中国科学 : 数学 2017 年 第 47 卷 第 10 期 : 1～24
SCIENTIA SINICA Mathematica

《中国科学》杂志社
SCIENCE CHINA PRESS

论 文

基于近似一阶信息的加速的 bundle level 算法

献给李大潜教授 80 华诞

陈韵梅*, 张维

Department of Mathematics, University of Florida, Gainesville, FL 32611, USA
E-mail: yun@ufl.edu, weizhang657@ufl.edu

收稿日期: 2016-10-02; 接受日期: 2017-06-23; 网络出版日期: 2017-07-XX; * 通信作者
美国国家科学基金 (批准号: DMS-1319050 和 DMS-1719932) 资助项目

摘要 本文提出了四种加速的 BL (bundle level) 算法 (IFAPL、IFAPLS、IFUSL 和 IFUSLS) 来分别求解凸光滑函数、强凸光滑函数的极小值问题和一类鞍点 (saddle-point) 问题. 这些算法可以运用目标函数的近似的一阶信息来得到上述几类问题的近似解. 本文重点研究了在一阶信息误差上界可自由选取和给定不变的两种情形下, 所提出的算法中近似解能达到的最佳精度以及相应的迭代复杂度分析.

关键词 加速算法 bundle level 算法 光滑优化

MSC (2010) 主题分类 XXXX

1 引言

给定凸函数 $f : \mathbb{R}^n \to \mathbb{R}$, 本文考虑以下优化问题:

$$f^* := \min_{x\in X} f(x), \tag{1.1}$$

其中 $X \subseteq \mathbb{R}^n$ 为 n 维 Euclid 空间上的非空凸紧集, $f : X \to \mathbb{R}$ 是一个凸闭函数. 假设上述优化问题的解集 X^* 非空, 同时对任意 $x \in X$, 存在一个模型可以给出目标函数 f 在此点的函数值 $f(x)$ 和次梯度 $f'(x) \in \partial f(x)$, 且存在常数 $M > 0$ 和 $\rho \in [0, 1]$ 使得 f 满足以下不等式:

$$f(y) - f(x) - \langle f'(x), y - x\rangle \leqslant \frac{M}{1+\rho}\|y - x\|^{1+\rho}, \quad \forall\, x, y \in X, \tag{1.2}$$

其中 $\|\cdot\|$ 是定义在 $\mathbb{R}^n$ 上的范数, $\langle\cdot,\cdot\rangle$ 是相应的内积. 此类函数包含非光滑函数 ($\rho = 0$)、光滑函数 ($\rho = 1$) 和半光滑函数 ($0 < \rho < 1$). 文献 [1–4] 证明了对任意 $\epsilon > 0$, 任何一阶算法为找到 (1.1) 的一个 ϵ- 解, 即找到一个点 $\hat{x}$ 使得 $f(\hat{x}) - f^* \leqslant \epsilon$, 所需要调用 f 一阶模型的次数 (即计算 f 次梯度的次数) 无法小于 $\mathcal{O}(1/\epsilon^{\frac{2}{1+3\rho}})$.

本文主要讨论 bundle level (BL) 类算法, 该类算法与通常的 (次) 梯度下降类 (sub/gradient descent) 算法有显著区别. 此类算法使用目标函数的一阶信息生成一个分段线性的切平面模型来近似原

英文引用格式: Chen Y M, Zhang W. Accelerated bundle level methods with inexact oracle (in Chinese). Sci Sin Math, 2017, 47: 1–24, doi: 10.1360/N012016-00173

目标函数, 并利用此切平面模型来产生一系列关于极小值 f^* 的上界和下界. BL 类算法就是通过不断缩小上下界的差值来找到原问题的一个近似解. 第一个 BL 类算法是在文献 [5] 中提出的, 随后在文献 [6–10] 中进一步发展, 其中文献 [9] 最先提出了加速的 BL 算法. 此文中提出了三种算法来求解光滑函数、非光滑函数和半光滑函数的极小值凸优化问题以及一类鞍点问题, 这些算法均有最优的迭代复杂度. 与梯度下降类算法相比, 加速 BL 算法[9,10] 具有不需要任何与目标函数相关的参数输入, 也不涉及步长的选择问题, 更适用于大规模数据计算等优势.

然而在很多问题里, 函数在某点的精确一阶信息、次梯度和函数值, 可能很难甚至无法计算. 在这种情形下, 设计能有效利用近似的一阶信息求解目标函数, 并具有理论上最优迭代复杂度的近似算法, 至关重要. 目前, 关于不加速近似 BL 类算法已经有了很多研究, 但都集中在解决非光滑问题上, 在一定条件下该类算法对非光滑问题能保证 $\mathcal{O}(1/\epsilon^2)$ 的迭代复杂度. 本文致力于发展加速的近似 BL 算法, 来解决光滑问题里使用不精确的一阶信息的问题, 并着重讨论近似解的精度与近似一阶信息精度的依赖关系, 以及这些算法的迭代复杂度.

1.1　背景介绍

首先简要回顾 BL 类算法和近似 BL 类算法的发展历史, 并介绍其中的主要思想和收敛性结果.

1.1.1　BL 类算法

BL 类算法起源于 1960 年 Kelly 的 cutting plane 方法[11], 其基本想法是, 利用目标函数 f 的凸性, 即对于任意点 x_i, 有 $f(x) \geqslant f(x_i) + \langle f'(x_i), x - x_i\rangle$, 产生一系列切平面模型 $m_k(x)$ 用来逼近原函数 f. 具体来讲, 给定 $x_1, x_2, \dots, x_k$, 切平面模型 $m_k(x)$ 定义如下:

$$m_k(x) := \max\{h(x_i, x), 1 \leqslant i \leqslant k\}, \tag{1.3}$$

其中

$$h(z, x) := f(z) + \langle f'(z), x - z\rangle. \tag{1.4}$$

Kelly 的 cutting plane 方法通过求解当前切平面模型 $m_k(x)$ 的最小值点产生下一个迭代点, 即

$$x_{k+1} \in \operatorname*{Argmin}_{x\in X} m_k(x). \tag{1.5}$$

Kelly 的 cutting plane 方法形式简单且保证收敛, 然而由于线性模型 $m_k(x)$ 的不稳定及对误差较为敏感, 导致该算法在实践中收敛很慢. 此后, bundle 算法[12] 为了改进线性模型的不稳定性, 通过加入一个二次项来改进下一个点的选取:

$$x_{k+1} \in \operatorname*{argmin}_{x\in X} m_k(x) + \frac{\rho_k}{2}\|x - x^+\|^2, \tag{1.6}$$

其中 $\rho_k > 0$, x^+ 是从当前已产生点列 $\{x_1, x_2 \dots, x_k\}$ 中选取的 prox-center. 进而, 1995 年 Lemaréchal 等[5] 引进了水平集 (level set) 的思想并提出了经典的 BL 算法. 该算法通过将当前的点投影到所选取的水平集中来产生下一个迭代点. 由于该方法奠定了 BL 类方法的基本框架, 这里介绍一下其详细步骤. 该方法在每次迭代中首先计算当前的关于 f^* 的上界 $\overline{f}_k$ 和下界 $\underline{f}_k$, 然后定义此时的 level l_k 和水平集 X_k, 最后将 x_k 投影到当前水平集来产生下一个点, 具体如下: 给定 $x_1, x_2, \dots, x_k$,

(i) 令 $\overline{f}_k := \min\{f(x_i), 1 \leqslant i \leqslant k\}$, 并计算

$$\underline{f}_k = \min_{x\in X} m_k(x); \tag{1.7}$$

(ii) 取 $l_k = \beta \underline{f}_k + (1-\beta)\overline{f}_k$, 其中 $\beta \in (0,1)$;

(iii) 定义 $X_k := \{x \in X : m_k(x) \leqslant l_k\}$ 并投影得到下一个点

$$x_{k+1} = \operatorname*{argmin}_{x\in X_k} \|x - x_k\|^2. \tag{1.8}$$

绝大部分已有的 BL 类算法均是用来解决凸紧集上的非光滑优化问题, 并可以得到最优的迭代复杂度 $O(1/\epsilon^2)$. 但是在实际计算中, 随着迭代次数 k 的增长, X_k 中约束条件的数量也线性地增多, 这会导致两个子问题 (1.7) 和 (1.8) 越来越难解. 随后, 文献 [8] 提出了 NERML (non-Euclidean restricted memory level) 算法, 对经典的 BL 算法作出了重要改进. NERML 算法首先可以控制切平面模型 $m_k(x)$ 中所使用的切平面数量, 使得存储 $m_k(x)$ 所需的内存不会随着迭代次数线性递增, 因而, 两个子问题 (1.7) 和 (1.8) 中的约束条件不会太多. 其次其将子问题 (1.8) 中的 $\|\cdot\|^2$ 函数扩展到任意迫近函数 (prox-function), 且可以选取相应的迫近函数来达到利用 X 的几何特征的目的.

相比较于其他一阶算法, BL 类算法有一个显著的优点. 大多数一阶算法在每步迭代时都需要确定该步的步长, 而步长的选取依赖于目标函数的 Lipschitz 常数值, 在高维的问题中, 函数的 Lipschitz 常数值往往难以精确计算. 不恰当的估计往往会导致步长过小, 函数值收敛速度较慢, 或者步长过大无法保证收敛. 在 BL 类算法中, 并不涉及步长选取的问题, 因此不要求输入 Lipschitz 常数值, 且在实际问题中, BL 类算法往往在求解具有较大的 Lipschitz 常数的目标函数的优化问题中有优势.

为了进一步将 BL 类算法拓展到光滑问题和半光滑问题, 文献 [9] 首次结合了 Nesterov 对于光滑函数的加速方法[1], 提出了两个加速的 BL 类算法: ABL (accelerated bundle-level) 和 APL (accelerated prox-level) 算法, 其中 ABL 算法可以看成是经典的 BL 方法的加速版本, 而 APL 则是结合了 Nesterov 的加速技巧和 NERML 算法的优点. ABL 和 APL 的主要特征是均使用三个点列来分别更新 f 的切平面模型 (进而更新 f^* 下界)、当前 prox-center 和 f^* 的上界以达到对光滑函数和半光滑函数的加速效果. 这两个算法对 (1.2) 中光滑问题、非光滑问题和半光滑问题均得到最优迭代复杂度 $\mathcal{O}(1/\epsilon^{\frac{2}{1+3\rho}})$, 其中 ρ 为 (1.2) 中目标函数的光滑参数.

然而, ABL 和 APL 算法均涉及求解两个子问题: 一个线性优化类似于 (1.7) 和一个二次优化 (1.8). 由于线性子问题 (1.7) 需要确保存在有限解来确定 f^* 的下界, 因此 X 必须有界. 随后, 文献 [10] 提出的 FAPL (fast APL) 方法在保持与 APL 算法同样最优迭代复杂度的情形下对上述问题作了改进. 首先, FAPL 算法去掉了类似于 (1.7) 的线性优化子问题, 所以 X 可以是无界的. 其次, 当 X 是一个 Euclid 球时, FAPL 算法可以得到二次优化问题 (1.8) 的精确解, 并通过恰当地选取 (1.8) 中的迫近函数, 将更新 prox-center 和 f^* 的下界合并在一个子问题中. 然后利用一个不断扩大球半径的策略将 FAPL 算法的适用范围扩展到了 $\mathbb{R}^n$ 上的无限制的优化问题, 且证明了对无限制优化问题的最优迭代复杂度.

1.1.2 近似 BL 类算法

近似 BL 类算法是假设对于目标函数, 存在一个近似的一阶模型, 它可以给出目标函数在限制集上任一点的近似的一阶信息, 近似算法可以利用这些不精确的一阶信息得到原优化问题一定精度的近似解. 此类算法在很多实际问题中有应用, 特别是在一些大规模的问题中, 精确的一阶信息可能无法

得到或者其计算代价会很高. 由于 BL 类算法的一些特征适应于求解大规模的优化问题, 因此近些年有很多关于近似 BL 类算法的研究, 如文献 [13–25]. 然而到目前为止, 所有这些研究都是针对于求解非光滑函数的优化问题, 未涉及求解光滑问题和半光滑问题, 并且所使用的是非加速的 BL 类算法. 这些研究可以分类为近似 PB (proximal bundle) 算法和近似 BL 算法.

文献 [26,27] 率先分别将 PB 和 BL 算法拓展到使用近似的一阶信息来求解非光滑函数的优化问题, 其主要区别是 PB 算法用 (1.6) 更新下一个迭代点, 而 BL 算法使用 (1.8). 考虑 (1.1), 其中 f 为非光滑函数 (即 $\rho = 0$), 假设存在一个给定的近似一阶模型, 对任意 $u \in X$, 可以给出 f 在此点的满足以下条件的近似一阶信息:

$$\begin{cases} f_u = f(u) - \eta_u, \\ g_u \in \mathbb{R}^n \text{ 使得 } f(x) \geqslant f_u + \langle g_u, x-u \rangle - \eta_u^g, \quad \eta_u^g \geqslant 0, \quad \forall\, x \in X, \end{cases} \tag{1.9}$$

其中 f_u 和 g_u 分别为 f 在 u 点的近似函数值和近似次梯度. 若 $\eta_u^g = 0$, 则此类模型称为下确界模型; 若 $\eta_u^g > 0$, 则称为上确界模型; 精确模型等价于 $\eta_u^g = \eta_u = 0$. 对于下确界模型, 对任意 $u \in X$, 在点 u 的近似切平面 $p(u,x) := f_u + \langle g_u, x-u \rangle$ 完全在 $f(x)$ 的曲面之下. 但是对于上确界模型, $f(x)$ 的曲线可能会与某些点的近似切平面相交.

文献 [27] 假设 $\eta_{x_j}^g = 0$, $\eta_{x_j} \leqslant \kappa \Delta_j$, $\forall\, j \geqslant 1$, 其中 $\kappa > 0$, x_j 是第 j 次迭代得到的点, Δ_j 为当前上下界差值, 该近似 BL 算法可以保证收敛到原问题的一个 ϵ- 解且具有 $\mathcal{O}(M/\epsilon^2)$ 的最优迭代复杂度. 文献 [13,20,28] 则证明了近似 PB 算法在 η_u 有上界 η, 即 $\eta_u \leqslant \eta, \forall\, u \in X$, 或者渐近趋于 0 时, 相应可以得到 (1.1) 的 η- 解或者 ϵ- 解.

文献 [19] 引进了下降目标 γ_x 的概念, 其假设

$$f(x) \in [f_x, f_x + \eta_{\gamma_x}], \quad \eta_{\gamma_x} \leqslant \epsilon_{\gamma_x}, \quad \text{当 } f_x \leqslant \gamma_x. \tag{1.10}$$

这意味着只有在函数值 f_x 达到其下降目标 γ_x 时, 才要求一阶模型误差不超过 ϵ_{γ_x}, 而在一些比较差的点上, 则对其精度要求可以更松. 在此概念下, 文献 [19] 将近似 BL 类算法按其所使用的近似一阶模型进行了分类: (i) 精确模型: $\epsilon_{\gamma_x} \equiv 0$ 且 $\gamma_x \equiv +\infty$; (ii) 半精确模型: $\epsilon_{\gamma_x} \equiv 0$, $\gamma_x < +\infty$ (参见文献 [19,20]); (iii) 不精确模型: $\gamma_x \equiv +\infty$ 但 $\epsilon_{\gamma_x} > 0$ (参见文献 [14,19,26,29,30]); (iv) 渐近精确模型: $\gamma_x \equiv +\infty$ 且 $\epsilon_{\gamma_x} \to 0$ (参见文献 [19,21,27,31]); (v) 半渐近精确模型: $\gamma_x < +\infty$ 且 $\epsilon_{\gamma_x} \to 0$ (参见文献 [19]), 其中文献 [19] 通过选取适当的 γ_x 和 ϵ_{γ_x}, 统一证明了近似 BL 算法在上述 (i)–(v) 情形下对非光滑优化问题均可以达到最优迭代复杂度.

近似 PB 算法也已经被用来减少很多优化问题的计算时间. 文献 [21] 中考虑的目标函数是一系列子函数的和, 通过每步仅仅计算其中一小部分子函数的一阶信息来迭代产生下一个点, 进而减少了计算时间, 且能保证在误差有界或者渐近趋于零时收敛到一定精度的解, 但没有提供迭代复杂度证明. 文献 [32] 中使用两个不同的近似一阶模型, 一个误差是有界的且小于 η, 一个误差是无界的, 该算法利用误差无界的一阶模型来构建切平面模型, 而使用误差有界的一阶模型来确定下一个迭代点. 该算法可以证明收敛到原问题的一个 η- 解, 且被应用到一类 two-stage stochastic 线性问题上, 参见文献 [14,19,23,31,32].

然而所有上述的近似 PB 算法和近似 BL 算法都集中在解决非光滑优化问题, 没有涉及用加速 BL 类算法来求解光滑函数的优化问题以得到更好的迭代复杂度. 实际上对于光滑问题, 近似的加速梯度下降类算法已经有了很多研究, 参见文献 [4,33–37]. 由于加速一阶算法无法避免受到误差累积效应的影响, 因此其与非加速近似算法有显著区别.

1.2 文章结构安排

第 2 节先给出解决普通凸光滑函数的优化问题的近似 IFAPL (fast accelerated prox-level) 算法, 然后讨论在目标函数为强凸函数时如何改进该算法, 从而提出求解强凸函数的近似 FAPL 算法, 即 IFAPLS. 第 3 节结合文献 [38] 中的光滑化方法, 将 IFAPL 和 IFAPLS 算法进一步拓展到一类鞍点问题上, 并提出近似 fast uniform smoothing level (IFUSL) 算法和对强凸函数的 IFUSLS 算法. 对这 4 种种算法, 在每节的最后都将分别讨论在目标函数的一阶模型误差可由用户选取和给定不变时, 该算法所能达到的最佳精度及其迭代复杂度.

2 加速近似 BL 法

本节先提出 IFAPL 算法用于解决 Euclid 球上 ($X = B(\bar{x}, R)$) 的光滑优化问题, 然后进一步扩展该算法来求解 $\mathbb{R}^n$ 上的强凸光滑函数的极小值问题, 并提出 IFAPLS 算法.

IFAPL 和 IFAPLS 算法均使用目标函数近似的一阶信息, 假设目标函数 (1.2) 的近似一阶信息满足下面的条件, 即存在一个 (L, δ)- 模型, 使得对于任意点 $x \in X$, f 在这点的近似一阶信息 $(f_\delta(x), g_\delta(x))$ 满足以下条件:

$$0 \leqslant f(y) - (f_\delta(x) + \langle g_\delta(x), y - x\rangle) \leqslant \frac{L}{2}\|y - x\|^2 + \delta, \quad \forall\, x, y \in X. \tag{2.1}$$

我们将分别考虑在 δ 可以由用户自由选取和 δ 给定不变两种情形下, IFAPL 和 IFAPLS 算法所能达到的最佳精度以及相应的迭代复杂度.

2.1 IFAPL 算法

本节考虑任意 Euclid 球上的光滑优化问题

$$f^*_{\bar{x},R} := \min_{x \in B(\bar{x},R)} f(x), \tag{2.2}$$

其中

$$B(\bar{x}, R) := \{x \in \mathbb{R}^n : \|x - \bar{x}\| \leqslant R\},$$

f 为光滑函数且 Lipchitz 常数为 L, 即 f 满足 (1.2), 其中 $\rho = 1$, $M = L$. f 在任意点 x 的近似一阶信息 $(f_\delta(x), g_\delta(x))$ 满足条件 (2.1).

IFAPL 算法的基本思想是扩展文献 [10] 中所提出的 FAPL 算法使其能有效利用目标函数的近似一阶信息来求解 (2.2). 与 FAPL 算法类似, IFAPL 算法利用 f 的不精确一阶信息来构建其切平面模型, 产生一系列 f^* 的上界和下界, 并不断缩小上界和下界之间的差值直到其差值小于或等于所要求的目标精度. 具体来讲, IFAPL 集法具有外迭代 - 内迭代的结构, 每次外迭代都检查当前上下界的差值是否大于给定的目标精度. 如果大于给定的精度, 则调用其内迭代子程序 $\mathcal{G}_{\mathrm{IFAPL}}$ 来缩小差值; 若差值已经小于或等于目标精度, 则 (2.2) 的一个符合目标精度的近似解已经找到, 输出该近似解并终止 IFAPL 算法. 其中内迭代子程序包含很多步骤, 每次调用内迭代子程序都可以将上下界的差值缩小一个固定的比例. 下面先给出 IFAPL 的内迭代子程序.

过程 1 IFAPL 算法内迭代子程序: $(x^+, \mathrm{ub}^+, \mathrm{lb}^+) = \mathcal{G}_{\mathrm{IFAPL}}(\hat{x}, \mathrm{ub}, \mathrm{lb}, R, \overline{x}, \beta, \theta, \delta)$

0: 令 $k=1$, $\overline{f}_0 = \mathrm{ub}$, $l = \beta \cdot \mathrm{lb} + (1-\beta) \cdot \mathrm{ub}$, $Q_0 = \mathbb{R}^n$, 取 $x_0^u = \hat{x}$, $x_0 = \overline{x}$.

1: 更新切平面模型:

$$x_k^l = (1-\alpha_k) x_{k-1}^u + \alpha_k x_{k-1}, \tag{2.3}$$

$$h_\delta(x_k^l, x) = f_\delta(x_k^l) + \langle g_\delta(x_k^l), x - x_k^l \rangle, \tag{2.4}$$

$$\underline{Q}_k := \{x \in Q_{k-1} : h_\delta(x_k^l, x) \leqslant l\}. \tag{2.5}$$

2: 更新 prox-center 和下界:

$$x_k = \underset{x \in \underline{Q}_k}{\operatorname{argmin}} \left\{ d(x) := \frac{1}{2}\|x - \overline{x}\|^2 \right\}. \tag{2.6}$$

若 $\underline{Q}_k = \emptyset$ 或 $\|x_k - \overline{x}\| > R$, 则终止迭代并输出 $x^+ = x_{k-1}^u$, $\mathrm{ub}^+ = \overline{f}_{k-1}$, $\mathrm{lb}^+ = l$.

3: 更新上界: 设

$$\tilde{x}_k^u = (1-\alpha_k) x_{k-1}^u + \alpha_k x_k, \tag{2.7}$$

$$x_k^u = \begin{cases} \tilde{x}_k^u, & \text{若 } f_\delta(\tilde{x}_k^u) + \delta < \overline{f}_{k-1}, \\ x_{k-1}^u, & \text{其他}, \end{cases} \tag{2.8}$$

令 $\overline{f}_k = f_\delta(x_k^u) + \delta$. 若 $\overline{f}_k \leqslant l + \theta(\overline{f}_0 - l)$, 则终止迭代并输出 $x^+ = x_k^u$, $\mathrm{ub}^+ = \overline{f}_k$, $\mathrm{lb}^+ = \mathrm{lb}$.

4: 选取任意多面集 Q_k 满足 $\underline{Q}_k \subseteq Q_k \subseteq \overline{Q}_k$, 其中

$$\overline{Q}_k := \{x \in \mathbb{R}^n : \langle x_k - \overline{x}, x - x_k \rangle \geqslant 0\}. \tag{2.9}$$

令 $k = k+1$, 转至第 1 步.

为了保证 $\mathcal{G}_{\mathrm{IFAPL}}$ 能终止迭代和 IFAPL 算法的最优迭代复杂度, 步长 $\{\alpha_k\}$ 的选取需满足下面的限制条件:

$$\alpha_1 = 1, \quad 0 < \alpha_k \leqslant 1, \quad \alpha_k \leqslant \frac{C_1}{k}, \quad \frac{1-\alpha_{k+1}}{\alpha_{k+1}^2} \leqslant \frac{1}{\alpha_k^2} \quad \text{且} \quad \alpha_k^2 \sum_{i=1}^{k} \frac{1}{\alpha_i^2} \leqslant C_2 k, \quad \forall\, k \geqslant 1, \tag{2.10}$$

其中常数 $C_1, C_2 > 0$. 在此, 我们提供两个具体 $\{\alpha_k\}$ 选取的例子, 可用类似于文献 [10] 中的方法来验证这两个例子均满足 (2.10), 这里不再给出证明.

(1) 令 $\alpha_k = 2/(k+1)$, $k = 1, 2, \ldots$, 取 $C_1 = 2$, $C_2 = 2$, 条件 (2.10) 满足.

(2) 若 $\{\alpha_k\}$ 由以下递归关系确定:

$$\alpha_1 = 1, \quad \alpha_{k+1}^2 = (1-\alpha_{k+1})\alpha_k^2, \quad \forall\, k \geqslant 1, \tag{2.11}$$

则取 $C_1 = 2$, $C_2 = 2$, 条件 (2.10) 满足.

在讨论 $\mathcal{G}_{\mathrm{IFAPL}}$ 收敛性之前, 我们需要以下两个引理, 这两个引理与文献 [10, 引理 3.2–3.4] 十分相似, 其基本思想是一致的. 为了文章的完整性, 我们再次给出这两个引理及其证明.

引理 1 令 $\{x_k\}$ 为 $\mathcal{G}_{\mathrm{IFAPL}}$ 所产生的点列, 对任意 $K \geqslant 1$, 若 $\mathcal{G}_{\mathrm{IFAPL}}$ 没有在 K 次迭代时终止,

则以下结论成立:

$$\frac{1}{2}\sum_{k=1}^{K}\|x_k-x_{k-1}\|^2\leqslant d(x_K)=\frac{1}{2}\|x_K-\overline{x}\|^2\leqslant\frac{1}{2}R^2. \tag{2.12}$$

证明 由 (2.5)、(2.6) 和 (2.9) 容易得到 $x_k\in\underline{Q}_k\subseteq Q_{k-1}\subseteq\overline{Q}_{k-1}, \forall\, 2\leqslant k\leqslant K$, 同时由 $x_1\in Q_0$, $x_0=\mathrm{argmin}_{x\in Q_0}d(x)$, 有

$$\langle\nabla d(x_{k-1}),x_k-x_{k-1}\rangle\geqslant 0,\quad \forall\, 1\leqslant k\leqslant K. \tag{2.13}$$

利用 $d(x)$ 的强凸性质, 可以得到

$$d(x_k)\geqslant d(x_{k-1})+\langle\nabla d(x_{k-1}),x_k-x_{k-1}\rangle+\frac{1}{2}\|x_k-x_{k-1}\|^2. \tag{2.14}$$

结合以上两个不等式便有 $\frac{1}{2}\|x_k-x_{k-1}\|^2\leqslant d(x_k)-d(x_{k-1}), \forall\, 1\leqslant k\leqslant K$, 不等式两边分别对 k 从 1 到 K 作和便得到 (2.12). □

引理 2 记 $\mathcal{E}_f(l):=\{x\in B(\overline{x},R):f(x)\leqslant l\}$, $\mathcal{G}_{\mathrm{IFAPL}}$ 中以下结论成立:

(1) 若 $\mathcal{E}_f(l)\neq\emptyset$, 则对任意 $k\geqslant 1$, 有 $\mathcal{E}_f(l)\subseteq\underline{Q}_k\subseteq Q_k\subseteq\overline{Q}_k$.

(2) 若 $\underline{Q}_k\neq\emptyset$, 则第 2 步中的子问题 (2.6) 有唯一解; 且如果 $\mathcal{G}_{\mathrm{IFAPL}}$ 在第 2 步满足条件终止循环, 那么有 $l\leqslant f^*_{\overline{x},R}$.

证明 只需注意到 $h_\delta(x_k^l,x)\leqslant f(x)$ 以及 $\mathcal{E}_f(l)\subseteq\{h_\delta(x_k^l,x)\leqslant l\}$, 引理 2 的其他部分的证明与文献 [10, 引理 3.2] 证明完全一致. □

定理 1 在 $\mathcal{G}_{\mathrm{IFAPL}}$ 中, 如果步长 $\{\alpha_k\}$ 满足条件 (2.10), 那么对任意 $K\geqslant 1$, 若 $\mathcal{G}_{\mathrm{IFAPL}}$ 在第 K 次迭代时没有终止, 则有以下结论:

$$\overline{f}_K-l\leqslant\frac{C_1^2LR^2}{2K^2}+2C_2K\delta. \tag{2.15}$$

证明 由 $\overline{f}_k$ 的定义及 (2.1) 和 (2.7), 对任意 $k\geqslant 1$, 有

$$\begin{aligned}\overline{f}_k&=f_\delta(x_k^u)+\delta\leqslant f_\delta(\tilde{x}_k^u)+\delta\leqslant f(\tilde{x}_k^u)+\delta\\&\leqslant f_\delta(x_k^l)+\langle g_\delta(x_k^l),\tilde{x}_k^u-x_k^l\rangle+\frac{L}{2}\|\tilde{x}_k^u-x_k^l\|^2+2\delta\\&=(1-\alpha_k)[f_\delta(x_k^l)+\langle g_\delta(x_k^l),x_{k-1}^u-x_k^l\rangle]+\alpha_k[f_\delta(x_k^l)+\langle g_\delta(x_k^l),x_k-x_k^l\rangle]\\&\quad+\frac{L\alpha_k^2}{2}\|x_k-x_{k-1}\|^2+2\delta\\&\leqslant(1-\alpha_k)f(x_{k-1}^u)+\alpha_kl+\frac{L\alpha_k^2}{2}\|x_k-x_{k-1}\|^2+2\delta\\&\leqslant(1-\alpha_k)(f_\delta(x_{k-1}^u)+\delta)+\alpha_kl+\frac{L\alpha_k^2}{2}\|x_k-x_{k-1}\|^2+2\delta\\&=(1-\alpha_k)\overline{f}_{k-1}+\alpha_kl+\frac{L\alpha_k^2}{2}\|x_k-x_{k-1}\|^2+2\delta,\end{aligned} \tag{2.16}$$

其中 (2.16) 由 f 的凸性及 (2.5) 和 (2.6) 得到. 将上述不等式两边同时减去 l, 即有

$$\overline{f}_k-l\leqslant(1-\alpha_k)(\overline{f}_{k-1}-l)+\frac{L\alpha_k^2}{2}\|x_k-x_{k-1}\|^2+2\delta. \tag{2.17}$$

两边同时除以 α_k^2, 然后对 k 从 1 到 K 作和, 有

$$\overline{f}_K-l\leqslant\frac{L\alpha_K^2}{2}\sum_{k=1}^{K}\|x_k-x_{k-1}\|^2+2\delta\alpha_K^2\sum_{k=1}^{K}\frac{1}{\alpha_k^2}. \tag{2.18}$$

进一步利用 (2.10) 和 (2.12), 可以得到 (2.15). □

由上述定理, 我们已经得到了 $\overline{f}_k - l$ 上界的一个估计, 此估计将用来进一步计算 $\mathcal{G}_{\text{IFAPL}}$ 满足第 3 步中终止条件所需要的迭代次数. 由于加速的一阶算法无法避免受到误差的累积效应的影响 (参见文献 [4]), 我们可以看到, 类似于近似加速梯度下降算法 (参见文献 [4]), 在 L、R 和 δ 固定的情形下, $\overline{f}_k - l$ 并不能无限递减至零, 而是存在着关于这些参数的一个最小值. 以下定理说明, 在 δ 满足一定的条件下, 每次调用 $\mathcal{G}_{\text{IFAPL}}$ 都可以将当前上下界的差值缩小一个固定比例.

定理 2 在 $\mathcal{G}_{\text{IFAPL}}$ 中, 若步长 $\{\alpha_k\}$ 满足条件 (2.10), 且 δ 满足条件

$$\delta \leqslant \delta_\Delta := \frac{(\frac{2}{3}\theta\beta\Delta)^{3/2}}{2C_1C_2R\sqrt{L}}, \tag{2.19}$$

其中 Δ 表示调用 $\mathcal{G}_{\text{IFAPL}}$ 时输入的上下界的差值, 即 $\Delta = \text{ub} - \text{lb}$, 则有以下结论成立:

(1) $\mathcal{G}_{\text{IFAPL}}$ 的迭代次数不超过

$$N(\Delta) := \frac{C_1}{\sqrt{\frac{2}{3}\theta\beta}} \cdot \frac{\sqrt{L}R}{\sqrt{\Delta}} + 1; \tag{2.20}$$

(2) $\mathcal{G}_{\text{IFAPL}}$ 迭代终止时, 我们有 $\text{ub}^+ - \text{lb}^+ \leqslant q(\text{ub} - \text{lb})$, 其中

$$q := \max\{\beta, 1 - (1-\theta)\beta\}. \tag{2.21}$$

证明 先证明 (1), 结合 (2.15) 和 (2.19) 可以得到

$$\overline{f}_K - l \leqslant \frac{C_1^2 L R^2}{2K^2} + 2C_2 K \delta_\Delta. \tag{2.22}$$

不等式右侧在

$$K^* := \left(\frac{C_1^2 L R^2}{2C_2\delta_\Delta}\right)^{\frac{1}{3}} = \frac{C_1}{\sqrt{\frac{2}{3}\theta\beta}} \cdot \frac{\sqrt{L}R}{\sqrt{\Delta}}$$

时达到其最小值. 因此有

$$\overline{f}_{K^*} - l \leqslant \frac{3}{2}(2C_1^2C_2^2LR^2\delta_\Delta^2)^{1/3} \leqslant \theta\beta\Delta. \tag{2.23}$$

注意到 $\mathcal{G}_{\text{IFAPL}}$ 第 3 步中终止条件等价于 $\overline{f}_k - l \leqslant \theta\beta\Delta$, 因此 $\mathcal{G}_{\text{IFAPL}}$ 或者在满足第 2 步中条件时提前终止循环或者在最多 K^* 次迭代后满足第 3 步中条件而终止. 因此, $\mathcal{G}_{\text{IFAPL}}$ 迭代次数有上界 $N(\Delta) = K^* + 1$.

对于 (2), 首先由 (2.8) 中 x_k^u 的定义可见 $\{\overline{f}_k\}$ 单调递减, 同时观察到 $\text{ub} = \overline{f}_0$, 我们有 $\text{ub}^+ \leqslant \text{ub}$. 由于 $\mathcal{G}_{\text{IFAPL}}$ 可能在满足第 2 或者 3 步条件下终止迭代, 我们对这两种情形分别讨论.

若 G_{IFAPL} 在第 K 次迭代中满足第 2 步条件终止迭代, 则有 $\text{lb}^+ = l = \beta \cdot \text{lb} + (1-\beta)\text{ub}$, 进一步可以得到

$$\text{ub}^+ - \text{lb}^+ \leqslant \text{ub} - \beta\,\text{lb} - (1-\beta)\text{ub} = \beta(\text{ub} - \text{lb}).$$

若 $\mathcal{G}_{\text{IFAPL}}$ 在第 K 次迭代中满足第 3 步中条件终止迭代, 则有 $\text{ub}^+ = \overline{f}_K \leqslant l + \theta(\text{ub} - l)$, 由 $\text{lb}^+ = \text{lb}$ 和 $l = \beta \cdot \text{lb} + (1-\beta)\text{ub}$, 可以得到

$$\text{ub}^+ - \text{lb}^+ \leqslant l + \theta\,(\text{ub} - l) - \text{lb} = [1 - (1-\theta)\beta](\text{ub} - \text{lb}).$$

综合上述两种情形, (2) 得证. □

下面给出 IFAPL 算法的外循环部分, 外循环的主要步骤是检查当前的上下界差值 Δ_s 是否已经小于目标精度 ϵ, 如果不是, 则调用 $\mathcal{G}_{\mathrm{IFAPL}}$ 来缩小差值; 否则就终止算法并输出已经找到的 ϵ- 解. 每次调用 $\mathcal{G}_{\mathrm{IFAPL}}$ 时需要输入当前的上下界 ub_s 和 lb_s 及目前找到的最优点 $\hat{x}_s$ 和当前的模型精度 δ_s.

算法 1　IFAPL 算法

0: 给定 $B(\overline{x},R)$, 选取初始点 $p_0\in B(\overline{x},R)$, 给定目标精度 $\epsilon>0$, 选取参数 $\beta,\theta\in(0,1)$ 以及初始模型精度 δ_0.

1: 令 $p_1\in\mathrm{Argmin}_{x\in B(\overline{x},R)}h_{\delta_0}(p_0,x)$, $\mathrm{lb}_1=h_{\delta_0}(p_0,p_1)$, $\mathrm{ub}_1=\min\{f_{\delta_0}(p_0),f_{\delta_0}(p_1)\}+\delta_0$, 取 $\hat{x}_1$ 为 p_0 或 p_1 使其满足 $f_{\delta_0}(\hat{x}_1)+\delta_0=\mathrm{ub}_1$, 设 $s=1$.

2: 若 $\mathrm{ub}_s-\mathrm{lb}_s\leqslant\epsilon$, 则终止算法并输出近似解 $\hat{x}_s$.

3: 调用 $(\hat{x}_{s+1},\mathrm{ub}_{s+1},\mathrm{lb}_{s+1})=\mathcal{G}_{\mathrm{IFAPL}}(\hat{x}_s,\mathrm{ub}_s,\mathrm{lb}_s,R,\overline{x},\beta,\theta,\delta_s)$, δ_s 为当前模型精度.

4: 令 $s=s+1$, 转至第 2 步.

以下讨论 IFAPL 算法的收敛性和迭代复杂度, 我们将对 δ 可由用户选取和 δ 给定不变这两种情形分别进行讨论.

定理 3　对于任意给定 $\epsilon>0$, 若 IFAPL 算法中 $\{\alpha_k\}$ 选取满足条件 (2.10), 且对 $s\geqslant1$ 均有 $\delta_s\leqslant\delta_{\Delta_s}$, 其中 δ_{Δ_s} 在 (2.19) 中定义, 则 IFAPL 算法将收敛到 (2.2) 的一个 ϵ- 解, 且以下结论成立:

(1) $\mathcal{G}_{\mathrm{IFAPL}}$ 的调用次数不超过

$$S_\epsilon:=\max\left\{0,\log_{\frac{1}{q}}\left(\frac{2LR^2+2\delta_0}{\epsilon}\right)\right\}+1;\tag{2.24}$$

(2) IFAPL 算法总迭代次数不超过

$$N_\epsilon:=S_\epsilon+\frac{\sqrt{3/2}}{1-\sqrt{q}}\frac{C_1\sqrt{L}R}{\sqrt{\theta\beta\epsilon}}.\tag{2.25}$$

证明　观察 IFAPL 算法的第 1 步, 并结合 $f_{\delta_0}(p_1)\leqslant f(p_1)$ 和 (2.1) 容易得到

$$\Delta_1=\mathrm{ub}_1-\mathrm{lb}_1\leqslant f_{\delta_0}(p_1)+\delta_0-h_{\delta_0}(p_0,p_1)\leqslant\frac{L}{2}\|p_0-p_1\|^2+2\delta_0\leqslant2LR^2+2\delta_0.\tag{2.26}$$

同时由定理 2 可知, $\Delta_{s+1}\leqslant q\Delta_s,\forall\, s\geqslant1$, 进一步可以得到

$$\Delta_{s+1}\leqslant q^s\Delta_1,\quad\forall\, s\geqslant1.\tag{2.27}$$

容易看到 $\{\Delta_s\}_{s=1}^{\infty}$ 几何级数递减并必将最终小于 ϵ, 因此可以假设在第 S 次调用 $\mathcal{G}_{\mathrm{IFAPL}}$ 后首次得到一个 ϵ- 解, 即

$$\Delta_{S+1}\leqslant\epsilon<\Delta_S.\tag{2.28}$$

结合 (2.26) 和 (2.27), 有

$$\epsilon<q^{S-1}\Delta_1\leqslant q^{S-1}(2LR^2+2\delta_0).\tag{2.29}$$

因此 (1) 得证.

对 (2), 结合 (2.27) 和 (2.28) 可以得到 $\Delta_s \geqslant \epsilon q^{s-S}$, 进一步利用定理 2 结论, 我们将每次调用 $\mathcal{G}_{\mathrm{IFAPL}}$ 的迭代次数相加, 可以得到总迭代次数不超过

$$\begin{aligned}\sum_{s=1}^{S} N_s \leqslant N(\Delta_s) &\leqslant S + \sum_{s=1}^{S} \frac{\sqrt{3/2}C_1}{\sqrt{\theta\beta}} \cdot \frac{\sqrt{L}R}{\sqrt{\Delta_s}} \\ &\leqslant S + \sum_{s=1}^{S} \frac{\sqrt{3/2}C_1\sqrt{L}R}{\sqrt{\theta\beta\epsilon}} \cdot q^{\frac{S-s}{2}} \leqslant S_\epsilon + \frac{\sqrt{3/2}}{1-\sqrt{q}} \frac{C_1\sqrt{L}R}{\sqrt{\theta\beta\epsilon}},\end{aligned} \tag{2.30}$$

其中 N_s 表示第 s 次调用 $\mathcal{G}_{\mathrm{IFAPL}}$ 时的迭代次数.

上述定理说明了在 δ 可以由用户自由选取时, IFAPL 算法可以以 (2.34) 中的迭代复杂度得到 (2.2) 的一个 ϵ- 解. 但是在有些问题中, 如果目标函数的一阶模型精度 δ 是给定不变的, 文献 [4] 证明了一阶加速算法由于受到误差累积效应的影响不能保证收敛到原问题的 ϵ- 解. 同样, IFAPL 算法也无法通过多次调用 $\mathcal{G}_{\mathrm{IFAPL}}$ 来无限缩小上下界差值 Δ_s 以得到一个 ϵ- 解. 这种情形下我们将在 $\mathcal{G}_{\mathrm{IFAPL}}$ 中恰当的时候检查当前的上下界之差是否已经达到了所能达到的一个与 δ 相关的最佳精度, 如果达到则终止算法即可. □

容易观察到, 在 (2.15) 中当

$$K = K_\delta := \left(\frac{C_1^2 L R^2}{2C_2\delta}\right)^{1/3}$$

时不等式右侧达到其最小值

$$\tau := \frac{3}{2}(2^2 C_1^2 C_2^2 L R^2 \delta^2)^{1/3}. \tag{2.31}$$

根据定理 1, 有 $\overline{f}_{K_\delta} - l \leqslant \tau$, 同时注意到 $\mathcal{G}_{\mathrm{IFAPL}}$ 第 3 步中条件 $\overline{f}_k \leqslant l + \theta(\overline{f}_0 - l)$ 等价于 $\overline{f}_k - l \leqslant \theta\beta\Delta$, 因此在 δ 给定不变时, IFAPL 算法需作以下调整:

(M1a) 在 $\mathcal{G}_{\mathrm{IFAPL}}$, 若条件 $\Delta = \mathrm{ub} - \mathrm{lb} \leqslant \frac{\tau}{\theta\beta}$ 满足, 则将其第 3 步中条件 $\overline{f}_k \leqslant l + \theta(\overline{f}_0 - l)$ 修改成 $\overline{f}_k - l \leqslant \tau$;

(M1b) 将 IFAPL 算法第 2 步中条件 $\mathrm{ub}_s - \mathrm{lb}_s \leqslant \epsilon$ 改成 $\mathrm{ub}_s - \mathrm{lb}_s \leqslant \epsilon_\delta$, 其中

$$\epsilon_\delta := \frac{1-\beta+\theta\beta}{\theta\beta}\tau = \frac{1-\beta+\theta\beta}{\theta\beta} \cdot \frac{3}{2}(2^2 C_1^2 C_2^2 L R^2 \delta^2)^{1/3}. \tag{2.32}$$

我们有以下定理.

定理 4 在 IFAPL 算法中, 若对所有 $s \geqslant 0$, $\delta_s = \delta$ 成立, $\{\alpha_k\}$ 选取满足条件 (2.10), 则 IFAPL 算法将收敛到 (2.2) 的一个 ϵ_δ- 解, 并且 $\mathcal{G}_{\mathrm{IFAPL}}$ 调用的次数不超过

$$S_\delta := \max\left\{0, \log_{\frac{1}{q}}\left(\frac{2LR^2 + 2\delta}{\epsilon_\delta}\right)\right\} + 2, \tag{2.33}$$

IFAPL 的总迭代次数不超过

$$N_\delta := \tilde{S}_1 + \frac{\sqrt{3/2}}{1-\sqrt{q}} \frac{C_1\sqrt{L}R}{\sqrt{\theta\beta\epsilon_\delta}} + \left(\frac{C_1 L R^2}{2C_2\delta}\right)^{1/3}. \tag{2.34}$$

证明 当 $\Delta > \frac{\tau}{\theta\beta}$ 时, 容易验证 (2.19) 满足, 由定理 2 可知, 此时每次 $\mathcal{G}_{\mathrm{IFAPL}}$ 迭代次数不超过 (2.20) 中的 $N(\Delta)$, 且有 $\Delta^+ \leqslant q\Delta$. 进而结合 (2.26) 便得到此时 $\mathcal{G}_{\mathrm{IFAPL}}$ 调用次数不超过

$$S_1 := \left\lceil \log_{\frac{1}{q}}\left(\frac{\Delta_1}{\epsilon_\delta}\right)\right\rceil \leqslant \log_{\frac{1}{q}}\left(\frac{2LR^2 + 2\delta}{\epsilon_\delta}\right) + 1.$$

注意到 $\Delta_{S_1} > \epsilon_\delta$, 我们有此时总的迭代次数

$$
\begin{aligned}
N_1 &\leqslant \sum_{s=1}^{S_1} N(\Delta_s) \leqslant S_1 + \sum_{s=1}^{S_1} \frac{\sqrt{3/2}C_1}{\sqrt{\theta\beta}} \cdot \frac{\sqrt{L}R}{\sqrt{\Delta_s}} \\
&\leqslant S_1 + \sum_{s=1}^{S_1} \frac{\sqrt{3/2}C_1\sqrt{L}R}{\sqrt{\theta\beta\epsilon_\delta}} \cdot q^{\frac{S_1-s}{2}} \leqslant S_1 + \frac{\sqrt{3/2}}{1-\sqrt{q}} \frac{C_1\sqrt{L}R}{\sqrt{\theta\beta\epsilon_\delta}}.
\end{aligned} \tag{2.35}
$$

当 $\Delta \leqslant \frac{\tau}{\theta\beta}$ 时, 由 $\overline{f}_{K_\delta} - l \leqslant \tau$ 和 (M1a), 有 $\mathcal{G}_{\text{IFAPL}}$ 的迭代次数不超过 K_δ. 对任意 $k \leqslant K_\delta$, 若 $\mathcal{G}_{\text{IFAPL}}$ 满足 $\overline{f}_k - l \leqslant \tau$ 而跳出循环, 则有

$$
\Delta^+ = \text{ub}^+ - \text{lb}^+ \leqslant l + \tau - \text{lb} \leqslant (1-\beta)(\text{ub} - \text{lb}) + \tau. \tag{2.36}
$$

结合 $\Delta \leqslant \frac{\tau}{\theta\beta}$, 便得到 $\Delta^+ \leqslant \frac{1-\beta+\theta\beta}{\theta\beta}\tau$, 即 IFAPL 算法达到最佳精度 ϵ_δ 而终止.

综合以上两点可知, IFAPL 调用 $\mathcal{G}_{\text{IFAPL}}$ 的次数不超过 $S_\delta := S_1 + 1$, 总的迭代次数不超过 $N_\delta := N_1 + K_\delta$. □

2.2　解强凸函数的 IFAPLS 算法

本节讨论在目标函数 f 是强凸光滑函数的情形下, 如何进一步改进 IFAPL 算法, 以得到对强凸函数的最优迭代复杂度. 具体来讲, 本节考虑以下无约束优化问题:

$$
f^* = \min_{x \in \mathbb{R}^n} f(x), \tag{2.37}
$$

其中 f 满足光滑条件 (1.2), 且为 μ- 强凸函数, 即 f 满足以下不等式:

$$
\begin{aligned}
\frac{\mu}{2}\|y-x\|^2 &\leqslant f(y) - f(x) - \langle f'(x), y-x \rangle \\
&\leqslant \frac{L}{2}\|y-x\|^2, \quad \forall\, x, y \in \mathbb{R}^n,
\end{aligned} \tag{2.38}
$$

其中 $\mu > 0$. 本节假设存在一个给定的 (L, μ, δ)- 模型, 它对于任意点 $x \in \mathbb{R}^n$ 可以给出 f 在此点满足以下关系式的近似一阶信息 $(f_\delta(x), g_\delta(x))$:

$$
\begin{aligned}
\frac{\mu}{2}\|y-x\|^2 &\leqslant f(y) - f_\delta(x) - \langle g_\delta(x), y-x \rangle \\
&\leqslant \frac{L}{2}\|y-x\|^2 + \delta, \quad \forall\, x, y \in \mathbb{R}^n.
\end{aligned} \tag{2.39}
$$

与第 2.1 小节不同, 本小节考虑的是 $\mathbb{R}^n$ 的无约束问题, 在此设定下, 我们需要假设存在并给出一个初始的 f^* 的下界 lb_1, 在大多数优化问题中, 这样一个下界很容易得到. 在此假设下, 由 f 的强凸性, 对于任意初始点 p_0, 有 $\|p_0 - x^*\|^2 \leqslant 2[f(p_0) - \text{lb}_1]/\mu$. 这意味着 (2.37) 的唯一解 x^* 是在 Euclid 球 $B(p_0, \sqrt{2[f(p_0) - \text{lb}_1]/\mu})$ 中, 因此可以将前一节中的 IFAPL 算法直接应用到此 Euclid 球上. 注意到每次调用内迭代子程序后, f^* 上下界的差值会单调缩小, 因此在每个内迭代子程序的开始, 我们都可以重新调整搜索 Euclid 球的球心和半径, 由于搜索的 Euclid 球不断变小, 我们便可以得到更好的迭代复杂度.

下面先给出对于强凸光滑函数的 IFAPLS 算法, 然后讨论其收敛及迭代复杂度的证明. 该算法只需改变 IFAPL 算法的外迭代和内迭代子程序的一些步骤, 以下为具体的改变:

过程 2 IFAPLS 内迭代子程序: $(x^+, \mathrm{ub}^+, \mathrm{lb}^+) = \mathcal{G}_{\mathrm{IFAPLS}}(\hat{x}, \mathrm{ub}, \mathrm{lb}, r, \beta, \theta)$

在过程 1 中, 令 $\overline{x} = \hat{x}$, 并将 (2.6) 中迫近函数 d 替换成 $\|x - \hat{x}\|^2/2$.

算法 2 IFAPLS 算法

在算法 1, 将第 0、1 和 3 步分别改成

0: 选取初始下界 $\mathrm{lb}_1 \leqslant f^*$, 初始点 $p_0 \in \mathbb{R}^n$, 初始模型精度 δ_0, 初始上界 $\mathrm{ub}_1 = f_{\delta_0}(p_0) + \delta_0$, 目标精度 $\epsilon > 0$ 以及参数 $\beta, \theta \in (0, 1)$.

1: 令 $\hat{x}_1 = p_0$, 设 $s = 1$.

3: 调用 $(\hat{x}_{s+1}, \mathrm{ub}_{s+1}, \mathrm{lb}_{s+1}) = \tilde{\mathcal{G}}_{\mathrm{IFAPLS}}(\hat{x}_s, \mathrm{ub}_s, \mathrm{lb}_s, \sqrt{2(\mathrm{ub}_s - \mathrm{lb}_s)/\mu}, \beta, \theta, \delta_s)$, δ_s 为当前模型精度.

以下具体讨论在 $\delta_s = \delta$ 给定不变和 δ_s 可自由选取时, IFAPLS 算法的收敛性和迭代复杂度. 可以看到对任意给定的 $\mathcal{G}_{\mathrm{IFAPLS}}$ 和 $\mathcal{G}_{\mathrm{IFAPL}}$, 两者唯一的区别是所搜索的 Euclid 球不一样, 前者是 $B(\hat{x}, R)$, 后者是 $B(\bar{x}, R)$. 对 IFAPLS 算法, 与 IFAPL 算法的主要不同点是, IFAPLS 每次调用 $\mathcal{G}_{\mathrm{IFAPLS}}$ 时所输入的球心 $\hat{x}$ 和半径 R 是不断变化的. 基于这些观察, 可以知道引理 1、2 和定理 1 对 $\mathcal{G}_{\mathrm{IFAPLS}}$ 算法同样成立. 对于定理 2, 只需要代入 $R = \sqrt{\frac{2\Delta}{\mu}}$, 我们便有如下类似定理.

定理 5 在 $\mathcal{G}_{\mathrm{IFAPLS}}$ 中, 若步长 $\{\alpha_k\}$ 满足条件 (2.10), 且 δ 满足以下条件:

$$\delta \leqslant \hat{\delta}_\Delta := \sqrt{\frac{\theta^3\beta^3\mu}{27C_1^2C_2^2L}}\Delta, \tag{2.40}$$

其中 C_1 和 C_2 在 (2.10) 中定义, μ 和 L 由 (2.38) 决定, θ 和 β 是算法所选定的参数, Δ 是调用 $\mathcal{G}_{\mathrm{IFAPLS}}$ 输入的上下界差值, 则有以下结论成立:

(1) $\mathcal{G}_{\mathrm{IFAPLS}}$ 的迭代次数不超过

$$\hat{N} := \sqrt{3}C_1\sqrt{\frac{L}{\theta\beta\mu}} + 1; \tag{2.41}$$

(2) $\mathcal{G}_{\mathrm{IFAPLS}}$ 迭代终止时, 我们有 $\mathrm{ub}^+ - \mathrm{lb}^+ \leqslant q(\mathrm{ub} - \mathrm{lb})$, 其中 q 在 (2.21) 中定义.

证明 利用定理 2, 代入 $R = \sqrt{\frac{2\Delta}{\mu}}$ 即得 (1), (2) 与定理 2(2) 一致. □

下面讨论在 δ 可以自由选取时, IFAPLS 算法的收敛性及迭代复杂度.

定理 6 对于任意给定 $\epsilon > 0$, 若 IFAPLS 算法中 $\{\alpha_k\}$ 选取满足条件 (2.10), 且对 $s \geqslant 1$ 均有 $\delta_s \leqslant \hat{\delta}_{\Delta_s}$, 其中 $\hat{\delta}_{\Delta_s}$ 在 (2.40) 中定义, 则 IFAPLS 算法将收敛到 (2.37) 的一个 ϵ- 解, 且以下结论成立:

(1) $\mathcal{G}_{\mathrm{IFAPLS}}$ 的调用次数不超过

$$\hat{S}_\epsilon := \log_{\frac{1}{q}}\left(\frac{\mathrm{ub}_1 - \mathrm{lb}_1}{\epsilon}\right) + 1; \tag{2.42}$$

(2) IFAPLS 总迭代次数不超过

$$\hat{N}_\epsilon := \hat{S}_\epsilon\left(\sqrt{3}C_1\sqrt{\frac{L}{\theta\beta\mu}} + 1\right). \tag{2.43}$$

证明 由于任意 $\mathcal{G}_{\mathrm{IFAPLS}}$ 均满足 (2.40), 由定理 5 可知, 每次调用 $\mathcal{G}_{\mathrm{IFAPLS}}$ 迭代次数不超过 $\hat{N}$, 且 $\Delta^+ \leqslant q\Delta$. 类似于定理 3(1) 中证明可以得到 (1), 从而, IFAPLS 总迭代次数不超过

$$\hat{N}_\epsilon := \hat{N} \cdot \hat{S}_\epsilon = \hat{S}_\epsilon\left(\sqrt{3}C_1\sqrt{\frac{L}{\theta\beta\mu}}+1\right). \tag{2.44}$$

这就证明了 (2). □

上述定理说明在一阶信息模型的误差随着当前上下界差值同步缩小时, 对任意 $\epsilon > 0$, IFAPLS 算法可以收敛到 ϵ- 解, 且算法的迭代复杂度为 $\mathcal{O}(\log\frac{1}{\epsilon})$. 下面讨论在 $\delta_s = \delta$ $(\forall\, s \geqslant 0)$ 时, IFAPLS 算法所能达到的最佳精度. 文献 [33] 证明了在 δ 给定不变的情形下, 近似加速梯度下降法所能达到的最佳精度为 $\mathcal{O}(\sqrt{\frac{L}{\mu}}\delta)$, IFAPLS 算法也可以达到同样的精度, 且有最优的迭代复杂度.

根据 IFAPLS 算法在 δ 给定不变的情形所能保证的近似解精度, IFAPLS 算法需要作以下调整:

(M2a) 在 $\mathcal{G}_{\text{IFAPLS}}$ 中, 若条件 $\text{ub} - \text{lb} \leqslant \frac{\hat{\tau}}{\theta\beta}$ 满足, 则将第 3 步中条件 $\overline{f}_k \leqslant l + \theta(\overline{f}_0 - l)$ 修改成 $\overline{f}_k - l \leqslant \hat{\tau}$;

(M2b) 将 IFAPLS 算法第 2 步中条件 $\text{ub}_s - \text{lb}_s \leqslant \epsilon$ 改成 $\text{ub}_s - \text{lb}_s \leqslant \hat{\epsilon}_\delta$, 其中

$$\hat{\tau} := \sqrt{\frac{27C_1^2C_2^2L}{\theta\beta\mu}}\delta, \quad \hat{\epsilon}_\delta := \frac{1-\beta+\theta\beta}{\theta\beta}\cdot\hat{\tau} = \frac{1-\beta+\theta\beta}{\theta\beta}\cdot\sqrt{\frac{27C_1C_2^2L}{\theta^3\beta^3\mu}}\delta. \tag{2.45}$$

定理 7　在 IFAPLS 算法中, 若 $\delta_s = \delta$ $(\forall\, s \geqslant 1)$ 给定不变, $\{\alpha_k\}$ 选取满足条件 (2.10), 则 IFAPL 算法将收敛到 (2.37) 的一个 $\hat{\epsilon}_\delta$- 解, 并且 $\mathcal{G}_{\text{IFAPLS}}$ 调用的次数不超过

$$\hat{S}_\delta := \log_{\frac{1}{q}}\left(\frac{\text{ub}_1 - \text{lb}_1}{\hat{\epsilon}_\delta}\right) + 2, \tag{2.46}$$

IFAPLS 的总迭代次数不超过

$$\hat{N}_\delta := \hat{S}_\delta\left(\sqrt{3}C_1\sqrt{\frac{L}{\theta\beta\mu}}+1\right). \tag{2.47}$$

证明　当 $\Delta > \frac{\hat{\tau}}{\theta\beta}$ 时, 容易验证 (2.40) 满足, 由定理 5 可知, 此时每次 $\mathcal{G}_{\text{IFAPLS}}$ 调用其迭代次数不超过 $\hat{N}$, 其中 $\hat{N}$ 在 (2.41) 中定义, 且 $\Delta^+ \leqslant q\Delta$. 因此, 此时 $\mathcal{G}_{\text{IFAPLS}}$ 的调用次数不超过

$$\hat{S}_1 := \log_{\frac{1}{q}}\left(\frac{\text{ub}_1 - \text{lb}_1}{\hat{\epsilon}_\delta}\right) + 1,$$

此时总迭代次数不超过

$$\hat{N}_1 := \hat{S}_1 \cdot \hat{N}.$$

当 $\Delta \leqslant \frac{\hat{\tau}}{\theta\beta}$ 时, 由定理 1 和 $R = \sqrt{\frac{2\Delta_s}{\mu}}$, 有

$$\begin{aligned}\overline{f}_K - l &\leqslant \frac{C_1^2L\Delta_s}{K^2\mu} + 2C_2K\delta \\ &\leqslant \frac{C_1^2L\hat{\tau}}{K^2\theta\beta\mu} + 2C_2K\delta.\end{aligned} \tag{2.48}$$

不等式右侧在

$$K = \hat{K}_\delta := \left(\frac{C_1^2L\hat{\tau}}{C_2\theta\beta\mu\delta}\right)^{\frac{1}{3}} = \sqrt{3}C_1\sqrt{\frac{L}{\theta\beta\mu}}$$

时取得其最小值 $\hat{\tau}$, 即有 $\overline{f}_{\hat{K}_\delta} - l \leqslant \hat{\tau}$, 由 (M2a) 可知, 此时 $\mathcal{G}_{\text{IFAPLS}}$ 的第 3 步中条件已满足, 因而终止 $\mathcal{G}_{\text{IFAPLS}}$. 进而仿照 (2.36) 可以得到 $\Delta^+ \leqslant \hat{\epsilon}_\delta$, 也就是 IFAPLS 算法达到最佳精度 $\hat{\epsilon}_\delta$ 而终止.

综合以上两点可知, $\mathcal{G}_{\mathrm{IFAPLS}}$ 的调用次数不超过 $\hat{S}_\delta := \hat{S}_1 + 1$, IFAPLS 算法总迭代次数不超过

$$\hat{N}_\delta := \hat{N}_1 + \hat{K}_\delta.$$

证毕. □

3 加速近似光滑化 BL 算法

本节结合文献 [38] 中提出的光滑化方法, 进一步拓展 IFAPL 和 IFAPLS 算法来求解一类鞍点问题. 我们将提出两个新算法: IFUSL 和 IFUSLS 算法, 前者适用于解决 Euclid 球上的此类鞍点问题, 后者适用于目标函数为 $\mathbb{R}^n$ 上强凸函数. 由于此类鞍点问题中包含的 Max 问题往往无法精确求解, 从而导致目标函数的一阶信息不精确, 所提出的两种算法均可以使用目标函数不精确的一阶信息. 我们将分别讨论在 Max 问题误差给定不变或者可以由用户选取时, 两种算法分别能达到的最佳精度及迭代复杂度.

3.1 IFUSL 算法

本小节考虑具有以下这种结构的鞍点问题:

$$f^*_{\bar{x},R} = \min_{x \in B(\bar{x},R)} f(x) := \hat{f}(x) + F(x), \tag{3.1}$$

其中 $\hat{f}$ 为光滑函数, 即存在 $L_{\hat{f}} > 0$ 使得

$$\hat{f}(y) - \hat{f}(x) - \langle \nabla \hat{f}(x), y - x \rangle \leqslant \frac{L_{\hat{f}}}{2} \|y - x\|^2, \tag{3.2}$$

同时

$$F(x) := \max_{y \in Y} \{ \langle Ax, y \rangle - \hat{g}(y) \}, \tag{3.3}$$

其中 $Y \subseteq \mathbb{R}^m$ 是一个紧致凸集, $\hat{g} : Y \to \mathbb{R}$ 为简单函数, $A : \mathbb{R}^n \to \mathbb{R}^m$ 是线性算子. 注意到 (3.1) 是一个复合优化问题, 目标函数是一个光滑函数与一个非光滑函数的和, 因此, 整体上它依然是一个非光滑优化问题. 文献 [38] 最先提出了一种光滑化的方法, 通过利用这类问题的特殊结构, 使用一系列光滑函数来一致逼近非光滑部分 $F(x)$, 该算法可以将此问题的迭代复杂度从 $\mathcal{O}(\frac{1}{\epsilon^2})$ 提高至 $\mathcal{O}(\frac{1}{\epsilon})$. 文献 [9,10] 进一步将此光滑化的思想应用到了 BL 类算法并得到了与文献 [38] 中同样的迭代复杂度.

令 $v : Y \to \mathbb{R}$ 是一个 σ_v- 强凸函数, 并记 $c_v := \mathrm{argmin}_{y \in Y} v(y)$, 运用文献 [38] 中光滑化方法, 我们可以用以下光滑函数逼近 $F(x)$:

$$F^\eta(x) := \max_{y \in Y} \Phi(x, y) := \langle Ax, y \rangle - \hat{g}(y) - \eta V(y), \tag{3.4}$$

其中 $\eta > 0$ 被称作光滑化参数, $V(\cdot)$ 为 v 所对应的 Bregman 距离, 定义如下:

$$V(y) := v(y) - v(c_v) - \langle \nabla v(c_v), y - c_v \rangle. \tag{3.5}$$

文献 [38] 证明了 $F^\eta(\cdot)$ 的梯度是 $\nabla F^\eta(x) = A^* y^*_x$, 且该梯度是 L_η-Lipschitz 连续的, 其中

$$L_\eta := \frac{\|A\|^2}{\eta \sigma_v}, \tag{3.6}$$

这里 $\|A\|$ 是 A 的算子范数, A^* 是其共轭算子, $y_x^* \in Y$ 是 (3.4) 的唯一精确解. $F^\eta(x)$ 一致逼近 $F(x)$, 且它们之间的距离与 η 线性相关, 即

$$F^\eta(x) \leqslant F(x) \leqslant F^\eta(x) + \eta D_{v,Y}, \quad \forall\, x \in B(\bar{x}, R), \tag{3.7}$$

其中

$$D_{v,Y} := \max_{y,z\in Y}\{v(y) - v(z) - \langle \nabla v(z), y - z\rangle\}. \tag{3.8}$$

因此, 记

$$f^\eta(x) := \hat{f}(x) + F^\eta(x), \tag{3.9}$$

我们有

$$f^\eta(x) \leqslant f(x) \leqslant f^\eta(x) + \eta D_{v,Y}. \tag{3.10}$$

可见, 通过光滑化 $f(x)$, 并逐渐将 η 缩小至接近零, 我们可以通过迭代求解 f^η 来得到原优化问题 (3.1) 的近似解. 然而很多问题中, 由于 Max 问题 (3.4) 往往只能得到近似解, 从而导致 F^η 及 f^η 的一阶信息无法精确计算, 因此需要发展能使用不精确的一阶信息的算法. 容易看到, F^η 及 f^η 一阶信息的精度与 (3.4) 的近似解的精度直接相关. 对于此类 Max 问题的近似解误差的定义, 文献 [4] 讨论了三种相互关联的形式, 本节采用其中一种. 令 y_x 为 (3.4) 的一个近似解, 其误差定义为

$$\delta_{y_x} := \Phi(x, y_x^*) - \Phi(x, y_x). \tag{3.11}$$

我们先定义如下条件:

条件 (C1) 给定 $\delta > 0$, 对任意 $x \in B(\bar{x}, R)$, $\eta \geqslant 0$, 可以找到一个 (3.4) 的近似解 y_x 使得 $\delta_{y_x} \leqslant \delta$.

文献 [4] 进一步证明了, 若 F^η 满足条件 (C1), 当 $\eta > 0$ 时,

$$\begin{aligned} F_\delta^\eta(z) + \langle G_\delta^\eta(z), x - z\rangle &\leqslant F^\eta(x) = \Phi(x, y_x^*) \\ &\leqslant F_\delta^\eta(z) + \langle G_\delta^\eta(z), x - z\rangle + L_\eta\|x - z\|^2 + 2\delta, \quad \forall\, z \in B(\bar{x}, R), \end{aligned} \tag{3.12}$$

其中

$$F_\delta^\eta(z) := \Phi(z, y_z), \quad G_\delta^\eta(z) := \nabla_1\Phi(z, y_z).$$

为简单起见, 本节假设 (3.1) 中 $\hat{f}$ 的一阶信息可以精确计算, 而 F^η 的一阶信息满足条件 (C1), 我们称这样的近似一阶信息模型为 f 的一个 $(2\delta, L)$- 模型, 即当 $\eta \geqslant 0$ 时,

$$f_\delta^\eta(x) \leqslant f^\eta(x) \leqslant f_\delta^\eta(x) + \delta, \quad \forall\, x \in B(\bar{x}, R), \tag{3.13}$$

其中记 $f(x) = f^0(x)$, $f_\delta(x) = f_\delta^0(x)$, 且当 $\eta > 0$ 时, $f^\eta(x)$ 满足 (2.1), 即

$$0 \leqslant f^\eta(y) - (f_\delta^\eta(x) + \langle g_\delta^\eta(x), y - x\rangle) \leqslant \frac{L}{2}\|y - x\|^2 + 2\delta, \quad \forall\, x, y \in X, \tag{3.14}$$

其中

$$L := L_{\hat{f}} + 2L_\eta, \quad f_\delta^\eta(x) := \hat{f}(x) + F_\delta^\eta(x), \quad g_\delta^\eta(x) := \nabla\hat{f}(x) + G_\delta^\eta(x). \tag{3.15}$$

下面介绍可以利用上述 $(2\delta, L)$- 模型求解 (3.1) 的 IFUSL 算法. IFUSL 算法依然为外迭代 - 内迭代的结构, 首先给出其内迭代子程序 $\mathcal{G}_{\text{IFUSL}}$.

过程 3 IFUSL 算法内迭代子程序: $(x^+, D^+, \mathrm{ub}^+, \mathrm{lb}^+) = \mathcal{G}_{\mathrm{IFUSL}}(\hat{x}, D, \mathrm{ub}, \mathrm{lb}, R, \overline{x}, \beta, \theta, \delta)$

0: 令 $k=1$, $\overline{f}_0 = \mathrm{ub}$, $l = \beta\cdot\mathrm{lb} + (1-\beta)\mathrm{ub}$, $Q_0 = \mathbb{R}^n$, $x_0^u = \hat{x}$, $x_0 = \overline{x}$, 并设

$$\eta := \frac{\theta(\overline{f}_0 - l)}{2D}. \tag{3.16}$$

1: 更新切平面模型: 按照 (2.3) 和 (2.5) 分别计算 x_k^l 和 $\underline{Q}_k$, 并记

$$h_\delta(x_k^l, x) = h_\delta^\eta(x_k^l, x) := f_\delta^\eta(x_k^l) + \langle g_\delta^\eta(x_k^l), x - x_k^l\rangle. \tag{3.17}$$

2: 更新 prox-center 和下界: 求解 (2.6) 来计算 x_k.
若 $\underline{Q}_k = \emptyset$ 或者 $\|x_k - \overline{x}\| > R$, 则终止迭代并输出 $x^+ = x_{k-1}^u$, $D^+ = D$, $\mathrm{ub}^+ = \overline{f}_{k-1}$, $\mathrm{lb}^+ = l$.

3: 更新上界和对 $D_{v,Y}$ 的估计: 按照 (2.7) 和 (2.8) 分别计算 $\tilde{x}_k^u$ 和 x_k^u, 更新 $\overline{f}_k = f_\delta(x_k^u) + \delta$.
并检查以下条件:
(3a) 若 $\overline{f}_k \leqslant l + \theta(\overline{f}_0 - l)$, 则终止迭代并输出 $x^+ = x_k^u$, $D^+ = D$, $\mathrm{ub}^+ = \overline{f}_k$, $\mathrm{lb}^+ = \mathrm{lb}$;
(3b) 若 $\overline{f}_k > l + \theta(\overline{f}_0 - l)$ 且 $f_\delta^\eta(x_k^u) + 2\delta \leqslant l + \frac{\theta}{2}(\overline{f}_0 - l)$, 则终止迭代并输出 $x^+ = x_k^u$, $D^+ = 2D$, $\mathrm{ub}^+ = \overline{f}_k$, $\mathrm{lb}^+ = \mathrm{lb}$.

4: 按 $\mathcal{G}_{\mathrm{IFAPL}}$ 第 4 步中同样方法定义 $\overline{Q}_k$ 和选取 Q_k. 令 $k = k+1$, 转至第 2 步.

在 $\mathcal{G}_{\mathrm{IFUSL}}$ 中, 为保证 IFUSL 算法的收敛性和最优迭代复杂度, 步长 $\{\alpha_k\}$ 的选取需要满足以下条件:

$$\alpha_1 = 1, \quad 0 < \alpha_k \leqslant 1, \quad \alpha_k \leqslant \frac{C_1}{k}, \quad \frac{1-\alpha_{k+1}}{\alpha_{k+1}^2} \leqslant \frac{1}{\alpha_k^2} \quad \text{且} \quad \alpha_k^2 \sum_{i=1}^{k} \frac{3+\alpha_i}{\alpha_i^2} \leqslant C_2 k, \quad \forall\, k \geqslant 1, \tag{3.18}$$

其中常数 $C_1, C_2 > 0$. 同样, 若取 $\alpha_k = 2/(k+1)$, $k \geqslant 1$, 则有 $C_1 = 2$, $C_2 = 4$; 若 α_k 由 (2.11) 选取, 则有 $C_1 = 2$, $C_2 = 4$.

我们补充以下几点说明. 首先, 在 $\mathcal{G}_{\mathrm{IFUSL}}$ 中, 由于原目标函数 f 被光滑函数 f^η 所替代, 所以该算法的切平面模型是建立在 f^η 上的, 并且所使用的是关于 f^η 不精确的一阶信息. 其次, 光滑化参数 η 是一个关于 D、θ 和当前上下界差值 Δ 的函数, 其中 D 是当前对 (3.8) 中 $D_{v,Y}$ 的估计, $\overline{f}_0 - l = \beta\Delta$. 最后, $\mathcal{G}_{\mathrm{IFUSL}}$ 运用与 $\mathcal{G}_{\mathrm{IFAPL}}$ 同样的技巧来选取 $\underline{Q}_k$、Q_k 和 $\overline{Q}_k$, 使得 Q_k 中线性限制的数量很小, 可以由用户决定, 进而保证了子问题 (2.6) 的对偶问题维数较小, 可以得到精确解, 具体细节可参见文献 [10].

与 USL 和 FUSL 算法类似, 观察 $\mathcal{G}_{\mathrm{IFUSL}}$ 的终止条件和输出, 我们有以下重要引理.

引理 3 对 $\mathcal{G}_{\mathrm{IFUSL}}$, 以下结论成立:

(1) 若 $\mathcal{G}_{\mathrm{IFUSL}}$ 在第 2 或者 (3a) 步终止迭代, 则有 $\mathrm{ub}^+ - \mathrm{lb}^+ \leqslant q(\mathrm{ub} - \mathrm{lb})$, 其中 q 在 (2.21) 定义.

(2) 若 $\mathcal{G}_{\mathrm{IFUSL}}$ 在第 (3b) 步终止迭代, 则有 $D < D_{v,Y}$ 且 $D^+ < 2D_{v,Y}$.

证明 (1) 的证明与定理 2(2) 的证明完全一致. 对 (2), 先由 (3.10) 有

$$D_{v,Y} \geqslant \frac{f(x) - f^\eta(x)}{\eta}, \quad \forall\, x \in B(\bar{x}, R). \tag{3.19}$$

取 $x = x_k^u$, 再结合 $f(x_k^u) \geqslant f_\delta(x_k^u) = \overline{f}_k - \delta$、(3.13) 和 (3.16) 以及第 (3b) 步的终止条件有

$$D_{v,Y} \geqslant \frac{f(x_k^u) - f^\eta(x_k^u)}{\eta} \geqslant \frac{\overline{f}_k - \delta - (f_\delta^\eta(x_k^u) + \delta)}{\eta} > \frac{\frac{\theta}{2}(\overline{f}_0 - l)}{\theta(\overline{f}_0 - l)/2D} = D. \tag{3.20}$$

因此, 由第 (3b) 步中 D^+ 的定义, 有 $D^+ < 2D_{v,Y}$. □

注意到在任意给定的 $\mathcal{G}_{\mathrm{IFUSL}}$ 中, 参数 η 在第 0 步中选定, 并在整个内迭代过程中保持不变, 因此, $\mathcal{G}_{\mathrm{IFUSL}}$ 可以看作将 $\mathcal{G}_{\mathrm{IFAPL}}$ 算法作用在光滑函数 f^η 上. 根据定理 1, 有以下结果.

引理 4 如果 $\mathcal{G}_{\mathrm{IFUSL}}$ 中 $\{\alpha_k\}$ 选取满足 (3.18), 那么对任意 $K \geqslant 1$, 若 $\mathcal{G}_{\mathrm{IFUSL}}$ 在第 K 次迭代时没有终止, 则有

$$f_\delta^\eta(x_K^u) + 2\delta - l \leqslant \frac{C_1^2 L R^2}{2K^2} + C_2 K\delta, \tag{3.21}$$

其中 L 在 (3.15) 中定义.

证明 注意到 f^η 是光滑函数, 即满足 (1.2), 其中 $\rho = 1$, $M = L$, 且 f^η 的近似一阶信息满足 $(2\delta, L)$- 模型, 仿照定理 1, 可以得到对 $\forall k \geqslant 1$,

$$\begin{aligned} f_\delta^\eta(x_k^u) + 2\delta &\leqslant f^\eta(\tilde{x}_k^u) + 2\delta \leqslant f_\delta^\eta(x_k^l) + \langle g_\delta^\eta(x_k^l), \tilde{x}_k^u - x_k^l\rangle + \frac{L}{2}\|\tilde{x}_k^u - x_k^l\| + 4\delta \\ &\leqslant (1-\alpha_k) f^\eta(x_{k-1}^u) + \alpha_k l + \frac{L\alpha_k^2}{2}\|x_k - x_{k-1}\|^2 + 4\delta \\ &\leqslant (1-\alpha_k)(f_\delta^\eta(x_{k-1}^u) + 2\delta) + \alpha_k l + \frac{L\alpha_k^2}{2}\|x_k - x_{k-1}\|^2 + (3+\alpha_k)\delta, \end{aligned} \tag{3.22}$$

其中第一和二个不等式分别由 (3.13) 和 (3.14) 得到, 第三个不等式应用了 f^η 的凸性及 (2.3) 和 (2.7), 第 4 个不等式利用了 (3.13). 将上述不等式两边减去 l, 再除以 α_k^2, 有

$$\frac{1}{\alpha_k^2}(f_\delta^\eta(x_k^u) + 2\delta - l) \leqslant \frac{1-\alpha_k}{\alpha_k^2}(f_\delta^\eta(x_{k-1}^u) + 2\delta - l) + \frac{L}{2}\|x_k - x_{k-1}\|^2 + \frac{3+\alpha_k}{\alpha_k^2}\delta. \tag{3.23}$$

进一步对 k 从 1 到 K 作和, 由 (3.18) 中 $\alpha_1 = 1$ 和 $\frac{1-\alpha_{k+1}}{\alpha_{k+1}^2} \leqslant \frac{1}{\alpha_k^2}$, 以及 $f_\delta^\eta(x_k^u) + 2\delta - l > 0, \forall\, 0 \leqslant k \leqslant K$, 可以得到

$$f_\delta^\eta(x_K^u) + 2\delta - l \leqslant \frac{L\alpha_K^2}{2}\sum_{k=1}^K \|x_k - x_{k-1}\|^2 + \delta\alpha_K^2 \sum_{k=1}^K \frac{3+\alpha_k}{\alpha_k^2} \leqslant \frac{C_1^2 L R^2}{2K^2} + C_2 K\delta, \tag{3.24}$$

其中第二个不等式应用了引理 1 和 (3.18). □

下面给出 IFUSL 算法的外循环部分, 其中外循环的主要作用是检查当前的上下界差值是否已达到目标精度, 若达到则终止算法并输出近似解, 否则调用内循环子程序 $\mathcal{G}_{\mathrm{IFUSL}}$ 来进一步缩小上下界差值.

算法 3 IFUSL 算法

0: 给定 $B(\overline{x}, R)$, 选取初始点 $p_0 \in B(\overline{x}, R)$, (3.4) 和 (3.5) 中的强凸函数 $v(\cdot)$, (3.8) 中对 $D_{v,Y}$ 的初始估计 D_1, 目标精度 $\epsilon > 0$ 和参数 $\beta, \theta \in (0, 1)$.

1: 令 $p_1 \in \mathrm{Argmin}_{x\in B(\overline{x},R)} h^0(p_0, x)$, 其中 $h^0(p_0, x) = f(p_0) + \langle g(p_0), x - p_0\rangle$, $\mathrm{lb}_1 = h^0(p_0, p_1), \mathrm{ub}_1 = \min\{f(p_0), f(p_1)\}$, 取 $\hat{x}_1$ 为 p_0 或者 p_1 使得 $f(\hat{x}_1) = \mathrm{ub}_1$, 设 $s = 1$.

2: 若 $\mathrm{ub}_s - \mathrm{lb}_s \leqslant \epsilon$, 则终止算法并输出近似解 $\hat{x}$.

3: 调用 $(\hat{x}_{s+1}, D_{s+1}, \mathrm{ub}_{s+1}, \mathrm{lb}_{s+1}) = \mathcal{G}_{\mathrm{IFUSL}}(\hat{x}_s, D_s, \mathrm{ub}_s, \mathrm{lb}_s, R, \overline{x}, \beta, \theta, \delta_s)$, 其中 δ_s 为模型精度.

4: 令 $s = s + 1$, 转至第 2 步.

由引理 4 可以看到, 由于 IFUSL 为光滑化的 IFAPL 算法, 所以也同样受到误差累积效应的影响. 在 δ 给定不变的情形下, (3.21) 右侧并不随 K 递增而收敛. 因此, 下面分别讨论在 δ 可以由用户选取

的情形下, IFUSL 算法需满足什么条件以保证收敛到 ϵ- 解; 以及在 δ 给定不变的情形下, IFUSL 算法所能达到的最佳精度. 对这两种情形, 我们都会给出 IFUSL 算法相应的迭代复杂度.

由引理 3 可以看到, 每次调用 $\mathcal{G}_{\text{IFUSL}}$ 或者将当前上下界差值缩小一定比例 q, 即 $\Delta^+ \leqslant q\Delta$, 或者将对 $D_{v,Y}$ 的估计 D 增加一倍, 即 $D^+ = 2D$. 为方便起见, 对每次调用 $\mathcal{G}_{\text{IFUSL}}$, 若 $\mathcal{G}_{\text{IFUSL}}$ 在第 2 或者 (3a) 步终止迭代, 则称这次调用为有效调用; 若 $\mathcal{G}_{\text{IFUSL}}$ 在第 (3b) 步终止迭代, 则称此次调用为无效调用.

对应于 IFAPL 算法中的定理 2, 有以下定理.

定理 8 在 $\mathcal{G}_{\text{IFUSL}}$ 中, 若步长 $\{\alpha_k\}$ 满足 (3.18), 且 δ 满足条件

$$\delta \leqslant \bar{\delta}_\Delta := \frac{(\frac{1}{3}\theta\beta\Delta)^{3/2}}{C_1C_2R\sqrt{L}}, \tag{3.25}$$

其中 L 在 (3.15) 中定义, C_1 和 C_2 在 (3.18) 中定义, θ 和 β 是 $\mathcal{G}_{\text{IFUSL}}$ 中选取的参数, Δ 是调用 $\mathcal{G}_{\text{IFUSL}}$ 时输入的上下界差值, 则有 $\mathcal{G}_{\text{IFUSL}}$ 的迭代次数不超过

$$\bar{N}(\Delta, D) := \frac{\sqrt{3}C_1R\sqrt{L_{\hat{f}}}}{\sqrt{\theta\beta\Delta}} + \frac{2\sqrt{3}C_1R\|A\|\sqrt{D}}{\theta\beta\Delta\sqrt{\sigma_v}} + 1. \tag{3.26}$$

证明 由引理 4 和 (3.25) 可以看到

$$f_\delta^\eta(x_K^u) + 2\delta - l \leqslant \frac{C_1^2LR^2}{2K^2} + C_2K\bar{\delta}_\Delta, \tag{3.27}$$

不等式右侧在 $K = \bar{K} := (\frac{C_1^2LR^2}{C_2\bar{\delta}_\Delta})^{\frac{1}{3}}$ 时取得其最小值, 即有

$$f_\delta^\eta(x_{\bar{K}}^u) + 2\delta - l \leqslant \frac{C_1^2LR^2}{2\bar{K}^2} + C_2\bar{K}\bar{\delta}_\Delta - 2\delta \leqslant \frac{3}{2}(C_1^2C_2^2LR^2\bar{\delta}_\Delta^2)^{1/3} - 2\delta \leqslant \frac{1}{2}\theta\beta\Delta. \tag{3.28}$$

由 $\frac{\theta}{2}(\overline{f}_0 - l) = \frac{1}{2}\theta\beta\Delta$, 有

$$f_\delta^\eta(x_{\bar{K}}^u) + 2\delta - l \leqslant \frac{\theta}{2}(\overline{f}_0 - l). \tag{3.29}$$

因此, 不超过 $\bar{K}$ 次迭代后 $\mathcal{G}_{\text{IFUSL}}$ 在第 (3b) 步中的条件必能满足而终止迭代. 由 (3.6)、(3.15) 和 (3.16), 可得

$$L_\eta = \frac{\|A\|^2}{\eta\sigma_v} = \frac{2D\|A\|^2}{\theta\beta\sigma_v\Delta}. \tag{3.30}$$

进而有 $\mathcal{G}_{\text{IFUSL}}$ 迭代次数不超过

$$\bar{K} = \sqrt{3}C_1R\sqrt{\frac{L}{\theta\beta\Delta}} \leqslant \frac{\sqrt{3}C_1R}{\sqrt{\theta\beta\Delta}}(\sqrt{L_{\hat{f}}} + \sqrt{2L_\eta}) = \frac{\sqrt{3}C_1R\sqrt{L_{\hat{f}}}}{\sqrt{\theta\beta\Delta}} + \frac{2\sqrt{3}C_1R\|A\|\sqrt{D}}{\theta\beta\Delta\sqrt{\sigma_v}}. \tag{3.31}$$

下面先讨论在 δ_s 可以由用户选取时, 对任意给定的 $\epsilon > 0$, 为保证 IFUSL 算法收敛到一个 ϵ- 解, 该算法需要满足的条件.

定理 9 对于任意给定 $\epsilon > 0$, 若 IFUSL 算法中 $\{\alpha_k\}$ 选取满足条件 (3.18), 且对 $s \geqslant 1$ 均有 $\delta_s \leqslant \frac{(\frac{1}{3}\theta\beta\Delta_s)^{3/2}}{C_1C_2R\sqrt{L}}$, 则 IFUSL 算法将收敛到 (3.1) 的一个 ϵ- 解, 且以下结论成立:

(1) $\mathcal{G}_{\text{IFUSL}}$ 调用的次数不超过

$$\bar{S}_\epsilon := \bar{S}_1 + \bar{S}_2, \tag{3.32}$$

其中 $\bar{S}_1$ 和 $\bar{S}_2$ 分别在 (3.35) 和 (3.36) 中定义;

(2) IFUSL 算法总迭代次数不超过

$$\bar{N}_\epsilon := \bar{N}_1 + \bar{N}_2, \tag{3.33}$$

其中 $\bar{N}_1$ 和 $\bar{N}_2$ 分别在 (3.38) 和 (3.39) 中定义.

证明 对 (1), 我们分别估计 IFUSL 算法得到 ϵ- 解所需要的 $\mathcal{G}_{\mathrm{IFUSL}}$ 有效调用次数和无效调用次数. 观察 IFUSL 算法的第 1 步及 $\|p_0 - p_1\| \leqslant 2R$, 并运用文献 [9, 引理 8], 有以下初始上下界差值的估计:

$$\begin{aligned}\Delta_1 &:= \mathrm{ub}_1 - \mathrm{lb}_1 \leqslant f(p_1) - h^0(p_0, p_1)\\ &\leqslant [F(p_0) - F(p_1) - \langle F'(p_1), p_0 - p_1\rangle] + [\hat{f}(p_0) - \hat{f}(p_1) - \langle \hat{f}'(p_1), p_0 - p_1\rangle]\\ &\leqslant 4\sqrt{2}R\|A\|\sqrt{\frac{D_{v,Y}}{\sigma_v}} + 2R^2 L_{\hat{f}}.\end{aligned} \tag{3.34}$$

运用引理 3, 由于每次 $\mathcal{G}_{\mathrm{IFUSL}}$ 的无效调用都有 $D^+ = 2D$, 且 $D^+ < 2D_{v,Y}$, 故无效调用的次数不超过

$$\bar{S}_1 := \max\left\{\log_2 \frac{D_{v,Y}}{D_1}, 0\right\} + 1. \tag{3.35}$$

同时由于每次 $\mathcal{G}_{\mathrm{IFUSL}}$ 的有效调用都有 $\Delta^+ \leqslant q\Delta$, 其中 q 在 (2.21) 定义, 因此 $\mathcal{G}_{\mathrm{IFUSL}}$ 的有效调用次数不超过

$$\bar{S}_2 := \max\left\{0, \log_{\frac{1}{q}}\left(\frac{\Delta_1}{\epsilon}\right)\right\} + 1 \leqslant \max\left\{0, \log_{\frac{1}{q}}\left(\frac{4\sqrt{2}R\|A\|\sqrt{\frac{D_{v,Y}}{\sigma_v}} + 2R^2 L_{\hat{f}}}{\epsilon}\right)\right\} + 1. \tag{3.36}$$

综上即有 $\mathcal{G}_{\mathrm{IFUSL}}$ 总的调用次数不超过 $\bar{S}_\epsilon = \bar{S}_1 + \bar{S}_2$.

对 (2), 仿照文献 [10, 定理 3.8] 的证明, 我们可以分别估计 IFUSL 算法中有效调用和无效调用 $\mathcal{G}_{\mathrm{IFUSL}}$ 的总迭代次数. 假设 $\{m_1, m_2, \ldots, m_{\bar{S}_1}\}$ 和 $\{n_1, n_2, \ldots, n_{\bar{S}_2}\}$ 分别为无效调用和有效调用, 即在 IFUSL 算法中, 若 $s = m_k$, $1 \leqslant k \leqslant \bar{S}_1$, 则该次调用为无效调用; 若 $s = n_k$, $1 \leqslant k \leqslant \bar{S}_2$, 则该次调用为有效调用. 令

$$\tilde{D} := \max\{D_1, 2D_{v,Y}\}, \tag{3.37}$$

注意到 $\forall m_k$, 其内循环迭代次数不超过 $\bar{N}(\Delta_{m_k}, D_{m_k})$, 且 $\Delta_{m_k} > \epsilon$, $D_{m_{k+1}} = 2D_{m_k}$; 同样 $\forall n_k$, 其内循环迭代次数不超过 $\bar{N}(\Delta_{n_k}, D_{n_k})$, 且 $\Delta_{n_{k+1}} \leqslant q\Delta_{n_k}, D_{n_k} < \tilde{D}$. 因此有无效调用总迭代次数不超过

$$\begin{aligned}\bar{N}_1 &:= \sum_{k=1}^{\bar{S}_1} \bar{N}(\Delta_{m_k}, D_{m_k}) \leqslant \sum_{k=1}^{\bar{S}_1} \bar{N}\left(\epsilon, \frac{\tilde{D}}{2^{\bar{S}_1 - k}}\right)\\ &\leqslant \bar{S}_1\left(\frac{\sqrt{3}C_1 R\sqrt{L_{\hat{f}}}}{\sqrt{\theta\beta\epsilon}} + 1\right) + \frac{2\sqrt{3}C_1 R\|A\|\sqrt{\tilde{D}}}{\theta\beta\epsilon\sqrt{\sigma_v}} \sum_{k=1}^{\bar{S}_1} 2^{-\frac{(\bar{S}_1 - k)}{2}}\\ &\leqslant \bar{S}_1\left(\frac{\sqrt{3}C_1 R\sqrt{L_{\hat{f}}}}{\sqrt{\theta\beta\epsilon}} + 1\right) + \frac{(4\sqrt{3} + 2\sqrt{6})C_1 R\|A\|\sqrt{\tilde{D}}}{\theta\beta\epsilon\sqrt{\sigma_v}}.\end{aligned} \tag{3.38}$$

有效调用总迭代次数不超过

$$\bar{N}_2 := \sum_{k=1}^{\bar{S}_2} \bar{N}(\Delta_{n_k}, D_{n_k}) \leqslant \sum_{k=1}^{\bar{S}_2} \bar{N}\left(\frac{\epsilon}{q^{\bar{S}_2 - k}}, \tilde{D}\right)$$

$$
\begin{aligned}
&\leqslant \bar{S}_2 + \frac{\sqrt{3}C_1R\sqrt{L_{\hat{f}}}}{\sqrt{\theta\beta\epsilon}}\sum_{k=1}^{\bar{S}_2} q^{\frac{\bar{S}_2-k}{2}} + \frac{2\sqrt{3}C_1R\|A\|\sqrt{\tilde{D}}}{\theta\beta\epsilon\sqrt{\sigma_v}}\sum_{k=1}^{\bar{S}_2} q^{\bar{S}_2-k} \\
&\leqslant \bar{S}_2 + \frac{\sqrt{3}C_1R\sqrt{L_{\hat{f}}}}{(1-\sqrt{q})\sqrt{\theta\beta\epsilon}} + \frac{2\sqrt{3}C_1R\|A\|\sqrt{\tilde{D}}}{(1-q)\theta\beta\epsilon\sqrt{\sigma_v}}. \qquad (3.39)
\end{aligned}
$$

因此总迭代次数不超过 $\bar{N}_\epsilon = \bar{N}_1 + \bar{N}_2$. □

下面讨论在 $\delta_s = \delta$ 给定不变的情形下, IFUSL 算法所能达到的最佳精度. 由 (3.10) 知, f 与 f^η 的距离与 $D_{v,Y}$ 直接相关, 而在 IFUSL 算法中, 为了保持该算法不需要输入任何与 f 相关参数的特性, 我们使用的是对 $D_{v,Y}$ 的估计 D, 而该估计只有在算法最终收敛到 ϵ- 解时才能保证 $D_{v,Y} \leqslant D < 2D_{v,Y}$. 观察 (3.21) 右侧在 L、R 和 δ 固定的情形下并不随 K 的递增而收敛, 因此, IFUSL 算法受到误差累积效应影响而不能保证收敛到 ϵ- 解, 且无法保证 D 与 $D_{v,Y}$ 接近. 基于上述两点, 在 δ 给定不变时, IFUSL 算法需作以下调整:

(M3a) 给定 $D_{v,Y}$ 作为 IFUSL 算法的输入, 即 IFUSL 第 3 步中 $D_s = D_{v,Y}$, $\forall s \geqslant 1$.

(M3b) 将 IFUSL 算法中第 2 步条件 $\mathrm{ub}_s - \mathrm{lb}_s \leqslant \epsilon$ 改成 $\mathrm{ub}_s - \mathrm{lb}_s \leqslant \bar{\epsilon}_\delta$, 其中

$$
\bar{\epsilon}_\delta := \frac{1}{\theta\beta}\max\left\{3(2C_1^2C_2^2R^2L_{\hat{f}}\delta^2)^{1/3}, \left(6^3C_1^2C_2^2R^2D_{v,Y}\|A\|^2\frac{\delta^2}{\sigma_v}\right)^{1/4}\right\}. \qquad (3.40)
$$

在上述改动下, 我们有以下定理.

定理 10 在 IFUSL 算法中, 若对所有 $s \geqslant 1$ 均有 $\delta_s = \delta$, $\{\alpha_k\}$ 的选取满足 (3.18), 则 IFUSL 算法将收敛到 (3.1) 的一个 $\bar{\epsilon}_\delta$- 解, 并且 $\mathcal{G}_{\mathrm{IFUSL}}$ 调用的次数不超过

$$
\bar{S}_\delta := \max\left\{0, \log_{\frac{1}{q}}\left(\frac{4\sqrt{2}R\|A\|\sqrt{\frac{D_{v,Y}}{\sigma_v}} + 2R^2L_{\hat{f}}}{\bar{\epsilon}_\delta}\right)\right\} + 1, \qquad (3.41)
$$

IFUSL 的总迭代次数不超过

$$
\bar{N}_\delta := \bar{S}_\delta + \frac{\sqrt{3}C_1R\sqrt{L_{\hat{f}}}}{(1-\sqrt{q})\sqrt{\theta\beta\bar{\epsilon}_\delta}} + \frac{2\sqrt{3}C_1R\|A\|\sqrt{D_{v,Y}}}{(1-q)\theta\beta\bar{\epsilon}_\delta\sqrt{\sigma_v}}. \qquad (3.42)
$$

证明 当 $\Delta = \mathrm{ub} - \mathrm{lb} < \bar{\epsilon}_\delta$ 时, IFUSL 算法满足 (M3b) 中条件而终止. 因此只需考虑 $\Delta \geqslant \bar{\epsilon}_\delta$ 的情形, 由 (3.6)、(3.15) 和 (3.16) 有 $C_1^2C_2^2R^2L\delta^2 = C_1^2C_2^2R^2\delta^2(L_f + \frac{4D\|A\|^2}{\theta\beta\sigma_v\Delta})$. 结合 (3.40), 进一步可得到

$$
C_1^2C_2^2R^2\delta^2L_f \leqslant \frac{(\theta\beta\Delta)^3}{54} \quad 和 \quad C_1^2C_2^2R^2\delta^2\cdot\frac{4D\|A\|^2}{\theta\beta\sigma_v\Delta} \leqslant \frac{(\theta\beta\Delta)^3}{54},
$$

因此定理 8 中 (3.25) 成立, 此时 $\mathcal{G}_{\mathrm{IFUSL}}$ 的迭代次数不超过 $\bar{N}(\Delta, D_{v,Y})$. 进一步, 由于 $D = D_{v,Y}$, $\mathcal{G}_{\mathrm{IFUSL}}$ 不可能在第 (3b) 步中终止迭代, 由引理 3 可知, 每次调用 $\mathcal{G}_{\mathrm{IFUSL}}$ 均为有效调用, 即有 $\Delta^+ \leqslant q\Delta$. 结合 (3.34), 便有 $\mathcal{G}_{\mathrm{IFUSL}}$ 调用次数不超过 $\bar{S}_\delta$. 进而, IFUSL 算法总迭代次数不超过

$$
\begin{aligned}
\sum_{s=1}^{\bar{S}_\delta}\bar{N}(\Delta_s, D_{v,Y}) &\leqslant \sum_{k=1}^{\bar{S}_\delta}\bar{N}\left(\frac{\bar{\epsilon}_\delta}{q^{\bar{S}_\delta-k}}, D_{v,Y}\right) \\
&\leqslant \bar{S}_\delta + \frac{\sqrt{3}C_1R\sqrt{L_{\hat{f}}}}{\sqrt{\theta\beta\bar{\epsilon}_\delta}}\sum_{k=1}^{\bar{S}_\delta} q^{\frac{\bar{S}_\delta-k}{2}} + \frac{2\sqrt{3}C_1R\|A\|\sqrt{D_{v,Y}}}{\theta\beta\bar{\epsilon}_\delta\sqrt{\sigma_v}}\sum_{k=1}^{\bar{S}_\delta} q^{\bar{S}_\delta-k} \\
&\leqslant \bar{S}_\delta + \frac{\sqrt{3}C_1R\sqrt{L_{\hat{f}}}}{(1-\sqrt{q})\sqrt{\theta\beta\bar{\epsilon}_\delta}} + \frac{2\sqrt{3}C_1R\|A\|\sqrt{D_{v,Y}}}{(1-q)\theta\beta\bar{\epsilon}_\delta\sqrt{\sigma_v}}. \qquad (3.43)
\end{aligned}
$$

3.2　解强凸函数的 IFUSLS 算法

本小节将 IFUSL 算法推广到强凸函数上, 来求解 $\mathbb{R}^n$ 上的一类具有特殊结构的鞍点问题:

$$f^* = \min_{x\in\mathbb{R}^n} f(x) := \hat{f}(x) + F(x), \tag{3.44}$$

其中 $\hat{f}(x)$ 和 $F(x)$ 分别在 (3.2) 和 (3.3) 中定义, 且 $\hat{f}$ 为光滑的 μ- 强凸函数, 即

$$\frac{\mu}{2}\|y-x\|^2 \leqslant \hat{f}(y) - \hat{f}(x) - \langle \hat{f}'(x), y-x\rangle \leqslant \frac{L_{\hat{f}}}{2}\|y-x\|^2, \quad \forall\, x, y \in \mathbb{R}^n, \tag{3.45}$$

其中 $\mu > 0$. 由 $\hat{f}$ 的强凸性容易得到 f 和 f^η 同样为 μ- 强凸函数. 与 IFUSL 算法一样, 假设 F^η 的近似一阶信息满足条件 (C1), 即有 $f(x)$ 满足 $(2\delta, L, \mu)$- 模型, 对 $\forall\,\eta > 0$, 有

$$\frac{\mu}{2}\|y-x\|^2 \leqslant f^\eta(y) - (f_\delta^\eta(x) + \langle g_\delta^\eta(x), y-x\rangle) \leqslant \frac{L}{2}\|y-x\|^2 + 2\delta, \quad \forall\, x, y \in \mathbb{R}^n, \tag{3.46}$$

其中 L、f_δ^η 和 g_δ^η 在 (3.15) 中定义. 为简单起见, 本节假设存在并给出一个 f^* 的初始下界 lb_1. 与 IFAPLS 算法一样, 对任意初始点 p_0, 利用 f 的强凸性, 有 $\|p_0 - x^*\|^2 \leqslant 2[f(p_0) - \mathrm{lb}_1]/\mu$, 这意味着 x^* 是在 Euclid 球 $B(p_0, \sqrt{2[f(p_0) - \mathrm{lb}_0]/\mu})$ 中. 因此, 在每次调用 $\mathcal{G}_{\text{IFUSLS}}$ 时, 我们可以利用当前的上下界来调整搜索球的球心和半径, 进而提高算法的收敛速度以得到更优的迭代复杂度.

我们先给出 IFUSLS 算法, 相比较于 IFUSL 算法, IFUSLS 算法需作以下改动:

过程 4　IFUSLS 内迭代子程序: $(x^+, D^+, \mathrm{ub}^+, \mathrm{lb}^+) = \mathcal{G}_{\text{IFUSLS}}(\hat{x}, D, \mathrm{ub}, \mathrm{lb}, r, \beta, \theta)$

在过程 3, 令 $\overline{x} = \hat{x}$, 并将 (2.6) 中 prox 函数 d 替换成 $\|x - \hat{x}\|^2/2$.

算法 4　IFUSLS 算法

在算法 3, 将第 0、1 和 3 步分别改成

0: 选取初始下界 $\mathrm{lb}_1 \leqslant f^*$, 选定 (3.4) 和 (3.5) 中的强凸函数 $v(\cdot)$, (3.8) 中对 $D_{v,Y}$ 的初始猜测 D_1, 初始点 $p_0 \in \mathbb{R}^n$, 初始模型精度 δ_0, 初始上界 $\mathrm{ub}_1 = f_{\delta_0}(p_0) + \delta_0$, 给定目标精度 $\epsilon > 0$ 以及参数 $\beta, \theta \in (0,1)$.

1: 令 $\hat{x}_1 = p_0$, 设 $s = 1$.

3: 调用 $(\hat{x}_{s+1}, D_{s+1}, \mathrm{ub}_{s+1}, \mathrm{lb}_{s+1}) = \mathcal{G}_{\text{IFUSLS}}(\hat{x}_s, D_s, \mathrm{ub}_s, \mathrm{lb}_s, \sqrt{2(\mathrm{ub}_s - \mathrm{lb}_s)/\mu}, \beta, \theta, \delta_s)$, δ_s 为当前模型精度.

以下讨论 IFUSLS 算法在 δ 可由用户选取和给定不变两种情形下的收敛性和迭代复杂度. 由于 IFUSLS 算法与 IFUSL 算法的内迭代子程序基本一致, 两者的主要区别是 IFUSLS 算法每次调用内迭代子程序时会调整当前搜索球的球心和半径. 因此, 引理 3、4 和定理 8 同样对 IFUSLS 算法成立. 对于 δ 可以由用户选取的情形, 有以下定理.

定理 11　对于任意给定 $\epsilon > 0$, 若 IFUSLS 算法中 $\{\alpha_k\}$ 选取满足条件 (3.18), 且对 $s \geqslant 1$ 均有

$$\delta_s \leqslant \tilde{\delta}_{\Delta_s} := \frac{(\frac{1}{3}\theta\beta)^{3/2}\sqrt{\mu}}{\sqrt{2}C_1C_2\sqrt{L}}\Delta_s, \tag{3.47}$$

则 IFUSLS 算法将收敛到 (3.44) 的一个 ϵ- 解, 且以下结论成立:

(1) 对任意 $\mathcal{G}_{\text{IFUSLS}}$, 其迭代次数不超过

$$\tilde{N}(\Delta, D) := \sqrt{6}C_1\sqrt{\frac{L_{\hat{f}}}{\theta\beta\mu}} + \frac{2\sqrt{6}C_1\|A\|}{\theta\beta}\sqrt{\frac{D}{\sigma_v\mu\Delta}} + 1; \tag{3.48}$$

(2) 内迭代子程序 $\mathcal{G}_{\text{IFUSLS}}$ 的调用次数不超过

$$\tilde{S}_\epsilon := \max\left\{0, \log_2\frac{D_{v,Y}}{D_1}\right\} + \max\left\{0, \log_{\frac{1}{q}}\left(\frac{\text{ub}_1 - \text{lb}_1}{\epsilon}\right)\right\} + 2; \tag{3.49}$$

(3) IFUSLS 算法总迭代次数不超过

$$\tilde{N}_\epsilon := \tilde{S}_\epsilon\left(\sqrt{6}C_1\sqrt{\frac{L_{\hat{f}}}{\theta\beta\mu}} + 1\right) + \left(\frac{\sqrt{2}}{\sqrt{2}-1} + \frac{1}{1-\sqrt{q}}\right)\frac{2\sqrt{6}C_1\|A\|}{\theta\beta}\sqrt{\frac{\tilde{D}}{\sigma_v\mu\epsilon}}, \tag{3.50}$$

其中 $\tilde{D}$ 在 (3.37) 中定义.

证明 对 (1), 仿照定理 8 的证明, 并利用 $R = \sqrt{\frac{2\Delta}{\mu}}$ 和 $\delta_s \leqslant \tilde{\delta}_{\Delta_s}$, 即有 (3.25) 成立. 因此, $\mathcal{G}_{\text{IFUSLS}}$ 的迭代次数不超过

$$\tilde{N}(\Delta, D) := \frac{\sqrt{3}C_1R\sqrt{L_{\hat{f}}}}{\sqrt{\theta\beta\Delta}} + \frac{2\sqrt{3}C_1R\|A\|\sqrt{D}}{\theta\beta\Delta\sqrt{\sigma_v}} + 1 = \sqrt{6}C_1\sqrt{\frac{L_{\hat{f}}}{\theta\beta\mu}} + \frac{2\sqrt{6}C_1\|A\|}{\theta\beta}\sqrt{\frac{D}{\sigma_v\mu\Delta}} + 1. \tag{3.51}$$

对 (2), 由定理 8 的证明可得无效调用次数不超过 $\bar{S}_1$, 其中 $\bar{S}_1$ 在 (3.35) 中定义. 有效调用次数不超过 $\tilde{S}_2 := \max\{0, \log_{\frac{1}{q}}(\frac{\text{ub}_1-\text{lb}_1}{\epsilon})\} + 1$, 因此 (2) 得证.

对 (3), 与 IFUSL 类似, 我们可以分别估计有效调用和无效调用总的迭代次数 $\tilde{N}_1$ 和 $\tilde{N}_2$. 我们有 IFUSLS 算法的总迭代次数不超过

$$\begin{aligned}\tilde{N}_1 + \tilde{N}_2 &\leqslant \tilde{S}_\epsilon\left(\sqrt{6}C_1\sqrt{\frac{L_{\hat{f}}}{\theta\beta\mu}} + 1\right) + \frac{2\sqrt{6}C_1\|A\|}{\theta\beta}\sqrt{\frac{\tilde{D}}{\sigma_v\mu\epsilon}}\left(\sum_{k=1}^{\bar{S}_1} 2^{-\frac{(\bar{S}_1-k)}{2}} + \sum_{k=1}^{\tilde{S}_2} q^{\frac{\tilde{S}_2-k}{2}}\right)\\ &\leqslant \tilde{S}_\epsilon\left(\sqrt{6}C_1\sqrt{\frac{L_{\hat{f}}}{\theta\beta\mu}} + 1\right) + \left(\frac{\sqrt{2}}{\sqrt{2}-1} + \frac{1}{1-\sqrt{q}}\right)\frac{2\sqrt{6}C_1\|A\|}{\theta\beta}\sqrt{\frac{\tilde{D}}{\sigma_v\mu\epsilon}}.\end{aligned} \tag{3.52}$$

由上述定理可以看到 IFUSLS 收敛到 (3.44) 的一个 ϵ- 解的迭代复杂度为 $\mathcal{O}(\frac{\|A\|}{\sqrt{\epsilon}})$, 若进一步 $D_{v,Y}$ 给定作为输入参数, 则所有 $\mathcal{G}_{\text{IFUSLS}}$ 的调用均为有效调用, 因此 (3.50) 中迭代次数 $\tilde{N}_\epsilon = \tilde{N}_2$, 此估计可以进一步缩紧. 下面讨论在 $\delta_s = \delta$ $(\forall\, s \geqslant 1)$ 时, IFUSLS 算法所能达到的最佳精度及其迭代复杂度. □

与 IFUSL 算法 δ 给定时的情形一样, 假设 $D_{v,Y}$ 作为给定参数输入. IFUSLS 算法需作以下调整:

(M4a) 给定 $D_{v,Y}$ 作为 IFUSLS 算法的输入, 即 IFUSLS 第 3 步中 $D_s = D_{v,Y}$, $\forall\, s \geqslant 1$.

(M4b) 将 IFUSL 算法中第 2 步条件 $\text{ub}_s - \text{lb}_s \leqslant \epsilon$ 改成 $\text{ub}_s - \text{lb}_s \leqslant \tilde{\epsilon}_\delta$, 其中

$$\tilde{\epsilon}_\delta := \max\left(\frac{6\sqrt{3}C_1C_2\delta\sqrt{L_{\hat{f}}}}{\mu(\theta\beta)^{3/2}}, 6\left(\frac{C_1^2C_2^2D_{v,Y}\|A\|^2\delta^2}{\mu\theta^4\beta^4\sigma_v}\right)^{1/3}\right). \tag{3.53}$$

定理 12 在 IFUSLS 算法中, 若 $\delta_s = \delta$ $(\forall\, s \geqslant 1)$ 给定不变, 且 $\{\alpha_k\}$ 选取满足条件 (3.18), 则 IFUSLS 算法将收敛到 (3.44) 的一个 $\tilde{\epsilon}_\delta$- 解, 并且 $\mathcal{G}_{\text{IFUSLS}}$ 调用的总次数不超过

$$\tilde{S}_\delta := \log_{\frac{1}{q}}\left(\frac{\text{ub}_1 - \text{lb}_1}{\tilde{\epsilon}_\delta}\right) + 1, \tag{3.54}$$

IFUSLS 的总迭代次数不超过

$$\tilde{N}_\delta := \tilde{S}_\delta\left(\sqrt{6}C_1\sqrt{\frac{L_{\hat{f}}}{\theta\beta\mu}}+1\right)+\frac{2\sqrt{6}C_1\|A\|}{\theta\beta(1-\sqrt{q})}\sqrt{\frac{D_{v,Y}}{\sigma_v\mu\tilde{\epsilon}_\delta}}. \tag{3.55}$$

证明 当 $\Delta=\mathrm{ub}-\mathrm{lb}\leqslant\tilde{\epsilon}_\delta$ 时, IFUSLS 算法满足 (M4b) 而终止, 因此只需考虑 $\Delta>\tilde{\epsilon}_\delta$ 的情形. 由 (3.6)、(3.15) 和 (3.16) 有 $C_1^2C_2^2\delta^2L\leqslant C_1^2C_2^2\delta^2(L_{\hat{f}}+\frac{4D_{v,Y}\|A\|^2}{\theta\beta\sigma_v\Delta})$, 结合 (3.53) 即得 $4C_1^2C_2^2\delta^2L_{\hat{f}}\leqslant(\frac{1}{3}\theta\beta)^3\mu\Delta^2$ 和 $4C_1^2C_2^2\delta^2\cdot\frac{4D_{v,Y}\|A\|^2}{\theta\beta\sigma_v\Delta}\leqslant(\frac{1}{3}\theta\beta)^3\mu\Delta^2$. 因此, 定理 11 中条件 (3.47) 满足, 此时每次调用 $\mathcal{G}_{\mathrm{IFUSLS}}$ 的迭代次数不超过 (3.48) 中 $\tilde{N}(\Delta,D_{v,Y})$. 由于 $D=D_{v,Y}$, 每次调用均为有效调用, $\Delta^+\leqslant q\Delta$, 因此, $\mathcal{G}_{\mathrm{IFUSLS}}$ 的调用次数不超过 $\tilde{S}_\delta$.

进而, IFUSLS 算法为得到一个 $\tilde{\epsilon}_\delta$- 解, 总的迭代次数不超过

$$\begin{aligned}\sum_{s=1}^{\tilde{S}_\delta}\tilde{N}(\Delta_s,D_{v,Y})&\leqslant\sum_{k=1}^{\tilde{S}_\delta}\tilde{N}\left(\frac{\tilde{\epsilon}_\delta}{q^{\tilde{S}_\delta-k}},D_{v,Y}\right)\leqslant\tilde{S}_\delta\left(\sqrt{6}C_1\sqrt{\frac{L_{\hat{f}}}{\theta\beta\mu}}+1\right)+\frac{2\sqrt{6}C_1\|A\|}{\theta\beta}\sqrt{\frac{D_{v,Y}}{\sigma_v\mu\tilde{\epsilon}_\delta}}\sum_{k=1}^{\tilde{S}_\delta}q^{(\tilde{S}_\delta-k-1)/2}\\&\leqslant\tilde{S}_\delta\left(\sqrt{6}C_1\sqrt{\frac{L_{\hat{f}}}{\theta\beta\mu}}+1\right)+\frac{2\sqrt{6}C_1\|A\|}{\theta\beta(1-\sqrt{q})}\sqrt{\frac{D_{v,Y}}{\sigma_v\mu\tilde{\epsilon}_\delta}}.\end{aligned}$$

证毕. □

参考文献

1 Nemirovskii A, Yudin D B, Dawson E R. Problem Complexity and Method Efficiency in Optimization. New York: A Wiley-Interscience Publication, 1983

2 Nemirovski A S, Nesterov Y E. Optimal methods of smooth convex minimization. Zh Vychisl Mat Mat Fiz, 1985, 25: 356–369

3 Nesterov Y. On an approach to the construction of optimal methods of minimization of smooth convex functions. Ekonomika Mat Metody, 1988, 24: 509–517

4 Devolder O, Glineur F, Nesterov Y. First-order methods of smooth convex optimization with inexact oracle. Math Program, 2014, 146: 37–75

5 Lemaréchal C, Nemirovskii A, Nesterov Y. New variants of bundle methods. Math Program, 1995, 69: 111–147

6 Kiwiel K C. Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. Math Program, 1995, 69: 89–109

7 Brännlund U, Kiwiel K C, Lindberg P O. A descent proximal level bundle method for convex nondifferentiable optimization. Oper Res Lett, 1995, 17: 121–126

8 Ben-Tal A, Nemirovski A. Non-euclidean restricted memory level method for large-scale convex optimization. Math Program, 2005, 102: 407–456

9 Lan G H. Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization. Math Program, 2013, 149: 1–45

10 Chen Y M, Lan G H, Ouyang Y Y, et al. Fast bundle level type methods for unconstrained and ball-constrained convex optimization. ArXiv:1412.2128, 2014

11 Kelley Jr J E. The cutting-plane method for solving convex programs. J Soc Ind Appl Math, 1960, 8: 703–712

12 Kiwiel K C. Proximity control in bundle methods for convex nondifferentiable minimization. Math Program, 1990, 46: 105–122

13 van Ackooij W, Sagastizábal C. Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. SIAM J Optim, 2014, 24: 733–765

14 Oliveira W, Sagastizábal C, Scheimberg S. Inexact bundle methods for two-stage stochastic programming. SIAM J Optim, 2011, 21: 517–544

15 de Oliveira W, Sagastizábal C, Lemaréchal C. Convex proximal bundle methods in depth: A unified analysis for inexact oracles. Math Program, 2014, 148: 241–277

16 Richtárik P. Approximate level method for nonsmooth convex minimization. J Optim Theory Appl, 2012, 152: 334–350

17 Noll D. Bundle method for non-convex minimization with inexact subgradients and function values. In: Computational and Analytical Mathematics. New York: Springer, 2013, 555–592

18 Kiwiel K C. Bundle methods for convex minimization with partially inexact oracles. Comput Optim Appl, 2012, 1702: 1703
19 de Oliveira W, Sagastizábal C. Level bundle methods for oracles with on-demand accuracy. Optim Methods Software, 2014, 29: 1180–1209
20 Kiwiel K C, Lemaréchal C. An inexact bundle variant suited to column generation. Math Program, 2009, 118: 177–206
21 Emiel G, Sagastizábal C. Incremental-like bundle methods with application to energy planning. Comput Optim Appl, 2010, 46: 305–332
22 Kiwiel K C. An inexact bundle approach to cutting-stock problems. INFORMS J Comput, 2010, 22: 131–143
23 Fábián C I, Szőke Z. Solving two-stage stochastic programming problems with level decomposition. Comput Manag Sci, 2007, 4: 313–353
24 Lin H L. An inexact spectral bundle method for convex quadratic semidefinite programming. Comput Optim Appl, 2012, 53: 45–89
25 Pang L P, Shen J. An approximate bundle method for solving variational inequalities. Commun Optim Theory, 2012, 1: 1–18
26 Hintermüller M. A proximal bundle method based on approximate subgradients. Comput Optim Appl, 2001, 20: 245–266
27 Fábián C I. Bundle-type methods for inexact data. CEJOR Cent Eur J Oper Res, 2000, 8: 35–55
28 Kiwiel K C. A method of centers with approximate subgradient linearizations for nonsmooth convex optimization. SIAM J Optim, 2008, 18: 1467–1489
29 Solodov M V. On approximations with finite precision in bundle methods for nonsmooth optimization. J Optim Theory Appl, 2003, 119: 151–165
30 Kiwiel K C. A proximal bundle method with approximate subgradient linearizations. SIAM J Optim, 2006, 16: 1007–1023
31 Zakeri G, Philpott A B, Ryan D M. Inexact cuts in benders decomposition. SIAM J Optim, 2000, 10: 643–657
32 Malick J, de Oliveira W, Zaourar S. Uncontrolled inexact information within bundle methods. Euro J Comput Optim, 2017, 5: 5–29
33 Devolder O, Glineur F, Nesterov Y, et al. First-Order Methods with Inexact Oracle: The Strongly Convex Case. Technical Report, UCL, 2013
34 Villa S, Salzo S, Baldassarre L, et al. Accelerated and inexact forward-backward algorithms. SIAM J Optim, 2013, 23: 1607–1633
35 Beck A, Teboulle M. Smoothing and first order methods: A unified framework. SIAM J Optim, 2012, 22: 557–580
36 Schmidt M, Roux N L, Bach F. Minimizing finite sums with the stochastic average gradient. Math Program, 2017, 162: 83–112
37 Jiang K F, Sun D F, Toh K C. An inexact accelerated proximal gradient method for large scale linearly constrained convex sdp. SIAM J Optim, 2012, 22: 1042–1064
38 Nesterov Y. Smooth minimization of non-smooth functions. Math Program, 2005, 103: 127–152

Accelerated bundle level methods with inexact oracle

CHEN YunMei & ZHANG Wei

Abstract In this paper, four accelerated bundle level methods are proposed to solve smooth and convex, smooth and strongly convex optimization problems and a class of saddle-point problems, respectively, by using the inexact first-order information of the objective functions. For each method two cases, where the accuracy of the oracle is chosen by the user and where the accuracy of the oracle is fixed in advance, are studied. The desired accuracy of the approximate solution and its corresponding iteration complexity of each proposed algorithm in each case are analyzed.

Keywords **accelerated algorithm, bundle level method, smooth optimization**

MSC(2010) **XXXX**

doi: 10.1360/N012016-00173