



南京大學

本科畢業論文

院 系 _____ 匡亞明學院

專 業 _____ 理強（統計學）

題 目 非光滑复合优化与拓展流形采样算法

年 级 2017 级 学 号 171240510

学生姓名 _____ 马宇恒

指导老师 顾国勇 职 称 副教授

提交日期 _____ 2021 年 6 月 8 日

Nonsmooth Nonconvex Compositions Optimization and Generalized Manifold Sampling

by
Yuheng Ma

Supervised by
Guoyong Gu

An Undergraduate Dissertation Submitted to
Kuang Yaming Honors School, DII(Statistics)



Kuang Yaming Honors School
DII(Statistics)

June 8, 2021



南京大学本科毕业论文（设计）

诚信承诺书

本人郑重承诺：所呈交的毕业论文（设计）（题目：非光滑复合优化与拓展流形采样算法）是在指导教师的指导下严格按照学校和院系有关规定由本人独立完成的。本毕业论文（设计）中引用他人观点及参考资源的内容均已标注引用，如出现侵犯他人知识产权的行为，由本人承担相应法律责任。本人承诺不存在抄袭、伪造、篡改、代写、买卖毕业论文（设计）等违纪行为。

作者签名：马宇恒

学号：171240510

日期：2024.6.8

南 京 大 学

本科生毕业论文（设计、作品）指导教师评阅意见

指导教师评语：（不少于 300 字）

论文拓展了基于局部模型的无导数优化模型——流形采样算法至更加广泛的形式，允许 f 的形式为 $f = h \circ F + \varphi$ ，其中 φ 为光滑函数，其中 h 为非凸非光滑函数， F 为光滑映射，但其雅可比矩阵很难获得。论文的拓展使得算法能够对一类常用形式的非光滑、非凸复合函数，即机器学习中常用的损失加惩罚形式的目标函数，进行无导数优化。在证明算法收敛性的同时，论文完成了丰富的数值实验，在标准测试集上检验了算法，并经验的讨论了实验结果。最后，文章运用该算法训练浅层神经网络，展示了算法的有效性。该毕业设计涉及分析、计算的理论技巧，同时需要强大的编程能力与实验能力，同时还需要对数值优化相关进展有充分的了解。论文说明该生具有很好的本科数学基础和较强的分析问题和解决问题的能力。

指导教师签名：



2021 年 5 月 26 日

南 京 大 学

本科生毕业论文（设计、作品）评阅教师评阅意见

评阅教师评语：（不少于 300 字）

马宇恒同学的毕业设计研究并拓展了基于局部模型的无导数优化模型——流形采样算法，其工作既有严谨的理论证明，也有详尽的数值实验。最后，文章应用了该算法处理了带有非光滑权重的神经网络的优化问题，为算法的实用性提供了保障。该毕业设计理论与实验相互佐证，并且针对实验中出现的异常情况给出了相应的分析。同时，文章最后还提出了未来进一步的研究方向，设想合理，证明该生对文章工作有深入的思考。文章排版优美，结构严谨，语言条理清晰，逻辑性强。结果展示清晰生动，实验条件细节详尽，具有可复现性。同时，该文章所开发的软件包已经开源在 GitHub 平台。论文体现了该生具有不错的本科数学基础和较强的分析问题和解决问题的能力，论文符合本科毕业学位论文的要求。

评阅教师签名：杨俊峰

2021 年 5 月 26 日

南 京 大 学

本科生毕业论文（设计、作品）答辩记录、成绩评定

答辩记录：

问：实验中的两个 Eta 参数作用各自是什么？

Eta1 决定了该循环是否可以接受， Eta2 决定了该循环是否成功。

问：函数 h 是否是光滑的？和连续选择函数集合的关系是什么？

不是光滑的，是分段光滑的。 h 在每个小邻域内与连续选择函数集合中的某个元素相等。

答辩记录人签名：

答辩小组评语：（不少于 100 字）

论文拓展了已有的基于局部模型的无导数优化模型的流形采样算法至更加广泛的形式，论文理论翔实，数值结果正确，达到了本科毕业论文的要求。马宇恒同学数学基础扎实，对文章细节把握清晰。答辩过程流利，展示过程清晰生动。

答辩小组成员：

杨俊祥 钱志 陈峰

成绩

86

组长签名：

杨俊祥

答辩时间：2021 年 5 月 28 日

南京大学本科生毕业论文(设计、作品)中文摘要

题目：非光滑复合优化与拓展流形采样算法

院系：匡亚明学院

专业：理强（统计学）

本科生姓名：马宇恒

指导老师（姓名、职称）：顾国勇副教授

摘要：

非光滑无导数优化是数值优化领域里富有前景但充满挑战的方向。流形采样算法是一类基于局部模型的无导数优化模型，其可以优化函数 $f = h \circ F$ ，其中 h 为非凸非光滑函数， F 为光滑映射，但其雅可比矩阵很难获得。算法通过在 x 附近随机采样，建立 F 在 x 附近的模型，结合 h 的信息，得到下降点，从而生成收敛到 f 的 Clark 稳定点的序列。我们拓展已有的流形采样算法至更加广泛的形式，允许 f 的形式为 $f = h \circ F + \phi$ ，其中 ϕ 为光滑函数，使得算法能够对一类常用形式的非光滑、非凸复合函数，即机器学习中常用的损失 + 惩罚形式的目标函数，进行无导数优化。在证明了算法收敛性的同时，我们完成了丰富的数值试验，在标准测试集上检验了算法，并经验的讨论了实验结果。最后，我们采用该算法训练浅层神经网络，展示了算法的有效性。

关键词：非光滑优化；无导数优化；复合优化；流形采样

南京大学本科生毕业论文(设计、作品) 英文摘要

THESIS: Nonsmooth Nonconvex Compositions Optimization and Generalized Manifold Sampling

DEPARTMENT: Kuang Yaming Honors School

SPECIALIZATION: DII(Statistics)

UNDERGRADUATE: Yuheng Ma

MENTOR: Guoyong Gu

ABSTRACT:

Non-smooth derivative free optimization method is a challenging but promising topic in optimization. Manifold sampling is a type of model based derivative free algorithm that optimize the composition $f = h \circ F$, where h is a non-smooth non-convex function, F is a smooth map without inexpensive Jacobian. Algorithm build local model of F around by sampling, which is combined with information of h to acquire decrease point. It generates a sequence converging to Clark stationary point of f . We modified manifold sampling algorithm to a more general case, allowing $f = h \circ F + \phi$, such that it can optimize non-smooth, non-convex compositions in a derivative free way. As convergence guarantee for the generalized algorithm provided, we also implement various numerical experiments and discussed results empirically. Meanwhile, we train neural network using manifold sampling, illustrating its power.

KEY WORDS: Non-smooth Optimization, Derivative Free Optimization, Composite Optimization, Manifold Sampling

目 录

1	绪论	1
1.1	研究背景	1
1.2	相关工作	1
1.3	本文主要工作	2
1.4	本文结构	2
2	算法概述	3
2.1	研究问题	3
2.2	记号	3
2.3	定义与假设	3
2.4	算法思路	5
2.5	算法组成	5
2.5.1	无导数模型	5
2.5.2	主模型	6
2.5.3	充分下降条件	6
2.5.4	流形采样循环	6
2.5.5	检验	7
2.6	伪代码	7
2.7	重要事实	7
3	理论保障	9
3.1	准备工作	9
3.2	收敛性	11
4	仿真实验	15
4.1	L^1 范数	15
4.2	基准测试集	17
4.3	非凸性	17
4.4	实证研究	20
5	总结与展望	21
	参考文献	23
	致 谢	27
A	材料	29
A.1	代码	29
A.2	测试结果	30
A.3	基准数据集	31

第一章 绪论

1.1 研究背景

无约束的复合优化是非光滑优化中一类重要的问题，其通常形式如下：

$$\underset{x}{\text{minimize}} \quad f(x) \triangleq \phi(x) + h(F(x)), \quad x \in \mathbb{R}^n. \quad (1-1)$$

其中 ϕ 为光滑函数，并且在很多情形下为 0。 $F: \mathbb{R}^n \rightarrow \mathbb{R}^p$ 为光滑函数， $h: \mathbb{R}^p \rightarrow \mathbb{R}$ 为连续但不可导，而且通常被假设为凸函数。这类问题经常伴随着优化与范数有关的损失函数。

在机器学习中，添加惩罚项能够促使参数朝着理想的方向靠近，如减小参数尺度的岭回归和神经网络中的权重衰减使用了 L^2 惩罚，Lasso 回归^[1] 是 L^1 惩罚的结果，ElasticNet 惩罚^[2] 则是二者的结合，还有缩减秩的 L^0 范数。当范数算子为非光滑，如 L^1 或 L^∞ 范数^[3] 时，优化问题即为复合优化。

复合优化在控制、信号处理领域有很多实际应用^[4]，如分散式控制系统^[5] 和低复杂性建模^[6]。John Duchi 和 Ruan Feng^[7] 考虑在相位恢复问题中应用复合优化，考虑真实信号 $x^* \in \mathbb{R}^n$ 和量测向量 (a_1, \dots, a_n) ，观察信号为 $b_i = \phi(\langle a_i, x^* \rangle) + \varepsilon_i$, $i = 1, \dots, n$ ，其中 ϕ 为非线性函数。则 x^* 的估计可以通过 $\min_x \frac{1}{n} \sum_{i=1}^n (\phi(\langle a_i, x \rangle) - b_i)^2$ 得到。然而实际过程中观察到的往往是 $b_i = \langle a_i, x^* \rangle^2$ ，所以需要优化 $\frac{1}{n} \sum_{i=1}^n |\langle a_i, x \rangle^2 - b_i|$ ，从而成为复合优化问题。

许多经典优化问题同样可以转化为复合优化。考虑一组非线性方程 $F_i(x) = 0$, $i = 1, \dots, n$ ，方程的求解等价于优化 $\|F(x)\| = \|(F_1, \dots, F_n)\|$ ， $\|\cdot\|$ 为任意的范数算子^[8]。当算子为非光滑，问题即为复合优化。寻找非线性系统的可行点也可以视为复合优化问题。考虑一系列非线性不等式约束 $F_i(x) \leq 0$, $i = 1, \dots, n$ ，则可行点的寻找等价于求解 $\min \Sigma \|F_i^+(x)\|$ ，其中 $F_i^+(x) = \max(F_i(x), 0)$ 。

1.2 相关工作

复合优化的理论最先由 A. D. Ioffe^[9] 建立，并且给出了 h 为次线性的局部极小条件。而后，J. V. Burke^[10] 将 h 的限制条件放宽至凸函数。当 F 为局部 Lipschitz 且 Gâteaux 可微时，Jeyakumar 和 X. Q.^[11] 给出了一阶拉格朗日必要条件和二阶充分条件。在 F 的导数容易取得的情况下，^[12-14] 都给出了理论分析。而就数值算法而言，在直接使用梯度的算法失效后，近端梯度下降^[15,16] 成为了很好的替代品，即便它们需要非光滑项关于变量可分离这一很强的假设。通过将光滑项与非光滑项的变量分离并添加等式约束，将问题转化为约束优化，交替方向乘子法 (ADMM)^[17] 等一系列方法便可以使用。此时问题被分解为光滑或者容易求解的非光滑子问题，便于分布式计算。但是

在 ϕ 非凸的情形下，这些方法均没有很好的收敛性保障。而乘子法 (Method of Multipliers)^[18] 虽然可以提供很好的收敛保障，但由于其需要针对所有变量优化增广拉格朗日函数，计算速度被大大限制。Dhingra 和 Khong^[19] 通过非光滑项进行最小化找到相应流型，再将增广拉格朗日映射至该流型，找到了光滑的近端增广拉格朗日函数，使得问题同时具有分布式算法、明确的参数调整方法和凸情形下的理论收敛性。束方法^[20-22] 是非光滑优化中常用的方法，并且在非凸情形下也有很好的表现，当条件放宽至无法得到 F 的导数 (Derivative-free) 时依然可以使用^[23,24]。

然而，正如很多文章中要求 h 为某个范数，如 L^2 范数^[25]、 L^∞ 范数^[14,26]，几乎所有已知的算法都要求 h 为凸函数。Larson, Menickelly 和 Wild^[27] 提出了流型采样的方法 (Manifold Sampling)，在置信域方法的框架下，通过 $F(x)$ 在附近的流型上的估计和 h 的 Clarke 方向导数来估计 $f(x)$ 。虽然文章采用了 L^1 范数作为分析对象，但其理论与实现均不依赖 h 的凸性。同时，当 F 的导数难以取得时，流型采样方法仍然可以使用，并且相比梯度采样方法^[28,29] 有非随机性的优势。Khan, Larson 和 Wild^[30] 拓展了这一工作，使得 h 可以是任意的分段线性函数。Larson, Menickelly 和 Zhou^[31] 最近将结果拓展至 h 为任意连续函数并且其 Clarke 次导数可计算的情形，同时要求 ϕ 为 0。这是目前唯一允许 h 为任意连续函数且无需 F 一阶信息的工作。

1.3 本文主要工作

在本课题中，我们将 Larson, Menickelly 和 Zhou 的工作^[31] 扩展到的 ϕ 是一个光滑函数的情况。具体工作包括：

1. 提出了在拓展情况下的算法形式。
2. 证明了该形式理论上的收敛性。
3. 实现了基于 Python 的流形采样软件包。
4. 在简单样例上击败了经典优化方法，在标准测试函数集上进行了算法测试，得到了良好的结果。
5. 使用了流形采样算法训练神经网络。

1.4 本文结构

本文的各章节组织结构如下：

第一章：绪论。简要说明了非光滑非凸复合优化的研究意义与背景。并概括地描述了这篇文章的工作，总结了本文结构。第二章：算法概述。阐述了算法思路与主要组成部分，给出后续证明所需的定义与假设。第三章：理论保障。证明了算法在一定条件下的收敛性。第四章：仿真实验。展示了算法在测试集上的表现。第五章：总结与展望。

第二章 算法概述

2.1 研究问题

我们希望优化

$$\underset{x}{\text{minimize}} f(x) \triangleq \phi(x) + h(F(x)), x \in \mathbb{R}^n.$$

其中 $\phi: \mathbb{R}^n \rightarrow \mathbb{R}$ 是光滑并且二阶 Lipschitz 连续的函数。 $F: \mathbb{R}^n \rightarrow \mathbb{R}^p$ 是可微函数, $h: \mathbb{R}^p \rightarrow \mathbb{R}$ 是连续函数, 但允许其非光滑、非凸。

2.2 记号

若不做特殊约定, 所有范数均为 l^2 范数。集合 S 的闭包记为 $\mathbf{cl}(S)$ 。集合 S 的内点集记为 $\mathbf{int}(S)$ 。集合 S 的凸包记为 $\mathbf{co}(S)$ 。集合 S 关于 F 的像集记为 $\mathbf{Im}(S) =: \{F(x) | x \in S\}$ 。球 $B(x, \delta)$ 为闭球。局部 Lipschitz 函数 f 在 x 点的广义克拉克导数为

$$\partial_C f(x) \triangleq \mathbf{co} \left(\left\{ \lim_{y^j \rightarrow x} \nabla f(y^j) : y^j \in D \right\} \right)$$

其中 D 为 x 的某个邻域。若 $0 \in \partial_C f(x)$, 则称 x 为 f 的 Clark 稳定点。一般我们使用 k 作为角标来代指变量在第 k 个循环里的值。

2.3 定义与假设

这一节中包含我们接下来需要用到的所有假设与定义。

定义 2.1 对某个常数 δ_{\max} 和 \mathbb{R}^n 中的点 x_0 , 定义

$$\mathcal{L}_{\max} \triangleq \bigcup_{x \in \mathcal{L}(x^0)} B(x, \Delta_{\max})$$

其中 $\mathcal{L}(x^0)$ 是某个我们所需要的集合。这一定义给出了我们接下来考虑的函数定义域。

假设 2.2 我们对 f 和 F 在定义域 \mathcal{L}_{\max} 上做下列假设:

- 对任意的 $x_0 \in \mathbb{R}^n$, 集合 $\mathcal{L}(x_0) =: \{x : f(x) \leq f(x_0)\}$ 是有界的。
- F_i 以常数 L_{F_i} 在 \mathcal{L}_{\max} 上 Lipschitz 连续。
- ∇F_i 以常数 $L_{\nabla F_i}$ 在 \mathcal{L}_{\max} 上连续。

假设 2.3 我们对光滑函数 ϕ 做如下假设:

- ϕ 和 $\nabla\phi$ 的求解是简单的。
- 对任意 $x_0 \in R^n$, 集合 $\{x : \phi(x) \leq \phi(x_0)\}$ 是有界的。
- ϕ 以常数 L_ϕ 在 \mathcal{L}_{\max} 上 Lipschitz 连续。
- $\nabla\phi$ 以常数 $L_{\nabla\phi}$ 在 \mathcal{L}_{\max} 上 Lipschitz 连续。

定义 2.4 称函数 h 是 $U \subseteq R^p$ 上的一个连续选择, 若 h 在 U 上连续并且

$$h(z) \in \{h_j(z) : h_j \in \mathfrak{H}\}, \quad \forall z \in U$$

其中 \mathfrak{H} 是一个选择函数的有限集合, 其中的每个元素均为光滑函数。例如 $h(z) = \|z\|_1$ 是一个连续选择, 若 $h(z) \in \{\sum \xi_i z_i | \xi_i = \pm 1\}$ 。

定义 2.5 若 h 为一个连续选择, 则

$$S_j \triangleq \{z : h(z) = h_j(z)\}, \quad \tilde{S}_j \triangleq \text{cl}(\text{int}(S_j)), \quad \mathbb{A}(z) \triangleq \{j : z \in \tilde{S}_j\}$$

$\mathbb{A}(z)$ 是 z 点处的活跃集合。我们称任意 h_j 为 z 点处的活跃选择, 若 $j \in \mathbb{A}(z)$ 是 z 处的活跃集合。对任意 z 的集合 Z , 记 $\mathbb{A}(Z) = \bigcup_{z \in Z} \mathbb{A}(z)$ 。

假设 2.6 我们对 h 做如下假设。

- $\mathbb{A}(z)$ 在任意 z 处是可计算的。
- h_i 以常数 L_{h_i} 在 \mathcal{L}_{\max} 上 Lipschitz 连续。
- ∇h_i 以常数 $L_{\nabla F_i}$ 在 \mathcal{L}_{\max} 上连续。

定义 2.7 函数 m^{F_i} 是一个 F_i 在 $B(x, \Delta)$ 上的完全线性模型, 若存在常数 $\kappa_{i,ef}$ 和 $\kappa_{i,eg}$ 使得

$$\begin{aligned} |F_i(x+s) - m^{F_i}(x+s)| &\leq \kappa_{i,ef} \Delta^2, \quad \forall s \in B(0; \Delta) \\ \|\nabla F_i(x+s) - \nabla m^{F_i}(x+s)\| &\leq \kappa_{i,eg} \Delta, \quad \forall s \in B(0; \Delta) \end{aligned}$$

假设 2.8 对 $F = (F_1, \dots, F_p)$ 及其完全线性模型 $m^F = (m^{F_1}, \dots, m^{F_p})$, m^F 二阶连续可微并且 Lipschitz 连续, $\|\nabla^2 m^{F_i}(x)\| \leq \kappa_{i,mH}$ 。

定义 2.9 这里我们定义一些常数, 大多数为 Lipschitz 常量。

$$\begin{aligned} L_F &\triangleq \sqrt{\sum_{i=1}^p L_{F_i}^2} \quad L_{\nabla F} \triangleq \sqrt{\sum_{i=1}^p L_{\nabla F_i}^2}, \quad L_h \triangleq \max_{j \in \{1, \dots, |\mathfrak{H}|\}} \{L_{h_j}\}, \\ \kappa_F &\triangleq \sum_{i=1}^p \kappa_{i,ef}, \quad \kappa_g \triangleq \sum_{i=1}^p \kappa_{i,eg}, \quad \kappa_H \triangleq \sum_{i=1}^p \kappa_{i,mH} \\ C &\triangleq (2L_h \kappa_g + 2L_{\nabla h} L_F^2 + L_h L_{\nabla F} + L_{\nabla \phi}) \end{aligned}$$

定义 2.10 记 \mathbb{Z}^k 为采样集

$$\mathbb{Z}^k = \text{co}(\{F(y) : y \in B(x^k, \Delta_k)\})$$

这即是说, \mathbb{Z}^k 包含所有具有形式 $\alpha F(x^k) + (1 - \alpha)F(y), \alpha \in [0, 1]$ 的点。所以, $\mathbb{Z}^k \subset \text{co}(\text{Im}(\mathcal{L}_{\max}))$ 。令 \mathbb{D}^k 为选择集合的梯度集。

$$\mathbb{D}^k = \{\nabla h_j(z) : z \in \mathbb{Z}^k, j \in \mathbb{A}(z)\}$$

记 \mathbb{G}^k 为 \mathbb{Z}^k 的梯度估计集合 (或者同^[31] 称为 generator)。

$$\mathbb{G}^k = \{\nabla M(x^k) \nabla h_j(z) + \nabla \phi : z \in \mathbb{Z}^k, j \in \mathbb{A}(z)\}$$

记 G^k 为 D^k 列向量为相应集合元素的矩阵。显然, $G^k = \nabla M(x^k) D^k + (\nabla \phi, \dots, \nabla \phi)$ 。

定义 2.11 定义梯度估计 g^k

$$g^k \triangleq \text{proj}(0, \text{co}(\mathbb{G}^k)) \in \text{co}(\mathbb{G}^k) \quad (2-1)$$

即 \mathbb{G}^k 的最小模元素。

假设 2.12 对 2.9 中的 L_F , 我们有

$$F(x^k) \subseteq \mathbb{Z}^k \subset \mathcal{B}(F(x^k); L_F \Delta_k)$$

2.4 算法思路

在具体介绍流形采样算法的细节之前, 我们先陈述算法的大致思路以便理解。首先, 从直觉上来说, 流形采样算法是基于模型的 (Model-based) 无导数优化模型。由模型 m^F 对 F 进行估计, 由选择集合中的元素 h_j 对 h 进行估计, 于是我们有

$$\nabla f = \nabla \phi + \nabla h(F(x)) = \nabla \phi + \nabla F(x) \nabla h_j(F(x)) \approx \nabla \phi + \nabla m^F(x) \nabla h_j(F(x))$$

此时, 基于梯度下降或信赖域方法的优化思路, 局部最小值可以迭代得到。在 2.10 中, \mathbb{Z}^k 与无导数模型建模相联系。 \mathbb{D}^k 代表了 ∇h 的潜在可能。 \mathbb{G}^k 是所有 ∇f 的可能估计。2.11 中的 g^k 是最终对梯度的估计。

2.5 算法组成

2.5.1 无导数模型

为了获得 F 的局部估计, 我们采用基于模型的无导数建模思路。^[32] 对函数进行局部完全线性估计, 估计的目标是最小化二次损失。值得注意的是, 在样本为 n 维时, ^[32] 只需 $n+1$ 个点进行建模。

2.5.2 主模型

为了获得投影 $g^k \triangleq \text{proj}(0, \text{co}(\mathbb{G}^k)) \in \text{co}(\mathbb{G}^k)$ ，我们使用凸二次优化的技巧，即转变优化问题为

$$\begin{aligned} & \underset{\lambda}{\text{minimize}} && \lambda^\top (G^k)^\top G^k \lambda \\ & \text{subject to} && e^\top \lambda = 1, \lambda \geq 0 \end{aligned} \quad (2-2)$$

解 2-2 并且令

$$g^k \triangleq G^k \lambda^* + \nabla \phi, \quad d^k \triangleq D^k \lambda^* \quad (2-3)$$

便获得了梯度估计。我们还可以使用 d^k 作为光滑元素 m^{F_i} 的权重，有

$$m_k^f(x) \triangleq \sum_{i=1}^P [d^k]_i m^{F_i}(x) + \phi(x)$$

简单的观察后便可以得到：

$$\nabla m_k^f(x^k) = \sum_{i=1}^P [d^k]_i \nabla m^{F_i}(x^k) + \nabla \phi(x^k) = (\nabla m^F(x^k) D^k + \nabla \phi) \lambda^* = G^k \lambda^* = g^k$$

2.5.3 充分下降条件

建模之后，主模型可以被信赖域方法局部优化。

$$\underset{s \in B(0; \Delta_k)}{\text{minimize}} \quad m_k^f(x^k + s) \quad (2-4)$$

此时，求精确解是不必要的。一方面，主模型本身就是估计的结果。另一方面，精确解会极大的降低效率。我们给出一个使得模型收敛的充分下降条件。

$$\begin{aligned} & \phi(x^k) - \phi(x^k + s^k) + \langle m^F(x^k) - m^F(x^k + s^k), d^k \rangle \\ & \geq \frac{\kappa_d}{2} \|g^k\| \min \left\{ \Delta_k, \frac{\|g^k\|}{L_h \kappa_H + L_{\nabla \phi}} \right\} \end{aligned} \quad (2-5)$$

注意到充分下降条件 2-5 与传统的信赖域方法中所使用的不同。传统方法衡量 $m^f(x)$ 与 $m^f(x+s)$ 差值的大小。2-5 中，等式左边衡量的是对 m^F 的下降在 \mathcal{D} 的凸组合上的投影，也就是对 h 进行了线性分解之后的下降结果。

2.5.4 流形采样循环

在 2-5 中，基于样本集合 \mathbb{Z}^k 所建的局部模型决定了 s^k 的选取。然而如果 \mathbb{Z}^k 本身选取的不好， s^k 的选取可能也是不恰当的。一个很显然的现象是 \mathbb{Z}^k 决定了 $\mathbb{A}(\mathbb{Z}^k)$ ，而 $\mathbb{A}(\mathbb{Z}^k)$ 与 $\mathbb{A}(F(x))$ 差异太大时， \mathbb{Z}^k 需要被重新加强。

我们在 \mathbb{Z}^k 中加入新的 z ，直到该集合满足下列条件： $\exists z \in \text{co}(F(x^k), F(x^k +$

s^k)) 和下标 j 使得

$$\begin{aligned} j \in \mathbb{A}(z), \quad \nabla h_j(z)^\top (F(x^k) - F(x^k + s^k)) - \nabla \phi(x^z) \cdot s^k \\ \leq h(F(x^k)) + \phi(x^k) - h(F(x^k + s^k)) - \phi(x^k + s^k) \end{aligned} \quad (2-6)$$

其中 x^z 是某个 x 使得 $\|F(x) - z\| \leq \frac{L_{\nabla F}}{4} \|s^k\|^2$ 。这样的 x 的存在性在 3.1 中被证明了。条件 2-6 具体说的是应当存在下标 j 使得 h_j 比 d 糟糕。我们还要求第二个 $z' \in \mathbb{Z}^k$ 的存在，它应当使得

$$j \in \mathbb{A}(z') \text{ and } (s^k)^\top \nabla M(x^k) (\nabla h_j(z') - d^k) \leq 0 \quad (2-7)$$

从几何的角度来说，这是说从 s^k 到梯度估计的投影与 d^k 更加相合（内积值更大）。

2.5.5 检验

与传统的信赖域方法相同，流形采样含有一个比值检验的步骤，用来确保每一步优化的效果。流形采样中使用的检验量 ρ_k 是衡量这样两个量的比值：分子是模型的实际下降，分母是模型 m^F 的估计下降，这两个量均由梯度估计的凸组合 d^k 进行衡量。

$$\rho_k \triangleq \frac{\langle F(x^k) - F(x^k + s^k), d^k \rangle + \phi(x^k) - \phi(x^k + s^k)}{\langle M(x^k) - M(x^k + s^k), d^k \rangle + \phi(x^k) - \phi(x^k + s^k)} \quad (2-8)$$

2.6 伪代码

附录 A.1 中包含算法的所有伪代码。

2.7 重要事实

我们有必要陈述算法的几个重要事实。 η_2 的上界需要满足

$$\eta_{\max} \triangleq \min \left\{ \frac{1}{L_\phi + L_h \kappa_H}, \frac{\eta_1 \kappa_d}{4C} \right\} \quad (2-9)$$

值得注意的是，在实现过程中，更大的 η 可以带来更快的收敛，这一点有可能无法满足。根据循环完成的质量，我们定义可接受循环与成功循环。定义一个循环为可接受的，如果这个循环中有 $\Delta_k < \eta_2 \|g^k\|$ 。如果一个循环中， $\rho_k > \eta_1$ ，那么该循环是成功的。

第三章 理论保障

在这一章中，我们为算法提供理论保障。具体来说，我们将证明，在合适的参数设定和初始值选取下，算法 A.1产生的序列 $(x^k, f(x^k))$ 会收敛到局部极小 $(x^*, f(x^*))$ 。我们首先介绍每个引理的作用。引理 3.1为样本集合的选取提供合适的上界。引理 3.2确保我们建立主模型时，模型的误差是“一阶梯度可控”的。引理 3.3和 3.4确保对样本集合的合理加强是存在并且可以实现的。引理 3.5确保我们永远可以找到合适的 s 使得流形采样循环结束。引理 3.6衡量目标函数每一轮的下降程度。引理 3-5为成功循环提供了充分条件。引理 3.8, 3.9, 3.10一起为序列 $(x^k, f(x^k))$ 找到了收敛子列。定理 3.11是最主要的结果，它证明了这一子列收敛到 Clark 稳定点（次导数集合包含 0 的点）。证明过程与^[31] 基本相同，只有部分涉及 ϕ 的部分进行了调整，所以我们仅提供引理与定理的陈述与调整过的证明。

3.1 准备工作

引理 3.1 对 $z = \alpha F(x) + (1 - \alpha)F(y)$, $\alpha \in [0, 1]$ ，也是一条 $\text{co}(\text{Im}(\mathcal{L}_{\max}))$ 中的路径，我们可以在 \mathcal{L}_{\max} 中找到另外一条路径 $\sigma(\alpha)$, $\alpha \in [0, 1]$, $\sigma(0) = x$, $\sigma(1) = y$ ，使得

$$\|\alpha F(x) + (1 - \alpha)F(y) - F(\sigma(\alpha))\| \leq \frac{1}{4}L_{\nabla F}\|x - y\|^2$$

证明：我们取 $\sigma(\alpha) = \alpha x + (1 - \alpha)y$ 为例。

$$\begin{aligned} & \|\alpha F(x) + (1 - \alpha)F(y) - F(\alpha x + (1 - \alpha)y)\| \\ &= \|F(x) + (1 - \alpha)\nabla F(x)^T(y - x) + (1 - \alpha)^2(y - x)^T\nabla^2 F(x)(y - x) \\ & \quad - (\alpha F(x) + (1 - \alpha)F(y)) + o(\|y - x\|^2)\| \\ &= \|(1 - \alpha)(\nabla F(t(y - x)) - \nabla F(x))^T(x - y) \\ & \quad + (1 - \alpha)^2(y - x)^T\nabla^2 F(x)(y - x) + o(\|y - x\|^2)\| \\ &= \|((1 - \alpha)t - (1 - \alpha)^2)(y - x)^T\nabla^2 F(x)(y - x) + o(\|y - x\|^2)\| \\ &\leq \frac{1}{4}L_{\nabla F}\|x - y\|^2 \end{aligned} \tag{3-1}$$

□

引理 3.2 当所有的假设成立，给定 $x, y \in \mathcal{L}_{\max}$ s.t. $\|x - y\| \leq \Delta$ 。则对于

$$\mathbb{G} \triangleq \{\nabla M(x)\nabla h_i(F(x)) + \nabla \phi : i \in I\}$$

$$\mathbb{H} \triangleq \text{co}(\{\nabla F(y)\nabla h_j(F(y)) + \nabla \phi : j \in I\})$$

我们有

$$\|g - v(g)\| \leq c_2 \Delta$$

其中

$$c_2 \triangleq L_h (L_{\nabla F} + \kappa_g) + L_F^2 L_{\nabla h_i}$$

引理 3.3 当所有的假设成立，我们可以找到 z 和 j 使得他们满足 2-6。

证明： 我们定义

$$z(\alpha) \triangleq \alpha F(x) + (1 - \alpha)F(x + s)$$

采用反证法。若不存在这样的 α 使得 $z(\alpha)$ 满足要求，则

$$\begin{aligned} & \phi(x) + h(F(x)) - \phi(x + s) - h(F(x + s)) \\ & \geq \int_0^1 \inf_{j \in \mathbb{A}(z(\alpha))} \{ \nabla h_j(z(\alpha))^\top (F(x) - F(x + s)) \} - \nabla \phi(x + (1 - \alpha)s) \cdot s d\alpha \\ & > \int_0^1 (\phi(x) + h(F(x)) - \phi(x + s) - h(F(x + s))) d\alpha \\ & = \phi(x) + h(F(x)) - \phi(x + s) - h(F(x + s)) \end{aligned}$$

从而得到矛盾。 \square

引理 3.4 当所有的假设成立，算法 A.2 会在有限时间内得到 z 和 j 。

引理 3.5 取 d^k 和 $g^k = \nabla m^F d^k + \nabla \phi$ 如 2-3。当所有的假设成立，并且 2-9 被满足，则

$$\hat{s} = -\Delta_k \frac{g^k}{\|g^k\|}$$

可以满足 2-5。

证明 显然有

$$\|d^k\| \leq L_h \quad (3-2)$$

对于假设 2.8 和定义 2.9,

$$\begin{aligned} & \left\| M(x^k) + \nabla M(x^k)^\top s - M(x^k + s) \right\| \\ & \leq \sum_{i=1}^p \left| m^{F_i}(x^k) + \nabla m^{F_i}(x^k)^\top s - m^{F_i}(x^k + s) \right| \\ & \leq \sum_{i=1}^p \frac{1}{2} \kappa_{i, \text{mH}} \|s\|^2 = \frac{1}{2} \kappa_H \|s\|^2 \end{aligned} \quad (3-3)$$

结合 3-2 和 3-3, 我们有

$$\begin{aligned}
& (d^k)^\top \left(M(x^k) + \nabla M(x^k)^\top \hat{s} - M(x^k + \hat{s}) \right) \\
& \geq -\|d^k\| \left\| M(x^k) + \nabla M(x^k)^\top \hat{s} - M(x^k + \hat{s}) \right\| \\
& \geq -\frac{1}{2} L_h \kappa_H \|\hat{s}\|^2 = -\frac{1}{2} L_h \kappa_H \Delta_k^2
\end{aligned}$$

使用 2-9, 可以得到

$$\begin{aligned}
& \phi(x^k) - \phi(x^k + s^k) + \langle M(x^k) - M(x^k + \hat{s}), d^k \rangle \\
& \geq -\nabla \phi(x^k) \cdot \hat{s} - \frac{1}{2} L_{\nabla \phi} \hat{s}^\top \hat{s} - (d^k)^\top \nabla M(x^k)^\top \hat{s} - \frac{1}{2} L_h \kappa_H \Delta_k^2 \\
& = - (g^k)^\top \hat{s} - \frac{1}{2} L_h \kappa_H \Delta_k^2 = \|g^k\| \Delta_k - \frac{1}{2} L_h \kappa_H \Delta_k^2 - \frac{1}{2} L_{\nabla \phi} \Delta_k^2 \\
& \geq \|g^k\| \Delta_k - \frac{1}{2} \|g^k\| \Delta_k = \frac{1}{2} \|g^k\| \Delta_k \\
& \geq \frac{\kappa_d}{2} \|g^k\| \min \left\{ \Delta_k, \frac{\|g^k\|}{L_{\nabla \phi} + L_h \kappa_H} \right\}
\end{aligned}$$

■

3.2 收敛性

引理 3.6 当所有的假设成立, 若算法 A.1 的第 k 个循环是可接受的, 则对于 2.9 中的 C , 有

$$\begin{aligned}
& \phi(x^k) + h(F(x^k)) - \phi(x^k + s^k) - h(F(x^k + s^k)) \\
& \geq (d^k)^\top (F(x^k) - F(x^k + s^k)) - \nabla \phi(x^k) \cdot s - C \Delta_k^2
\end{aligned} \tag{3-4}$$

证明 我们首先证明 2-7 必然成立。在 2-7 中, 若 z' 被找到, 则必然成立。若没有, 则 $s^k = -\Delta_k \frac{g^k}{\|g^k\|}$ 。从经典的投影理论和 2-1, 我们有

$$\begin{aligned}
& (\nabla h_j(z') - d^k)^\top \nabla m^F(x^k)^\top s^k \\
& = -(\nabla h_j(z') - d^k)^\top \nabla m^F(x^k)^\top \Delta_k \frac{g^k}{\|g^k\|} \\
& = \text{positiveconst} \cdot [(\nabla \phi(x^k) + d^k \nabla m^F(x^k)^\top) g^k - (\nabla \phi(x^k) + \nabla h_j(z') \nabla m^F(x^k)^\top) g^k] \\
& \leq 0
\end{aligned}$$

接下来由假设 2.2 和假设 2.9,

$$|F(x^k + s^k) - F(x^k) - \nabla F(x^k)^\top s^k| \leq \frac{L_{\nabla F}}{2} \|s^k\|^2$$

所以

$$\begin{aligned}
& (\nabla h_j(z) - d^k)^\top (F(x^k + s^k) - F(x^k) - \nabla F(x^k)^\top s^k) \\
& \leq \|\nabla h_j(z) - d^k\| \|F(x^k + s^k) - F(x^k) - \nabla F(x^k)^\top s^k\| \\
& \leq (2L_h) \frac{L_{\nabla F}}{2} \|s^k\|^2
\end{aligned}$$

于是有

$$\begin{aligned}
& \phi(x^k + s^k) + h(F(x^k + s^k)) - \phi(x^k) - h(F(x^k)) \\
& - \nabla \phi(x^k) \cdot s - (d^k)^\top (F(x^k + s^k) - F(x^k)) \\
& \leq (\nabla h_j(z) - d^k)^\top (F(x^k + s^k) - F(x^k)) + \phi(x^k + s^k) - \phi(x^k) - \nabla \phi(x^k) \cdot s \\
& = (\nabla h_j(z) - d^k)^\top (F(x^k + s^k) - F(x^k)) + s^\top \nabla^2 \phi(s - x^z + x) \\
& \leq (\nabla h_j(z) - d^k)^\top \nabla F(x^k)^\top s^k + (L_h L_{\nabla F} + L_{\nabla \phi}) \|s^k\|^2 \\
& = (\nabla h_j(z') - d^k)^\top \nabla F(x^k)^\top s^k + (\nabla h_j(z) - \nabla h_j(z'))^\top \nabla F(x^k)^\top s^k + (L_h L_{\nabla F} + L_{\nabla \phi}) \|s^k\|^2 \\
& \leq (\nabla h_j(z') - d^k)^\top (\nabla m^F(x^k)^\top s^k + (\nabla F(x^k) - \nabla m^F(x^k))^\top s^k) \\
& \quad + 2L_{\nabla h} L_F^2 \Delta_k^2 + L_{\nabla \phi} \Delta_k^2 + L_h L_{\nabla F} \Delta_k^2 \\
& = (\nabla h_j(z') - d^k)^\top \nabla m^F(x^k)^\top s^k + (\nabla h_j(z') - d^k)^\top (\nabla F(x^k) - \nabla m^F(x^k))^\top s^k \\
& \quad + 2L_{\nabla h} L_F^2 \Delta_k^2 + L_{\nabla \phi} \Delta_k^2 + L_h L_{\nabla F} \Delta_k^2 \\
& \leq 0 + 2L_h \kappa_g \Delta_k^2 + 2L_{\nabla h} L_F^2 \Delta_k^2 + L_{\nabla \phi} \Delta_k^2 + L_h L_{\nabla F} \Delta_k^2 \\
& = C \Delta_k^2
\end{aligned}$$

■

引理 3.7 当所有的假设成立，对可接受的循环，若

$$\Delta_k < \frac{\kappa_d(1 - \eta_1)}{4\kappa_f L_h} \|g^k\| \quad (3-5)$$

则 $\rho_k > \eta_1$ ，即该循环是成功的。

引理 3.8 当所有的假设成立，若 $\{x^k, \Delta_k\}$ 是算法 A.1 生成的序列，则 $\{f(x^k)\}$ 是非减的，且 $\lim_k \Delta_k = 0$ 。

证明 我们首先证明其非减。余下的证明与^[31] 相似，不过多赘述。若一个循

环是不成功的, 则 $x^{k+1} = x^k$, 即 $f(x^{k+1}) = f(x^k)$ 。若它是成功的, 则

$$\begin{aligned}
f(x^k) - f(x^{k+1}) &\geq (d^k)^\top (F(x^k) - F(x^{k+1})) - \nabla \phi(x^z) \cdot s - C\Delta_k^2 \\
&\geq (d^k)^\top (F(x^k) - F(x^{k+1})) + \phi(x^k) - \phi(x^k + s^k) - L_{\nabla \phi} \Delta_k^2 - C\Delta_k^2 \\
&= \rho_k \left[(d^k)^\top (m^F(x^k) - m^F(x^{k+1})) + \phi(x^k) - \phi(x^k + s^k) \right] - L_{\nabla \phi} \Delta_k^2 - C\Delta_k^2 \\
&\geq \rho_k \frac{\kappa_d}{2} \|g^k\| \min \left\{ \Delta_k, \frac{\|g^k\|}{L_h \kappa_H + L_{\nabla \phi}} \right\} - L_{\nabla \phi} \Delta_k^2 - C\Delta_k^2 \\
&= \rho_k \frac{\kappa_d}{2} \|g^k\| \Delta_k - C\Delta_k^2 > \eta_1 \frac{\kappa_d}{2} \|g^k\| \Delta_k - L_{\nabla \phi} \Delta_k^2 - C\Delta_k^2 \\
&> \eta_1 \frac{\kappa_d}{2} \frac{4C}{\eta_1 \kappa_d} \Delta_k^2 - L_{\nabla \phi} \Delta_k^2 - C\Delta_k^2 = C\Delta_k^2 - L_{\nabla \phi} \Delta_k^2 > 0
\end{aligned}$$

■

引理 3.9 对于引理 3.8 中的序列, 我们有

$$\liminf_{k \rightarrow \infty} \|g^k\| = 0$$

引理 3.10 对引理 3.8 中的序列, 存在一个可接受循环的无穷子列 $\{k_j\}$ 使得

$$\lim_{j \rightarrow \infty} \|g^{k_j}\| = 0$$

并且 $\{x^{k_j}\}$ 收敛到某个 x^* 。同时, $0 \in \partial_C f(x^*)$ 。

定理 3.11 当所有的假设成立, 由算法 A.1 生成的序列的聚点里包含一个 Clark 稳定点。

第四章 仿真实验

这一章中，我们展示使用流形采样进行的数值仿真实验结果。我们首先使用简单的例子展示流形采样在计算速度和精确度上的优势。然后，我们采用^[33]提供的基准测试集对流形采样进行测试，该测试集包含十个非光滑无导数优化问题。我们特别注意流形采样跳出初始值附近局部极小的能力，并且用一个小小的例子展现这一点。最后，我们使用流形采样优化非凸、非光滑无导数优化问题。这是流形采样算法独有的能力。

4.1 L^1 范数

我们首先考虑简单的情形。

$$\phi(x) + h(F(x)) = \|x\|_2 + \|((x_2 - x_1^2)^2 + (1 - x_1)^2, x_3)\|_1, x \in \mathbb{R}^3 \quad (4-1)$$

此时 h 为 L^1 范数， $\phi: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ 为 L^2 范数， F 在二维向量第一个位置为 Rosenbrock 测试函数，第二个位置为恒同映射。注意到 x 的前两个维度与第三个维度是可以分别优化的，函数以 0 作为全局最优。函数值关于 x_1 和 x_2 的绘图如图 4-1。

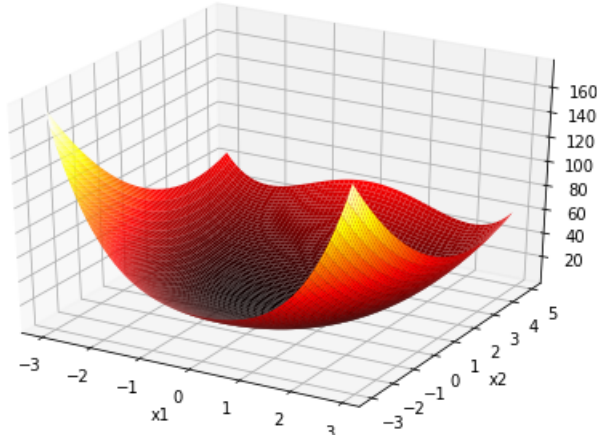


图 4-1: 函数值关于 x_1 和 x_2 的绘图。

注意到这是一个凸问题，关于局部收敛的讨论是无意义的。在经过简单的尝试后，我们决定设定超参数为 $\eta_1 = 0.1$, $\eta_2 = 1$, $\kappa_d = 10^{-4}$, $\kappa_H = 1$, $\gamma_d = 0.5$, $\gamma_i = 1.5$, $\Delta_{max} = 10^4$, $\Delta_{min} = 10^{-13}$ 。初始点 x_0 由标准正态分布生成。限制算法至多运行 3000 个循环。考虑到实验效率，我们也设定中止条件。若 $|f(x^k) - f(x^{k+1})| < conv_tol = 10^{-9}$ 连续若干个轮次（我们使用的是 3 个），则代表序列很可能已经收敛，所以算法被中止。我们特别关心每一循环中，用于

建立局部无导数模型的点的数量。点的数量很大程度上影响了效率与精度。图 4-2 展示了不同点的数量对两者的影响。综合考虑下，我们设定点的数量为 10，这一选择能够同时使得模型具有合理的精度和速度。对于大维问题，点的数量应当大于维度，所以点的数量 $npt = \max\{10, n \cdot [1 + \frac{1}{\log n}]\}$ 。

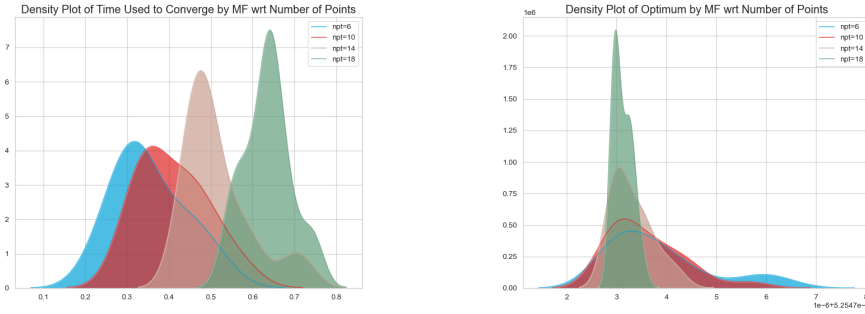


图 4-2: (彩色模式浏览) 时间与函数值关于不同样本点个数的分布。毫无意外的，增加点的个数会使得模型更稳健，同时会增加求解时间。

接下来我们对比流形采样与基准模型。我们选择两个基准模型。第一个是无导数优化模型“Least Square Derivative Free Optimization Tool Box”，由^[32]开发。选择该模型的目的是比较流形采样与其他无导数优化模型。第二个是基于 PyTorch 的简单梯度下降。这一方法能够使用的前提是 h 是分段光滑的，且 $F(x)$ 落入非光滑的部分的支集为零测集。选取这一基准的目的是为后文结合流形采样与神经网络训练作准备。图 4-3 展示了结果对比与流形采样的优越性。

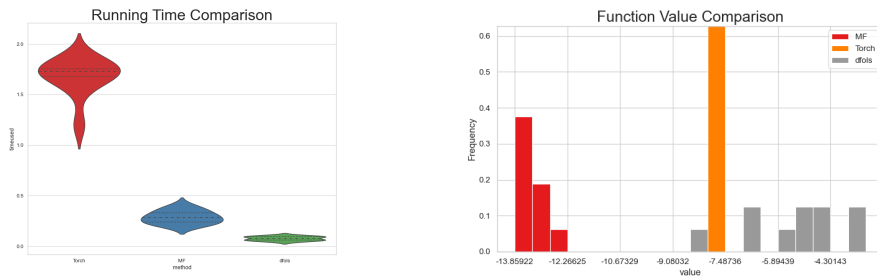


图 4-3: (彩色模式浏览) 三种模型的结果比较。函数值所取的是优化结果与理论最优的差值的 \log 尺度。流形采样得到了最高的精度，同时其运行时间相对较短。

然而，Rosenbrock 函数 4-1 是非负的，也就是说 L^1 范数失去了意义。我们将问题变为

$$\phi(x) + h(F(x)) = \|x\|_2 + \|((x_2 - x_1^2)^2 - (1 - x_1)^2, x_3)\|_1, x \in \mathbb{R}^3 \quad (4-2)$$

这是一个非凸的问题，它在原点附近的行为如图 4-4。

算法在这一测试问题上的表现，请查附录 A.2。

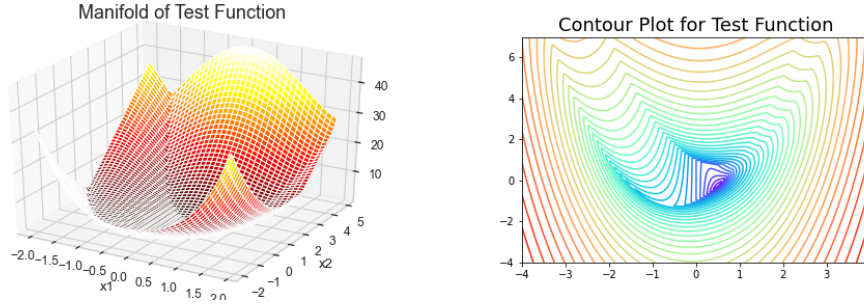


图 4-4: (彩色模式浏览) 测试函数绘图。左图是原点附近的函数曲面，右图为 \log 尺度的函数值等高线图。

4.2 基准测试集

我们采用^[33]作为基准测试集。这一测试集是为了非光滑优化特别设计的，并且可以在任意数据维度下进行测试。测试集的细节，请查附录 A.3。实验结果在表 4-1 中进行了展示。需要强调的是，大多数的测试问题原始维度为 2，流形采样几乎出色地完成了所有该维度的问题。高维情况的探索实验证明了流形采样能够完成高维的任务，是一类强大的算法。

对所有的测试问题，函数有唯一的全局最优，所以我们期待“收敛”与“到达最值”是等价的。然而，在问题“Chained CB3 1”和“Chained LQ”中，这一推测被推翻了。这并不是偶然的数值爆炸。我们进一步检查了“曲率”这一变量，发现这两个问题中，曲率在最值附近很大。我们将原因归结于该量。

我们注意到在实验中，有些发散序列会存在收敛子列收敛到全局最优。关于这一现象我们有两种猜测，一是数值爆炸，二是函数 F 本身的性质。我们相信第二个原因占了主导。例如，在误差比较大（或者不收敛）但误差下极限很小的实验中，比如“Chained CB3 1”“Number of Active Faces”和“Chained Crescent 2”，曲率分别为 $12(n-1)$, ∞ 和 $2(n-1)$ ，均随着 n 的增大而迅速增大，使得曲率对于函数良好性质的限制作废。更多的，从直觉上来讲，即使三个问题均对于 n 为 2 收敛，“Chained CB3 1”和“Chained Crescent 2”有更小的曲率，所以在 n 为 5 时它们仍然有很小的下极限，而“Number of Active Faces”没有。算法表现对维度的敏感性也被曲率所体现。即使 n 为 100 时，“MAXQ”，“MXHILB”，“CB3 2”和“Crescent 1”仍然有很好的收敛，我们推测这正是因为他们的曲率都相对较小。

4.3 非凸性

我们强调，虽然基准测试集中函数有非凸的性质，但使用的 h 均为凸函数，无法体现流形采样的独特优越性。我们接下来尝试优化非凸的 h 。

$$\phi(x) + h(F(x)) = -\exp(-\|x\|^2) + h((x_2 - x_1^2)^2, (1 - x_1)^2), x \in \mathbb{R}^2 \quad (4-3)$$

表 4-1: 基准测试集上的实验结果。我们总共在九个问题上测试了流形采样算法（测试集包含十个问题，然而问题 7 不符合流形采样要求的问题形式 1-1，其非光滑性必须出现在复合的两个阶段）。我们设定自变量的维度为 $n=2, 5, 10$ 和 100 来测试各种维度下的表现。F 的像空间维度是自蕴的，我们在 A.3 中展示。为了方便接下来的讨论，我们也列出在最值点某个邻域内 $h(L_{\nabla F_i}, \dots)$ 的上界，记为“曲率”。所有的实验结果记录的均为单次实验值，但每种情况都至少有五次平行实验保障类似结果的获得。收敛指 x^k 和 $f(x^k)$ 是收敛的，到达最值指的是 x^k 和 $f(x^k)$ 收敛到理论最值附近。误差为 \log 尺度的优化结果与理论最优差值。问题“Chained Milfflin”无法获取理论最优，所以收敛时采用柯希极限，不收敛时留白。误差下极限为潜在的收敛做出了提示。时间单位为秒。留白的位置为无意义。

问题	曲率	n	收敛	到达最值	误差	误差下极限	时间	循环
General MAXQ	2	2	Y	Y	-9	-9	1.71	150
		5	Y	Y	-9	-9	3.59	300
		10	Y	Y	-9	-9	6.57	500
		100	Y	Y	-5	-5	140	3000
General MXHILB	1	2	Y	Y	-6	-6	5.46	300
		5	Y	Y	-5	-5	51	3000
		10	Y	Y	-5	-5	107	3000
		100	Y	Y	-3	-5	5598	3000
Chained LQ	$2(n-1)$	2	Y	Y	-6	-6	2.74	100
		5	Y	Y	-3	-5	112	3000
		10	Y	N		-5	180	3000
		100	N	Y		-5		
Chained CB3 1	$12(n-1)$	2	Y	Y	-8	-8	21.4	1000
		5	Y	N	-1	-4	20.0	250
		10	N	N		-1		
		100	N	N		-1		
Chained CB3 2	12	2	Y	Y	-9	-9	27.3	1450
		5	Y	Y	-3	-4	61.6	3000
		10	Y	Y	-3	-3	80.4	3000
		100	Y	Y	-3	-3	590	3000
Number of Active Faces	∞	2	Y	Y	-9	-9	3.30	130
		5	N	N		-1		
		10	N	N		-1		
		100	N	N		-1		
Chained Milfflin	$2(n-1)$	2	Y	Y	-9	-9	31.3	1600
		5	Y	Y	-6	-6	71.7	3000
		10	N	N				
		100	N	N				
Chained Crescent 1	2	2	Y	Y	-9	-9	3.14	150
		5	Y	Y	-8	-8	22	700
		10	Y	Y	-7	-7	30	1200
		100	Y	Y		-3	90	3000
Chained Crescent 2	$2(n-1)$	2	Y	Y	-7	-7	2.11	150
		5	N	N		-4		
		10	N	N		-1		
		100	N	N		1		

其中

$$h(z) = \sum_{i=1}^m |1 - z_i^2|$$

函数在零点周围的性质如图 4-5。

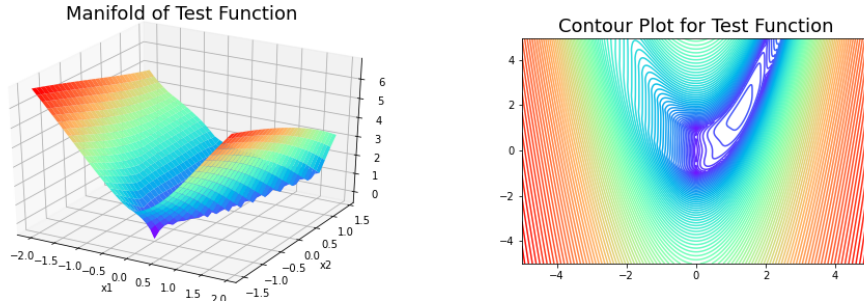


图 4-5: (彩色模式浏览) 测试函数绘图。左图是原点附近的函数曲面，右图为 log 尺度的函数值等高线图。该函数有五个局部极小， $(0,0)(0,1)(0,-1)(2,3)(2,5)$ 。

我们以 $[-3, 3] \times [-3, 3]$ 上的均匀分布生成 1000 个初始点来测试流形采样算法的局部收敛性。在大多数实验中，序列成功的收敛到附近的极小值，我们在图 4-6 中展示这一结果。由于计算资源的限制，在如此大的样本量下我们无法使用小步长搜索。

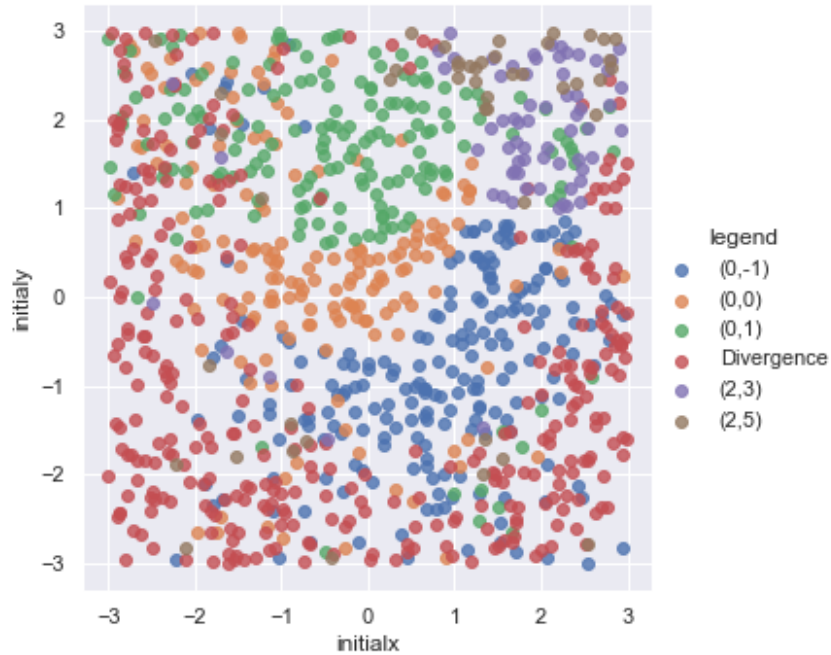


图 4-6: (彩色模式浏览) 初始值点关于收敛结果的分类图。我们注意到仍有许多发散点。这是由于收到计算资源限制，我们只能选取大步长来进行 2-4，导致 x^k 在两个“斜面”来回跳跃。给予足够的时间，这些点仍然会收敛到局部极小。

4.4 实证研究

实证研究的部分，我们采用流形采样算法来优化前向神经网络，从而完成回归任务。我们使用^[34]中提供的糖尿病数据集。该数据集总共有十个基本变量，包括年龄、性别、BMI、平均血压和六个血液指标。响应变量是糖尿病一年后病情发展的量化指标。该数据集包括 442 个样本。我们建立前向神经网络，网络包含一个隐藏层，共 10 个隐藏单元，采用 sigmoid 函数作为激活函数。优化目标是添加了 L^1 惩罚的回归损失

$$\sum (f(x; w) - y)^2 + \lambda \|w\|_1$$

考虑到样本量相对较小，我们仅采用梯度下降作为基准，在 4-7 中记录两种方法的损失函数随时间的下降。虽然结果并不稳定，但流形采样能够更快的收敛到损失相对小的点。

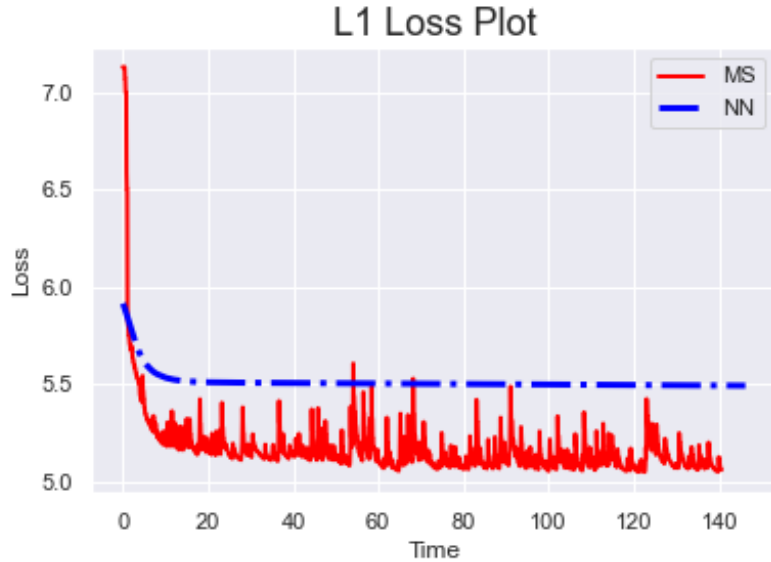


图 4-7: (彩色模式浏览) log 尺度损失随着时间的变化。流形采样迅速的下降至小于梯度下降的点。作为相对原始的基准方法，梯度下降没有采用 batch norm 或者 learning rate decay 等技巧。相应的，流形采样也没有系统化的调参方法。流形采样损失的震荡可以归结于神经网络损失曲面的高曲率。我们相信，如果存在合理的自动调参方法，损失函数可以稳定的趋于极小。

第五章 总结与展望

在本文中，我们拓展已有的流形采样算法至更加广泛的形式，允许 f 的形式为 $f = h \circ F + \phi$ ，其中 ϕ 为光滑函数，使得算法能够对一类常用形式的非光滑、非凸复合函数进行优化。在证明了算法收敛性的同时，我们完成了丰富的数值试验，在标准测试集上检验了算法，讨论了实验结果。最后，我们采用该算法训练浅层神经网络，为未来工作提供可能性。

在后续研究中，我们认为有以下方向值得探索：

- 对“曲率”与算法是否收敛的关系的探索。我们合理的猜测，在不合适的参数设定下，算法仍然有 Minimax 收敛，且收敛速率与曲率有关。
- 自动调参方法。在手动调节参数的过程中，我们能够发现特定参数对于算法行为的影响。比如， η_1 能够控制通过检验的循环质量，从而控制收敛速度与精度的动态平衡。在运行一段时间后，显然我们应当逐渐减小 η_1 ，以获得更精确的收敛。这启发我们寻找与学习率衰减类似的自动调参方法。
- 使用算法进行神经网络训练。非光滑正则项，如 $L^q, 0 < q < 1$ ，被相信能够为神经网络参数提供更好的惩罚。常见的优化方法并不能很好的处理这类问题。

参考文献

- [1] TIBSHIRANI R. Regression Shrinkage and Selection via the lasso[J]. Journal of the Royal Statistical Society. Series B (methodological) 58, 1996.
- [2] ZOU H, TREVOR H. Regularization and Variable Selection via the Elastic Net[J]. Journal of the Royal Statistical Society, Series B. 67, 2005.
- [3] GHARIBI W. A Dual Approach for Solving Nonlinear Infinite-Norm Minimization Problems with Applications in Separable Cases[J]. <https://arxiv.org/pdf/1106.0851.pdf>, 2011.
- [4] C C. Acceleration of the least pth- algorithm for minimax optimization with engineering applications[J]. Math Program, 1979.
- [5] ARASTOO R. Optimal Sparse Output Feedback Controller Design: A Rank Constrained Optimization Approach[J]. <https://arxiv.org/pdf/1412.8236.pdf>, 2014.
- [6] ZHOU. X, YANG. C, ZHAO. H, et al. Low-Rank Modeling and Its Applications in Image Analysis[J]. ACM Computing Surveys 47, 2014.
- [7] DUCHI. J C, RUAN F. Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval[J]. <https://arxiv.org/abs/1705.02356>, 2017.
- [8] KELLER A A. lp-Norm Minimization Method for Solving Nonlinear Systems of Equations[J]. WSEAS TRANSACTIONS on MATHEMATICS, 2014.
- [9] IOFFE A. Necessary and sufficient conditions for a local minimum, 1: A reduction theorem and first-order con- ditions[J]. SIAM J Control Optim 17:245-250, 1979.
- [10] BERKE J V. Second-order necessary and sufficient con- ditions for convex composite NDO.[J]. Math Program 38:287-302, 1987.
- [11] JEYAKUMAR V, YANG X. Convex composite multi- objective nonsmooth programming[J]. Math Program 59, 1993.
- [12] FLETCHER R. A model algorithm for composite nondifferentiable optimization problems, in Nondifferential and Variational Techniques in Optimization[J]. D. C. Sorensen and R. J.-B. Wets, eds., vol. 17 of Mathematical Programming Studies, Springer, 1982.

- [13] FLETCHER R. Second order corrections for non-differentiable optimization[J]. Numerical Analysis, Springer, 1982.
- [14] WOMERSLEY R, FLETCHER R. An algorithm for composite nonsmooth optimization problems[J]. Journal of Optimization Theory and Applications, 1986.
- [15] PARIKH. N, BOYD S. Proximal Algorithms[J]. Foundations and Trends in Optimization, 2013.
- [16] BECK. A, TEBoulLE M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems[J]. SIAM J. Imaging Sci, 2009.
- [17] BOYD. S, PARIKH. N, CHU. E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Found. Trends Mach. Learning, 2011.
- [18] BERTSEKAS D P. Constrained optimization and Lagrange multiplier methods[J]. New York Academic Press, 1982.
- [19] K. N, SEI D, KHONG Z, et al. The proximal augmented Lagrangian method for nonsmooth composite optimization[J]. <https://arxiv.org/pdf/1610.04514.pdf>, 2016.
- [20] MAKELA M. Survey of bundle methods for nonsmooth optimization[J]. Optimization Methods and Software, 2002.
- [21] HARE W L, BAL C S. A redistributed proximal bundle method for nonconvex optimization[J]. SIAM Journal on Optimization, 2010.
- [22] HARE. W L, BAL C S, SOLODOV M. A proximal bundle method for nonsmooth nonconvex functions with inexact information[J]. Computational Optimization and Applications, 63, 2016.
- [23] KARMITSA N, BAGIROV A M. Limited memory discrete gradient bundle method for nonsmooth derivative-free optimization[J]. Optimization, 61, 2012.
- [24] LIUZZI G, LUCIDI S, RINALDI F, et al. Trust-region methods for the derivative-free optimization of nonsmooth black-box functions[J]. SIAM Journal on Optimization 29, 2019.
- [25] WILD S M. Solving derivative-free nonlinear least squares problems with POUNDERS, in Advances and Trends in Optimization with Engineering Applications[J]. SIAM, 2017.

- [26] MENICKELLY M, WILD S M. Derivative-free robust optimization by outer approximations[J]. Mathematical Programming, 179, 2020.
- [27] LARSON. J, MENICKELLY M, WILD S M. Manifold sampling for l1 non-convex optimization[J]. SIAM Journal on Optimization, 26, 2016.
- [28] BURKE. J V, CURTIS. F E, LEWIS. A S, et al. Gradient sampling methods for nonsmooth optimization[J]. Numerical Nonsmooth Optimization, 2020.
- [29] BURKE. J V, LEWIS A S, OVERTON M L. Approximating subdifferentials by ran- dom sampling of gradients[J]. Mathematics of Operations Research, 2002.
- [30] KHAN. K A, LARSON J, WILD S M. Manifold sampling for optimization of nonconvex functions that are piecewise linear compositions of smooth components[J]. SIAM Journal on Optimization 28, 2018.
- [31] LARSON. J, MENICKELLY. M, ZHOU B. Manifold Sampling for Optimizing Nonsmooth Nonconvex Compositions[J]. <https://arxiv.org/abs/2011.01283>, 2020.
- [32] CARTIS C, FIALA J, MARTEAU B, et al. Improving the Flexibility and Robustness of Model-Based Derivative-Free Optimization Solvers[J]. ACM Transactions on Mathematical Software, 2019, 45:3 : 32:1 – 32:41.
- [33] KARMITSA N, MIETTINEN K, MAKELA M. New limited memory bundle method for large-scale nonsmooth optimization[J/OL]. Optimization Methods and Software - OPTIM METHOD SOFTW, 2004, 19 : 673 – 692. <http://dx.doi.org/10.1080/10556780410001689225>.
- [34] EFRON B, HASTIE T, JOHNSTONE I, et al. Least angle regression[J/OL]. The Annals of Statistics, 2004, 32(2) : 407 – 499. <https://doi.org/10.1214/0090536040000000067>.

致 谢

感谢帮助我完成这一毕业课题的顾国勇老师，以及和我讨论的两位同学。感谢南京大学四年来的培养；感谢杜克大学让我度过了精彩的一学期。感谢南大友鸭，夜间一块肉，快活似神仙。感谢皇家马德里俱乐部，我会继续买马塞洛的球衣。感谢宜宾燃面，如果让我选三张照片纪念在南京的日子，我一定和牛杂面自拍一张。不会加磨皮，面会丑。感谢亚瑟士，让我的脚底得到极致享受。讲实话，落地声音有点大。但这又怎么能怪孩子呢？是我的步子太沉重。感谢实况足球，没有社交生活的人终于有事情可做。感谢农夫山泉，让我有水喝。感谢 California Dreaming，在仙二 503 引吭高歌的时光总是从这首歌开始。但其实，我唱得挺难听的。感谢莱特币，暴跌时没有带走我，暴涨时却带上了我。感谢 La La Land，在六个没有能量的日子里拉了我一把。很难想象洛杉矶的爱情故事有这样的作用，只能说形成了相当稳定的看电影-变开心的反射过程。感谢今田美樱，福冈第一美少女的笑永远能让人开心起来，祝新剧大热。感谢在我生命中出现的每个人，生命因你们而美丽。

附录 A 材料

A.1 代码

算法的主要伪代码为 A.1。A.2是 A.1需要调用的函数。

算法 A.1 流形采样

```

1: Set  $\eta_1, \kappa_d \in (0, 1), \kappa_H \geq 0, \eta_2 \in (0, \eta_{\max}), 0 < \gamma_d < 1 \leq \gamma_i$ , and  $\Delta_{\max}, \Delta_{\min} > 0$ 
2: Choose  $\Delta_0$  in  $(0, \Delta_{\max}]$  and initialize  $x^0$ 
3: for  $k = 0 \rightarrow K$  do
4:   Build master model  $m^F$ 
5:   Initialize  $\mathbb{Z}^k$  and Form  $D^k$ 
6:   while true do
7:     Form  $\nabla m^F$ 
8:      $G^k \leftarrow \nabla M(x^k) D^k + (\nabla \phi(x^k), \dots, \nabla \phi(x^k))$ 
9:     Solve 2-2 for  $\lambda^*$ 
10:     $d^k \leftarrow D^k \lambda^*, g^k \leftarrow G^k \lambda^*$ 
11:    if  $\Delta_k < \Delta_{\min}$  then
12:      Break
13:    end if
14:    if  $\Delta_k < \eta_2 \|g^k\|$  then
15:      Solve 2-4 to get  $s^k$ 
16:      Get  $z$  and  $j$  satisfying 2-6 using Algorithm 2
17:      if  $j \in \mathbb{A}(\mathbb{Z}^k)$  and no  $z'$  satisfies 2-7 then
18:         $s^k \leftarrow -\Delta_k \frac{g^k}{\|g^k\|}$ 
19:        Get  $z$  and  $j$  satisfying 2-6 using Algorithm 2
20:      end if
21:      if  $j \in \mathbb{A}(\mathbb{Z}^k)$  then
22:        Calculate  $\rho_k$  using 2-8, Break
23:      else
24:        Update  $m^F$  and  $\mathbb{Z}\mathbb{Z}^k$ , Form  $D^k$ 
25:      end if
26:    else
27:       $\rho_k \leftarrow 0$ , Break
28:    end if
29:  end while
30:  if  $\rho_k > \eta_1 > 0$  then
31:     $x^{k+1} \leftarrow x^k + s^k$ 
32:     $\Delta_{k+1} \leftarrow \min\{\gamma_i \Delta_k, \Delta_{\max}\}$ 
33:  else
34:     $x^{k+1} \leftarrow x^k$ 
35:     $\Delta_{k+1} \leftarrow \gamma_d \Delta_k$ 
36:  end if
37: end for

```

算法 A.2 网格搜索求解 z 与 j

```

1: if  $x^z = x$ ,  $F(x)$  and  $j$  satisfy 2-6 then
2:   return  $F(x)$  and  $j$ 
3: end if
4: if  $x^z = x + s$ ,  $F(x+s)$  and  $j$  satisfy 2-6 then
5:   return  $F(x+s)$  and  $j$ 
6: end if
7: for  $l = 0, \dots$  do
8:   Generate  $\{\frac{2^k-1}{2^l} : k = 1, \dots, 2^{l-1}\}$ 
9:   for  $k = 1, 2, \dots, 2^{l-1}$  do
10:     $\alpha \leftarrow \frac{2^k-1}{2^l}$ 
11:     $z(\alpha) \leftarrow \alpha F(x) + (1-\alpha)F(x+s)$ 
12:    if  $x^z = x + (1-\alpha)s$ ,  $z(\alpha)$  and  $j$  satisfy 2-6 then
13:      return  $z(\alpha)$  and  $j$ 
14:    end if
15:  end for
16: end for
17: end for

```

A.2 测试结果

我们采取与之前相同的测试方法，结果展示在图 A-1与 A-2中。

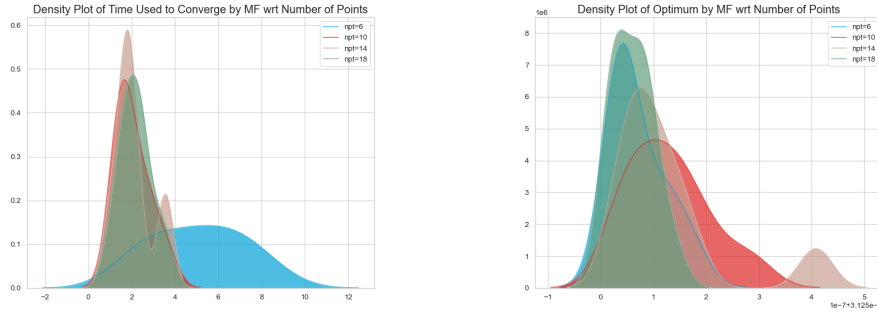


图 A-1: (彩色模式浏览) 时间与函数值关于不同样本点个数的分布。在样本数为 14 时有一次实验遭遇了数值爆炸，所以该类别表现很差。

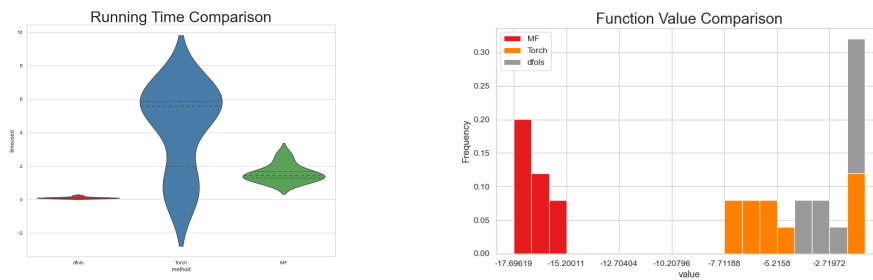


图 A-2: (彩色模式浏览) 三种模型的结果比较。函数值所取的是优化结果与理论最优的差值的 \log 尺度。流形采样得到了最高的精度，同时其运行时间相对较短。

A.3 基准数据集

我们提供^[33] 的细节。

1. Generalization of MAXQ

$$\begin{aligned} f(\mathbf{x}) &= \max_{1 \leq i \leq n} x_i^2 \\ x_i^{(1)} &= i, \quad \text{for } i = 1, \dots, n/2 \text{ and} \\ x_i^{(1)} &= -i, \quad \text{for } i = n/2 + 1, \dots, n. \end{aligned}$$

2. Generalization of MXHILB

$$\begin{aligned} f(\mathbf{x}) &= \max_{1 \leq i \leq n} \left| \sum_{j=1}^n \frac{x_j}{i+j-1} \right| \\ x_i^{(1)} &= 1, \quad \text{for all } i = 1, \dots, n \end{aligned}$$

3. Chained LQ

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{n-1} \max \{ -x_i - x_{i+1}, -x_i - x_{i+1} + (x_i^2 + x_{i+1}^2 - 1) \} \\ x_i^{(1)} &= -0.5, \quad \text{for all } i = 1, \dots, n \end{aligned}$$

4. Chained CB3 1

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{n-1} \max \{ x_i^4 + x_{i+1}^2, (2 - x_i)^2 + (2 - x_{i+1})^2, 2e^{-x_i + x_{i+1}} \} \\ x_i^{(1)} &= 2, \quad \text{for all } i = 1, \dots, n \end{aligned}$$

5. Chained CB3 2

$$\begin{aligned} f(\mathbf{x}) &= \max \left\{ \sum_{i=1}^{n-1} (x_i^4 + x_{i+1}^2), \sum_{i=1}^{n-1} ((2 - x_i)^2 + (2 - x_{i+1})^2) \right. \\ &\quad \left. \sum_{i=1}^{n-1} (2e^{-x_i + x_{i+1}}) \right\} \\ x_i^{(1)} &= 2, \quad \text{for all } i = 1, \dots, n \end{aligned}$$

6. Number of Active Faces

$$\begin{aligned} f(\mathbf{x}) &= \max_{1 \leq i \leq n} \{ g(-\sum_{i=1}^n x_i), g(x_i) \} \\ \text{where } g(y) &= \ln(|y| + 1) \\ x_i^{(1)} &= 1, \quad \text{for all } i = 1, \dots, n \end{aligned}$$

7. Chained Mifflin

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{n-1} (-x_i + 2(x_i^2 + x_{i+1}^2 - 1) + 1.75|x_i^2 + x_{i+1}^2 - 1|) \\ x_i^{(1)} &= -1, \quad \text{for all } i = 1, \dots, n \end{aligned}$$

表 A-1: 问题集的性质

序号	最小值	凸性	非光滑函数 (h)	L_h	m
1	0	Y	$\max(\cdot)$	1	n
2	0	Y	$\max(\cdot)$		n
3	$-(n-1)\sqrt{2}$	Y	$\sum \max(\cdot)$		$2(n-1)$
4	$2(n-1)$	Y	$\sum \max(\cdot)$		$3(n-1)$
5	$2(n-1)$	Y	$\max(\cdot)$		3
6	0	N	$\max(\cdot)$		n+1
7	Varies*	N	$ \cdot $		n-1
8	0	N	$\max(\cdot)$		2
9	0	N	$\sum \max(\cdot)$		$2(n-1)$

* n=10 时最小值约-6.51

n=100 时约为-70.15, n=1000 时约为-706.55

8. Chained Crescent 1

$$f(\mathbf{x}) = \max \left\{ \sum_{i=1}^{n-1} \left(x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1 \right), \sum_{i=1}^{n-1} \left(-x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1 \right) \right\}$$

$$x_i^{(1)} = -1.5, \text{ when } \text{mod}(i, 2) = 1, (i = 1, \dots, n) \text{ and}$$

$$x_i^{(1)} = 2.0, \text{ when } \text{mod}(i, 2) = 0, (i = 1, \dots, n).$$

9. Chained Crescent 2

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \max \left\{ x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1, -x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1 \right\}$$

$$x_i^{(1)} = -1.5, \text{ when } \text{mod}(i, 2) = 1, (i = 1, \dots, n) \text{ and}$$

$$x_i^{(1)} = 2.0, \text{ when } \text{mod}(i, 2) = 0, (i = 1, \dots, n).$$

注意到问题 6 其实是不符合要求的, 因为其不满足全线性 2.8, 但我们仍然保留。表 A-1 给出了问题的性质。