and L. N. Vicente, Pattern search methods for user-provided points: Application to molecular geometry problems, *SIAM Journal on Optimization*, 14(4):1216–1236, 2004.
- [5] E. Artin, *The Gamma Function*, Athena Series: Selected Topics in Mathematics, Holt, Rinehart and Winston, New York, 1964.
- [6] C. Audet and J. E. Dennis Jr., Pattern search algorithms for mixed variable programming, *SIAM Journal on Optimization*, 11(3):573–594, 2001.
- [7] C. Audet and J. E. Dennis Jr., Analysis of generalized pattern searches, *SIAM Journal on Optimization*, 13(3):889–903, 2003.
- [8] C. Audet and J. E. Dennis Jr., Mesh adaptive direct search algorithms for constrained optimization, *SIAM Journal on Optimization*, 17(1):188–217, 2006.
- [9] C. Audet and D. Orban, Finding optimal algorithmic parameters using derivative-free optimization, *SIAM Journal on Optimization*, 17(3):642–664, 2006.

- [10] A. Bandeira, K. Scheinberg, and L. N. Vicente, *Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization*, Master's thesis, Department of Mathematics, University of Coimbra, Coimbra, Portugal, 2010.
- [11] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific Optimization and Computation Series, Athena Scientific, 2003, ISBN 9781886529458.
- [12] S. C. Billups, J. Larson, and P. Graf, Derivative-free optimization of expensive functions with computational error using weighted regression, 2011, URL http://www.optimization-online.org/DB_HTML/2011/02/2924.html.
- [13] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. J. Torczon, and M. W. Trosset, A rigorous framework for optimization of expensive functions by surrogates, *Structural and Multidisciplinary Optimization*, 17(1):1–13, 1999.
- [14] A. J. Booker, J. E. Dennis Jr., P. Frank, D. B. Serafini, V. J. Torczon, and M. W. Trosset, Optimization using surrogate objectives on a helicopter test example, in: J. T. Borggaard, E. Cliff J. Burns, and S. Schreck, eds., *Computational Methods in Optimal Design and Control*, volume 24 of *Progress in Systems and Control Theory*, Birkhäuser, Cambridge, 1998.
- [15] A. J. Booker, P.D. Frank, J. E. Dennis Jr., D. W. Moore, and D. B. Serafini, Managing surrogate objectives to optimize helicopter rotor design—further experiments, in: *Proceedings of the Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, American Institute of Aeronautics and Astronautics, St. Louis, Missouri, USA, 1998.
- [16] D. Bortz and C. T. Kelley, The simplex gradient and noisy optimization problems, in: J. T. Borggaard, E. Cliff J. Burns, and S. Schreck, eds., *Computational Methods in Optimal Design and Control*, volume 24 of *Progress in Systems and Control Theory*, pp. 77–90, Birkhäuser, Boston, 1998.

-
- [17] G. E. P. Box, Evolutionary operation: A method for increasing industrial productivity, *Applied Statistics*, 6(2):81–101, 1957.
- [18] M. J. Box, A new method of constrained optimization and a comparison with other methods, *The Computer Journal*, 8(1):42–52, 1965.
- [19] R. P. Brent, *Algorithms for Minimization Without Derivatives*, Prentice-Hall Series in Automatic Computation, Prentice-Hall, Englewood Cliffs, New Jersey, 1973, ISBN 9780130223357.
- [20] K.M. Brown and J. E. Dennis Jr., Derivative free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximation, *Numerische Mathematik*, 18(4):289–297, 1971.
- [21] C. G. Broyden, Quasi-Newton methods and their application to function minimization, *Mathematics of Computation*, 21(99):368–381, 1967.
- [22] C. G. Broyden, J. E. Dennis Jr., and J. J. Moré, On the local and superlinear convergence of quasi-Newton methods, *IMA Journal of Applied Mathematics*, 12(3):223–245, 1973.
- [23] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*, volume 12 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press, Cambridge, England, 2003, ISBN 9780521633383.
- [24] R. H. Byrd, J. Nocedal, and Y. X. Yuan, Global convergence of a class of quasi-Newton methods on convex problems, *SIAM Journal on Numerical Analysis*, 24(5):1171–1190, 1987.
- [25] E. J. Candès and T. Tao, Decoding by linear programming, *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.
- [26] E. J. Candès and M. B. Wakin, An introduction to compressive sampling, *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

- [27] I. S. Chan, A. A. Goldstein, and J. B. Bassingthwaite, Sensop: a derivative-free solver for nonlinear least squares with sensitivity scaling, *Annals of Biomedical Engineering*, 21(6):621–631, 1993.
- [28] T. D. Choi and C. T. Kelley, Superlinear convergence and implicit filtering, *SIAM Journal on Optimization*, 10(4):1149–1162, 2000.
- [29] A. Cochocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, John Wiley & Sons, Inc., New York, NY, USA, 1993, ISBN 9780471930105.
- [30] B. Colson and Ph. L. Toint, Exploiting band structure in unconstrained optimization without derivatives, *Optimization and Engineering*, 2(4):399–412, 2001.
- [31] B. Colson and Ph. L. Toint, A derivative-free algorithm for sparse unconstrained optimization problems, *in*: A. H. Siddiqi and M. Kočvara, eds., *Trends in Industrial and Applied Mathematics: Proceedings of the First International Conference on Industrial and Applied Mathematics of the Indian Subcontinent*, volume 72 of *Applied Optimization*, pp. 131–148, Kluwer Academic, 2002.
- [32] B. Colson and Ph. L. Toint, Optimizing partially separable functions without derivatives, *Optimization methods and software*, 20(4-5):493–508, 2005.
- [33] A. R. Conn, N. Gould, A. Sartenaerf, and Ph. L. Toint, On iterated-subspace minimization methods for nonlinear optimization, *in*: L. Adams and L. Nazareth, eds., *Linear and nonlinear conjugate gradient-related methods*, number 85 in *Proceedings in Applied Mathematics Series*, pp. 50–78, Society for Industrial and Applied Mathematics, 1996, proceedings of the AMS-IMS-SIAM Summer Research Conference held at the University of Washington, July 1995.
- [34] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *LANCELOT: a Fortran Package for Large-scale Nonlinear Optimization (Release A)*,

- Springer series in computational mathematics, Springer-Verlag, 1992, ISBN 9780387554709.
- [35] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, volume 1 of *MPS-SIAM series on optimization*, Society for Industrial and Applied Mathematics, 2000, ISBN 9780898714609.
- [36] A. R. Conn, K. Scheinberg, and Ph. L. Toint, On the Convergence of Derivative-Free Methods for Unconstrained Optimization, *in: M. D. Buhmann and A. Iserles, eds., Approximation Theory and Optimization: Tributes to M. J. D. Powell*, pp. 83–108, Cambridge University Press, Cambridge, UK, 1997.
- [37] A. R. Conn, K. Scheinberg, and Ph. L. Toint, Recent progress in unconstrained nonlinear optimization without derivatives, *Mathematical Programming*, 79(1):397–414, 1997.
- [38] A. R. Conn, K. Scheinberg, and Ph. L. Toint, A derivative free optimization algorithm in practice, *in: Proceedings of the Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, American Institute of Aeronautics and Astronautics, St. Louis, Missouri, USA, 1998.
- [39] A. R. Conn, K. Scheinberg, and L. N. Vicente, Geometry of interpolation sets in derivative free optimization, *Mathematical Programming*, 111(1):141–172, 2008.
- [40] A. R. Conn, K. Scheinberg, and L. N. Vicente, Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation, *IMA Journal of Numerical Analysis*, 28(4):721–748, 2008.
- [41] A. R. Conn, K. Scheinberg, and L. N. Vicente, Global convergence of general derivative-free trust-region algorithms to first-and second-order critical points, *SIAM Journal on Optimization*, 20(1):387–415, 2009.

- [42] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, volume 8 of *MPS-SIAM Series on Optimization*, Society for Industrial and Applied Mathematics/Mathematical Programming Society, Philadelphia, PA, USA, 2009, ISBN 9780898716689.
- [43] A. R. Conn and Ph. L. Toint, An Algorithm Using Quadratic Interpolation for Unconstrained Derivative Free Optimization, *in*: G. Di Pillo and F. Giannesi, eds., *Nonlinear Optimization and Applications*, pp. 27–47, Kluwer Academic/Plenum Publishers, New York, 1996.
- [44] J. B. Conway, *A Course in Functional Analysis*, volume 96 of *Graduate Texts in Mathematics*, Springer-Verlag, 1990, ISBN 9780387972459.
- [45] A. L. Custódio, J. E. Dennis Jr., and L. N. Vicente, Using simplex gradients of nonsmooth functions in direct search methods, *IMA Journal of Numerical Analysis*, 28(4):770–784, 2008.
- [46] A. L. Custódio, H. Rocha, and L. N. Vicente, Incorporating minimum Frobenius norm models in direct search, *Computational Optimization and Applications*, 46(2):265–278, 2010.
- [47] A. L. Custódio and L. N. Vicente, SID-PSM: A pattern search method guided by simplex derivatives for use in derivative-free optimization, 2005.
- [48] A. L. Custódio and L. N. Vicente, Using sampling and simplex derivatives in pattern search methods, *SIAM Journal on Optimization*, 18(2):537–555, 2007.
- [49] Y. H. Dai, *Analyses of Conjugate Gradient Methods*, Ph.D. thesis, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing, CHINA, 1997.
- [50] W. C. Davidon, Variable metric method for minimization, Technical Report ANL 5990, Argonne National Laboratory, Illinois, USA, 1959.
- [51] L. Davis, *Genetic algorithms and simulated annealing*, Research notes in artificial intelligence, Pitman, 1987, ISBN 9780273087717.

-
- [52] J. E. Dennis Jr. and J. J. Moré, Quasi-Newton methods, motivation and theory, *SIAM Review*, 19(1):46–89, 1977.
- [53] J. E. Dennis Jr. and R. B. Schnabel, Least change secant updates for quasi-Newton methods, *SIAM Review*, 21(4):443–459, 1979.
- [54] J. E. Dennis Jr. and V. J. Torczon, Direct search methods on parallel machines, *SIAM Journal on Optimization*, 1(04):448–474, 1991.
- [55] M. A. Diniz-Ehrhardt, J. M. Martínez, and M. Raydán, A derivative-free nonmonotone line-search technique for unconstrained optimization, *Journal of Computational and Applied Mathematics*, 219(2):383–397, 2008.
- [56] L. C. W. Dixon, Quasi-Newton algorithms generate identical points, *Mathematical Programming*, 2(1):383–387, 1972.
- [57] L. C. W. Dixon and D. J. Mills, Effect of rounding errors on the variable metric method, *Journal of Optimization Theory and Applications*, 80(1):175–179, 1994.
- [58] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming*, 91(2):201–213, 2002.
- [59] D. L. Donoho, Compressed sensing, *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [60] E. Esser, Applications of Lagrangian-based alternating direction methods and connections to split Bregman, Technical Report CAM Report TR09-31, UCLA, 2009.
- [61] L. C. Evans, *Partial Differential Equations*, Graduate Studies in Mathematics, American Mathematical Society, 1998, ISBN 9780821807729.
- [62] G. Fasano, J. L. Morales, and J. Nocedal, On the geometry phase in model-based algorithms for derivative-free optimization, *Optimization Methods and Software*, 24(1):145–154, 2009.

-
- [63] E. Fermi and N. Metropolis, Los Alamos Unclassified Report LS-492, Technical report, Alamos National Laboratory, Los Alamos, USA, 1952.
- [64] R. Fletcher and M. J. D. Powell, A rapidly convergent descent method for minimization, *The Computer Journal*, 6(2):163–168, 1963.
- [65] P. E. Gill and W. Murray, Quasi-Newton methods for unconstrained optimization, *IMA Journal of Applied Mathematics*, 9(1):91–108, 1972.
- [66] P. Gilmore and C. T. Kelley, An implicit filtering algorithm for optimization of functions with many local minima, *SIAM Journal on Optimization*, 5(2):269–285, 1995.
- [67] N. I. M. Gould, D. Orban, and Ph. L. Toint, CUTeR and SifDec: A constrained and unconstrained testing environment, revisited, *ACM Transactions on Mathematical Software (TOMS)*, 29(4):373–394, 2003.
- [68] J. Greenstadt, A quasi-Newton method with no derivatives, *Mathematics of Computations*, 26(117):145–166, 1972.
- [69] J. Greenstadt, Revision of a derivative-free quasi-Newton method, *Mathematics of Computation*, 32(141):201–221, 1978.
- [70] A. Griewank and Ph. L. Toint, Local convergence analysis for partitioned quasi-Newton updates, *Numerische Mathematik*, 39(3):429–448, 1982.
- [71] A. Griewank and Ph. L. Toint, On the unconstrained optimization of partially separable objective functions, in: M. J. D. Powell, ed., *Nonlinear Optimization 1981*, pp. 301–312, Academic Press, London, England, 1982.
- [72] A. Griewank and Ph. L. Toint, Partitioned variable metric updates for large structured optimization problems, *Numerische Mathematik*, 39(1):119–137, 1982.
- [73] W. W. Hager and H. C. Zhang, A survey of nonlinear conjugate gradient methods, *Pacific Journal of Optimization*, 2(1):35–58, 2006.

-
- [74] L. Han and M. Neumann, Effect of dimensionality on the Nelder-Mead simplex method, *Optimization Methods and Software*, 21(1):1–16, 2006.
- [75] B. S. He and X. M. Yuan, On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method, *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.
- [76] N. J. Higham, *The Matrix Computation Toolbox*, URL www.ma.man.ac.uk/~higham/mctoolbox.
- [77] N. J. Higham, Optimization by direct search in matrix computations, *SIAM Journal on Matrix Analysis and Applications*, 14(2):317–333, 1993.
- [78] W. Hock and K. Schittkowski, Test examples for nonlinear programming codes, *Journal of Optimization Theory and Applications*, 30(1):127–129, 1980.
- [79] R. Hooke and T. A. Jeeves, “Direct search” solution of numerical and statistical problems, *Journal of the Association for Computing Machinery*, 8(2):212–229, 1961.
- [80] P. D. Hough, T. G. Kolda, and V. J. Torczon, Asynchronous parallel pattern search for nonlinear optimization, *SIAM Journal on Scientific Computing*, 23(1):134–156, 2002.
- [81] S. Karimi and S. Vavasis, Conjugate gradient with subspace optimization, *Arxiv preprint arXiv:1202.1479*, 2012.
- [82] W. Karush, *Minima of Functions of Several Variables with Inequalities as Side Constraints*, Master’s thesis, Department of Mathematics, University of Chicago, Chicago, Illinois, USA, 1939.
- [83] C. T. Kelley, Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition, *SIAM Journal on Optimization*, 10(1):43–55, 1999.

-
- [84] C. T. Kelley, *Iterative Methods for Optimization*, volume 18 of *Frontiers in applied mathematics*, Society for Industrial Mathematics, 1999, ISBN 9780898714333.
- [85] C. T. Kelley, A brief introduction to implicit filtering, Technical Report CRSC-TR02-28, North Carolina State Univ., Raleigh, NC, 2002.
- [86] C. T. Kelley, *Implicit filtering*, volume 23 of *Software, Environments and Tools*, Society for Industrial and Applied Mathematics, 2011, ISBN 9781611971897.
- [87] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, Optimization by simulated annealing, *Science*, 220(4598):671–680, 1983.
- [88] T. G. Kolda, Revisiting asynchronous parallel pattern search for nonlinear optimization, *SIAM Journal on Optimization*, 16(2):563–586, 2006.
- [89] T. G. Kolda, R. M. Lewis, and V. J. Torczon, Optimization by direct search: New perspectives on some classical and modern methods, *SIAM Review*, 45(3):385–482, 2003.
- [90] H. W. Kuhn and A. W. Tucker, Nonlinear programming, *in*: Jerzy Neyman, ed., *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492, 1950.
- [91] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, Convergence properties of the Nelder-Mead simplex method in low dimensions, *SIAM Journal on Optimization*, 9(1):112–147, 1998.
- [92] J. P. Lawrence and K. Steiglitz, Randomized pattern search, *IEEE Transactions on Computers*, 100(4):382–385, 1972.
- [93] T. Levina, Y. Levin, J. McGill, and M. Nediak, Dynamic pricing with online learning and strategic consumers: An application of the aggregating algorithm, *Operations Research*, 57(2):327–341, 2009.

-
- [94] R. M. Lewis and V. J. Torczon, Rank ordering and positive bases in pattern search algorithms, Technical Report ICASE Technical report TR 96-71, NASA Langley Research Center, Hampton, USA, 1996.
- [95] R. M. Lewis and V. J. Torczon, Pattern search algorithms for bound constrained minimization, *SIAM Journal on Optimization*, 9(4):1082–1099, 1999.
- [96] R. M. Lewis and V. J. Torczon, Pattern search methods for linearly constrained minimization, *SIAM Journal on Optimization*, 10(3):917–941, 2000.
- [97] R. M. Lewis, V. J. Torczon, and M. W. Trosset, Why pattern search works, Technical Report ICASE Report No. 98-57, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, USA, 1998.
- [98] R. M. Lewis, V. J. Torczon, and M. W. Trosset, Direct search methods: then and now, *Journal of Computational and Applied Mathematics*, 124(1):191–207, 2000.
- [99] X. Liu, J. F. Yang, and Y. Zhang, Towards a general convergence theory for alternating direction method, Technical report, Department of Computational and Applied Mathematics, RICE University, Houston, Texas, USA, 2011.
- [100] L. Lukšan, C. Matonoha, and J. Vlček, Modified CUTE problems for sparse unconstrained optimization, Technical Report No. 1081, Institute of Computer Science, Academy of Sciences of the Czech Republic, October 2010.
- [101] M. Marazzi and J. Nocedal, Wedge trust region methods for derivative free optimization, *Mathematical Programming*, 91(2):289–305, 2002.
- [102] K. I. M. McKinnon, Convergence of the Nelder-Mead simplex method to a nonstationary point, *SIAM Journal on Optimization*, 9(1):148–158, 1998.

- [103] J. C. Meza and M. L. Martinez, Direct search methods for the molecular conformation problem, *Journal of Computational Chemistry*, 15(6):627–632, 1994.
- [104] Z. Michalewicz and D. B. Fogel, *How to solve it: modern heuristics*, Springer, 2004, ISBN 9783540224945.
- [105] M. Momma and K. P. Bennett, A pattern search method for model selection of support vector regression, *in: Proceedings of the SIAM International Conference on Data Mining*, pp. 261–274, Society for Industrial and Applied Mathematics, 2002.
- [106] J. J. Moré, B. S. Garbow, and K. E. Hillstom, Testing unconstrained optimization software, *ACM Transactions on Mathematical Software (TOMS)*, 7(1):17–41, 1981.
- [107] J. J. Moré and S. M. Wild, Benchmarking derivative-free optimization algorithms, *SIAM Journal on Optimization*, 20(1):172–191, 2009.
- [108] G. Narkiss and M. Zibulevsky, Sequential subspace optimization method for large-scale unconstrained problems, Technical report, Department of Electrical Engineering Technion, Israel Institute of Technology Haifa 32000, Israel, 2005.
- [109] L. Nazareth and P. Tseng, Gilding the lily: A variant of the Nelder-Mead algorithm based on golden-section search, *Computational Optimization and Applications*, 22(1):133–144, 2002.
- [110] J. A. Nelder and R. Mead, A simplex method for function minimization, *The Computer Journal*, 7(4):308–313, 1965.
- [111] Q. Ni and Y. X. Yuan, SLMQN: A fortran code of subspace limited memory quasi-Newton method for large-scale bound constrained nonlinear optimization (in Chinese), *Journal on Numerical Methods and Computer Applications*, 18(3):173–179, 1997.

-
- [112] Q. Ni and Y. X. Yuan, A subspace limited memory quasi-Newton algorithm for large-scale nonlinear bound constrained optimization, *Mathematics of Computation*, 66(220):1509–1520, 1997.
- [113] Y. Nie and K. K. Ma, Adaptive rood pattern search for fast block-matching motion estimation, *IEEE Transactions on Image Processing*, 11(12):1442–1449, 2002.
- [114] J. Nocedal, Conjugate gradient methods and nonlinear optimization, *in*: L. Adams and L. Nazareth, eds., *Linear and nonlinear conjugate gradient-related methods*, number 85 in Proceedings in Applied Mathematics Series, pp. 9–23, Society for Industrial and Applied Mathematics, 1996, proceedings of the AMS-IMS-SIAM Summer Research Conference held at the University of Washington, July 1995.
- [115] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer series in operations research, Springer, 1999, ISBN 9780387987934.
- [116] R. Oeuvray, *Trust-region methods based on radial basis functions with application to biomedical imaging*, Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2005.
- [117] R. Oeuvray and M. Bierlaire, A new derivative-free algorithm for the medical image registration problem, *International Journal of Modelling and Simulation*, 27(2):115–124, 2007.
- [118] R. Oeuvray and M. Bierlaire, BOOSTERS: A derivative-free algorithm based on radial basis functions, *International Journal of Modelling and Simulation*, 29(1), 2009.
- [119] J. R. Palmer, An improved procedure for orthogonalising the search vectors in Rosenbrock’s and Swann’s direct search optimisation methods, *The Computer Journal*, 12(1):69–71, 1969.

- [120] M. J. D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives, *The Computer Journal*, 7(2):155–162, 1964.
- [121] M. J. D. Powell, A method for minimizing a sum of squares of non-linear functions without calculating derivatives, *The Computer Journal*, 7(4):303–307, 1965.
- [122] M. J. D. Powell, On the calculation of orthogonal vectors, *The Computer Journal*, 11(3):302–304, 1968.
- [123] M. J. D. Powell, A new algorithm for unconstrained optimization, *in*: J. B. Rosen, O. L. Mangasarian, and Ritter K, eds., *Nonlinear Programming*, pp. 31–66, Academic Press, New York, USA, 1970.
- [124] M. J. D. Powell, Some properties of the variable metric algorithm, *in*: F. A. Lootsma, ed., *Numerical Methods for Nonlinear Optimization*, pp. 1–17, London: Academic Press, 1972.
- [125] M. J. D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, *in*: R. W. Cottle and C. E. Lemke, eds., *Nonlinear programming*, volume 9, pp. 53–72, American Mathematical Society, 1976.
- [126] M. J. D. Powell, Variable metric methods for constrained optimization, *in*: R. Glowinski, J. Lions, and Iria Laboria, eds., *Computing Methods in Applied Sciences and Engineering, 1977, I*, volume 704 of *Lecture Notes in Mathematics*, pp. 62–72, Springer Berlin / Heidelberg, 1979.
- [127] M. J. D. Powell, Nonconvex minimization calculations and the conjugate gradient method, *in*: David Griffiths, ed., *Numerical Analysis*, volume 1066 of *Lecture Notes in Mathematics*, pp. 122–141, Springer Berlin / Heidelberg, 1984.
- [128] M. J. D. Powell, A direct search optimization method that models the objective and constraint functions by linear interpolation, *in*: *Advances in*

- Optimization and Numerical Analysis, Proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca Mexico*, volume 275, pp. 51–67, Kluwer Academic Publishers, 1994.
- [129] M. J. D. Powell, Direct search algorithms for optimization calculations, *Acta Numerica*, 7(1):287–336, 1998.
- [130] M. J. D. Powell, Recent research at Cambridge on radial basis functions, Technical Report DAMTP NA1998/05, CMS, University of Cambridge, 1998.
- [131] M. J. D. Powell, UOBYQA: unconstrained optimization by quadratic approximation, Technical Report DAMTP NA2000/14, CMS, University of Cambridge, 2000.
- [132] M. J. D. Powell, Radial basis function methods for interpolation to functions of many variables, Technical Report DAMTP NA2001/11, CMS, University of Cambridge, 2001.
- [133] M. J. D. Powell, Least Frobenius norm updating of quadratic models that satisfy interpolation conditions, Technical Report DAMTP NA2002/08, CMS, University of Cambridge, October 2002.
- [134] M. J. D. Powell, On trust region methods for unconstrained minimization without derivatives, Technical Report DAMTP NA2002/NA02, CMS, University of Cambridge, February 2002.
- [135] M. J. D. Powell, On the use of quadratic models in unconstrained minimization without derivatives, Technical Report DAMTP NA2003/03, CMS, University of Cambridge, March 2003.
- [136] M. J. D. Powell, On trust region methods for unconstrained minimization without derivatives, *Mathematical programming*, 97(3):605–623, 2003.
- [137] M. J. D. Powell, Least Frobenius norm updating of quadratic models that satisfy interpolation conditions, *Mathematical Programming*, 100:183–215, 2004.

-
- [138] M. J. D. Powell, The NEWUOA software for unconstrained optimization without derivatives, Technical Report DAMTP NA2004/08, CMS, University of Cambridge, November 2004.
- [139] M. J. D. Powell, On the use of quadratic models in unconstrained minimization without derivatives, *Optimization Methods and Software*, 19(3-4):399–411, 2004.
- [140] M. J. D. Powell, On updating the inverse of a KKT matrix, Technical Report DAMTP NA2004/01, CMS, University of Cambridge, January 2004.
- [141] M. J. D. Powell, *Five lectures on radial basis functions*, IMM, Informatik og Matematisk Modelling, DTU, 2005.
- [142] M. J. D. Powell, Developments of NEWUOA for minimization without derivatives, *IMA Journal of Numerical Analysis*, 28(4):649–664, 2008.
- [143] M. J. D. Powell, The BOBYQA algorithm for bound constrained optimization without derivatives, Technical Report DAMTP 2009/NA06, CMS, University of Cambridge, August 2009.
- [144] M. J. D. Powell, Beyond symmetric Broyden for updating quadratic models in minimization without derivatives, Technical Report DAMTP 2010/NA04, CMS, University of Cambridge, March 2010.
- [145] M. J. D. Powell, On the convergence of trust region algorithms for unconstrained minimization without derivatives, Technical Report DAMTP NA2011/01, CMS, University of Cambridge, January 2011.
- [146] M. J. D. Powell, Beyond symmetric Broyden for updating quadratic models in minimization without derivatives, *Mathematical Programming*, pp. 1–26, 2012.
- [147] C. J. Price, I. D. Coope, and D. Byatt, A convergent variant of the Nelder-Mead algorithm, *Journal of Optimization Theory and Applications*, 113(1):5–19, 2002.

- [148] C. J. Price and Ph. L. Toint, Exploiting problem structure in pattern search methods for unconstrained optimization, *Optimisation Methods and Software*, 21(3):479–491, 2006.
- [149] M. L. Ralston and R. I. Jennrich, Dud, a derivative-free algorithm for nonlinear least squares, *Technometrics*, 20(1):7–14, 1978.
- [150] H. H. Rosenbrock, An automatic method for finding the greatest or least value of a function, *The Computer Journal*, 3(3):175–184, 1960.
- [151] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill Book Co., New York, 3rd edition, 1976, ISBN 0070856133, International Series in Pure and Applied Mathematics.
- [152] T. Sauer and Y. Xu, On multivariate Lagrange interpolation, *Mathematics of Computation*, 64(211):1147–1170, 1995.
- [153] K. Scheinberg, Geometry in model-based algorithms for derivative-free unconstrained optimization, *Mathematical Programming Society Newsletter*, 79:1–5, 2009.
- [154] K. Scheinberg and Ph. L. Toint, Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization, *SIAM Journal on Optimization*, 20(6):3512, 2010.
- [155] K. Schittkowski, *More test examples for nonlinear programming codes*, number 282 in Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, 1987, ISBN 9783540171829.
- [156] K. Schittkowski, Examples for nonlinear programming codes — all problems from the Hock-Schittkowski-Collection, Technical report, Institut für Informatik, Universität Bayreuth, Deutschland, 2009.
- [157] C. S. Smith, The automatic computation of maximum likelihood estimates, Technical Report Report 864/MR/40, NCB Sci. Dept., 1960.

-
- [158] W. Spendley, G. R. Hext, and F. R. Himsworth, Sequential application of simplex designs in optimisation and evolutionary operation, *Technometrics*, 4(4):441–461, 1962.
- [159] G. W. Stewart III, A modification of Davidon’s minimization method to accept difference approximations of derivatives, *Journal of the Association for Computing Machinery*, 14(1):72–83, 1967.
- [160] W. H. Swann, Report on the development of a new direct search method of optimization, Technical Report Research Note 64/3, ICI Ltd. Cent. Inst. Lab., 1964.
- [161] W. H. Swann, Direct search methods, *in*: W. Murray, ed., *Numerical Methods for Unconstrained Optimization*, pp. 13–28, Academic Press, London, England, 1972.
- [162] Ph. L. Toint, Some numerical results using a sparse matrix updating formula in unconstrained optimization, *Mathematics of Computation*, 32(143):839–851, 1978.
- [163] Ph. L. Toint, Using problem structure in derivative-free optimization, *SIAG/OPT Views-and-News*, 79:11–18, 2006.
- [164] V. J. Torczon, *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*, Ph.D. thesis, Rice University, Houston, TX, USA, 1989.
- [165] V. J. Torczon, On the convergence of the multidirectional search algorithm, *SIAM Journal on Optimization*, 1(1):123–145, 1991.
- [166] V. J. Torczon, On the convergence of pattern search algorithms, *SIAM Journal on optimization*, 7(1):1–25, 1997.
- [167] M. W. Trosset, I know it when I see it: toward a definition of direct search methods, *SIAG/OPT Views and News*, 9:7–10, 1997.

- [168] F. Vanden Berghen and H. Bersini, CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm, *Journal of Computational and Applied Mathematics*, 181(1):157–175, 2005.
- [169] Z. H. Wang, Z. W. Wen, and Y. X. Yuan, A subspace trust region method for large scale unconstrained optimization, *in*: Y. X. Yuan, ed., *Numerical linear algebra and optimization*, pp. 265–274, Since Press, 2004.
- [170] Z. H. Wang and Y. X. Yuan, A subspace implementation of quasi-Newton trust region methods for unconstrained optimization, *Numerische Mathematik*, 104(2):241–269, 2006.
- [171] Z. Wen, D. Goldfarb, and W. Yin, Alternating direction augmented Lagrangian methods for semidefinite programming, *Mathematical Programming Computation*, 2(3):203–230, 2010.
- [172] S. M. Wild, MNH: a derivative-free optimization algorithm using minimal norm Hessians, Technical Report ORIE-1466, Cornell University, School of Operations Research and Information Engineering, Ithaca, NY, USA, 2008.
- [173] S. M. Wild, R. G. Regis, and C. A. Shoemaker, ORBIT: Optimization by radial basis function interpolation in trust-regions, *SIAM Journal on Scientific Computing*, 30(6):3197–3219, 2008.
- [174] D. H. Winfield, *Function and Functional Optimization by Interpolation in Data Tables*, Ph.D. thesis, Harvard University, Cambridge, MA, USA, 1969.
- [175] D. H. Winfield, Function minimization by interpolation in a data table, *IMA Journal of Applied Mathematics*, 12(3):339–347, 1973.
- [176] M. H. Wright, Direct search methods: Once scorned, now respectable, *in*: D. F. Griffiths and G. A. Watson, eds., *Numerical Analysis 1995*, volume 344 of *Pitman Research Notes in Mathematics Series*, pp. 191–208, Longman Scientific & Technical, 1996.

- [177] T. Wu, Y. Yang, L. Sun, and H. Shao, A heuristic iterated-subspace minimization method with pattern search for unconstrained optimization, *Computers and Mathematics with Applications*, 58(10):2051–2059, 2009.
- [178] J. Yang, Y. Zhang, and W. Yin, A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data, *IEEE Journal of Selected Topics in Signal Processing*, 4(2):288–297, 2010.
- [179] Y. X. Yuan, A review of trust region algorithms for optimization, *in*: J. M. Ball and J. C. R. Hunt, eds., *ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics*, pp. 271–282, Oxford University Press, Oxford, 2000.
- [180] Y. X. Yuan, Subspace techniques for nonlinear optimization, *in*: *Some Topics in Industrial and Applied Mathematics*, volume 8 of *Series on Concrete and Applicable Mathematics*, pp. 206–218, Higher Education Press, Beijing, 2007.
- [181] Y. X. Yuan, Subspace methods for large scale nonlinear equations and nonlinear least squares, *Optimization and Engineering*, 10(2):207–218, 2009.
- [182] Y. X. Yuan, Recent advances in numerical methods for nonlinear equations and nonlinear least squares, *Numerical Algebra Control and Optimization*, 1(1):15–34, 2011.
- [183] Y. X. Yuan and J. Stoer, A subspace study on conjugate gradient algorithms, *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 75(1):69–77, 1995.
- [184] W. I. Zangwill, Minimizing a function without calculating derivatives, *The Computer Journal*, 10(3):293–296, 1967.
- [185] H. C. Zhang and A. R. Conn, On the local convergence of a derivative-free algorithm for least-squares minimization, *Computational Optimization and Applications*, 51(2):481–507, 2012.

-
- [186] H. C. Zhang, A. R. Conn, and K. Scheinberg, A derivative-free algorithm for the least-squares minimization, *SIAM Journal on Optimization*, 20(6):3555–3576, 2010.
- [187] Zaikun Zhang, Sobolev seminorm of quadratic functions with applications to derivative-free optimization, November 2011, URL http://www.optimization-online.org/DB_HTML/2011/11/3245.html.
- [188] 丁晓东, 基于插值模型的无导数优化方法及其应用, Ph.D. thesis, 中国科学院计算数学与科学工程计算研究所, 北京, 2010.
- [189] 戴戡虹, 袁亚湘, 非线性共轭梯度法, 科学专著丛书, 上海科学技术出版社, 上海, 2000, ISBN 9787532355860.
- [190] 袁亚湘, 非线性优化计算方法, 运筹与管理科学丛书, 科学出版社, 北京, 2008, ISBN 9787030208835.

索引

- 测试问题, 37–38
- 插值
 - 欠定二次, 33
 - 完全二次, 11
 - 稀疏二次, 11
 - 最小范数, 65–66, 68–71
 - 最小 Frobenius 范数, 33
- 插值点集
 - 良定, 10
 - 适定, 10
- 方法
 - 单纯形, 5, 16
 - 模式搜索, 4–5, 14–16
 - 无导数优化, 2, 13–14
 - 基于线搜索的方法, 4, 6–8
 - 基于信赖域的方法, 4, 8–13, 96
 - 直接搜索方法, 4–5, 13–14
 - 线搜索, 2, 6
 - 信赖域, 2, 8–9
 - 直接, 2, 13–14
 - 子空间, 95
 - 坐标轮换, 6
- 函数
 - 目标, 1
 - 约束, 1
- 几何步, 12
- 模型步, 13
- Profile
 - Data , 21
 - Performance, 19–21
 - R-Sensitivity, 28
 - Sensitivity, 28
- Sobolev (半) 范数, 66–67
 - 二次函数的, 67–68, 83–92
- 算法
 - DFO , 10
 - 插值点集, 10, 35
 - 模型, 10, 33
 - 全局收敛性, 10
 - LFN , 45
 - NEWUOA , 11, 39
 - 参数, 38
 - 插值点集, 11, 35
 - 重新开始技术, 34, 36
 - 模型, 11, 34
 - 源代码的改进, 37–39
 - 终止条件, 38
 - NEWUOAm , 37, 39
 - NEWUOAs , 108
 - 解的质量, 123
 - 计算过程, 119–120
 - 求解大规模问题, 146
 - 求解坏初始点问题, 138–139
 - 求解较大规模问题, 129–131
 - 全局收敛性, 120–121

R-线性收敛速度, 121

实现细节, 124–125

有限终止性, 121–123

预条件技术, 125–129

UOBYQA , 10

修正

对称 Broyden, 34, 59–61

扩展的对称 Broyden, 65, 71–76

最小 Frobenius 范数, 34, 60

优化问题

非线性, 1

无导数, 2

无约束, 1

约束, 1

完成文章目录

- [1] Zaikun Zhang, Sobolev Seminorm with Applications to Derivative-Free Optimization, 2011, submitted.
- [2] Zaikun Zhang, Notes on the Sobolev (Semi)Norms of Quadratic Functions 2011, finished.
- [3] Zaikun Zhang, A Fast Derivative-Free Optimization Algorithm with Subspace Technique, 2012, finished.

简 历

张在坤，男，山东省临朐县大郝庄村人，1985年8月生。

张在坤 2003年9月至2007年7月就读于吉林大学数学学院信息与计算科学专业，获学士学位；2007年9月至2012年7月于中国计算数学与科学工程计算研究所攻读博士学位，研究方向为最优化理论与应用。

Email: zaikunzhang@gmail.com, zhangzk@lsec.cc.ac.cn。

致 谢

博士学位论文写完了，二十一年的求学生活随之进入尾声。回首往事，我心中充满感激。我有太多的人需要感谢。

首先，我要感谢我的导师、我最敬爱的袁老师。2006年保研的时候，我其实并没有想好要选择什么专业。六年前的那个下午，当我敲开袁老师的门，第一眼看到袁老师的时候，就被老师的学者风度打动了。我依然清晰地记得那天袁老师与我面对面交谈的情景，那是我第一次见到如此平易近人的老师。从那一刻起我就决定要做袁老师的学生。如今，即将从袁老师这里毕业，我为当初的选择感到幸运。感谢袁老师五年来提供的宽松的研究环境，让我可以研究自己喜欢的问题；感谢袁老师细心的指导，让我领略到优化之美，领略到数学家思考问题的方式；感谢袁老师一直以来的鼓励和提携；感谢袁老师给我出国访问的机会。我还要特别感谢袁老师在生活上对我的关心和照顾。五年里，袁老师一直给予我父亲一般的关怀。2010年夏天与袁老师在欧洲的三个月里，袁老师给我做饭，带我旅游，让我懂得了什么叫师徒如父子。感谢袁老师一次又一次对我的教诲，教给我生活的道理，分享给我人生的经验。是袁老师让我重新认识了生活。从袁老师身上学到的每一点都是我最宝贵的财富。谢谢您，袁老师！

感谢我的师母张焱女士。感谢师母一直以来在生活上的关心和照顾。师母和袁老师让我在北京感受到了家的温暖。我可能是被师母和袁老师叫到家里吃饭最多的学生。2010年夏天在欧洲的日子里，师母和袁老师像家人一样待我，让我度过了一段难忘的日子。那段时间师母做的汤是世界上最好喝的汤。师母无论多忙，都会时常关心我的生活和工作，让我心中多了一份前进的动力。

特别感谢袁老师的导师、英国皇家学会会员、美国科学院外籍院士、首届Dantzig奖获得者、剑桥大学的 M. J. D. Powell 教授给我提供的指导和帮助。Powell 教授对后辈的热情让我感动。每一次向 Powell 教授请教都让我获益匪浅。Powell 教授提供了 NEWUOA 算法的源代码和参考文献，这是我五年学习和研究中最重要素材。

感谢德国拜罗伊特大学的 Klaus Schittkowski 教授和夫人。在 2010 年夏天我访问拜罗伊特大学期间，教授和夫人给予了热情周到的接待，令我十分难

忘。在拜罗伊特的日子是这五年里我度过的最美好的时光。感谢洪堡基金资助我这次访问。本文第四章的部分内容就是在此次访问期间完成的。

感谢课题组的戴彧虹老师。戴老师在讨论班上的指点让我获益匪浅，这在论文里有直接体现。戴老师豁达的生活态度和平易近人的风度永远值得我学习。感谢课题组的刘歆老师。感谢刘老师五年以来像兄长一样的鼓励和帮助。

感谢课题组的夏勇博士、徐姿博士、牛凌峰博士、李在禾博士、付云姗博士、唐明筠博士、程明厚博士、宫鲁津博士、郝春林博士、寇彩霞博士、费存林、顾晓娟博士和李庆娜博士等师姐师兄们，感谢孙聪、张睿燕、姜波、Thanawath Niyamosoth、张文慧、刁瑞、刘田香、盛镇醴、崔春风、董乾和王树雄等师妹师弟和同学们。感谢我的师兄丁晓东博士。我接触无导数优化算法就是从听师兄的报告开始的；师兄在我研究的起步阶段提供了很多帮助，我使用的很多参考文献都是师兄提供的。感谢与我同一级的王晓、刘亚锋和吴乐秦，和他们一起度过的五年，点点滴滴都值得回忆。

Powell 教授、葡萄牙 Universidade de Coimbra 的 Luís Nunes Vicente 教授、美国 Louisiana State University 的张洪超博士 (我的师兄) 以及李庆娜师姐、孙聪师妹和姜波师弟审阅了我的第一篇论文 [187] 并提出了宝贵的意见。感谢他们对我的无私帮助。

感谢我在北京期间的室友张乔夫和张新雨、翟建梁两位师兄。我经常早出晚归，影响了他们休息，感谢他们对我的包容。

感谢与我一同入所的何莲花、梁珊、孙建青、王满、陈冲、陈耀、成杰、戴银云、黄记祖、马云飞、伍泽东、肖俊敏和张乔夫。五年与他们一起走过我很荣幸。

感谢吴继萍老师、白英老师、丁如娟老师、樊建荣老师、刘颖老师、钱莹老师、张纪平老师、魏敏老师、关华老师、王璐璐老师、邵欣老师和尹永华老师五年期间对我的帮助。

感谢我的母校、我深怀眷恋的吉林大学。人生最美好的四年能在吉大度过，我很幸运。特别感谢吉林大学数学学院的老师们。感谢谢敬然老师。谢老师并不认识我，但是是他的引导让我走入了数学的美妙世界。感谢马富明老师。马老师对学生的爱护和提携是我学习的榜样。感谢李永海老师。李老师对学生的真诚让人感动。感谢纪友清老师。纪老师讲授的泛函分析塑造了我的思维方式。感谢大学四年所有教过我的老师。从他们身上，我学到了很多很多。

特别感谢我的启蒙老师夏翠苓女士。回首漫漫求学路，我取得的每一点进步都深深地植根于您二十年前的教诲。

感谢中学和大学里与我一起笑过、哭过、奋斗过的兄弟们。有你们在，我活得很踏实。特别感谢王振宇、张明涛、石胜坤和张旭升。五年里，你们一直是我在北京最可靠的后盾。

最后，我要感谢我的家人。感谢我深爱的父亲母亲。做你们的儿子是我此生最幸运的事情；是你们教会我，诚实的劳动是获得成功的唯一方法；在我失落的时候，你们是我内心最大的慰藉；在我遭受挫折的时候，想到你们我总能找回前进的勇气。谢谢你们，我爱你们！感谢我的哥哥和嫂子，感谢他们一直以来对我的照顾。若没有哥哥当初的建议，我不会报考吉林大学数学学院，这是改变我一生的事情。

本文完成之后的第十天（壬辰年三月廿七）是母亲六十岁生日。我谨以此文作为献给母亲的生日礼物，祝母亲健康、平安。

張在坤

2012年4月7日
于北京保福寺