



一种求解非线性规划的半光滑 Newton 增广 Lagrange 函数方法

周宇昕¹, 吴嘉源², 王锐^{3*}, 宋洁², 文再文¹

1. 北京大学北京国际数学研究中心, 北京 100871;

2. 北京大学工学院工业工程与管理系, 北京 100871;

3. 西南财经大学数学学院, 成都 611130

E-mail: 2301213064@stu.pku.edu.cn, 1901110043@pku.edu.cn, ruiwang@bicmr.pku.edu.cn, songjie@coe.pku.edu.cn, wenzw@pku.edu.cn

收稿日期: 2023-09-23; 接受日期: 2024-09-22; 网络出版日期: 2025-03-03; * 通信作者

国家自然科学基金 (批准号: 12331010, 12288101 和 1240011671) 资助项目

摘要 增广 Lagrange 函数方法 (augmented Lagrangian method, ALM) 被广泛应用于求解非线性规划问题. 虽然在算法方面已经有很多深入研究, 但是对不等式约束的处理通常会在增广 Lagrange 函数子问题中保留盒子约束形式. 本文通过引入拆分将约束函数以及变量的盒子约束进行了等价重写, 从而借助投影算子在分离变量的基础上确立了增广 Lagrange 函数, 进而提出了一种子问题是无约束优化的 ALM 基本框架. 针对 ALM 子问题, 本文利用一种有效且计算代价低的技巧来逼近增广 Lagrange 函数的 Hessian 矩阵信息, 设计了求解此问题的近似半光滑 Newton 算法. 结合 KKT (Karush-Kuhn-Tucker) 条件以及外层 ALM 整体框架下变量变换的特性, 本文给出算法的收敛性和复杂度分析. 最后基于 CUTer (constrained and unconstrained testing environment) 数据集中的非线性规划问题实例, 对所提出的 ALM 与内点法和序列二次规划算法进行了数值实验和比较, 说明了算法的有效性.

关键词 非线性规划 增广 Lagrange 函数方法 KKT 条件 半光滑 Newton 法 收敛性 复杂度

MSC (2020) 主题分类 49M15, 65K05, 90C26

1 引言

非线性规划是最优化领域中的一个重要分支, 用于解决目标函数或约束条件存在非线性项的优化问题, 在计算机科学、机器学习、信号处理等领域的实际问题中有着广泛的应用. 本文考虑如下带有盒子约束的非线性规划问题:

$$\min_{x \in \mathbb{R}^n} f(x), \quad \text{s.t. } c(x) \in \mathcal{K}_c, \quad x \in \mathcal{K}_x, \quad (\text{P})$$

英文引用格式: Zhou Y X, Wu J Y, Wang R, et al. A semismooth Newton augmented Lagrangian method for solving nonlinear programming (in Chinese). Sci Sin Math, 2026, 56: 995–1016, doi: 10.1360/SSM-2023-0259

其中 $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $c(x) = [c_1(x), c_2(x), \dots, c_m(x)]^T: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 均是二次连续可微的函数, 且其中至少有一个函数为非线性函数. 约束集合

$$\mathcal{K}_c := \{w \in \mathbb{R}^m \mid \ell^c \leq w \leq u^c\}, \quad \mathcal{K}_x := \{v \in \mathbb{R}^n \mid \ell^x \leq v \leq u^x\},$$

其中 $\ell^c, u^c \in \mathbb{R}^m$ 和 $\ell^x, u^x \in \mathbb{R}^n$ 为 4 个常数向量, 在分量上允许有 $\ell = u$ 或 $\ell = \pm\infty$ 的无界情形出现. 问题形式 (P) 较为通用, 涵盖了一般的非线性规划问题. 该问题形式将原来关于变量 x 的限制单独分离出来进行了考虑, 且没有对等式和不等式进行特殊区分, 这为我们处理此类约束问题带来了挑战. 本文针对此问题特点, 借助增广 Lagrange 函数技术, 致力于设计一个与主流软件包相比具有竞争力的非线性规划求解器.

求解非线性规划问题的算法有很多, 如序列二次规划 (sequential quadratic programming, SQP)、内点法等, 这些算法各自有其适用的场合及不同的效果, 其中也包括本文所要重点讨论的增广 Lagrange 函数方法 (augmented Lagrangian method, ALM). ALM 最早于 1969 年由 Hestenes^[11] 和 Powell^[18] 提出, 近期因为机器学习等领域中各种新问题形式的出现又重新得到了国内外众多研究者的广泛关注. 总体上来说, ALM 形式简单, 其框架可以用来处理大规模约束优化问题, 在目标函数或约束集合凸或非凸情形下均可适用, 适用情境十分广泛. 与内点法相比, ALM 在接近约束边界时不会因为障碍函数的梯度发散而引入数值病态问题. 由于没有引入障碍项, 它在边界附近的数值行为可能更为稳定. 此外, ALM 可以与多种已有算法或理论结合, 设计新的算法, 得到新的理论结果, 因而关于其研究的前景也十分广阔.

近期, 有许多研究学者针对不同的问题形式研究了其对应适用的 ALM, 一部分研究还在此基础上对算法的收敛性与复杂度等性质作了探究. 例如, 对于非凸的二次规划问题, Hermans 等^[10] 应用了近似点迭代的 ALM 形式, 设计得到了 QPALM (quadratic programming augmented Lagrangian method), 它被验证在非凸二次规划情形下比 IPOPT (interior point optimizer) 更有竞争力. 对于有结构化几何约束的问题, Jia 等^[13] 在设计 ALM 中显式保留了这些约束, 并在求解内部子问题时使用了非凸集的谱梯度方法, 所得 ALM 的收敛性得到了理论和数值实验的验证. 进一步地, Marchi 等^[16] 研究了带约束的复合优化问题, 对其设计了一种灵活的 ALM 形式, 得到了算法理论收敛结果, 并基于推导出的求解器 ALS 进行了数值实验. 继续拓宽应用范围, 若将约束也融进目标函数, 考虑非凸且非光滑的复合目标函数形式, Hallak 和 Teboulle^[9] 在该问题上应用了自适应的 ALM 框架, 所提出的算法被验证在一定条件下可以收敛至稳定点, 算法达到稳定点的复杂度也得到了相关说明.

此外, 有一部分研究学者对 ALM 的收敛性与复杂度展开了相关的分析研究, 例如, Andreani 等^[3] 为研究 ALM 的收敛性引入了新的序列最优性条件 PCAKKT (positive complementary approximate KKT), 将其与已有的 KKT (Karush-Kuhn-Tucker) 等最优性条件进行了关系间的比较, 说明了一定约束准则条件下 ALM 所得序列点会收敛至满足 PCAKKT 条件的可行点的结论, 并将其运用的约束准则与其他标准准则作了对比, 说明了其理论的适用性. Rockafellar^[20] 则将近似点算法收敛性质的最新进展应用于 ALM 框架, 得到了关于 ALM 步长选择及其线性收敛性质的新的见解. 至于 ALM 的复杂度, Birgin 和 Martínez^[4] 基于一种公认的 ALM 框架 Algencan 算法出了关于它的众多复杂度结果, 研究考虑了惩罚因子有界或越界停止时 Algencan 算法停止所需要的最坏迭代次数以及对应的收敛情形, 同时还基于一种特殊的盒子约束求解方案分析了它的复杂度.

本文基于问题 (P) 设计了一种全新的可应用的 ALM, 具体贡献如下. (i) 在算法设计方面, 不同于经典的 ALM 在求解子问题时保留了盒子约束, 本文所提出算法的思想是通过约束进行拆分, 并借助在约束集合上的投影算子使得 ALM 子问题变成了一个无约束优化问题. 尽管已有部分文献阐述

过这种方法,但在求解非线性规划问题上的研究应用较少.除此之外,不同于文献 [13, 16],本文引入的 Lagrange 乘子可以是无界的,关于算法的停止准则和 Lagrange 乘子的更新规则也有所不同. (ii) 在算法子问题求解方面,结合增广 Lagrange 函数性质,本文考虑了半光滑 Newton 方法,并结合矩阵修正和信赖域方法等技术来逼近增广 Lagrange 函数的 Hessian 矩阵,降低函数二阶信息的计算代价,这使得子问题求解更加有效. (iii) 在算法收敛性方面,本文证明了所提出算法的所有可行聚点都满足问题 (P) 的渐近 KKT (approximate KKT, AKKT) 条件 (具体定义见第 2 节). 除此之外,本文分析了外层算法框架在惩罚因子有界或无界情形下满足一定条件终止所需要的最大迭代次数. 通过在 CUTeR (constrained and unconstrained testing environment) 测试集中的非线性规划问题上的数值实验,将 ALM 与 MATLAB 自带函数中的内点法和 SQP 进行比较,进一步验证了所提算法的准确性和有效性.

本文余下内容的结构如下. 第 2 节对本文所要用到的各类定义与一些基本的稳定性概念作相关说明. 第 3 节呈现我们关于问题形式 (P) 所设计的 ALM, 包括利用半光滑 Newton 方法对 ALM 子问题进行求解, 并对算法的收敛性、复杂度等性质进行一定的分析. 第 4 节基于一些基本的非线性规划问题实例给出所设计算法的数值实验结果, 验证说明算法的有效性. 第 5 节给出本文结论.

2 定义与基本概念

本节对本文所要用到的一些符号、预先定义、基本概念和有用结论进行说明.

2.1 预备知识

对于 $\ell, u \in \mathbb{R}^m$, 定义 $[\ell, u] := \{y \in \mathbb{R}^m \mid \ell \leq y \leq u\}$. 给定常数 $\rho > 0$, 定义集合 $\rho\mathcal{K}_c := \{w \in \mathbb{R}^m \mid \rho\ell^c \leq w \leq \rho u^c\}$, $\rho\mathcal{K}_x := \{v \in \mathbb{R}^n \mid \rho\ell^x \leq v \leq \rho u^x\}$. 定义 I_n 为 $n \times n$ 维的单位矩阵, 一个对称矩阵 A 的最小特征值记为 $\lambda_{\min}(A)$. 符号 $[\cdot]$ 表示取整.

为了更好地分析问题并建立相应的算法, 有必要引入一些变分分析中的理论工具. 首先给出稳定点的定义. 给定一个连续可微的函数 $h: \mathbb{R}^n \rightarrow \mathbb{R}$. 对于函数 h 的最小化问题, 如果 $\nabla h(x^*) = 0$, 则称 $x^* \in \text{dom } h$ 是该问题的稳定点. 此外, 对于某个 $\varepsilon \geq 0$, 如果 $\|\nabla h(x^*)\| \leq \varepsilon$, 则称 x^* 是 ε -稳定点. 接下来定义一个闭凸集 Ω 上的投影算子为

$$\mathcal{P}_\Omega(v) := \arg \min\{\|u - v\| : u \in \Omega\}.$$

为叙述方便, 引入算子

$$\mathcal{P}_{\Omega^\circ}(v) := v - \mathcal{P}_\Omega(v), \quad (2.1)$$

表示点与其在 Ω 上投影点的差. 具体到问题形式 (P) 中的两个集合, 可以得到

$$\mathcal{P}_{\mathcal{K}_c}(v) = \max(\min(v, u^c), \ell^c), \quad \mathcal{P}_{\mathcal{K}_x}(v) = \max(\min(v, u^x), \ell^x) \quad (2.2)$$

和

$$\mathcal{P}_{\mathcal{K}_c^\circ}(v) = \min(\max(v - u^c, 0), v - \ell^c), \quad \mathcal{P}_{\mathcal{K}_x^\circ}(v) = \min(\max(v - u^x, 0), v - \ell^x). \quad (2.3)$$

给定 $v \in \Omega$, 定义 Ω 在 v 处的法锥为

$$\mathcal{N}_\Omega(v) := \{z \in \mathbb{R}^n \mid z^T(v' - v) \leq 0, \forall v' \in \Omega\}.$$

对于 $v \notin \Omega$, 令 $\mathcal{N}_\Omega(v) := \emptyset$. 关于闭凸集 Ω , 定义的法锥与投影间有如下关系:

$$v = \mathcal{P}_\Omega(z) \Leftrightarrow z - v \in \mathcal{N}_\Omega(v). \quad (2.4)$$

具体到问题形式 (P) 中的两个集合, 对于 $v \in \mathcal{K}_c \subseteq \mathbb{R}^m$, $v' \in \mathcal{K}_x \subseteq \mathbb{R}^n$, 有

$$z \in \mathcal{N}_{\mathcal{K}_c}(v) \Leftrightarrow \begin{cases} z_i = 0, & \text{若 } \ell_i^c < v_i < u_i^c, \\ z_i \geq 0, & \text{若 } v_i = u_i^c, \\ z_i \leq 0, & \text{若 } v_i = \ell_i^c, \end{cases} \quad i = 1, 2, \dots, m, \quad (2.5)$$

$$z' \in \mathcal{N}_{\mathcal{K}_x}(v') \Leftrightarrow \begin{cases} z'_j = 0, & \text{若 } \ell_j^x < v'_j < u_j^x, \\ z'_j \geq 0, & \text{若 } v'_j = u_j^x, \\ z'_j \leq 0, & \text{若 } v'_j = \ell_j^x, \end{cases} \quad j = 1, 2, \dots, n. \quad (2.6)$$

注意到投影算子 $\mathcal{P}_\Omega(\cdot)$ 是 Lipschitz 连续的, 它使得我们引入如下次微分的概念 (参见文献 [5]). 给定一个向量值函数 $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$, 用分量函数的形式表示为 $F(x) = [f_1(x), f_2(x), \dots, f_m(x)]$. 假设每个 f_i 在 x 附近都是 Lipschitz 连续的, S_F 表示 F 不可微分的点集. 令 $\partial F(x)$ 表示函数 F 在点 x 处的广义 Jacobi 矩阵, 则 $\partial F(x) = \text{co}\{\lim JF(x_k) : x_k \rightarrow x, x_k \notin S_F\}$, 其中 $JF(x_k)$ 表示在 x_k 处的 Jacobi 矩阵.

2.2 KKT 条件与准则条件

为了分析本文所提算法的收敛性, 有必要引入关于问题形式 (P) 的一些基本的 KKT 条件及其对应的准则条件.

定义 2.1 (可行性) 一个点 $x^* \in \mathbb{R}^n$ 关于问题 (P) 是可行的, 若 $c(x^*) \in \mathcal{K}_c$ 且 $x^* \in \mathcal{K}_x$.

定义 2.2 (KKT 条件) 令 $x^* \in \mathbb{R}^n$ 为 (P) 的一个可行点. 若存在 $\lambda^* \in \mathbb{R}^m$ 和 $\mu^* \in \mathbb{R}^n$ 使得

$$\nabla f(x^*) + \nabla c(x^*)\lambda^* + \mu^* = 0, \quad (2.7a)$$

$$\lambda^* \in \mathcal{N}_{\mathcal{K}_c}(c(x^*)), \quad c(x^*) \in \mathcal{K}_c, \quad (2.7b)$$

$$\mu^* \in \mathcal{N}_{\mathcal{K}_x}(x^*), \quad x^* \in \mathcal{K}_x, \quad (2.7c)$$

其中 $\nabla c(x) = [\nabla c_1(x), \nabla c_2(x), \dots, \nabla c_m(x)] \in \mathbb{R}^{n \times m}$, 则 x^* 满足关于问题 (P) 的 KKT 条件.

接着, 参照文献 [1] 定义 KKT 条件对应的渐近形式, 称为渐近 KKT (AKKT) 条件.

定义 2.3 (AKKT 条件) 令 $x^* \in \mathbb{R}^n$ 为 (P) 的一个可行点. 若存在 $\{x^k\}, \{\epsilon^k\}, \{\mu^k\}, \{\xi^k\} \subset \mathbb{R}^n$ 和 $\{\lambda^k\}, \{\zeta^k\} \subset \mathbb{R}^m$ 使得 $x^k \rightarrow x^*$, $\epsilon^k \rightarrow 0$, $\xi^k \rightarrow 0$, $\zeta^k \rightarrow 0$ 以及

$$\nabla f(x^k) + \nabla c(x^k)\lambda^k + \mu^k = \epsilon^k, \quad (2.8a)$$

$$\lambda^k \in \mathcal{N}_{\mathcal{K}_c}(c(x^k) - \zeta^k), \quad c(x^*) \in \mathcal{K}_c, \quad (2.8b)$$

$$\mu^k \in \mathcal{N}_{\mathcal{K}_x}(x^k - \xi^k), \quad x^* \in \mathcal{K}_x, \quad (2.8c)$$

则 x^* 满足关于问题 (P) 的渐近 KKT 条件.

我们无法保证问题 (P) 的局部极小值点都是满足 KKT 条件的, 但可以参照文献 [1] 证明问题 (P) 的局部极小值点都是满足 AKKT 条件的.

命题 2.1 令 x^* 为问题 (P) 的任一局部极小值点, 则 x^* 满足关于问题 (P) 的 AKKT 条件.

证明 如果 x^* 为局部极小值点, 则存在 $\delta > 0$, 使得对于任何关于问题 (P) 的可行点 $x \in \mathbb{B}_\delta(x^*) := \{x \in \mathbb{R}^n \mid \|x - x^*\| \leq \delta\}$, 都有 $f(x) \geq f(x^*)$, 即 x^* 是问题

$$\min_{x \in \mathbb{R}^n} f(x) + \frac{1}{2} \|x - x^*\|^2, \quad \text{s.t. } c(x) \in \mathcal{K}_c, \quad x \in \mathcal{K}_x, \quad x \in \mathbb{B}_\delta(x^*) \quad (2.9)$$

唯一确定的全局极小值点. 考虑带惩罚项的问题

$$\min_{x \in \mathbb{R}^n} f(x) + \frac{1}{2} \|x - x^*\|^2 + \frac{k}{2} \|\mathcal{P}_{\mathcal{K}_c^o}(c(x))\|^2 + \frac{k}{2} \|\mathcal{P}_{\mathcal{K}_x^o}(x)\|^2, \quad \text{s.t. } x \in \mathbb{B}_\delta(x^*), \quad (\text{P}(k))$$

其中 $k \in \mathbb{N}$ 为任意正整数. 结合问题 (P(k)) 中目标函数的连续性与可行集的紧性, 问题 (P(k)) 对于任意 $k \in \mathbb{N}$ 存在全局极小值点 $x^k \in \mathbb{R}^n$. 由紧性可知, 存在子列收敛, 不失一般性地, 假设 $x^k \rightarrow \tilde{x}$, $\tilde{x} \in \mathbb{B}_\delta(x^*)$. 下面证明 $\tilde{x} = x^*$.

回到问题 (P(k)), 由于 x^* 在其可行集内, 故有

$$f(x^k) + \frac{1}{2} \|x^k - x^*\|^2 + \frac{k}{2} \|\mathcal{P}_{\mathcal{K}_c^o}(c(x^k))\|^2 + \frac{k}{2} \|\mathcal{P}_{\mathcal{K}_x^o}(x^k)\|^2 \leq f(x^*) \quad (2.10)$$

对所有 $k \in \mathbb{N}$ 成立. 令 $k \rightarrow \infty$, 由 (2.10) 左侧的连续性及右侧的有界性, 可得 $\mathcal{P}_{\mathcal{K}_c^o}(c(\tilde{x})) = 0$, $\mathcal{P}_{\mathcal{K}_x^o}(\tilde{x}) = 0$, 即 $\tilde{x}$ 为可行点, 则 $f(x^*) \leq f(\tilde{x})$. 进一步地, 在 (2.10) 令 $k \rightarrow \infty$, 得到

$$f(\tilde{x}) + \frac{1}{2} \|\tilde{x} - x^*\|^2 \leq f(x^*) \leq f(\tilde{x}).$$

因此, $\tilde{x} = x^*$, $x^k \rightarrow x^*$.

由 $x^k \rightarrow x^*$, 不失一般性地, 设序列 $\{x^k\}$ 均在 $\mathbb{B}_\delta(x^*)$ 的内部, 则 x^k 可视为问题 (P(k)) 中目标函数的局部极小值点, 即对于任意 $k \in \mathbb{N}$, 关于目标函数求导, 有

$$\nabla f(x^k) + k \nabla c(x^k) \mathcal{P}_{\mathcal{K}_c^o}(c(x^k)) + k \mathcal{P}_{\mathcal{K}_x^o}(x^k) + x^k - x^* = 0.$$

此时, 取 $\epsilon^k := x^* - x^k$, $\lambda^k := k \mathcal{P}_{\mathcal{K}_c^o}(c(x^k))$, $\mu^k := k \mathcal{P}_{\mathcal{K}_x^o}(x^k)$, $\zeta^k := c(x^k) - \mathcal{P}_{\mathcal{K}_c}(c(x^k))$, $\xi^k := x^k - \mathcal{P}_{\mathcal{K}_x}(x^k)$, 即可得到 (2.8a). 由于

$$\lambda^k = k \mathcal{P}_{\mathcal{K}_c^o}(c(x^k)) = k(c(x^k) - \mathcal{P}_{\mathcal{K}_c}(c(x^k))) \in \mathcal{N}_{\mathcal{K}_c}(\mathcal{P}_{\mathcal{K}_c}(c(x^k))) = \mathcal{N}_{\mathcal{K}_c}(c(x^k) - \zeta^k),$$

其中属于关系成立是根据 (2.4) 以及在凸集上法锥的性质得到的. 于是 (2.8b) 成立, 同理可得到 (2.8c), 因此 x^* 满足关于问题 (P) 的 AKKT 条件. $\square$

为了确保问题 (P) 的局部极小值点不仅满足 AKKT 条件, 而且能满足 KKT 条件, 我们有必要引入类似线性独立约束规范条件 (linear independence constraint qualification, LICQ). 参照文献 [2] 中定义的 CCP (cone continuity property) 条件, 关于问题形式 (P), 可以定义其对应的 AKKT 正则性条件作为准则条件.

定义 2.4 (AKKT 正则性) 令 $x^* \in \mathbb{R}^n$ 为 (P) 的一个可行点. 定义集值映射 $\mathcal{M}: \mathbb{R}^n \rightrightarrows \mathbb{R}^n$,

$$\mathcal{M}(x) := \nabla c(x)^T \mathcal{N}_{\mathcal{K}_c}(c(x)) + \mathcal{N}_{\mathcal{K}_x}(x).$$

称 x^* 关于问题 (P) 满足 AKKT 正则性条件, 若

$$\limsup_{x \rightarrow x^*} \mathcal{M}(x) \subset \mathcal{M}(x^*).$$

作为命题 2.1 的一个推论, 即可得到如下结论:

推论 2.1 令 $x^* \in \mathbb{R}^n$ 满足关于问题 (P) 的 AKKT 条件和 AKKT 正则性条件, 则 x^* 满足关于问题 (P) 的 KKT 条件. 此外, 任一满足 AKKT 正则性条件的问题 (P) 的局部极小值点, 也满足其对应的 KKT 条件.

上述命题证明可参见文献 [2], 所定义的 AKKT 正则性条件是使 AKKT 条件同时达成 KKT 条件所需要满足的最弱的严格准则条件, 弱于常见的 LICQ 和 MFCQ (Mangasarian-Fromovitz constraint qualification) 等易便捷判断的约束准则. 因而, 在特定约束问题下, 若可行点满足 LICQ 或 MFCQ 条件, 则它同时也必定满足 AKKT 正则性条件.

3 增广 Lagrange 函数方法 (ALM)

如 (P) 形式的非线性规划问题可以借助 ALM 来解决. 通过分离变量的方法, 引入变量 $y \in \mathbb{R}^m$, $z \in \mathbb{R}^n$, 可以将问题 (P) 重写为

$$\min_{x \in \mathbb{R}^n} f(x), \quad \text{s.t. } c(x) = y, \quad x = z, \quad y \in \mathcal{K}_c, \quad z \in \mathcal{K}_x.$$

保持后两项的盒子约束不变, 惩罚其他的等式约束, 相应的增广 Lagrange 函数为

$$L_\rho(x, y, z; \lambda, \mu) = f(x) + \frac{1}{2\rho} \|\rho(c(x) - y) + \lambda\|^2 + \frac{1}{2\rho} \|\rho(x - z) + \mu\|^2, \quad (3.1)$$

其中 Lagrange 乘子 $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^n$, 惩罚因子 $\rho > 0$. 为了在函数 (3.1) 中消去额外的变量 y 和 z , 以使得增广 Lagrange 函数仅与变量 x 有关, 我们求解如下问题:

$$\min_{y, z} L_\rho(x, y, z; \lambda, \mu), \quad \text{s.t. } y \in \mathcal{K}_c, \quad z \in \mathcal{K}_x, \quad (3.2)$$

可以得到该问题的最优解 (y_x^*, z_x^*) 为

$$y_x^* = \mathcal{P}_{\mathcal{K}_c} \left(c(x) + \frac{\lambda}{\rho} \right), \quad z_x^* = \mathcal{P}_{\mathcal{K}_x} \left(x + \frac{\mu}{\rho} \right).$$

代入函数 (3.1) 中, 即得到最终仅与 x 有关的增广 Lagrange 函数

$$\begin{aligned} \mathcal{L}_\rho(x; \lambda, \mu) &= L_\rho(x, y_x^*, z_x^*; \lambda, \mu) \\ &= f(x) + \frac{1}{2\rho} \|\rho c(x) + \lambda - \mathcal{P}_{\rho\mathcal{K}_c}(\rho c(x) + \lambda)\|^2 + \frac{1}{2\rho} \|\rho x + \mu - \mathcal{P}_{\rho\mathcal{K}_x}(\rho x + \mu)\|^2 \\ &= f(x) + \frac{1}{2\rho} \|\mathcal{P}_{\rho\mathcal{K}_c^c}(\rho c(x) + \lambda)\|^2 + \frac{1}{2\rho} \|\mathcal{P}_{\rho\mathcal{K}_x^c}(\rho x + \mu)\|^2. \end{aligned} \quad (3.3)$$

此时参照经典的增广 Lagrange 函数方法流程, 整个 ALM 的迭代步骤可简单总结为

$$x^{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^n} \mathcal{L}_\rho(x; \lambda^k, \mu^k), \quad (3.4)$$

$$\lambda^{k+1} = \mathcal{P}_{\rho\mathcal{K}_c^c}(\rho c(x^{k+1}) + \lambda^k), \quad (3.5)$$

$$\mu^{k+1} = \mathcal{P}_{\rho\mathcal{K}_x^c}(\rho x^{k+1} + \mu^k). \quad (3.6)$$

在具体算法细节上, 为控制惩罚因子的更新, 有必要引入额外的辅助函数

$$V_{\rho}(x; \lambda, \mu) = \max \left\{ \left\| c(x) - \mathcal{P}_{\mathcal{K}_c} \left(c(x) + \frac{\lambda}{\rho} \right) \right\|, \left\| x - \mathcal{P}_{\mathcal{K}_x} \left(x + \frac{\mu}{\rho} \right) \right\| \right\}. \quad (3.7)$$

V 中两项的设计其实参照的是原始分离变量后等式约束 $c(x) = y$ 与 $x = z$ 的满足程度, 它实际上要强于仅考虑关于变量 x 的约束违反度. 具体地, 当辅助函数 V 的值足够小, 即所得点满足一定可行性要求时, 算法保持罚因子不变, 更新乘子; 否则, 算法保持乘子不变, 增大罚因子. 因此, 本文给出了一个新的求解问题 (P) 的实际 ALM 算法, 如算法 1 所示.

算法 1 求解问题 (P) 的实际 ALM 算法

1. 输入收敛的最优性容差 η_* 和 ω_* ;
2. 初始化选取 $x^0 \in \mathbb{R}^n$, $\lambda^0 \in \mathbb{R}^m$, $\mu^0 \in \mathbb{R}^n$, $\rho_0 > 1$, $\gamma > 1$ 以及 $\tau_1, \tau_2 \in (0, 1)$;
3. 令 $k = 0$, $\omega_0 = 1/\rho_0$, $\eta_0 = 1/\rho_0^{\tau_2}$;
4. **for** $k = 0, 1, 2, \dots$ **do**
5. 求解子问题

$$\min_x \mathcal{L}_{\rho_k}(x; \lambda^k, \mu^k); \quad (3.8)$$

6. 找到 (近似) 极小值点 x^{k+1} 使其满足

$$\|\nabla_x \mathcal{L}_{\rho_k}(x^{k+1}; \lambda^k, \mu^k)\| \leq \omega_k; \quad (3.9)$$

7. **if** $V_{\rho_k}(x^{k+1}; \lambda^k, \mu^k) \leq \eta_k$ **then** 情形 1
 8. $\lambda^{k+1} = \mathcal{P}_{\rho_k \mathcal{K}_c^o}(\rho_k c(x^{k+1}) + \lambda^k)$;
 9. $\mu^{k+1} = \mathcal{P}_{\rho_k \mathcal{K}_x^o}(\rho_k x^{k+1} + \mu^k)$;
 10. **if** $V_{\rho_k}(x^{k+1}; \lambda^k, \mu^k) \leq \eta_*$ 及 $\|\nabla_x \mathcal{L}_{\rho_k}(x^{k+1}; \lambda^k, \mu^k)\| \leq \omega_*$ **then**
 11. 停止算法并得到近似解 x^{k+1} ;
 12. **end if**
 13. $\rho_{k+1} = \rho_k$, $\eta_{k+1} = \eta_k / \rho_{k+1}^{\tau_1}$, $\omega_{k+1} = \omega_k / \rho_{k+1}$;
 14. **else** 情形 2
 15. $\lambda^{k+1} = \lambda^k$;
 16. $\mu^{k+1} = \mu^k$;
 17. $\rho_{k+1} = \gamma \rho_k$, $\eta_{k+1} = 1/\rho_{k+1}^{\tau_2}$, $\omega_{k+1} = 1/\rho_{k+1}$.
 18. **end if**
 19. **end for**
-

3.1 子问题的求解

ALM 中最重要的步骤是求解增广 Lagrange 函数极小化的子问题 (3.4), 它的求解效率极大地决定了整个算法的运行效率, 采用何种方法求解子问题, 如何高效地对其求解都是值得研究的问题. 本小节对子问题 (3.4) 讨论应用近似半光滑 Newton 方法. 为防止混淆, 记外层迭代次数为上标 k , 内层迭代次数为下标 t . 具体而言, 设子问题求解的初始点 $x_0 := x^k$, 子问题求解得到的迭代序列为 $\{x_t\}$, 当满足 $\|\nabla_x \mathcal{L}_{\rho_k}(x_t; \lambda^k, \mu^k)\| \leq \omega_k$ 时, 我们退出子问题的求解, 并得到 $x^{k+1} := x_t$.

首先给出增广 Lagrange 函数的一阶和二阶信息. 定义 $\psi_{\rho}(x; \lambda) := \mathcal{P}_{\rho \mathcal{K}_c^o}(\rho c(x) + \lambda)$, 则增广 Lagrange 函数 (3.3) 关于变量 x 的梯度表达式为

$$\nabla_x \mathcal{L}_{\rho}(x; \lambda, \mu) = \nabla f(x) + \nabla c(x) \psi_{\rho}(x; \lambda) + \mathcal{P}_{\rho \mathcal{K}_x^o}(\rho x + \mu). \quad (3.10)$$

由于增广 Lagrange 函数是非光滑的, 为了刻画函数的广义 Hessian 矩阵, 我们给出如下半光滑定义 (参见文献 [17, 19]). 设 $F: \mathcal{O} \rightarrow \mathbb{R}^m$ 为一个局部 Lipschitz 连续函数. 如果它满足以下条件:

- (a) F 在 x 处是方向可导的;
- (b) 对于任意的 $d \in \mathcal{O}$ 和 $J \in \partial F(x + d)$, 有

$$\|F(x + d) - F(x) - Jd\|_2 = o(\|d\|), \quad d \rightarrow 0,$$

则称 F 在 $x \in \mathcal{O}$ 处是半光滑的. 若 F 在每个点处都是半光滑的, 则称其为半光滑函数.

为探究目标函数 (3.3) 的一些二阶信息, 首先考虑算子

$$\hat{\partial}^2 \mathcal{L}_\rho(x; \lambda, \mu) := \nabla^2 f(x) + \nabla^2 c(x)[\psi_\rho(x; \lambda)] + \nabla c(x) \partial \psi_\rho(x; \lambda) \nabla c(x)^T + \partial \mathcal{P}_{\rho \mathcal{K}_x^\circ}(\rho x + \mu),$$

其中 $\partial \mathcal{P}_{\rho \mathcal{K}_x^\circ}(\rho x + \mu)$ 是 $\partial \mathcal{P}_{\rho \mathcal{K}_x^\circ}(\cdot)$ 在 $\rho x + \mu$ 处的广义 Jacobi 矩阵, 且

$$\nabla^2 c(x)[\psi_\rho(x; \lambda)] := \sum_{i=1}^m (\psi_\rho(x; \lambda))_i \nabla^2 c_i(x).$$

设函数 $\mathcal{L}_\rho(x; \lambda, \mu)$ 的广义 Hessian 矩阵为 $\partial^2 \mathcal{L}_\rho(x; \lambda, \mu)$ (参见文献 [5, 命题 2.3.3 和定理 2.6.6]), 则它满足

$$\partial^2 \mathcal{L}_\rho(x; \lambda, \mu) \subseteq \hat{\partial}^2 \mathcal{L}_\rho(x; \lambda, \mu).$$

为计算广义的 Hessian 矩阵, 首先定义

$$W(x) := \nabla^2 f(x) + \nabla^2 c(x)[\psi_\rho(x; \lambda)] + \rho \nabla c(x) D_c \nabla c(x)^T + \rho D_x, \quad (3.11)$$

取 $\rho D_c \in \partial \psi_\rho(x)$ 和 $\rho D_x \in \partial \mathcal{P}_{\rho \mathcal{K}_x^\circ}(\rho x + \mu)$, 则 $W(x) \in \hat{\partial}^2 \mathcal{L}_\rho(x; \lambda, \mu)$. 考虑到算子 $\mathcal{P}_{\mathcal{K}_c^\circ}$ 和 $\mathcal{P}_{\mathcal{K}_x^\circ}$ 的定义, 选择

$$D_c = \text{Diag}(d^c) \quad \text{和} \quad D_x = \text{Diag}(d^x), \quad (3.12)$$

其中矩阵的对角元分别为 d_i^c ($i = 1, 2, \dots, m$) 和 d_j^x ($j = 1, 2, \dots, n$), 满足

$$d_i^c = \begin{cases} 0, & \text{若 } c_i(x) + \lambda_i/\rho \in [\ell_i^c, u_i^c], \\ 1, & \text{否则,} \end{cases} \quad d_j^x = \begin{cases} 0, & \text{若 } x_j + \mu_j/\rho \in [\ell_j^x, u_j^x], \\ 1, & \text{否则.} \end{cases}$$

将其代入 (3.11) 后对应得到的 $W(x)$ 即可认为是目标函数的 Hessian 矩阵. 由于 $\partial \psi_\rho(x)$ 和 $\partial \mathcal{P}_{\rho \mathcal{K}_x^\circ}(\cdot)$ 是半光滑的, 因此设计求解子问题的半光滑 Newton 方法.

值得注意的是, 在计算 (3.11) 中的第一项 $\nabla^2 f(x)$ 时, 如果 f 具有某种特殊形式, 我们可采取一定的近似技巧来计算其 Hessian 矩阵. 例如, 在最小二乘问题中, 目标函数 f 具有以下特殊形式:

$$f(x) = \frac{1}{2} \|r(x)\|^2 = \frac{1}{2} \sum_{i=1}^m r_i(x)^2,$$

其中每个 r_i 是从 $\mathbb{R}^n$ 到 $\mathbb{R}$ 的光滑函数. 函数 f 的梯度和 Hessian 矩阵表示如下:

$$\nabla f(x) = J(x)^T r(x), \quad \nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^m r_i(x) \nabla^2 r_i(x),$$

其中 $J(x)$ 为函数 $f(x)$ 的 Jacobi 矩阵. 当 $r_j(x)$ 的二阶导数项不易求出时, 为节省计算时间, 可以采用 Gauss-Newton 方法的思想, 使用近似矩阵 $\nabla^2 f(x) \approx J(x)^T J(x)$, 这样避免了计算每个残差 Hessian 矩阵 $\nabla^2 r_i$ ($i = 1, 2, \dots, m$). 对于某些实际问题, 在数值实验中计算 $W(x)$ 时忽略前两项中比较复杂的部分可以获得更好的结果, 这一点在文献 [12] 中得到了很好的支持: 当目标函数的 Hessian 矩阵可以自然地分解为两部分时, 第一部分较容易计算且在问题中占据主导地位; 第二部分虽然计算复杂, 但对问题的贡献较小. 在这种情形下, 精确计算第一部分并对第二部分进行近似 (或在某些情形下直接忽略第二部分), 往往可以获得比整体近似更优的效果, 同时显著加速计算. 尤其当 $f(x)$ 是凸的而 $c(x)$ 非凸时, 忽略第二项 $\nabla^2 c(x)[\psi_\rho(x; \lambda)]$ 常常能提高算法效率并保证算法的准确性 (参见文献 [15]). 使用增广 Lagrange 函数的真正 Hessian 矩阵更为合适, 因为它更好地表示了真实问题的结构. 然而在求解增广 Lagrange 函数极小化子问题时, 可用的算法在函数是凸的情形下更为高效, 并且近似复杂结构的 Hessian 矩阵也可以提高增广 Lagrange 函数方法的整体性能.

接下来给出求解子问题的半光滑 Newton 算法. 在子问题内部迭代的第 t 步, 为方便起见, 记 $g(x_t) := \nabla_x \mathcal{L}_{\rho_k}(x_t; \lambda^k, \mu^k)$, 则半光滑 Newton 方向的计算方式为

$$W(x_t)p_t^N = -g(x_t). \quad (3.13)$$

通常情形下, 在迭代过程中通过 (3.11) 求得 Hessian 矩阵 $W(x_t)$ 可能是非正定甚至奇异的, 这会使得上述的 p_t^N 并非下降方向或者无法求得. 因此我们开发了一种自适应正则化的近似半光滑 Newton 方法对 $W(x_t)$ 进行一定的修正, 使其能始终保持正定. 具体而言, 我们使用二阶 Taylor 模型来近似子问题的原始目标函数. 此外, 为了控制函数 Hessian 矩阵的正定性, 我们添加了一个小的正修正项. 因此本文构建子问题 (3.4) 的如下二次近似模型:

$$\min m_t(d) := \langle g(x_t), d \rangle + \frac{1}{2} d^T W(x_t) d + \frac{\sigma_t}{2} \|d\|^2, \quad (3.14)$$

其中正则化参数 $\sigma_t > 0$ 相当于在子问题中起到类似信赖域半径的作用, 可以很好地保证 Hessian 矩阵的正定性. 我们的整体思路是通过求解一系列形式为 (3.14) 的子问题来解决和替代初始子问题 (3.4), 因此需要求解方程

$$(W(x_t) + \sigma_t I) d = -g(x_t) \quad (3.15)$$

得到问题 (3.14) 的最优解 d_t . 我们采用如下方式对上述方程进行求解. 首先对矩阵 $W(x_t) + \sigma_t I$ 进行 Cholesky 分解 [7], 如果分解失败, 我们估计 $W(x_t)$ 的最小特征值 λ_w , 然后调整 (3.15) 中的参数 $\sigma_t = \max\{-2\lambda_w, 10^{-6}\}$, 以保证 $W(x_t) + \sigma_t I$ 正定, 再对方程重新计算. 这样使得我们总是可以找到搜索方向 d_t , 再进一步更新实验点 $\tilde{x}_{t+1} = x_t + d_t$. 除此之外, 对 $W(x_t)$ 还可以应用基于对称不定因子分解和特征值修正策略 [14] 等. 这样处理的好处是, 即使子问题是无界的, 但该实验点始终是良好定义的.

为了决定是否应当接受 $\tilde{x}_{t+1}$ 作为下一个迭代, 以及是否应当更新正则化参数 σ_t , 我们计算目标函数的实际减少与预测减少之间的比率:

$$\pi_t = \frac{\mathcal{L}_\rho(\tilde{x}_{t+1}; \lambda, \mu) - \mathcal{L}_\rho(x_t; \lambda, \mu)}{m_t(d_t)}. \quad (3.16)$$

如果 $\pi_t \geq \varrho_1 > 0$, 则迭代成功, 设置 $x_{t+1} = \tilde{x}_{t+1}$; 否则, 迭代失败, 设置 $x_{t+1} = x_t$, 即

$$x_{t+1} = \begin{cases} \tilde{x}_{t+1}, & \text{如果 } \pi_t \geq \varrho_1, \\ x_t, & \text{否则.} \end{cases} \quad (3.17)$$

正则化参数 σ_t 的更新如下:

$$\sigma_{t+1} \in \begin{cases} (0, \gamma_0 \sigma_t], & \text{如果 } \pi_t \geq \varrho_2, \\ [\gamma_0 \sigma_t, \gamma_1 \sigma_t], & \text{如果 } \varrho_1 \leq \pi_t < \varrho_2, \\ [\gamma_1 \sigma_t, \gamma_2 \sigma_t], & \text{否则,} \end{cases} \quad (3.18)$$

其中 $0 < \eta_1 \leq \eta_2 < 1$ 且 $0 < \gamma_0 < 1 < \gamma_1 \leq \gamma_2$. 因此我们将求解子问题的近似半光滑 Newton 算法总结在算法 2 中.

算法 2 求解子问题 (3.8) 的半光滑 Newton 算法

1. **Input:** 初始点 x_0 , 乘子 λ , μ 和惩罚因子 ρ ;
 2. **Output:** 最优解 x_{t+1} ;
 3. 令 $t = 0$, 给定参数满足 $0 < \varrho_1 \leq \varrho_2 < 1$ 和 $0 < \gamma_0 < 1 < \gamma_1 \leq \gamma_2$;
 4. **while** 没有满足停机条件 **do**
 5. 通过求解 (3.15) 得到 d_t , 计算实验点 $\tilde{x}_{t+1} = x_t + d_t$;
 6. 通过 (3.16) 计算比率 π_t ;
 7. 通过 (3.17) 更新 x_{t+1} , 通过 (3.18) 更新参数 σ_t . 设置 $t = t + 1$.
 8. **end while**
-

除了上述算法, 还可以考虑标准的信赖域算法或者三次正则方法对子问题进行求解. 值得注意的是, 根据 W 的特殊结构, 这里不需要直接计算 Hessian 矩阵, 而是找到计算成本较高部分的近似矩阵, 这种技巧不仅可以很好地保留矩阵的部分性质, 而且也方便快速有效求解子问题.

3.2 收敛性分析

本小节对算法 1 的收敛性进行一些基本的分析, 主要参考了文献 [13, 16] 中关于 ALM 收敛性的证明思路. 首先证明在适当的正则性条件下, ALM 具有全局收敛性, 并通过逐步更新 Lagrange 乘子和调整惩罚参数, 确保算法逐渐逼近最优解. 其次, 进一步证明该方法在迭代过程中能够满足 AKKT 条件, 即使在解接近约束边界时, 仍能保持较好的逼近精度. 此外, 还推导了算法产生的聚点是描述约束违反度的可行性函数的稳定点.

在收敛性分析中, 我们不考虑算法中的终止性条件 (步骤 10 和 11), 默认算法会得到无穷步迭代序列. 基于此, 我们首先证明一个引理和基本的结论.

引理 3.1 在算法 1 下, 任取 $\tau_0 \in (1 - \tau_2, 1)$, 序列 $\{\frac{\lambda^k}{\rho_k}\}$ 和 $\{\frac{\mu^k}{\rho_k}\}$ 有界, 即存在 $M_{\tau_0} > 0$ 使得对任意 $k \in \mathbb{N}$, 有 $\frac{\|\lambda^k\|}{\rho_k} \leq M_{\tau_0}$ 和 $\frac{\|\mu^k\|}{\rho_k} \leq M_{\tau_0}$.

证明 我们针对序列 $\{\rho_k\}$ 有界和无界两种情形展开讨论. 若 $\rho_k \rightarrow \infty$, 说明算法运行中有无穷多步进入了情形 2. 取序列 $\{b_K\}_{K=0}^\infty$, $k = b_K$ 时算法进行了情形 2 中的一系列参数变换. 最本地, 有 $\rho_{b_K} = \gamma^K \rho_0$, $\rho_{b_K+1} = \gamma^{K+1} \rho_0$, $\eta_{b_K+1} = 1/\rho_{b_K+1}^{\tau_2}$. 而对于 $b_K + 1 \leq k \leq b_{K+1} - 1$ (如果存在的话), 算法在这些迭代步上进行了情形 1 的一系列步骤, 此时 $\rho_k = \rho_{b_K+1}$, $\eta_k = \rho_{b_K+1}^{-\tau_1(k-b_K-1)} \eta_{b_K+1}$.

以 λ^k 为例说明, 当 $b_K + 1 \leq k \leq b_{K+1} - 1$ 时,

$$\lambda^{k+1} = \mathcal{P}_{\rho_k \mathcal{K}_c^\circ}(\rho_k c(x^{k+1}) + \lambda^k) = \lambda^k + \rho_k \left(c(x^{k+1}) - \mathcal{P}_{\mathcal{K}_c} \left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k} \right) \right).$$

由于此时 $\|c(x^{k+1}) - \mathcal{P}_{\mathcal{K}_c}(c(x^{k+1}) + \frac{\lambda^k}{\rho_k})\| \leq V_{\rho_k}(x^{k+1}; \lambda^k, \mu^k) \leq \eta_k$, 所以 $\|\lambda^{k+1}\| \leq \|\lambda^k\| + \eta_k \rho_k$, 将 η_k 和 ρ_k 具体值代入, 并进行累加, 利用 ρ_k 递增的特性即得

$$\begin{aligned} \|\lambda^{k+1}\| &\leq \|\lambda^k\| + \rho_{b_K+1}^{1-\tau_2} \rho_{b_K+1}^{-\tau_1(k-b_K-1)} \leq \dots \\ &\leq \|\lambda^{b_K+1}\| + \rho_{b_K+1}^{1-\tau_2} \sum_{i=b_K+1}^k \rho_{b_K+1}^{-\tau_1(i-b_K-1)} \\ &\leq \|\lambda^{b_K+1}\| + \rho_{b_K+1}^{1-\tau_2} \sum_{i=0}^{\infty} \rho_{b_K+1}^{-\tau_1 i} \\ &= \|\lambda^{b_K+1}\| + \rho_{b_K+1}^{1-\tau_2} \frac{1}{1 - \rho_{b_K+1}^{-\tau_1}} \\ &\leq \|\lambda^{b_K+1}\| + \rho_{b_K+1}^{1-\tau_2} \frac{1}{1 - \rho_0^{-\tau_1}}, \quad b_K + 1 \leq k \leq b_{K+1} - 1. \end{aligned} \quad (3.19)$$

取 $k = b_{K+1} - 1$, 并利用 $\lambda^{b_K+1} = \lambda^{b_K}$, 可得

$$\|\lambda^{b_{K+1}}\| \leq \|\lambda^{b_K}\| + \rho_{b_K+1}^{1-\tau_2} \frac{1}{1 - \rho_0^{-\tau_1}}. \quad (3.20)$$

对 (3.20) 两边同除 $\rho_{b_K}^{\tau_0}$, 则有

$$\frac{\|\lambda^{b_{K+1}}\|}{\rho_{b_{K+1}}^{\tau_0}} \leq \frac{\|\lambda^{b_K}\|}{\rho_{b_K}^{\tau_0}} \leq \frac{\|\lambda^{b_K}\|}{\rho_{b_K}^{\tau_0}} + \gamma^{K(1-\tau_2-\tau_0)} \gamma^{1-\tau_2} \rho_0^{1-\tau_2-\tau_0} \frac{1}{1 - \rho_0^{-\tau_1}}.$$

同样进行累加, 可得

$$\frac{\|\lambda^{b_{K+1}}\|}{\rho_{b_{K+1}}^{\tau_0}} \leq \frac{\|\lambda^{b_0}\|}{\rho_0^{\tau_0}} + \gamma^{1-\tau_2} \rho_0^{1-\tau_2-\tau_0} \frac{1}{1 - \rho_0^{-\tau_1}} \sum_{i=0}^K \gamma^{i(1-\tau_2-\tau_0)}, \quad (3.21)$$

而由于 $1 - \tau_2 - \tau_0 < 0$, 所以 $\sum_{i=0}^K \gamma^{i(1-\tau_2-\tau_0)} < \sum_{i=0}^{\infty} \gamma^{i(1-\tau_2-\tau_0)} < \infty$, 说明序列 $\{\frac{\|\lambda^{b_K}\|}{\rho_{b_K}^{\tau_0}}\}_{K=0}^{\infty}$ 有界, 即存在 $M > 0$, 使得 $\frac{\|\lambda^{b_K}\|}{\rho_{b_K}^{\tau_0}} \leq M$ 对所有 K 恒成立. 对于 $k = b_K + 1$, $\frac{\|\lambda^{b_{K+1}}\|}{\rho_{b_{K+1}}^{\tau_0}} \leq \frac{\|\lambda^{b_K}\|}{\rho_{b_K}^{\tau_0}} \leq M$ 有界; 对于 $b_K + 2 \leq k \leq b_{K+1}$, 由 (3.19) 及 $\rho_k = \rho_{b_K+1}$,

$$\frac{\|\lambda^k\|}{\rho_k^{\tau_0}} \leq \frac{\|\lambda^{b_K}\|}{\rho_{b_K}^{\tau_0}} + \rho_{b_K+1}^{1-\tau_2-\tau_0} \frac{1}{1 - \rho_0^{-\tau_1}} \leq \frac{\|\lambda^{b_K}\|}{\rho_{b_K}^{\tau_0}} + \rho_0^{1-\tau_2-\tau_0} \frac{1}{1 - \rho_0^{-\tau_1}} \leq M + \rho_0^{1-\tau_2-\tau_0} \frac{1}{1 - \rho_0^{-\tau_1}}$$

同样有界. 由此即说明对于任意 k , $\{\frac{\lambda^k}{\rho_k^{\tau_0}}\}$ 有界.

若 $\{\rho_k\}$ 有界, 则说明算法中只有有限多次进入了情形 2, 设为 N . 与无界时一样取序列 $\{b_K\}_{K=0}^N$, 取 $k = N$ 代入 (3.19), 则有

$$\|\lambda^{k+1}\| \leq \|\lambda^{b_N+1}\| + \rho_{b_N+1}^{1-\tau_2} \frac{1}{1 - \rho_0^{-\tau_1}}, \quad k \geq b_N + 1.$$

说明此时 $\{\lambda^k\}$ 有界, 再由 $\{\rho_k\}$ 有界, 即得 $\{\frac{\lambda^k}{\rho_k^{\tau_0}}\}$ 有界.

关于 μ^k , 与 λ^k 证明时原理类似, 可证 $\{\frac{\mu^k}{\rho_k^{\tau_0}}\}$ 有界, 此处不再赘述. $\square$

命题 3.1 若算法 1 中的 $\{\rho_k\}$ 无界, 即 $\rho_k \rightarrow \infty$, 则有 $\frac{\lambda^k}{\rho_k}, \frac{\mu^k}{\rho_k} \rightarrow 0$ 成立.

证明 由于 $\frac{\|\lambda^k\|}{\rho_k} = \frac{\|\lambda^k\|}{\rho_k^{\tau_0}} \rho_k^{\tau_0-1}$, 所以由引理 3.1 可知 $\frac{\|\lambda^k\|}{\rho_k}$ 在 $k \rightarrow \infty$ 时保持有界, 当 $\rho_k \rightarrow \infty$ 时有 $\rho_k^{\tau_0-1} \rightarrow 0$, 因此 $\frac{\|\lambda^k\|}{\rho_k} \rightarrow 0$. 对于 μ^k , 在 $\rho_k \rightarrow \infty$ 时, 借助引理 3.1, 同理可证 $\frac{\mu^k}{\rho_k} \rightarrow 0$. $\square$

有了命题 3.1 作铺垫, 下面先说明在何种情形下算法所得序列的聚点是可行的.

命题 3.2 设 $\{x^k\}$ 为由算法 1 生成的迭代点序列, 若以下两种情形之一成立, 则该序列的任一聚点 x^* 是关于问题 (P) 的可行点:

- (1) $\{\rho_k\}$ 是有界的;
- (2) 存在 $B \in \mathbb{R}$ 使得对于任意 $k \in \mathbb{N}$, 有 $\mathcal{L}_{\rho_k}(x^{k+1}; \lambda^k, \mu^k) \leq B$.

证明 设 x^* 为序列 $\{x^k\}$ 的任一聚点, 令 $\{x^{k+1}\}_K$ 为其子列且满足 $x^{k+1} \rightarrow_K x^*$. 对命题中的两种情形分别进行讨论.

(1) 若 $\{\rho_k\}$ 有界, 说明算法中情形 2 的各步骤只进行了有限多次, 即存在 K_0 , 当 $k \geq K_0$ 时, 算法进行的都是情形 1 中各步骤. 此时 ρ_{k+1} 保持不变, 则序列 $\{\eta_k\}$ 和 $\{\omega_k\}$ 均形成等比数列, 公比小于 1, 即有 $\eta_k \rightarrow 0$, $\omega_k \rightarrow 0$. 对应地, $V_{\rho_k}(x^{k+1}; \lambda^k, \mu^k) \rightarrow 0$, 并由辅助函数 V 的定义, 有 $\|c(x^{k+1}) - \mathcal{P}_{\mathcal{K}_c}(c(x^{k+1}) + \frac{\lambda^k}{\rho_k})\| \rightarrow 0$ 和 $\|x^{k+1} - \mathcal{P}_{\mathcal{K}_x}(x^{k+1} + \frac{\mu^k}{\rho_k})\| \rightarrow 0$, 则

$$\|\mathcal{P}_{\mathcal{K}_c^o}(c(x^{k+1}))\| = \|c(x^{k+1}) - \mathcal{P}_{\mathcal{K}_c}(c(x^{k+1}) + \frac{\lambda^k}{\rho_k})\| \leq \left\| c(x^{k+1}) - \mathcal{P}_{\mathcal{K}_c}\left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k}\right) \right\| \rightarrow 0, \quad (3.22)$$

$$\|\mathcal{P}_{\mathcal{K}_x^o}(x^{k+1})\| = \|x^{k+1} - \mathcal{P}_{\mathcal{K}_x}(x^{k+1} + \frac{\mu^k}{\rho_k})\| \leq \left\| x^{k+1} - \mathcal{P}_{\mathcal{K}_x}\left(x^{k+1} + \frac{\mu^k}{\rho_k}\right) \right\| \rightarrow 0. \quad (3.23)$$

由连续性, 令 $x^{k+1} \rightarrow_K x^*$, 即得 $\|\mathcal{P}_{\mathcal{K}_c^o}(c(x^*))\| = \|\mathcal{P}_{\mathcal{K}_x^o}(x^*)\| = 0$, 于是 $c(x^*) = \mathcal{P}_{\mathcal{K}_c}(c(x^*))$, $x^* = \mathcal{P}_{\mathcal{K}_x}(x^*)$, 说明 $c(x^*) \in \mathcal{K}_c$, $x^* \in \mathcal{K}_x$, x^* 为可行点.

(2) 若 $\{\rho_k\}$ 无界, 假设 $\rho_k \rightarrow \infty$. 根据条件, 可得

$$B \geq \mathcal{L}_{\rho_k}(x^{k+1}; \lambda^k, \mu^k) = f(x^{k+1}) + \frac{1}{2\rho_k} \|\mathcal{P}_{\rho_k \mathcal{K}_c^o}(\rho_k c(x^{k+1}) + \lambda^k)\|^2 + \frac{1}{2\rho_k} \|\mathcal{P}_{\rho_k \mathcal{K}_x^o}(\rho_k x^{k+1} + \mu^k)\|^2.$$

重新整理, 并将 ρ_k 分离出来, 即得

$$\left\| \mathcal{P}_{\mathcal{K}_c^o}\left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k}\right) \right\|^2 + \left\| \mathcal{P}_{\mathcal{K}_x^o}\left(x^{k+1} + \frac{\mu^k}{\rho_k}\right) \right\|^2 \leq \frac{2(B - f(x^{k+1}))}{\rho_k}, \quad \forall k \in \mathbb{N}. \quad (3.24)$$

在上式中取 $k \rightarrow_K \infty$, 由 f 的连续性可知 $f(x^{k+1}) \rightarrow_K f(x^*)$ 有界, 而 $\rho_k \rightarrow_K \infty$, 则上述不等式右边 $\rightarrow_K 0$. 结合上述命题 3.1, 有

$$\mathcal{P}_{\mathcal{K}_c^o}(c(x^*)) = \lim_{k \rightarrow_K \infty} \mathcal{P}_{\mathcal{K}_c^o}\left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k}\right) = 0,$$

$$\mathcal{P}_{\mathcal{K}_x^o}(x^*) = \lim_{k \rightarrow_K \infty} \mathcal{P}_{\mathcal{K}_x^o}\left(x^{k+1} + \frac{\mu^k}{\rho_k}\right) = 0.$$

同样说明 x^* 为问题 (P) 的可行点. $\square$

接着考虑算法的收敛性结论, 即由算法得到的任何可行聚点都满足 AKKT 条件.

定理 3.1 设 $\{x^k\}$ 为由算法 1 生成的迭代点序列, 则该序列关于问题 (P) 的任意一个可行聚点 x^* 均满足其对应的 AKKT 条件.

证明 定义 $\{x^{k+1}\}_K$ 为子序列满足 $x^{k+1} \rightarrow_K x^*$. 对于任意 $k \in \mathbb{N}$, 按照算法中各变量的变化定义

$$u^{k+1} := \mathcal{P}_{\rho_k \mathcal{K}_c^o}(\rho_k c(x^{k+1}) + \lambda^k), \quad v^{k+1} := \mathcal{P}_{\rho_k \mathcal{K}_x^o}(\rho_k x^{k+1} + \mu^k),$$

$$\begin{aligned}\epsilon^{k+1} &:= \nabla f(x^{k+1}) + \nabla c(x^{k+1})u^{k+1} + v^{k+1}, \\ s^{k+1} &:= \mathcal{P}_{\mathcal{K}_c}\left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k}\right), \quad \zeta^{k+1} := c(x^{k+1}) - s^{k+1}, \\ r^{k+1} &:= \mathcal{P}_{\mathcal{K}_x}\left(x^{k+1} + \frac{\mu^k}{\rho_k}\right), \quad \xi^{k+1} := x^{k+1} - r^{k+1}.\end{aligned}$$

接下来证明上述定义的序列 $\{x^{k+1}\}_K, \{u^{k+1}\}_K, \{v^{k+1}\}_K, \{\epsilon^{k+1}\}_K, \{\zeta^{k+1}\}_K$ 和 $\{\xi^{k+1}\}_K$ 均满足 AKKT 条件定义中的性质, 从而说明 x^* 满足 AKKT 条件.

首先由梯度表达式 (3.10) 和算法中的第 1 步可以得出 $\|\epsilon^{k+1}\| \leq \omega_k$. 同时考虑到 $\mathcal{N}_{\mathcal{K}_c}$ 和 $\mathcal{N}_{\mathcal{K}_x}$ 的定义及它们与投影 $\mathcal{P}_{\mathcal{K}_c}$ 和 $\mathcal{P}_{\mathcal{K}_x}$ 间的关系, 有

$$\begin{aligned}u^{k+1} &= \rho_k \left[c(x^{k+1}) + \frac{\lambda^k}{\rho_k} - s^{k+1} \right] \in \mathcal{N}_{\mathcal{K}_c}(s^{k+1}) = \mathcal{N}_{\mathcal{K}_c}(c(x^{k+1}) - \zeta^{k+1}), \\ v^{k+1} &= \rho_k \left[x^{k+1} + \frac{\mu^k}{\rho_k} - r^{k+1} \right] \in \mathcal{N}_{\mathcal{K}_x}(r^{k+1}) = \mathcal{N}_{\mathcal{K}_x}(x^{k+1} - \xi^{k+1}).\end{aligned}$$

则还需要证明 $\epsilon^{k+1} \rightarrow_K 0, \zeta^{k+1} \rightarrow_K 0, \xi^{k+1} \rightarrow_K 0$. 为此, 针对 $\{\rho_k\}$ 有界和无界的两种情形分别进行讨论.

若 $\{\rho_k\}$ 有界, $\omega_k \rightarrow 0$, 则由 $\|\epsilon^{k+1}\| \leq \omega_k \rightarrow 0$, 有 $\epsilon^{k+1} \rightarrow_K 0$. 同时, 由

$$\max\{\|\zeta^{k+1}\|, \|\xi^{k+1}\|\} = V_{\rho_k}(x^{k+1}; \lambda^k, \mu^k) \leq \eta_k \rightarrow 0,$$

即得 $\zeta^{k+1} \rightarrow_K 0, \xi^{k+1} \rightarrow_K 0$, 说明此时的 x^* 满足 AKKT 条件.

若 $\{\rho_k\}$ 无界, 则 $\rho_k \rightarrow \infty$. 由算法中关于 ω_k 的定义与变换可知 $\omega_k \leq 1/\rho_k$, 则当 $\rho_k \rightarrow \infty$ 时, $\omega_k \rightarrow 0$. 同样由 $\|\epsilon^{k+1}\| \leq \omega_k \rightarrow 0$ 得 $\epsilon^{k+1} \rightarrow_K 0$. 此外, 由命题 3.1 可知 $\frac{\lambda^k}{\rho_k}, \frac{\mu^k}{\rho_k} \rightarrow 0$, 结合假定 x^* 的可行性, 有

$$s^{k+1} := \mathcal{P}_{\mathcal{K}_c}\left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k}\right) \rightarrow_K \mathcal{P}_{\mathcal{K}_c}(c(x^*)) = c(x^*), \quad (3.26)$$

$$r^{k+1} := \mathcal{P}_{\mathcal{K}_x}\left(x^{k+1} + \frac{\mu^k}{\rho_k}\right) \rightarrow_K \mathcal{P}_{\mathcal{K}_x}(x^*) = x^*. \quad (3.27)$$

因此, $\zeta^{k+1} = c(x^{k+1}) - s^{k+1} \rightarrow_K 0, \xi^{k+1} = x^{k+1} - r^{k+1} \rightarrow_K 0$, 说明在 $\{\rho_k\}$ 无界情形下算法所得的 x^* 也满足 AKKT 条件. $\square$

考虑结合 AKKT 正则性条件, 以下结论是推论 2.1 和定理 3.1 直接推导所得到的结果.

推论 3.1 设 $\{x^k\}$ 为由算法 1 生成的迭代点序列, 该序列关于问题 (P) 的任意一个可行聚点为 x^* . 若 x^* 关于问题 (P) 满足 AKKT 正则性条件, 则它满足问题 (P) 对应的 KKT 条件.

以上考虑的是聚点可行的情形, 但在实际运行算法时, 也有可能会出现算法得不到可行点的情形. 然而, 即使在这种情形下, 也可以说明算法所得的聚点是描述约束违反度的所谓可行性函数的稳定点.

命题 3.3 设算法 1 所得迭代点序列为 $\{x^k\}$, 其任意一个聚点 x^* 都是所谓可行性函数

$$\frac{1}{2}\|\mathcal{P}_{\mathcal{K}_c^c}(c(x))\|^2 + \frac{1}{2}\|\mathcal{P}_{\mathcal{K}_x^c}(x)\|^2 \quad (3.28)$$

的一个稳定点.

证明 同样按照 $\{\rho_k\}$ 有界和无界的情形分类进行讨论. 若 $\{\rho_k\}$ 有界, 由之前命题的陈述, 此时 $\|\mathcal{P}_{\mathcal{K}_c^o}(c(x^*))\| = \|\mathcal{P}_{\mathcal{K}_x^o}(x^*)\| = 0$, 即 x^* 为 (3.28) 的全局极小值点, 自然为稳定点.

若 $\{\rho_k\}$ 无界, 则 $\rho_k \rightarrow \infty$. 设 $\{x^{k+1}\}_K$ 为子列满足 $x^{k+1} \rightarrow_K x^*$. 由算法子问题的求解精度可知, 此时的 x^{k+1} 满足

$$\|\nabla f(x^{k+1}) + \nabla c(x^{k+1})\mathcal{P}_{\rho_k \mathcal{K}_c^o}(\rho_k c(x^{k+1}) + \lambda^k) + \mathcal{P}_{\rho_k \mathcal{K}_x^o}(\rho_k x^{k+1} + \mu^k)\| \leq \omega_k.$$

两边同除 ρ_k , 即得

$$\left\| \frac{\nabla f(x^{k+1})}{\rho_k} + \nabla c(x^{k+1})\mathcal{P}_{\mathcal{K}_c^o}\left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k}\right) + \mathcal{P}_{\mathcal{K}_x^o}\left(x^{k+1} + \frac{\mu^k}{\rho_k}\right) \right\| \leq \frac{\omega_k}{\rho_k}.$$

此时令 $k \rightarrow_K \infty$, 由 $\rho_k \rightarrow \infty$, $x^{k+1} \rightarrow_K x^*$, $\omega_k \rightarrow 0$, $\frac{\lambda^k}{\rho_k}, \frac{\mu^k}{\rho_k} \rightarrow 0$, 且 f 和 c 连续, $\nabla f(x^*)$ 有界, 即得

$$\nabla c(x^*)\mathcal{P}_{\mathcal{K}_c^o}(c(x^*)) + \mathcal{P}_{\mathcal{K}_x^o}(x^*) = 0.$$

该式左边恰为函数 (3.28) 关于 x^* 的梯度, 即说明此时的 x^* 为可行性函数的稳定点. 命题得证. $\square$

3.3 复杂度分析

本小节着重分析算法 1 的一系列复杂度结果, 主要是基于算法整体的外层架构. 考虑在某些假设情境下, 算法 1 满足停止条件所需要的的最多的外层迭代次数. 本小节的分析方法与证明思路主要参见文献 [4]. 首先考虑算法中惩罚因子 $\{\rho_k\}$ 始终保持有界的情形, 可以说明此时的算法在有限步后一定会终止得到近似解. 而关于该有限值的最坏情形, 可以参照以下定理.

定理 3.2 给定容差 ω_* 和 η_* . 假定算法运行过程中对所有 $k = 0, 1, 2, \dots$, 存在 $\bar{\rho}$ 满足 $\rho_k \leq \bar{\rho}$, 则算法在运行至多

$$\left(\left\lceil \frac{\log(\bar{\rho}/\rho_0)}{\log \gamma} \right\rceil + 1 \right) \times \max \left\{ \left\lceil \frac{\tau_1 - \tau_2}{\tau_1} - \frac{\log \eta_*}{\tau_1 \log \rho_0} \right\rceil, \left\lceil -\frac{\log \omega_*}{\log \rho_0} \right\rceil \right\} \quad (3.29)$$

次迭代后, 会满足终止条件, 得到 $x^{k+1} \in \mathbb{R}^n$, $\lambda^{k+1} \in \mathbb{R}^m$ 和 $\mu^{k+1} \in \mathbb{R}^n$ 满足

$$\|\nabla f(x^{k+1}) + \nabla c(x^{k+1})\lambda^{k+1} + \mu^{k+1}\| \leq \omega_*, \quad (3.30)$$

$$\|\mathcal{P}_{\mathcal{K}_c^o}(c(x^{k+1}))\| \leq \eta_*, \quad \|\mathcal{P}_{\mathcal{K}_x^o}(x^{k+1})\| \leq \eta_*. \quad (3.31)$$

证明 由于 $\{\rho_k\}$ 有界, 说明算法 1 中仅有限多步进入了情形 2. 令 N 为算法运行情形 2 中一系列步骤的次数. 由于情形 1 内 ρ 值保持不变, 所以 ρ 的取值构成序列 $\{\rho_K\}_{K=0}^N$, $\rho_K = \gamma^K \rho_0$. 由 $\rho_N \leq \bar{\rho}$, 有 $N \leq \lceil \log(\bar{\rho}/\rho_0)/\log(\gamma) \rceil$.

下面考虑在 $\rho_k = \rho_K$ 取值固定不变时, 算法能连续运行情形 1 中步骤且不终止的最大次数. 在 ρ 最初取得 ρ_K 时, 其对应的容差 $\eta_K = 1/\rho_K^{\tau_2}$, $\omega_K = 1/\rho_K$, 则此时对应地, 如果算法想要不终止, 则最多连续运行情形 1 中步骤的次数为

$$\max \left\{ \left\lceil \frac{\log(\eta_*/\eta_K)}{\log(1/\rho_K^{\tau_1})} \right\rceil, \left\lceil \frac{\log(\omega_*/\omega_K)}{\log(1/\rho_K)} \right\rceil \right\}.$$

则综合 N 次 ρ_K 取值的变化, 整个算法在终止前的最多迭代次数为

$$N + \sum_{K=0}^N \max \left\{ \left\lceil \frac{\log(\eta_*/\eta_K)}{\log(1/\rho_K^{\tau_1})} \right\rceil, \left\lceil \frac{\log(\omega_*/\omega_K)}{\log(1/\rho_K)} \right\rceil \right\}.$$

代入 ρ_K, η_K 和 ω_K 的具体值, 可得迭代次数至多为

$$\begin{aligned} N + \sum_{K=0}^N \max & \left\{ \left[-\frac{\tau_2}{\tau_1} - \frac{\log \eta_*}{\tau_1 K \log \gamma + \tau_1 \log \rho_0} \right], \left[-1 - \frac{\log \omega_*}{K \log \gamma + \log \rho_0} \right] \right\} \\ & \leq (N+1) \times \max \left\{ \left[\frac{\tau_1 - \tau_2}{\tau_1} - \frac{\log \eta_*}{\tau_1 \log \rho_0} \right], \left[-\frac{\log \omega_*}{\log \rho_0} \right] \right\} \\ & \leq \left(\left\lceil \frac{\log(\bar{\rho}/\rho_0)}{\log \gamma} \right\rceil + 1 \right) \times \max \left\{ \left[\frac{\tau_1 - \tau_2}{\tau_1} - \frac{\log \eta_*}{\tau_1 \log \rho_0} \right], \left[-\frac{\log \omega_*}{\log \rho_0} \right] \right\}. \end{aligned}$$

终止得到了对应的 x^{k+1}, λ^{k+1} 及 μ^{k+1} , 由梯度的表达式展开, 显然此时有 (3.30) 成立, 而 (3.31) 则由 (3.22) 和 (3.23) 中呈现的不等关系以及 V 的定义可得出. $\square$

另外, 由于算法所得的迭代点序列最终并不一定会收敛至可行点, 现有终止条件可能永远无法满足. 则此时结合之前的命题 3.3, 可知算法所得的迭代点序列一定会收敛至可行性函数 (3.28) 的稳定点. 所以不限制 $\{\rho_k\}$ 有界, 考虑算法最终要么收敛至满足 AKKT 条件的可行点, 要么收敛至函数 (3.28) 的稳定点, 于是可以得到如下的复杂度定理.

定理 3.3 给定容差 $\omega_*, \eta_* > 0$ 及 $\delta > 0$, 给定常数 τ_0 满足 $1 - \tau_2 < \tau_0 < 1$, 假定算法运行过程中 $\|\nabla f(x)\|$ 和 $\|\nabla c(x)\|$ 都有界, 即存在 $\bar{c} > 0$, $\max\{\|\nabla f(x^{k+1})\|, \|\nabla c(x^{k+1})\|\} \leq \bar{c}$ 对所有 $k = 0, 1, 2, \dots$ 成立, 则算法在运行至多

$$\left(\left\lceil \frac{\log(\rho_{\max}/\rho_0)}{\log \gamma} \right\rceil + 1 \right) \times \max \left\{ \left[\frac{\tau_1 - \tau_2}{\tau_1} - \frac{\log \eta_*}{\tau_1 \log \rho_0} \right], \left[-\frac{\log \omega_*}{\log \rho_0} \right] \right\} \quad (3.32)$$

次迭代, 其中

$$\rho_{\max} = \max \left\{ \frac{1}{\omega_*}, \frac{2}{\sqrt{\delta}}, \frac{4\bar{c}}{\delta}, \left(\frac{8\bar{c}M_{\tau_0}}{\delta} \right)^{\frac{1}{1-\tau_0}}, \left(\frac{8M_{\tau_0}}{\delta} \right)^{\frac{1}{1-\tau_0}} \right\}, \quad (3.33)$$

M_{τ_0} 按之前引理 3.1 方式定义得到, 之后必可以得到迭代 k 使得它满足以下两种情形之一:

(1) 算法满足终止条件, 得到 $x^{k+1} \in \mathbb{R}^n$, $\lambda^{k+1} \in \mathbb{R}^m$ 和 $\mu^{k+1} \in \mathbb{R}^n$ 满足

$$\|\nabla f(x^{k+1}) + \nabla c(x^{k+1})\lambda^{k+1} + \mu^{k+1}\| \leq \omega_*, \quad (3.34)$$

$$\|\mathcal{P}_{\mathcal{K}_c^o}(c(x^{k+1}))\| \leq \eta_*, \quad \|\mathcal{P}_{\mathcal{K}_x^o}(x^{k+1})\| \leq \eta_*; \quad (3.35)$$

(2) 得到 x^{k+1} 为可行性函数 (3.28) 的 δ -稳定点, 即

$$\|\nabla c(x^{k+1})\mathcal{P}_{\mathcal{K}_c^o}(c(x^{k+1})) + \mathcal{P}_{\mathcal{K}_x^o}(x^{k+1})\| \leq \delta, \quad (3.36)$$

其不满足可行性条件:

$$\max\{\|\mathcal{P}_{\mathcal{K}_c^o}(c(x^{k+1}))\|, \|\mathcal{P}_{\mathcal{K}_x^o}(x^{k+1})\|\} > \eta_*. \quad (3.37)$$

证明 首先说明当 $\rho_k \geq \rho_{\max}$ 时, (3.34) 和 (3.36) 成立. 对于 (3.34), 由于 $\omega_k \leq 1/\rho_k \leq 1/\rho_{\max} \leq \omega_*$, 显然成立. 关于 (3.36), 考虑此时将 (3.9) 中左侧的梯度表达式展开, 并同除 ρ_k , 得到

$$\left\| \frac{\nabla f(x^{k+1})}{\rho_k} + \nabla c(x^{k+1})\mathcal{P}_{\mathcal{K}_c^o} \left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k} \right) + \mathcal{P}_{\mathcal{K}_x^o} \left(x^{k+1} + \frac{\mu^k}{\rho_k} \right) \right\| \leq \frac{\omega_k}{\rho_k}.$$

结合函数 $\mathcal{P}_{\mathcal{K}_c^\circ}$ 和 $\mathcal{P}_{\mathcal{K}_x^\circ}$ 的定义以及投影的非扩张性, 对于任意 $a, b \in \mathbb{R}^m$, $c, d \in \mathbb{R}^n$, 有

$$\begin{aligned}\|\mathcal{P}_{\mathcal{K}_c^\circ}(a) - \mathcal{P}_{\mathcal{K}_c^\circ}(b)\| &= \|a - \mathcal{P}_{\mathcal{K}_c}(a) - (b - \mathcal{P}_{\mathcal{K}_c}(b))\| \\ &\leq \|a - b\| + \|\mathcal{P}_{\mathcal{K}_c}(a) - \mathcal{P}_{\mathcal{K}_c}(b)\| \\ &\leq 2\|a - b\|.\end{aligned}$$

同理有

$$\|\mathcal{P}_{\mathcal{K}_x^\circ}(c) - \mathcal{P}_{\mathcal{K}_x^\circ}(d)\| \leq 2\|c - d\|,$$

则此时,

$$\begin{aligned}\|\nabla c(x^{k+1})\mathcal{P}_{\mathcal{K}_c^\circ}(c(x^{k+1})) + \mathcal{P}_{\mathcal{K}_x^\circ}(x^{k+1})\| &\leq \left\| \nabla c(x^{k+1}) \left(\mathcal{P}_{\mathcal{K}_c^\circ}(c(x^{k+1})) - \mathcal{P}_{\mathcal{K}_c^\circ} \left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k} \right) \right) + \mathcal{P}_{\mathcal{K}_x^\circ}(x^{k+1}) - \mathcal{P}_{\mathcal{K}_x^\circ} \left(x^{k+1} + \frac{\mu^k}{\rho_k} \right) \right\| \\ &\quad + \left\| \nabla c(x^{k+1})\mathcal{P}_{\mathcal{K}_c^\circ} \left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k} \right) + \mathcal{P}_{\mathcal{K}_x^\circ} \left(x^{k+1} + \frac{\mu^k}{\rho_k} \right) \right\| \\ &\leq \|\nabla c(x^{k+1})\| \left\| \mathcal{P}_{\mathcal{K}_c^\circ} \left(c(x^{k+1}) + \frac{\lambda^k}{\rho_k} \right) - \mathcal{P}_{\mathcal{K}_c^\circ}(c(x^{k+1})) \right\| + \left\| \mathcal{P}_{\mathcal{K}_x^\circ} \left(x^{k+1} + \frac{\mu^k}{\rho_k} \right) - \mathcal{P}_{\mathcal{K}_x^\circ}(x^{k+1}) \right\| \\ &\quad + \frac{\omega_k}{\rho_k} + \left\| \frac{\nabla f(x^{k+1})}{\rho_k} \right\| \\ &\leq 2\bar{c} \left\| \frac{\lambda^k}{\rho_k} \right\| + 2 \left\| \frac{\mu^k}{\rho_k} \right\| + \frac{1}{\rho_k^2} + \frac{\bar{c}}{\rho_k} \leq 2M_{\tau_0}\rho_k^{\tau_0-1} + 2\bar{c}M_{\tau_0}\rho_k^{\tau_0-1} + \frac{1}{\rho_k^2} + \frac{\bar{c}}{\rho_k} \\ &\leq 2M_{\tau_0}\rho_{\max}^{\tau_0-1} + 2\bar{c}M_{\tau_0}\rho_{\max}^{\tau_0-1} + \frac{1}{\rho_{\max}^2} + \frac{\bar{c}}{\rho_{\max}}.\end{aligned}$$

由 $\rho_{\max}$ 的定义可知上述加法的每一项均不超过 $\delta/4$, 即有

$$\|\nabla c(x^{k+1})\mathcal{P}_{\mathcal{K}_c^\circ}(c(x^{k+1})) + \mathcal{P}_{\mathcal{K}_x^\circ}(x^{k+1})\| \leq \frac{\delta}{4} + \frac{\delta}{4} + \frac{\delta}{4} + \frac{\delta}{4} = \delta.$$

因此 (3.36) 成立.

下面考虑算法进行了如 (3.32) 数量次数的迭代, 若此时已有 $\rho_k \geq \rho_{\max}$, (3.34) 和 (3.36) 成立, 则此时不管 $\max\{\|\mathcal{P}_{\mathcal{K}_c^\circ}(c(x^{k+1}))\|, \|\mathcal{P}_{\mathcal{K}_x^\circ}(x^{k+1})\|\}$ 与 η_* 的关系如何, (3.35) 与 (3.37) 中必有一个成立, 原定理两种情形之一也必成立. 若在 (3.32) 数量次数的迭代中, 算法始终保持 $\rho_k \leq \rho_{\max}$, 则结合定理 3.2, 在定理 3.2 中取 $\bar{\rho} = \rho_{\max}$, 它说明了在至多 (3.32) 次迭代后, 算法会终止, 会有本定理中的第一个情形成立, 同样得证. $\square$

上述定理 3.3 说明了算法在原有终止条件外, 若考虑算法所得序列点可能始终是不可行的情形后, 可以引入容差 δ , 将得到可行性函数 (3.28) 的 δ -稳定点同样作为终止条件, 这样算法在一定有限步的迭代后也是会终止的. 注意, 这是在算法无法得到可行解的情形下, 才有必要这么做.

需要强调的是, 尽管在第 3.2 和 3.3 小节的理论分析中参考了一些已有文献的证明方法, 但由于我们在算法设计上, 尤其是在一些算法参数和 Lagrange 乘子的更新机制上有所不同, 因此证明过程中也有一些差异. 具体而言, 我们的参数动态调整策略与文献中的处理方式不同, 对 Lagrange 乘子的有界性也没有进行限制, 这些改动对收敛性分析产生了新的要求. 因此, 虽然核心思路相似, 但我们对算法证明的某些细节进行了相应的调整, 以适应这种更新带来的影响.

4 数值实验

基于第 3 节中确立的 ALM 框架和子问题求解方案, 我们在 MATLAB 上实现了算法 1 来解决一些非线性规划问题, 并将其与 MATLAB 自带的非线性规划求解函数进行了比较. 所有实验均在 PC (personal computer) 集群的单个节点上进行, 每个节点有两个 12 核 Intel Xeon 2.40 GHz CPU 和 256 GB RAM.

我们用非线性规划中广泛使用的 CUTEr 数据集^[8]作测试. CUTEr 数据集是一个广泛使用的优化测试基准, 包含多种约束和无约束优化问题, 问题规模从小规模到大规模不等, 存在着各种差异, 同时还在随着时间的推移而不断扩充. CUTEr 的特点在于它的多样性和复杂性, 适合测试算法的鲁棒性和效率, 其由英国研究人员从最初的 CUTE 数据集收集发展而来, 旨在为优化算法的研究提供标准的测试基准. 因此我们选取了部分大小合适且可求解的例子, 以便在给定时间限制内找到最优解. 同时也通过运行 MATLAB 自带的优化求解函数 fmincon, 运用其中的内点 (interior point, IP) 法与 SQP 算法解决这些问题来作为基准. 我们从算法产生的目标函数值、约束违反度、函数计算次数和计算时间这四个准则进行评估.

为了更加直观地展示算法的优势, 我们使用文献 [6] 中提出的性能分析工具, 比较三种方法的准确性和效率. 给定一个包含 n_p 个问题的测试集, 以及在每个问题上测试的算法总数 n_s , 这里由于比较了三个算法, 因此 $n_s = 3$. 这些配置提供了一种以图形方式呈现数量 $t_{p,s}$ 比较, 假设 $t_{p,s}$ 是通过第 s 个算法解决问题 p 时的某种性能指标, 如计算所需要的迭代次数或 CPU 时间. 然后计算 $t_{p,s}$ 相对于第 s 个算法在问题 p 上获得的性能比 $r_{p,s}$, 即 $r_{p,s} := \frac{t_{p,s}}{\min\{t_{p,s}: 1 \leq s \leq n_s\}}$. 如果算法 s 在问题 p 上求解失败, 定义 $r_{p,s} := \infty$. 对于任意的 $\tau \geq 0$, 定义

$$\pi_s(\tau) := \frac{\text{满足 } \log_2(r_{p,s}) \leq \tau \text{ 的问题数量}}{\text{总问题数量}}.$$

它表示算法 s 与最佳算法性能相差 $2^\tau \geq 1$ 倍. 性能分析图是每个算法 s 的关于 τ 变化的 $\pi_s(\tau)$ 曲线. 图 1 展示了比较的三种方法的目标函数值计算次数和 CPU 时间的性能表现. 特别地, 每个子图的“问题比率”轴的截点是三个算法中和最快算法相比的百分比. 从整体性能图上看, 在 CPU 时间和函数计算次数两方面, ALM 算法整体优于 MATLAB fmincon 内点法和 SQP 算法.

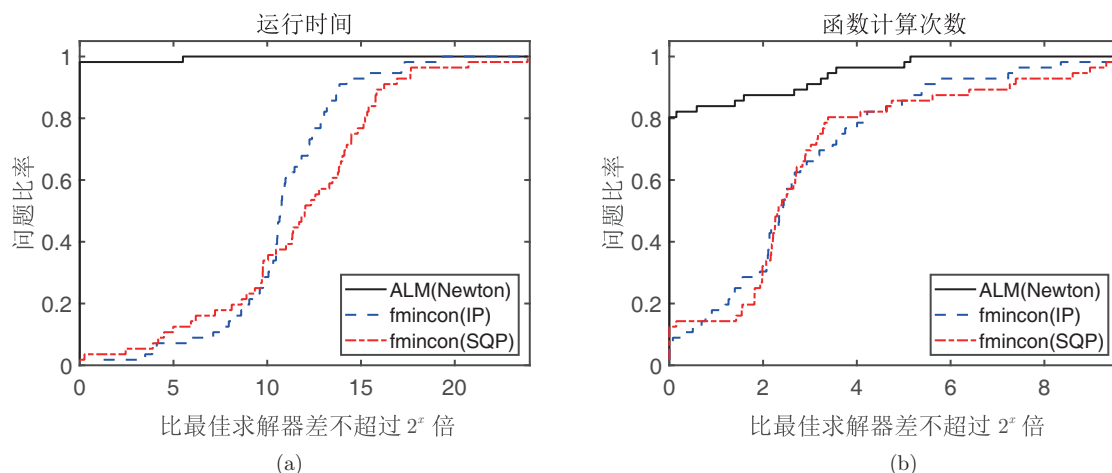


图 1 (网络版彩图) ALM、内点法与 SQP 算法在部分 CUTEr 带有盒子约束问题集上的性能表现. (a) 运行时间; (b) 函数计算次数

在本节之后的数值实验中, 取定算法 1 中的参数为 $\rho_0 = 5$, $\gamma = 2$, $\tau_1 = 0.9$, $\tau_2 = 0.6$, $\eta_* = \omega_* = 10^{-6}$. 在求解子问题时, 设置算法 2 中的参数为 $\varrho_1 = 0.01$, $\varrho_2 = 0.9$, $\gamma_0 = 0.5$, $\gamma_1 = 1$, $\gamma_2 = 10$. 我们在运行 fmincon 时, 使用默认选项和终止容差对测试集运行. 除了前面提到的停止准则, 所有算法迭代限制为 500, 时间限制为 72 小时.

4.1 带盒子约束的 CUTEr 问题集

为了评估 ALM 的性能, 首先从 CUTEr 集中选择了 56 个带盒子约束的非线性问题来构成测试集, 其中测试集的规模均在 600–10000 之间, 结果如表 1–3 和图 1 所示, 其中问题按首字母顺序排列. 符号 n 表示变量的数量, fobj 是目标函数值, feas 是约束违反度, 定义为

$$\text{feas} := \max\{\|c(x) - \mathcal{P}_{\mathcal{K}_c}(c(x))\|, \|x - \mathcal{P}_{\mathcal{K}_x}(x)\|\}.$$

feval 是目标函数计算的次数, 计算时间是以秒为单位的 CPU 时间.

从表 1–3 (表中加粗数据表示其在三个方法中表现最好) 和图 1 中的显示结果可以发现, 对于带有盒子约束的 CUTEr 数据集, 忽略机器计算产生的误差问题, 三个算法最后都可以找到可行解. 在求解质量方面, 我们的算法在大多数例子下找到可行解所对应的目标函数值是最小的, 同时花费的时间在三个算法中也是最少的. 在一些维数较大的例子上, 如 NCVXBQP1 和 NCVXBQP2, 我们算法的表现更为突出. 在找到最优解的同时, 运行时间相对于其他两个算法都有大幅度缩减.

表 1 ALM、内点法和 SQP 算法在第一部分 CUTEr 带有盒子约束问题集上的数值实验结果

问题名称	n	ALM				IP				SQP			
		fobj	feas	时间	feval	fobj	feas	时间	feval	fobj	feas	时间	feval
BDEXP	5000	4.60E-06	0.00E+00	6.74E-01	92	0.00E+00	0	3.91E+01	12	2.22E-04	0.00E+00	1.52E+01	35
BIGGSB1	5000	1.50E-02	0.00E+00	1.29E-01	89	1.64E-02	0	4.89E+03	1607	1.87E-02	0.00E+00	9.93E+03	1501
BQPGAUSS	2003	-3.63E-01	7.94E-07	4.83E-01	89	-2.07E-01	0	5.88E+02	3000	-3.55E-01	0.00E+00	2.02E+03	7494
CLPLATEA	5041	3.64E-04	0.00E+00	1.24E-02	2	-1.26E-02	0	2.03E+03	653	-1.26E-02	0.00E+00	1.53E+03	1278
CLPLATEB	5041	-5.10E-03	1.10E-07	5.45E-01	83	-5.10E-03	0	1.41E+03	374	-5.10E-03	0.00E+00	4.83E+02	822
CLPLATEC	5041	-5.02E-03	9.64E-07	4.59E-01	72	-5.88E-05	0	4.51E+03	3000	-4.14E-04	0.00E+00	6.49E+02	12080
CVXBQP1	10000	2.25E+06	8.40E-10	9.11E-01	52	2.25E+06	0	3.56E+02	6	2.25E+06	0.00E+00	2.65E+04	5
DIAGIQB	1000	-1.18E+15	0.00E+00	5.50E-01	707	-1.02E+15	0	1.92E+03	1001	-1.18E+15	0.00E+00	4.01E+01	4062
DIAGIQE	1000	-6.24E+14	0.00E+00	7.92E-01	677	-6.24E+14	0	1.10E+02	1269	-2.69E+16	2.02E-10	2.18E+02	6092
DIAGNQE	1000	-2.50E+15	0.00E+00	1.86E-01	549	-1.98E+15	0	1.37E+03	881	-2.50E+15	0.00E+00	5.89E+00	17
DIAGNQT	1000	-1.67E+15	0.00E+00	1.41E+00	3214	-1.23E+15	0	1.24E+03	1018	-1.67E+15	0.00E+00	7.71E+00	91
DIAGPQE	1000	-3.74E+00	0.00E+00	6.60E-03	20	-3.74E+00	0	1.27E+02	3000	-3.74E+00	0.00E+00	6.97E+00	7780

表 2 ALM、内点法和 SQP 算法在第二部分 CUTEr 带有盒子约束问题集上的数值实验结果

问题名称	n	ALM				IP				SQP			
		fobj	feas	时间	feval	fobj	feas	时间	feval	fobj	feas	时间	feval
TORSIONA	5476	-4.18E-01	5.08E-08	6.16E-01	66	-4.18E-01	0	8.73E+02	279	-4.18E-01	0	3.73E+03	314
TORSIONB	5476	-4.18E-01	8.12E-08	5.80E-01	64	-4.18E-01	0	7.41E+02	244	-4.18E-01	0	1.12E+04	322
TORSIONC	5476	-1.20E+00	1.38E-07	4.91E-01	54	-1.20E+00	0	8.74E+02	274	-1.20E+00	0	1.78E+03	214
TORSIOND	5476	-1.20E+00	5.50E-07	5.37E-01	58	-1.20E+00	0	7.92E+02	255	-1.20E+00	0	8.85E+03	268
TORSIONE	5476	-2.85E+00	5.89E-07	3.75E-01	42	-2.85E+00	0	7.32E+02	228	-2.85E+00	0	9.56E+02	161
TORSIONF	5476	-2.85E+00	8.33E-08	4.47E-01	49	-2.85E+00	0	6.67E+02	214	-2.85E+00	0	6.47E+03	221

表 3 ALM、内点法和 SQP 算法在第三部分 CUTEr 带有盒子约束问题集上的数值实验结果

问题名称	n	ALM				IP				SQP			
		fobj	feas	时间	feval	fobj	feas	时间	feval	fobj	feas	时间	feval
EXPLIN	1200	-7.19E+07	0.00E+00	6.67E-02	161	-7.19E+07	0	6.01E+01	145	-7.19E+07	0	5.60E+01	673
EXPLIN2	1200	-7.20E+07	0.00E+00	4.94E-02	129	-7.20E+07	0	5.30E+01	86	-7.20E+07	0	1.97E+01	304
EXPQUAD	1200	-3.69E+09	1.93E-08	3.07E-01	171	-3.69E+09	0	7.58E+01	292	-3.69E+09	0	4.55E+01	602
HARKERP2	1000	-5.00E-01	7.07E-07	2.95E+02	133	-5.00E-01	0	9.50E+01	193	-5.00E-01	0	6.49E+00	21
JNLBRNG1	10000	-1.81E-01	6.10E-07	1.13E+00	62	-1.80E-01	0	1.47E+04	1526	-1.81E-01	0	2.33E+05	1661
JNLBRNG2	10000	-4.15E+00	1.77E-07	1.29E+00	72	-4.12E+00	0	1.90E+04	3000	-4.15E+00	0	4.90E+04	3527
JNLBRNGA	10000	-7.86E-03	0.00E+00	2.26E-02	2	-2.70E-01	0	1.53E+04	1559	-2.71E-01	0	3.93E+04	1458
LINVERSE	1999	6.81E+02	2.41E-12	4.14E-01	174	6.81E+02	0	7.34E+02	2343	6.81E+02	0	2.88E+03	1246
LMINSURF	5625	9.00E+00	8.12E-07	2.84E+00	239	9.00E+00	0	4.90E+03	706	9.00E+00	0	5.83E+03	1001
MCCORMCK	5000	-4.57E+03	1.83E-07	7.11E-02	26	-4.57E+03	0	3.82E+02	166	-4.57E+03	0	3.98E+03	195
MINSURFO	5306	2.51E+00	2.48E-07	6.68E-01	57	2.52E+00	0	6.19E+03	1776	2.51E+00	0	1.36E+05	1423
NCVBQP1	10000	-1.99E+10	3.97E-15	2.50E+00	47	-1.99E+10	0	7.55E+02	12	-1.99E+10	0	2.54E+04	5
NCVBQP2	10000	-1.32E+10	8.16E-15	4.08E+01	950	-1.33E+10	0	3.23E+05	1001	-1.33E+10	0	3.34E+04	1053
NCVBQP3	10000	-6.45E+09	1.49E-11	4.92E+01	1111	-6.55E+09	0	1.79E+05	369	-6.56E+09	0	4.26E+04	2991
NLMSURF	5625	3.90E+01	4.35E-07	4.11E+00	343	3.90E+01	0	6.28E+03	1004	3.91E+01	0	3.58E+03	1009
NOBNDTOR	5476	-4.50E-01	8.70E-08	6.09E-01	66	-4.50E-01	0	1.71E+03	503	-4.50E-01	0	2.49E+03	434
NONSCOMP	5000	2.17E-14	0.00E+00	2.78E-02	18	3.99E-02	0	4.06E+03	3000	1.76E-10	0	4.32E+05	2764
OBSTCLAE	10000	1.89E+00	1.90E-07	1.33E+00	71	1.89E+00	0	7.92E+03	640	1.89E+00	0	7.76E+04	625
OBSTCLAL	10000	1.89E+00	5.25E-07	1.02E+00	54	1.89E+00	0	8.61E+03	638	1.89E+00	0	3.69E+04	402
OBSTCLBL	10000	7.27E+00	4.01E-07	1.13E+00	62	7.27E+00	0	5.50E+03	384	7.27E+00	0	6.33E+04	388
OBSTCLBM	10000	7.27E+00	1.27E-07	1.15E+00	66	7.27E+00	0	5.36E+03	368	7.27E+00	0	4.72E+04	349
OBSTCLBU	10000	7.27E+00	1.24E-07	1.30E+00	73	7.27E+00	0	5.95E+03	393	7.27E+00	0	5.63E+04	360
ODC	5184	-1.14E-02	1.96E-09	1.35E+00	128	-1.14E-02	0	7.09E+02	337	-1.14E-02	0	6.40E+02	613
ODNAMUR	11130	9.24E+03	5.83E-12	1.52E+04	449	9.24E+03	0	3.67E+04	2738	9.24E+03	0	2.76E+05	1206
PENTDI	5000	-7.50E-01	2.10E-07	8.67E-02	29	-7.49E-01	0	1.23E+03	205	-7.50E-01	0	1.35E+03	11
QR3DLS	610	1.43E-12	0.00E+00	2.46E+00	381	1.59E-03	0	4.24E+01	1931	1.91E-02	0	2.93E+00	2478
QRTQUAD	5000	-2.65E+11	5.13E-11	1.15E+01	623	-2.48E+11	0	4.71E+03	1004	-2.65E+11	0	2.65E+05	6510
QUDLIN	5000	-1.25E+09	1.13E-08	7.79E-02	59	-1.25E+09	0	1.36E+02	13	-1.25E+09	0	1.78E+03	5
RAYBENDL	2050	9.62E+01	5.50E-07	7.70E-01	432	9.63E+01	0	5.99E+02	3000	9.65E+01	0	4.70E+01	2780
RAYBENDS	2050	9.62E+01	7.34E-08	3.33E+00	120	9.62E+01	0	6.51E+02	1545	9.63E+01	0	4.97E+01	1162
SINEALI	1000	-9.99E+04	0.00E+00	5.24E-01	19	-9.99E+04	0	5.88E+00	305	-9.98E+04	0	1.36E+03	9546
SSC	5184	-2.08E+00	4.41E-08	7.50E-01	76	-2.08E+00	0	4.72E+02	173	-2.08E+00	0	4.85E+02	340
TORSION1	5476	-4.30E-01	7.75E-08	6.39E-01	66	-4.30E-01	0	9.05E+02	283	-4.30E-01	0	3.31E+03	315
TORSION2	5476	-4.30E-01	2.02E-07	5.38E-01	57	-4.30E-01	0	7.70E+02	246	-4.30E-01	0	9.55E+03	323
TORSION3	5476	-1.22E+00	1.37E-07	5.08E-01	54	-1.22E+00	0	8.74E+02	272	-1.22E+00	0	1.69E+03	214
TORSION4	5476	-1.22E+00	2.24E-08	5.08E-01	58	-1.22E+00	0	8.19E+02	255	-1.22E+00	0	7.08E+03	264
TORSION5	5476	-2.86E+00	5.90E-07	3.72E-01	42	-2.86E+00	0	7.48E+02	231	-2.86E+00	0	9.94E+02	148
TORSION6	5476	-2.86E+00	3.96E-08	4.92E-01	53	-2.86E+00	0	7.38E+02	223	-2.86E+00	0	5.62E+03	209

4.2 带一般约束的 CUTEr 问题集

接下来, 从 CUTEr 集合中选择 30 个带有一般约束的非线性问题来构成测试集, 其中这些约束包括等式、不等式和盒子约束. 测试集的变量规模高达至万, 约束个数高达 50 多万. 测试结果显示在表 4 和图 2 中, 其中符号 m 表示约束个数, 表中的 “—” 表示在给定时间限制下没有输出结果. 可以发现,

表 4 ALM、内点法和 SQP 算法在部分 CUTEr 带有一般约束问题集上的数值实验结果

问题名称	n	m	ALM				IP				SQP			
			fobj	feas	时间	feval	fobj	feas	时间	feval	fobj	feas	时间	feval
ACOPP30	72	142	5.77E+02	4.60E-10	3.81E-01	206	5.77E+02	1.81E-14	4.10E-01	62	5.77E+02	1.16E-13	1.12E-01	96
AIRPORT	84	42	4.80E+04	5.07E-09	6.20E-02	91	4.80E+04	0.00E+00	1.04E+00	652	4.80E+04	9.02E-12	6.06E-01	892
CAR2	5999	4996	2.67E+00	1.67E-09	1.96E+02	3178	2.67E+00	2.95E-12	5.71E+04	999	2.67E+00	3.35E-09	5.31E+03	809
CHARDIS1	2000	999	1.84E-13	1.04E-08	1.95E+02	85	4.59E-01	0.00E+00	5.15E+02	3000	1.06E-01	0.00E+00	5.83E+03	7927
CLEUVEN7	360	946	6.83E+02	3.48E-07	4.49E+00	180	6.84E+02	0.00E+00	7.86E+00	226	6.84E+02	1.01E-14	2.03E+01	570
COSHFUN	6001	2000	-7.73E-01	4.01E-08	5.33E+00	738	-7.64E-01	0.00E+00	1.36E+03	3000	-7.73E-01	2.87E-07	3.85E+03	2851
DEGENQPC	50	19625	4.34E-08	1.30E-10	4.92E+00	53	6.29E-03	0.00E+00	4.00E+00	21	1.36E-29	2.72E-30	1.06E+01	5
GMNCASE1	175	300	2.67E-01	6.58E-07	2.79E-01	112	2.67E-01	0.00E+00	7.98E-01	25	2.67E-01	1.31E-16	1.35E-01	33
GMNCASE2	175	1050	-9.95E-01	7.32E-07	1.78E+00	106	-9.94E-01	0.00E+00	2.57E+00	105	-9.94E-01	9.14E-17	1.50E+00	145
GMNCASE3	175	1050	1.53E+00	8.41E-09	1.70E+00	102	1.53E+00	0.00E+00	3.55E+00	191	1.53E+00	1.86E-16	2.70E+00	291
GMNCASE4	175	350	5.95E+03	2.04E-07	1.41E-01	51	5.95E+03	4.48E-07	4.98E-01	32	5.95E+03	7.82E-15	8.99E-02	15
GPP	1000	1998	2.32E+05	2.61E-10	4.27E+02	471	2.32E+05	0.00E+00	3.55E+02	2408	2.32E+05	4.13E-08	4.18E+03	6477
HAIFAM	99	150	-4.50E+01	6.59E-12	2.59E-01	549	-4.50E+01	0.00E+00	3.77E-01	139	-4.50E+01	3.93E-07	4.33E-01	105
HANGING	3600	2330	-3.15E+04	1.51E-10	5.83E+00	635	-3.15E+04	0.00E+00	4.45E+03	3000	-3.15E+04	5.71E-10	3.76E+04	3672
HIMMELBI	100	12	-1.74E+03	2.10E-07	1.48E-01	575	-1.74E+03	0.00E+00	6.57E-01	309	-1.74E+03	1.28E-14	3.07E-01	497
JANNSON4	10000	2	9.80E+03	4.81E-09	6.70E+01	61	9.80E+03	0.00E+00	1.05E+03	129	9.80E+03	3.09E-11	2.97E+04	86
KSIP	20	1001	5.76E-01	7.22E-07	9.12E-02	118	5.76E-01	0.00E+00	3.67E-01	63	5.76E-01	1.23E-16	1.13E-01	77
LEUVEN7	360	946	6.95E+02	3.68E-07	5.12E+00	206	6.95E+02	0.00E+00	9.66E+00	253	6.95E+02	1.46E-14	2.32E+01	581
LUKVL13	10000	2	1.16E+01	1.03E-08	3.81E-01	59	1.16E+01	0.00E+00	1.51E+03	309	1.16E+01	1.22E-10	4.03E+03	5262
LUKVL19	10000	6	9.99E+02	8.05E-07	5.10E-01	118	9.99E+02	0.00E+00	6.77E+03	907	9.99E+02	0.00E+00	3.56E+03	1630
MOSARQP1	2500	700	-3.82E+03	1.79E-09	4.48E-01	93	-3.82E+03	0.00E+00	3.68E+02	411	-3.82E+03	7.35E-16	2.87E+03	736
MOSARQP2	2500	700	-5.05E+03	1.43E-07	4.49E-02	33	-5.05E+03	0.00E+00	3.11E+02	444	-5.05E+03	1.70E-15	8.17E+02	641
OET5	5	1002	2.65E-03	6.63E-07	2.69E-01	768	2.69E-03	0.00E+00	1.69E+00	1234	2.65E-03	1.00E-07	6.63E-02	49
POLYGEVEN	400	20296	-7.85E-01	9.24E-07	8.69E+01	880	-7.85E-01	1.22E-15	1.17E+02	613	-7.85E-01	7.68E-08	8.48E+01	1002
POLYGON	200	5049	-7.85E-01	7.01E-07	1.08E+01	625	-7.20E-01	0.00E+00	8.81E+00	432	-7.85E-01	2.30E-10	7.52E+00	653
SAROMM	5120	5110	5.74E+01	2.24E-07	2.97E+01	514	5.74E+01	7.11E-08	1.97E+04	1011	5.74E+01	1.18E-12	6.62E+03	692
SINROSNB	1000	999	1.41E+00	0.00E+00	8.17E-02	97	1.95E+05	0.00E+00	1.74E+02	3000	1.00E-03	0.00E+00	4.26E+02	4401
SMBANK	117	64	-7.13E+06	3.32E-07	3.98E-02	98	-7.13E+06	1.72E-09	3.11E+00	1023	-6.88E+06	1.75E-10	2.06E+00	1097
TAX213322	1512	570781	-1.20E+10	9.23E-07	2.66E+03	1199	-5.21E+03	5.87E-08	1.60E+04	3000	-	-	-	-
TAX53322	360	32221	-2.93E+09	7.48E-07	5.46E+01	1215	-1.30E+03	2.55E-03	2.62E+02	2081	-	-	-	-

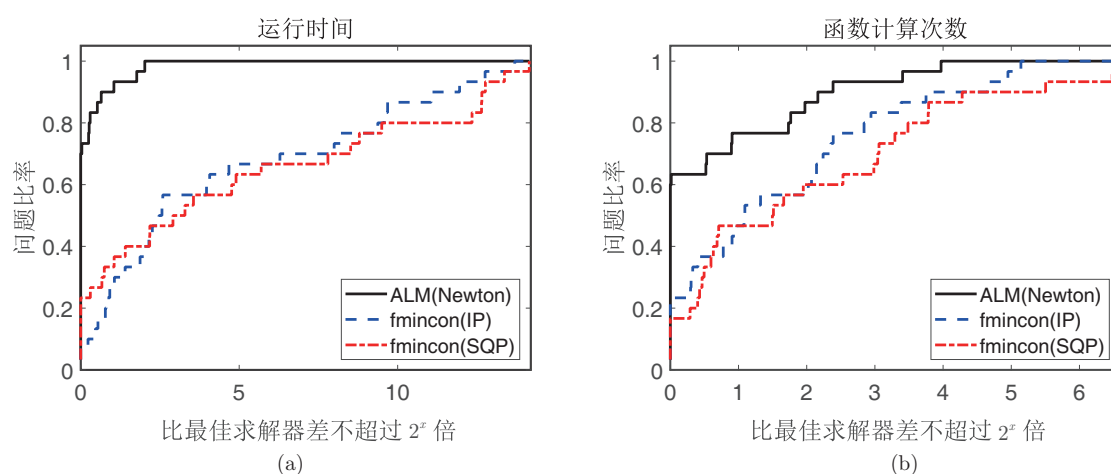


图 2 (网络版彩图) ALM、内点法与 SQP 算法在 CUTEr 带有一般约束问题集上的性能表现. (a) 运行时间; (b) 函数计算次数

即使是对于一些包含等式和不等式的约束问题, 我们的算法在这类问题上的表现依旧优异, 可以在较短时间内得到更为精确的解.

5 结论

本文致力于研究带有盒子约束的非线性规划问题. 以往的 ALM 方法需要在盒子约束下近似求解增广 Lagrange 函数子问题, 而不同于这些方法, 我们通过对约束进行拆分, 得到了一个子问题是无约束优化的 ALM. 为了快速有效求解此无约束子问题, 我们利用二阶近似等技巧得到该增广 Lagrange 函数的二阶信息, 并应用半光滑 Newton 方法来加速子问题的求解. 借助变分分析等方法, 本文对所确立的 ALM 外层框架进行了基本的收敛性、复杂度分析, 即使在 Lagrange 乘子是无界的情形下. 通过数值实验验证了在 CUTEr 测试集中的某些大规模问题上, 我们所提出的 ALM 收敛速度快, 结果精确, 可以与内点法和 SQP 方法相提并论.

参考文献

- 1 Andreani R, Haeser G, Martínez J M. On sequential optimality conditions for smooth constrained optimization. *Optimization*, 2011, 60: 627–641
- 2 Andreani R, Martínez J M, Ramos A, et al. A cone-continuity constraint qualification and algorithmic consequences. *SIAM J Optim*, 2016, 26: 96–110
- 3 Andreani R, Ramos A, Ribeiro A A, et al. On the convergence of augmented Lagrangian strategies for nonlinear programming. *IMA J Numer Anal*, 2022, 42: 1735–1765
- 4 Birgin E G, Martínez J M. Complexity and performance of an augmented Lagrangian algorithm. *Optim Methods Softw*, 2020, 35: 885–920
- 5 Clarke F H. *Optimization and Nonsmooth Analysis*. New York: John Wiley & Sons, 1983
- 6 Dolan E D, Moré J J. Benchmarking optimization software with performance profiles. *Math Program*, 2002, 91: 201–213
- 7 Gill P E, Murray W, Wright M H. *Practical Optimization*. London: Academic Press, 1981
- 8 Gould N I M, Orban D, Toint P L. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Trans Math Software*, 2003, 29: 373–394
- 9 Hallak N, Teboulle M. An adaptive Lagrangian-based scheme for nonconvex composite optimization. *Math Oper Res*, 2023, 48: 2337–2352

- 10 Hermans B, Themelis A, Patrinos P. QPALM: A proximal augmented Lagrangian method for nonconvex quadratic programs. *Math Program Comput*, 2022, 14: 497–541
- 11 Hestenes M R. Multiplier and gradient methods. *J Optim Theory Appl*, 1969, 4: 303–320
- 12 Hu J, Jiang B, Lin L, et al. Structured quasi-Newton methods for optimization with orthogonality constraints. *SIAM J Sci Comput*, 2019, 41: A2239–A2269
- 13 Jia X X, Kanzow C, Mehlitz P, et al. An augmented Lagrangian method for optimization problems with structured geometric constraints. *Math Program*, 2023, 199: 1365–1415
- 14 Jorge N, Stephen J W. *Numerical Optimization*. New York: Springer, 2006
- 15 Krejić N, Martínez J M, Mello M, et al. Validation of an augmented Lagrangian algorithm with a Gauss-Newton Hessian approximation using a set of hard-spheres problems. *Comput Optim Appl*, 2000, 16: 247–263
- 16 Marchi A D, Jia X X, Kanzow C, et al. Constrained composite optimization and augmented Lagrangian methods. *Math Program*, 2023, 201: 863–896
- 17 Mifflin R. Semismooth and semiconvex functions in constrained optimization. *SIAM J Control Optim*, 1977, 15: 959–972
- 18 Powell M J D. A method for nonlinear constraints in minimization problems. In: *Optimization*. London-New York: Academic Press, 1968, 283–298
- 19 Qi L Q, Sun J. A nonsmooth version of Newton’s method. *Math Program*, 1993, 58: 353–367
- 20 Rockafellar R T. Convergence of augmented Lagrangian methods in extensions beyond nonlinear programming. *Math Program*, 2023, 199: 375–420

A semismooth Newton augmented Lagrangian method for solving nonlinear programming

Yuxin Zhou, Jiayuan Wu, Rui Wang, Jie Song & Zaiwen Wen

Abstract The augmented Lagrangian method (ALM) has been widely used to solve nonlinear programming problems. Although there has been extensive research in the algorithmic aspect, the box constraints are typically preserved during the minimization of the augmented Lagrangian function. In this paper, we propose an equivalent reformulation of the problem by splitting the variables in the constraint. We then establish the augmented Lagrangian function based on projection operators to propose an ALM framework, where the subproblem is an unconstrained optimization problem. For the ALM subproblem, we utilize efficient techniques to approximate the Hessian matrix of the augmented Lagrangian function and design an inexact semismooth Newton method as the solver. We also provide convergence and complexity properties of the algorithm. Finally, we conduct numerical experiments and comparisons with the interior point and sequential quadratic programming methods on nonlinear programming instances from the CUTer (constrained and unconstrained testing environment) dataset. The results demonstrate the effectiveness of the proposed ALM.

Keywords nonlinear programming, augmented Lagrangian method, KKT condition, semismooth Newton method, convergence, complexity

MSC(2020) 49M15, 65K05, 90C26

doi: 10.1360/SSM-2023-0259