

分类号 O224

密 级

U D C

编 号 10486

武汉大学

硕 士 学 位 论 文

线性约束的凸二次规划求解算法研究

研 究 生 姓 名 : 陈文标

学 号 : 2019202010014

指导教师姓名、职称 : 洪桂祥 教授

专 业 名 称 : 基础数学

研 究 方 向 : 非线性优化

二〇二二年五月

Research the Algorithms Solving Convex Quadratic Programming with Linear Constraints

Candidate: WENBIAO CHEN

Student Number: 2019202010014

Supervisor: PROF. GUIXIANG HONG

Major: Pure Mathematics

Speciality: Nonlinear Optimization



School of Mathematics and Statistics
WUHAN UNIVERSITY

May, 2022

论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的研究成果。除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

学位论文作者（签名）：

年 月 日

摘 要

二次规划问题是非线性优化的一个重要研究领域，其在现实生活中有着诸多运用，而本文的主要研究对象则是线性约束下的凸二次规划问题求解算法。

本文从等式约束凸二次规划问题入手，探究其求解方法，并以此为基础进一步介绍三种求解非等式约束凸二次规划问题的方法，包括有效集法、内点法和梯度投影法，并根据对各种求解算法思路的分析总结，提出了基于 CQP 图象特性的梯度投影方法。

通过数值实验，本文验证了有效集法、原始-对偶路径跟随法和梯度投影法的可行性、对不同维度问题的适用性和在可行域内的收敛性，以及基于 CQP 图象特性的梯度投影法的可行性，并初步确定改进方法确实可以降低迭代次数，提高计算效率。

关键词：凸二次规划；线性约束；CQP 图象特性；梯度投影法

Abstract

The quadratic programming problem is an important research field of nonlinear optimization, and it has many applications in real life. The main research object of the thesis is the algorithm of solving convex quadratic programming problem with linear constraints.

The thesis starts with the equality-constrained convex quadratic programming problem, explores its solution, and further introduces three methods to solve the non-equality-constrained convex quadratic programming problem, including the active set method, interior point method and gradient projection method. What's more, the gradient projection method based on the properties of CQP image is proposed by the analysis and summary of various algorithm ideas.

Through numerical experiments, the thesis verifies the feasibility of the active set method, the primal-dual path following method and the gradient projection method, the adaptability to problems of different dimensions and the convergence in the feasible region. Moreover, it also verifies the feasibility of the gradient projection method based on the properties of CQP image, and preliminarily verifies that the improved method can indeed reduce the number of iterations and improve the computational efficiency.

Key Words: convex quadratic programming; linear constraints; properties of CQP image; gradient projection method

目 录

摘 要	I
Abstract	III
第 1 章 绪论.....	1
1.1 凸二次规划的研究意义和现状	1
1.2 本文的研究内容	2
第 2 章 线性约束凸二次规划预备知识.....	5
2.1 相关理论及概念	5
2.2 二次规划的数学模型	7
第 3 章 等式约束凸二次规划求解方法.....	9
3.1 变量消去法.....	9
3.2 KKT 方法	10
3.2.1 KKT 法的原理.....	10
3.2.2 求解 KKT 系统的零空间法.....	11
3.2.3 基于零空间求解 KKT 系统算法步骤.....	12
3.3 交替方向乘子法	13
第 4 章 非等式约束凸二次规划求解方法.....	15
4.1 有效集法	16
4.1.1 有效集法的原理.....	16
4.1.2 有效集算法步骤.....	18
4.2 内点法.....	18
4.2.1 内点罚函数法的原理.....	19
4.2.2 内点罚函数算法步骤.....	20
4.2.3 原始-对偶路径跟随法的原理.....	20
4.2.4 原始-对偶路径跟随算法步骤.....	22
4.3 梯度投影法.....	23
4.3.1 梯度投影法的原理.....	23
4.3.2 梯度投影算法步骤.....	25
第 5 章 基于 CQP 图象特性的梯度投影法.....	27
5.1 改进算法的思路和步骤	27
5.2 算法的适用性和收敛性说明	29
第 6 章 数值实验.....	31
6.1 实验目的及设置.....	31
6.2 实验结果及分析.....	31
第 7 章 总结与展望.....	37
参考文献	39
致 谢	43

第 1 章 绪论

1.1 凸二次规划的研究意义和现状

凸二次规划通常作为非线性规划中的子问题出现^[1, 2], 随着社会经济的不断发展, 基于凸二次规划的模型被运用到许多领域, 包括组合投资^[3, 4]、最优控制^[5]、参数选择^[6]、信号处理^[7]和机器学习^[8]等。自上世纪 50 年代^[9]起, 学术界对凸二次规划问题的研究就没有停止过, 各领域的研究者陆续提出来了多种数值求解方法, 例如有效集法^[10, 11]、内点法^[12-15]、梯度投影法^[16, 17]等, 它们在处理不同维度和类型的二次规划问题上各具特点, 并且这些方法还构成了非线性规划中的顺序二次规划方法^[18]和混合整数优化的分支定界法^[19]的基础。由于凸二次规划问题在实践和理论方面的广泛运用, 使得当前社会对更高效可靠的求解方法有着不断增加的需求。

有效集法起源于线性规划的单纯形法^[20], 这种方法根据给定的可行解来确定一组有效约束, 进而构建对应的等式约束子问题, 并在迭代中不断调整子问题约束条件, 同时限制迭代点可行, 通过求解子问题来间接获取原问题的最优解。对于中小型问题, 有效集方法通常可以非常快地获得最优解, 而且可以很容易热启动, 并能做到有限收敛。此外, 通过在约束集合中添加和删除约束, 可以采用分解更新来求解连续线性系统^[21]。然而, 这种方法在求解过程中可能需要多次迭代来识别正确的有效约束集, 并且它们不能很好地扩展, 因为它们难以利用稀疏性, 同时对提前终止不具有鲁棒性, 即中间迭代不能以有意义的方式逼近最优解^[22]。近些年 Bartlett^[23]、Forsgren^[24]、Fält^[25]等人通过引入对偶理论继续优化该算法, 并将其运用到了混合约束二次规划问题的求解。

内点法是求解凸二次规划的一类方法, 目前常用的包括基于 log-barrier 函数的内点罚函数法^[15]和原始-对偶路径跟随法^[12, 13]。其中内点罚函数法主要是通过 log-barrier 函数将原问题的不等式约束构造为罚函数, 进而将带约束的凸二次规划问题转化为无约束的增广目标函数来求解; 而原始-对偶路径跟随法则是由原问题与对偶问题之间的对偶间隙入手, 通过引入松弛变量和对偶参数, 运用 Newton 法将松弛化的 KKT 条件转化为一个线性系统, 进而通过求解线性系统, 来不断迭代缩小对偶间隙至 0, 最终逼近原问题的最优解。内点法的出现使得二次规划问题能在多项式时间内得到解决, 促进了二次规划模型在大规模问题上的运用。它是一种快速、高效、对提前终止具有鲁棒性的算法, 并且相比于增广拉格朗日法, 内点法可以充分利用数值软件库中的稀疏矩阵子程序来解决大型稀疏问题, 降低算法的编程难度, 不过它不能像有效集方法一样进行热启动^[26]。由于内点法可在多项式时间内求解问题以及对提前终止具有鲁棒性的特点, 使得它一直得到学界的

重视^[27-29]，近些年 Absil^[30]、Friedlander^[31]、Poukakiotis^[32]通过结合其他方法或引入正则化技术进一步改进了内点法，优化了其计算性能。

梯度投影法是一种求解约束优化问题的通用方法，主要思想是通过将负梯度或者负梯度的投影作为搜索方向，迭代逼近二次规划问题的最优解。相比于有效集方法在每次迭代中，最多向有效集内添加或删除一个约束，梯度投影法允许有效集在迭代之间快速变化，这使得它比起有效集法更适合求解大规模的优化问题，而且该方法特别适用于有界约束问题，能保证在有限次迭代中快速识别出最优解对应的有效集。该方法不仅可以求解凸二次规划问题，还可以求解非凸二次规划问题^[1]，因此具有良好的适应性。不过为了高效地进行投影，该算法一般需要约束条件足够简单。近些年 Crisci^[33, 34]、Naik^[35, 36]进一步研究了梯度投影法的步长选择以及在嵌入式混合整数二次规划问题的运用，完善了该方法的理论基础，并验证了该方法在求解非凸二次规划的良好性能。

除了上述方法外，求解凸二次规划的方法还包括早期提出的扩展的单纯形法^[37]、椭球法^[38]、增广拉格朗日法^[39-41]、共轭梯度法^[42, 43]、一阶方法^[9]等，其中像拓展的单纯形法和椭球法已经日渐式微，不过像增广拉格朗日法和一阶方法近些年正在复兴，Bueno^[44]、Necoara^[45, 46]等人相继提出了相关的改进方法，为旧方法注入了新活力。近一段时间 Dominic^[22]、Marchi^[21]等人提出运用半光滑牛顿法来求解凸二次规划问题，并取得了不错的效果。

以上的各位学者都做出了出色的研究工作，奠定了本文关于凸二次规划问题研究的理论基础，同时也加快了凸二次规划问题求解方法的传播，促进了凸二次规划模型在实践中的运用与发展，其中部分方法的求解思想构成了本文改进方法的基础。

1.2 本文的研究内容

本文的研究对象是线性约束的凸二次规划问题，目前学界针对该方法的研究已有数十年，相关理论知识也较为完备。本文的主要工作是将求解方法分为等式约束和非等式约束两类，对这些方法的求解思路和算法步骤进行介绍和分析，最后提出了一种改进方法。

本文内容安排如下：在第二章，介绍线性约束优化问题的相关理论和二次规划问题的模型；在第三章，介绍等式约束下的凸二次规划问题的求解方法，为后续介绍非等式约束凸二次规划问题求解方法奠定基础；在第四章，介绍三种常见的非等式约束凸二次规划问题求解方法，包括有效集法、内点法和梯度投影法，而这也是本文的主体；在第五章，介绍基于 CQP 图象特性的梯度投影法的理论和步骤，并对算法的适应性和收敛性进行必要说明；在第六章，将进行数值试验，

验证有效集法、原始-对偶路径跟随法、梯度投影法、改进方法的相关性能，并基于实验数据来分析这些算法；在最后一章，将对全文做出总结，同时对二次规划问题的研究进行展望。

本文提出的改进算法充分利用了线性约束的凸二次规划问题在图象上的凸性以及目标函数最优解的特点，针对大规模问题，可以在理论上进一步降低 Rosen 梯度投影法的迭代次数，间接提高算法效率。而文中的数值实验在一定程度上验证了三种常用求解方法的可行性、全局收敛性和适应性，以及改进方法的可行性和有效性，这些研究也将为后续研究者提供参考思路。

第 2 章 线性约束凸二次规划预备知识

2.1 相关理论及概念

(1) 无约束问题的最优性充要条件：设 $f(x)$ 是 R^n 上的一阶连续可微的凸函数，则存在 $x^* \in R^n$ 为如下优化问题

$$\min f(x) \quad (2.1)$$

的全局最优解的充要条件是 $\nabla f(x^*) = 0$ 。

(2) KKT 必要条件^[47, 48]：考虑一般约束的凸优化问题

$$\begin{aligned} \min & f(x), \\ \text{s.t.} & h_i(x) = 0, \quad i = 1, 2, \dots, l, \\ & g_i(x) \geq 0, \quad i = 1, 2, \dots, m, \end{aligned} \quad (2.2)$$

其中目标函数 $f(x)$ 和约束函数 $h_i(x)$ 、 $g_i(x)$ 都可微。记问题 (2.2) 的可行域为 D ，则局部最优解 x^* 满足下列条件

$$\begin{cases} h_i(x^*) = 0, & i = 1, 2, \dots, l, \\ g_i(x^*) \geq 0, & i = 1, 2, \dots, m, \\ \nabla f(x^*) - \sum_{i=1}^l v_i^* \nabla h_i(x^*) - \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) = 0, \\ \lambda_i^* \geq 0, & i = 1, 2, \dots, m, \\ \lambda_i^* g_i(x^*) = 0, & i = 1, 2, \dots, m, \end{cases} \quad (2.3)$$

其中 $v = (v_1, v_2, \dots, v_l)$ ， $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ 被称为 Lagrange 乘子，满足条件 (2.3) 的 $x^* \in D$ 被称为 KKT 点。对于可微的凸规划问题，KKT 点即为其全局最优解，记最优值 $p^* = f(x^*)$ 。

(3) 有效集定义：对于如下的不等式约束优化问题

$$\begin{aligned} \min & f(x), \\ \text{s.t.} & g_i(x) \geq 0, \quad i = 1, 2, \dots, m. \end{aligned} \quad (2.4)$$

设 \hat{x} 是问题 (2.4) 的可行解，若 $g_i(\hat{x}) = 0$ ，则称不等式约束 $g_i(x) \geq 0$ 为 \hat{x} 的有效约束；否则，若 $g_i(\hat{x}) > 0$ ，则称不等式约束 $g_i(x) \geq 0$ 为 \hat{x} 的非有效约束。称集合

$$\mathcal{A}(\hat{x}) = \{i \mid g_i(\hat{x}) = 0, i = 1, 2, \dots, m\} \quad (2.5)$$

为可行解 \hat{x} 处的有效约束指标集，简称为 \hat{x} 处的有效集或活动集，最优解对应的有效集被称为最优有效集。

(4) 投影矩阵定义：若矩阵 $P \in R^{n \times n}$ 满足条件

$$P = P^T, \quad P^2 = P, \quad (2.6)$$

则称矩阵 P 为投影矩阵。

(5) 凸优化问题的对偶问题：考虑一般约束的凸优化问题 (2.2) 的对偶问题

$$\begin{aligned} \max g(v, \lambda) = \inf_{x \in D} & \left\{ f(x) - \sum_{i=1}^l v_i h_i(x) - \sum_{i=1}^m \lambda_i g_i(x) \right\}, \\ \text{s.t. } & \lambda_i \geq 0, \quad i = 1, 2, \dots, m. \end{aligned} \quad (2.7)$$

设 (v^*, λ^*) 为问题 (2.7) 的最优解，则对应的最优值 $d^* = g(v^*, \lambda^*)$ ，且与问题 (2.2) 的最优值 p^* 满足如下弱对偶关系：

$$d^* \leq p^*. \quad (2.8)$$

(6) 相对内部：定义集合 D 的相对内部为

$$\text{relint } D = \{x \in D \mid B(x, r) \cap \text{aff } D \subseteq D, r > 0\}, \quad (2.9)$$

其中 $B(x, r)$ 表示以 x 为中心， r 为半径的球， $\text{aff } D$ 表示集合 D 的仿射包。

(7) Slater 条件^[49]：如果原问题(2.2)是一个凸优化问题，且存在 $x \in \text{relint } D$ 使得 $h_i(x) = 0, i = 1, 2, \dots, l$ 和 $g_i(x) < 0, i = 1, 2, \dots, m$ 成立，则原问题(2.2)和对偶问题(2.7)是强对偶的，即 $p^* = d^*$ 。

(8) 弱 Slater 条件：如果原问题(2.2)是一个凸优化问题，并且有不等式约束 $g_i(x) < 0, i = 1, 2, \dots, m$ 为仿射集，则只要可行域 D 非空，必有 $p^* = d^*$ 。

(9) 算法收敛速判别^[50]

算法的收敛速度一般由算法产生的点列 $\{x_k\}$ 的收敛速度来度量。设迭代算法产生的点列为 $\{x_k\}$ ，且该点列收敛于点 x^* ，如果满足以下判别式

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} = q, \quad (p \in \mathbb{Z}, p \geq 1). \quad (2.10)$$

当 $p = 1$ 且 $q = 0$ 时，则该算法是超线性收敛的；

当 $p = 1$ 且 $0 < q < 1$ 时，则该算法是线性收敛的；

当 $p > 1$ 且 $q = 0$ 时。则该算法是超 p 阶收敛的；

当 $p > 1$ 且 $0 < q < 1$ 时，则该算法是 p 阶收敛的；

其中 $p \geq 1$ 为整数， $q \geq 0$ 为实数。

(10) 迭代算法的终止条件^[50]

为了避免迭代算法陷入死循环，需给算法设定合适的终止条件，一般终止条件可以分为下列三类：

基于变量 $\|x_{k+1} - x_k\| < \varepsilon$ ；

基于目标函数 $\|f(x_{k+1}) - f(x_k)\| < \varepsilon$ ；

基于目标函数梯度 $\|g(x_k)\| = \|\nabla f(x_k)\| < \varepsilon$ ；

其中 $0 < \varepsilon \ll 1$ 。

2.2 二次规划的数学模型

二次规划是一种特殊的非线性规划，其目标函数是二次的，而约束条件是线性的，一般可以表示为下列形式：

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T Q x + c^T x, \\ \text{s.t. } x &\in D, \end{aligned} \quad (2.11)$$

其中 x 是一个 $n \times 1$ 的列向量， Q 是一个 $n \times n$ 的对称矩阵， D 是由一系列线性等式约束或不等式约束所确定的多面体^[2]。根据 Q 是否为半正定，可以对二次规划问题进行分类：当 Q 不是半正定矩阵，则问题 (2.11) 为非凸二次规划问题，这也是一个 NP 难问题；当 Q 是半正定矩阵时，则问题 (2.11) 为凸二次规划问题，特别的若 Q 为严格正定矩阵，则问题 (2.11) 为严格凸二次规划问题。目前针对凸二次规划问题的理论基础较为完备，不过在提高算法求解效率的研究上却一直没有停止。

本文探讨重点将放在凸二次规划上，主要介绍等式约束和非等式约束下的常用求解方法，以及基于 CQP 图象特性的改进方法。

第3章 等式约束凸二次规划求解方法

对于等式约束凸二次规划，本章将介绍三种求解方法，即变量消去法、KKT法和交替方向乘子法。变量消去法主要是利用等式约束求解出变量 x ，再将求解出的结果代入目标函数 $f(x)$ ，进而化约束优化问题为无约束优化问题来进行求解；而KKT法则是运用了目标函数的凸性，通过二次规划问题的KKT点来间接获得目标函数的全局最优解；最后的交替方向乘子法则是根据目标函数的可分性，将原问题分解成一系列子问题，通过求解这些子问题来间接求出原问题的最优解，而且分解后的子问题可以运用并行计算技术，提高求解效率。

3.1 变量消去法

考虑如下的等式约束凸二次规划问题：

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T Q x + c^T x, \\ \text{s.t. } a_i^T x &= b_i \quad (i=1, 2, \dots, m), \end{aligned} \quad (3.1)$$

其中 x 是 $n \times 1$ 的列向量， Q 是 $n \times n$ 的对称正定矩阵， a_i 是 $n \times 1$ 的列向量， b_i 是实数。为了简化说明，记 $A = (a_1, a_2, \dots, a_m)^T_{m \times n}$ ， $b = (b_1, b_2, \dots, b_m)^T$ ，则问题(3.1)可以化为

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T Q x + c^T x, \\ \text{s.t. } Ax &= b, \end{aligned} \quad (3.2)$$

其中 A 为 $m \times n$ 的行满秩的矩阵($\text{rank } A = m$)， b 是一个 $n \times 1$ 的列向量。由于矩阵 A 不是一个方阵，因此不能直接对矩阵 A 求逆。可以将矩阵 A 分块，变为一个 $m \times m$ 的方阵 B 和 $m \times (n-m)$ 矩阵 N ，变量 x 也分块为对应的 x_B 和 x_N ，所以问题(3.2)的约束条件可以转化为

$$Ax = \begin{pmatrix} B & N \end{pmatrix} \begin{pmatrix} x_B \\ x_N \end{pmatrix} = Bx_B + Nx_N = b. \quad (3.3)$$

由于 B 是一个可逆的方阵，因此可进一步求得

$$\begin{aligned} x_B &= B^{-1}b - B^{-1}Nx_N, \\ x &= \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}b - B^{-1}Nx_N \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix} + \begin{pmatrix} -B^{-1}N \\ E_{n-m} \end{pmatrix} x_N. \end{aligned}$$

为了便于说明，记 $x_0 = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}_{n \times 1}$ ， $P = \begin{pmatrix} -B^{-1}N \\ E_{n-m} \end{pmatrix}_{n \times (n-m)}$ ，则 $x = x_0 + Px_N$ 。将求出

的 x 代入目标函数，则可得到一个关于 x_N 的等价无约束二次规划问题：

$$\begin{aligned}\min f(x_N) &= \frac{1}{2}(x_0 + Px_N)^T Q(x_0 + Px_N) + c^T(x_0 + Px_N) \\ &= \frac{1}{2}x_N^T P^T Q P x_N + (x_0^T Q P + c^T P)x_N + (x_0^T Q + c^T)x_0.\end{aligned}\quad (3.4)$$

对 (3.4) 中的目标函数求梯度可得

$$\nabla f(x_N) = P^T Q P x_N + x_0^T Q P + c^T P. \quad (3.5)$$

因为 $P^T Q P$ 正定，且 $f(x_N)$ 一阶连续可微，运用无约束问题的最优性条件，可得问题 (3.4) 的存在唯一最优解 x_N^* ，使得 $\nabla f(x_N^*) = 0$ ，进一步可求出问题 (3.2) 的唯一最优解 $x^* = x_0 + Px_N^*$ 。

3.2 KKT 方法

3.2.1 KKT 法的原理

由于在凸二次规划问题中，局部最优解即为全局最优解，所以可以通过求解凸二次规划问题的 KKT 点，间接求得下面优化问题的全局最优解。

$$\begin{aligned}\min f(x) &= \frac{1}{2}x^T Q x + c^T x, \\ \text{s.t. } Ax &= b,\end{aligned}\quad (3.6)$$

其中 Q 是 $n \times n$ 的对称正定矩阵， x 是 $n \times 1$ 的列向量， A 为 $m \times n$ 的行满秩的矩阵。设 x^* 为问题 (3.6) 的最优解，则满足 KKT 条件：

$$\begin{cases} Qx^* + c - A^T \lambda^* = 0, \\ Ax^* = b, \end{cases} \quad (3.7)$$

其中 λ^* 为对应的 Lagrange 乘子。将 (3.7) 中的等式组表示为如下矩阵形式：

$$\begin{pmatrix} Q & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} -c \\ b \end{pmatrix}. \quad (3.8)$$

下面说明 KKT 条件 (3.7) 的解是唯一的。令 $x^* = x + p$ ，其中 x 是问题 (3.7) 的预估解， p 表示步长，则可进一步将方程 (3.8) 转化为

$$\begin{pmatrix} Q & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} -p \\ \lambda^* \end{pmatrix} = \begin{pmatrix} g \\ h \end{pmatrix}, \quad (3.9)$$

其中 $g = c + Qx$ ， $h = Ax - b$ ，(3.9) 被称为 KKT 系统，系统中的矩阵被称为 KKT 矩阵。

定理 3.1 若矩阵 A 是行满秩的，且矩阵 $Z^T Q Z$ 是正定的，则 KKT 矩阵

$$K = \begin{pmatrix} Q & A^T \\ A & 0 \end{pmatrix} \quad (3.10)$$

是非奇异的, 故存在唯一的 KKT 对 (x^*, λ^*) 满足 (3.7) 中的 KKT 条件, 其中 Z 是一个 $n \times (n-m)$ 的满秩矩阵, 且满足 $AZ=0$ 。

证明: 假设 $n \times 1$ 向量 s 和 $m \times 1$ 维向量 t 满足条件

$$\begin{pmatrix} Q & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix} = 0, \quad (3.11)$$

所以 $As=0$, 即 s 在矩阵 A 的零空间里, 于是

$$\begin{pmatrix} s \\ t \end{pmatrix}^T \begin{pmatrix} Q & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix} = s^T Qs = 0.$$

又因为 s 在矩阵 A 的零空间, 故可令 $s = Zu, u \in R^{n-m}$, 所以

$$s^T Qs = u^T Z^T QZ u = 0,$$

根据定理给定的前提 $Z^T QZ$ 是正定的, 可知 $u=0$, 所以 $s=0$, 结合 (3.11) 可得

$$Qs + A^T t = A^T t = 0.$$

由于矩阵 A 是行满秩的, 所以 $t=0$ 。因此当且仅当 $s=0, t=0$ 时, 等式 (3.11) 成立, 所以矩阵 K 是非奇异的, 证毕。

当定理 3.1 中的条件满足时, 利用高斯消元法可知 (3.9) 中的 KKT 系统存在如下最优解:

$$\begin{aligned} p &= -(A^T A)^{-1} A^T h, \\ x^* &= x + p, \\ \lambda^* &= (AA^T)^{-1} A(Qp + g). \end{aligned}$$

不过这样直接求解涉及到计算矩阵的逆, 特别是当矩阵 A 的维度较大或者较稀疏时, 可能导致 $A^T A$ 或 AA^T 为奇异矩阵, 进而无法计算出结果, 下面介绍其他直接求解 KKT 系统的方法。

3.2.2 求解 KKT 系统的零空间法

目前求解 KKT 系统 (3.9) 的方法除了高斯消元法, 常用的直接求解方法还包括 Schur-Complement 方法、零空间法等^[1, 50], 其中零空间法不需要对矩阵 Q 的奇异性做要求, 只需满足定理 3.1 中的条件即可求解 KKT 系统, 因此相比高斯消元法和 Schur-Complement 方法具有更广泛的适用性。

本节介绍重点将放在零空间法上, 其求解思路是: 利用 KKT 系统中的块结构将 (3.9) 分解为两个较小的系统。假设 KKT 系统 (3.9) 中的向量 p 可以分解成如下两部分:

$$p = Yp_Y + Zp_Z, \quad (3.12)$$

其中 Z 是 $n \times (n-m)$ 零空间矩阵, Y 是任意 $n \times m$ 矩阵, 使得 (Y, Z) 是非奇异的, p_Y 是 $m \times 1$ 的向量, p_Z 是 $(n-m) \times 1$ 的向量。易知 $AZ = 0$, 而 Yx_Y 是 $Ax = b$ 的一个特殊解, 因此求解出 p 的关键在于求出 p_Y 和 p_Z 。

通过将 p 的分解式 (3.12) 代入到 (3.9) 的等式中, 结合 $AZ = 0$ 这一已知条件, 可得

$$\begin{cases} -Q(Yp_Y + Zp_Z) + A^T \lambda^* = g, \\ -AYp_Y = h. \end{cases} \quad (3.13)$$

由于矩阵 A 是行满秩的且 (Y, Z) 是非奇异的, 所以 AY 是非奇异的 $m \times m$ 矩阵, 于是由等式 (3.14) 可解出 $p_Y = -(AY)^{-1}h$ 。接着求解 p_Z , 在等式 (3.13) 的两边右乘矩阵 Z^T 可得

$$Z^T Q Z p_Z = -Z^T Q Y p_Y - Z^T g, \quad (3.15)$$

于是由 (3.15) 可以求出 p_Z 。

结合 p_Y 和 p_Z 可计算总步长 $p = Yp_Y + Zp_Z$, 间接求出 KKT 系统的最优解 $x^* = x + p$, 其中 x 为已知的预估解。接着继续计算拉格朗日乘子, 只需将 (3.9) 的第一个等式右乘矩阵 Y , 则

$$\lambda^* = ((AY)^T)^{-1} Y^T (Qp + g). \quad (3.16)$$

以上即为零空间法的原理, 该方法相比于高斯消元法, 可以有效降低矩阵求逆的维度, 提升计算速度, 并且更加稳定。

3.2.3 基于零空间求解 KKT 系统算法步骤

Step 0 根据凸二次规划问题, 确定 (3.9) 式中的矩阵 Q 、 A 和向量 g 、 h 。

Step 1 确定可行点向量 x_0 , 并根据 (3.12) 式的要求, 确定矩阵 A 的零空间矩阵 Z , 以及矩阵 Y 。

Step 2 根据 (3.13)、(3.14) 和 (3.15) 式求出 p_Y 和 p_Z , 进而由 (3.12) 式求出向量 p 。

Step 3 计算全局最优解 $x^* = x_0 + p$ 和对应的乘子 $\lambda^* = ((AY)^T)^{-1} Y^T (Qp + g)$ 。

如果 KKT 系统的规模比较大, 直接求解方法在效率上则不如迭代求解, 此时可采用 CG 迭代方法^[42, 43]、GMRES 迭代方法、QMR 迭代方法、LSQR 迭代方法等^[1]。

3.3 交替方向乘子法

交替方向乘子法（以下简称 ADMM）最早分别由 Glowinski、Marrocco 于 1975 年^[51]及 Gabay、Mercier 于 1976 年^[52]提出，并被 Boyd 等人于 2011 年^[53]重新发现。相比于上述 3.1 和 3.2 部分的求解算法，ADMM 方法由于可以利用计算机进行并行计算，提高了计算效率，因此更适用于大规模分布式优化问题。

该方法是由增广拉格朗日法延伸出的，而且可以处理部分非凸二次规划问题。其主要思想是将原问题中的目标函数进行分解，转化为等价的若干个可求解的子问题，接着并行求解每一个子问题，最后协调子问题的解得到原问题的全局解。

运用凸二次规划问题的目标函数具有可分性的特点，即（3.6）的二次规划模型可以转化为下列形式：

$$\begin{aligned} \min & f_1(y) + f_2(z), \\ \text{s.t.} & A_1 y + A_2 z = b, \end{aligned} \quad (3.17)$$

其中 $f_1(y)$ 和 $f_2(z)$ 是由（3.6）的目标函数 $f(x)$ 对偶分解得到的凸函数， A_1 和 A_2 是由（3.6）中的 A 分块产生的矩阵。记问题（3.17）的最优值

$$p^* = \inf \{ f_1(y) + f_2(z) \mid A_1 y + A_2 z = b \}, \quad (3.18)$$

于是可以求出问题（3.17）的增广拉格朗日函数：

$$L_\sigma(y, z, \lambda) = f_1(y) + f_2(z) + \lambda(A_1 y + A_2 z - b) + \frac{\sigma}{2} \|A_1 y + A_2 z - b\|_2^2, \quad (3.19)$$

其中 σ 为罚参数。进而可通过固定 $L_\sigma(y, z, \lambda)$ 中的部分变量，而去求解剩余的变量，这个过程即为 ADMM 方法的迭代步骤：

$$\begin{aligned} y^{k+1} &= \arg \min_y L_\sigma(y, z^k, \lambda^k), \\ z^{k+1} &= \arg \min_z L_\sigma(y^{k+1}, z, \lambda^k), \\ \lambda^{k+1} &= \lambda^k + \sigma(A_1 y^{k+1} + A_2 z^{k+1} - b). \end{aligned} \quad (3.20)$$

在 2011 年 Boyd^[53]还证明了这种算法是收敛的，具体过程在此不作赘述。

第 4 章 非等式约束凸二次规划求解方法

本章的研究对象为非等式约束凸二次规划问题，包括不等式约束问题和一般约束问题，将主要介绍有效集法、内点法、梯度投影法的求解原理和算法步骤，并对以上算法的优缺点进行说明分析。

为了便于本章后续的介绍，先结合第 2 章的预备知识，对凸二次规划问题的 KKT 条件做出必要说明。给定如下的标准方程：

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T Q x + c^T x, \\ \text{s.t. } a_i^T x - b_i &= 0, \quad i \in \mathcal{E} = \{1, 2, \dots, l\}, \\ a_i^T x - b_i &\geq 0, \quad i \in \mathcal{F} = \{l+1, l+2, \dots, m\}, \end{aligned} \quad (4.1)$$

可以得到对应的 Lagrange 方程为

$$\mathcal{L}(x, \lambda) = \frac{1}{2} x^T Q x + c^T x - \sum_{i \in \mathcal{E} \cup \mathcal{F}} \lambda_i (a_i^T x - b_i), \quad (4.2)$$

于是

$$\nabla_x \mathcal{L}(x, \lambda) = Qx + c - \sum_{i \in \mathcal{E} \cup \mathcal{F}} \lambda_i a_i.$$

设 \hat{x} 是问题 (4.1) 的任意可行解，定义 \hat{x} 的有效指标集 $\mathcal{A}(\hat{x})$ 为

$$\mathcal{A}(\hat{x}) = \{i \in \mathcal{F} \mid a_i^T \hat{x} = b_i\}, \quad (4.3)$$

则对于任意局部最优解 \bar{x} ，存在对应的乘子 $\bar{\lambda}$ 满足如下 KKT 条件：

$$\begin{cases} \nabla_x \mathcal{L}(\bar{x}, \bar{\lambda}) = 0, \\ a_i^T \bar{x} = b_i, \quad \forall i \in \mathcal{E}, \\ a_i^T \bar{x} \geq b_i, \quad \forall i \in \mathcal{F}, \\ \bar{\lambda}_i \geq 0, \quad \forall i \in \mathcal{A}(\bar{x}), \\ \bar{\lambda}_i = 0, \quad \forall i \in \mathcal{F} \setminus \mathcal{A}(\bar{x}). \end{cases} \quad (4.4)$$

由于问题 (4.1) 是一个凸问题，因此满足上述 (4.4) 中 KKT 条件的局部最优解 \bar{x} 亦为全局最优解。

4.1 有效集法

4.1.1 有效集法的原理

鉴于等式约束凸二次规划问题早已有了比较成熟的求解方法，因此可以引入先验知识，通过控制约束条件，将一个不等式约束凸二次规划问题转化为等式约束二次规划问题来处理。基于这种思路，有效集方法被提出，并在近些年里得到了广泛的应用^[24, 25, 54]，其对中小型问题有较好的效果，并能有效地检测解的不可行性。考虑如下问题：

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T Q x + c^T x, \\ \text{s.t. } a_i^T x - b_i &= 0, \quad i \in \mathcal{E} = \{1, 2, \dots, l\}, \\ a_i^T x - b_i &\geq 0, \quad i \in \mathcal{F} = \{l+1, l+2, \dots, m\}, \end{aligned} \quad (4.5)$$

其中 Q 为半正定矩阵，故问题 (4.5) 的 KKT 点即为其全局最优解。通过引入最优解 x^* 的有效集 $\mathcal{A}(x^*) = \{i \in \mathcal{F} \mid a_i^T x^* = b_i\}$ ，可以将 (4.5) 中的不等式约束二次规划问题简化为如下等式约束凸二次规划子问题：

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T Q x + c^T x, \\ \text{s.t. } a_i^T x - b_i &= 0, \quad i \in \mathcal{E} \cup \mathcal{A}(x^*). \end{aligned} \quad (4.6)$$

下面探讨问题 (4.5) 最优解与子问题 (4.6) 最优解的关系。由 (4.4) 中给定的问题 (4.5) 的 KKT 条件，可知下式成立。

$$\begin{cases} Qx^* + c - \sum_{i \in \mathcal{E} \cup \mathcal{F}} \lambda_i^* a_i = 0, \\ \sum_{i \in \mathcal{F} \setminus \mathcal{A}(x^*)} \lambda_i^* a_i = 0, \\ a_i^T x^* = b_i, \quad \forall i \in \mathcal{E}, \\ a_i^T x^* = b_i, \quad \forall i \in \mathcal{A}(x^*), \end{cases}$$

进而可以推知

$$\begin{cases} Qx^* + c - \sum_{i \in \mathcal{E} \cup \mathcal{A}(x^*)} \lambda_i^* a_i = 0, \\ a_i^T x^* = b_i, \quad \forall i \in \mathcal{E} \cup \mathcal{A}(x^*). \end{cases} \quad (4.7)$$

易知 (4.7) 中的方程组是子问题 (4.6) 的 KKT 条件，所以问题 (4.5) 的 KKT 点必是子问题 (4.6) 的 KKT 点，进而可得定理 4.1。

定理 4.1 若 x^* 是不等式约束凸二次规划问题 (4.5) 的最优解, 则 x^* 必是等式约束凸二次规划子问题 (4.6) 的最优解。

进一步探讨在什么条件下, 等式约束子问题 (4.6) 的最优解会是不等式约束问题 (4.5) 的最优解。

定理 4.2 设 \hat{x} 为凸二次规划子问题 (4.6) 的最优解, $\hat{\lambda}$ 为最优解对应的乘子, 若 \hat{x} 为凸二次规划问题 (4.5) 的可行解, 且 $\hat{\lambda}_i \geq 0, i \in \mathcal{A}(\hat{x})$, 则 \hat{x} 为凸二次规划问题 (4.5) 的最优解。

证明: 由于 \hat{x} 为凸二次规划问题 (4.6) 的最优解, 所以 \hat{x} 满足问题 (4.6) 对应的 KKT 条件:

$$\begin{cases} Q\hat{x} + c - \sum_{i \in \mathcal{E} \cup \mathcal{A}(\hat{x})} \hat{\lambda}_i a_i = 0, \\ a_i^T \hat{x} = b_i, \quad \forall i \in \mathcal{E} \cup \mathcal{A}(\hat{x}). \end{cases} \quad (4.8)$$

又因为 $\hat{\lambda}_i \geq 0, i \in \mathcal{A}(\hat{x})$, 所以可得

$$\begin{cases} Q\hat{x} + c - \sum_{i \in \mathcal{E} \cup \mathcal{A}(\hat{x})} \hat{\lambda}_i a_i - \sum_{i \in \mathcal{F} \setminus \mathcal{A}(\hat{x})} \hat{\lambda}_i a_i = 0, \\ a_i^T \hat{x} - b_i \geq 0, \quad \forall i \in \mathcal{F} \setminus \mathcal{A}(\hat{x}), \\ \hat{\lambda}_i = 0, \quad \forall i \in \mathcal{F} \setminus \mathcal{A}(\hat{x}). \end{cases} \quad (4.9)$$

结合 \hat{x} 为凸二次规划问题 (4.5) 的可行解, 进一步可推知

$$\begin{cases} Q\hat{x} + c - \sum_{i \in \mathcal{E} \cup \mathcal{F}} \hat{\lambda}_i a_i = 0, \\ a_i^T \hat{x} = b_i, \quad \forall i \in \mathcal{E}, \\ a_i^T \hat{x} \geq b_i, \quad \forall i \in \mathcal{F}, \\ \hat{\lambda}_i \geq 0, \quad \forall i \in \mathcal{A}(\hat{x}), \\ \hat{\lambda}_i = 0, \quad \forall i \in \mathcal{F} \setminus \mathcal{A}(\hat{x}), \end{cases} \quad (4.10)$$

故在 \hat{x} 为问题 (4.5) 的可行解, $\hat{\lambda}_i \geq 0, i \in \mathcal{A}(\hat{x})$ 条件下, \hat{x} 满足问题 (4.5) 的 KKT 条件, 即 \hat{x} 为问题 (4.5) 的最优解, 证毕。

根据上述分析, 有效集法求解过程大致可分为两个阶段: 第一阶段, 选择问题 (4.5) 一个可行解 \hat{x} , 确定可行解 \hat{x} 对应的有效集和等式约束凸二次规划子问题 (4.6), 并对子问题求解得到对应的乘子 $\hat{\lambda}$, 判断 $\hat{\lambda}$ 是否符合定理 4.2 中的条件,

若满足则说明该可行解即是最优解；第二阶段，当 $\hat{\lambda}$ 不符合定理 4.2 中的条件时，对可行解 \hat{x} 和有效集按照一定规则更新，然后重复第一阶段。两个阶段不断进行下去，即可以找到原问题 (4.5) 的最优解。

4.1.2 有效集算法步骤

有效集算法的最大难点是一般无法预先知道最优有效集 $\mathcal{A}(x^*)$ ，因此只能通过迭代来构造一系列可行解 $\{x_k\}$ 的有效集 $\{\mathcal{A}(x_k)\}$ ，使得 $\mathcal{A}(x_k) \rightarrow \mathcal{A}(x^*)$ ，最终获得原问题 (4.5) 的最优解，具体算法步骤如下所示：

Step 0 初始化。选定问题 (4.5) 的初始可行解 x_0 ，初始迭代次数 $k=0$ ，并设置最大容许误差 $0 < \varepsilon \ll 1$ 和最大迭代次数为 k_{\max} 。

Step 1 确定对应的有效集 $\mathcal{A}(x_k)$ 和等式约束凸二次规划子问题：

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T Q x + c^T x, \\ \text{s.t. } a_i^T x - b_i &= 0, \quad i \in \mathcal{E} \cup \mathcal{A}(x_k). \end{aligned} \quad (4.11)$$

运用 3.2 中的 KKT 方法求得子问题的解 \tilde{x}_k 和对应的 Lagrange 乘子 λ^k 。

Step 2 设置终止条件。如果 $\|x_k - \tilde{x}_k\| \leq \varepsilon$ ，判断乘子 $\lambda_i^k \geq 0, \forall i \in \mathcal{A}(x_k)$ 是否成立，若成立则终止循环，最优解 $x^* = \tilde{x}_k$ ；否则令 $\lambda_p^k = \min\{\lambda_i^k \mid \forall i \in \mathcal{A}(x_k)\} < 0$ ，更新有效集 $\mathcal{A}(x_{k+1}) = \mathcal{A}(x_k) \setminus \{p\}$ ，迭代次数 $k = k + 1$ ，转 Step 1。

Step 3 如果 $\|x_k - \tilde{x}_k\| > \varepsilon$ 且 \tilde{x}_k 为可行解，则更新迭代点 $x_{k+1} = \tilde{x}_k$ ，有效集为 $\mathcal{A}(x_{k+1})$ ，迭代次数 $k = k + 1$ ，转 Step 1；

Step 4 如果 $\|x_k - \tilde{x}_k\| > \varepsilon$ 且 \tilde{x}_k 为不可行解，则令 $x_{k+1} = x_k + \alpha_k d_k$ ，使得 x_{k+1} 为可行解，其中 d_k 和 α_k 满足条件：

$$\begin{aligned} d_k &= \tilde{x}_k - x_k, \\ a_i^T (x_k + \alpha_k d_k) &\geq b_i, \quad \forall i \in \mathcal{F} \setminus \mathcal{A}(x_k). \end{aligned} \quad (4.12)$$

不妨取 $\alpha_k = \min\{\tilde{\alpha}_k, 1\}$ ，其中

$$\tilde{\alpha}_k = \min \left\{ \frac{b_i - a_i^T x_k}{a_i^T d_k} \mid a_i^T d_k < 0, i \in \mathcal{F} \setminus \mathcal{A}(x_k) \right\}, \quad (4.13)$$

更新有效集为 $\mathcal{A}(x_{k+1})$ ，迭代次数 $k = k + 1$ ，转 Step 1。

4.2 内点法

内点法及其改进方法非常适合处理大型凸二次规划问题，而且相比于有效集法，它的适用范围更广，并能处理一些非凸问题。von Neumann 最早提出了线性规划的内点方法，但这种内点法不仅不是多项式时间方法，在实际中的计算效率甚

至比不过单纯形法。苏联数学家 II Dikin^[55]于 1967 年重新发现了该方法，1984 年 Narendra Karmarkar^[12]在美国开发了一种用于求解线性规划的 Karmarkar 算法，该算法与单纯形法相反，是通过遍历可行域的内部来求出最优解，而且它和椭球法一样能在可证明的多项式时间内运行，并在实践中具有远优于椭球法的计算效果，这使得它能解决超出单纯形法能力的大型线性规划问题，也因此该方法获得了许多学者的关注。在上世纪 90 年代内点法被广泛用于求解二次规划问题，Boggs^[14]、Melman^[15]、Vanderbei^[29]、Mészáros^[27]等人先后提出了改进的内点法来求解二次规划问题，并进一步验证了该方法对于求解大型问题具有良好的适应性。

目前常见的内点法包括内点罚函数法和原始-对偶路径跟随法，其中内点罚函数法通过引入罚函数，化不等式约束为等式约束来进行求解；而原始-对偶路径跟随法则是通过引入中心路径的概念，运用迭代方法，不断缩小对偶间隙，最终求出 KKT 点。

相比而言，原始-对偶路径跟随算法在稳定性和适用范围上更具优势，被认为是最成功的内点法，并为大多数此类方法的编程实现提供了基础^[28]，故本节将重点说明原始-对偶路径跟随法的原理和步骤。

4.2.1 内点罚函数法的原理

内点罚函数法一般适用于不等式约束二次规划问题，并且可以求解一些非凸问题。为了便于说明其原理，现将凸二次规划问题的约束条件限制为不等式形式，具体如下所示：

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T Q x + c^T x, \\ \text{s.t. } g_i(x) &= a_i x - b_i \geq 0, \quad i = 1, 2, \dots, m, \end{aligned} \quad (4.14)$$

其中 Q 为对称矩阵，可行域 $\mathcal{D} = \{x \in R^n \mid g_i(x) \geq 0, i = 1, 2, \dots, m\}$ 。为了确保迭代点在可行域内部，记 $\text{int } \mathcal{D} = \{x \in \mathcal{D} \mid B(x, \delta) \cap \mathcal{D} \neq \emptyset, \forall \delta > 0\}$ ，引入对数罚函数：

$$H(x) = -\sum_{i=1}^m \ln(g_i(x)), \quad x \in \text{int } \mathcal{D}, \quad (4.15)$$

进而可以构造出如下增广目标函数：

$$\min_{x \in \text{int } \mathcal{D}} P(x, \sigma) = f(x) + \sigma H(x), \quad (4.16)$$

其中 $\sigma > 0$ 为罚参数, 此时求解不等式优化问题 (4.14) 即转化为求解无约束优化问题 (4.16), 并且增广目标函数 $P(x, \sigma)$ 可微。运用无约束优化问题最优性条件, 可得最优解 x^* 满足

$$\nabla P(x^*, \sigma) = 0, \quad (4.17)$$

求解 (4.17) 式即可获得最优解。

4.2.2 内点罚函数算法步骤

根据 4.2.1 部分的原理说明, 可知内点罚函数法的关键在于引入 log-barrier 函数, 构造了罚函数以及对应的增广目标函数, 其具体步骤如下所示:

Step 0 初始化。在可行域内选定初始点 $x_0 \in \text{int } \mathcal{D}$, 初始迭代次数 $k = 0$, 设置罚参数 $\sigma > 0$ 和系数 $\theta \in (0, 1)$, 容许误差 $0 < \varepsilon \ll 1$, 最大迭代次数 k_{\max} 。

Step 1 求解子问题。将点 $x_k \in \text{int } \mathcal{D}$ 带入到无约束问题 (4.16), 求出极小点 x_{k+1} ;

Step 2 设置终止条件。当新的迭代点满足 $\sigma_k H(x_{k+1}) \leq \varepsilon$, 停止迭代, 输出最优解 $x^* = x_{k+1}$, 迭代次数 $k = k + 1$; 否则, 转 Step 3。

Step 3 减小罚参数 $\sigma_{k+1} = \theta \sigma_k$, 迭代次数 $k = k + 1$, 转 Step 1。

内点罚函数法在罚参数趋向其极限时, 增广目标函数会变得越来越病态, 而造成算法的不稳定。因此在实践中, 常常会运用同样基于罚函数原理的增广拉格朗日法来求解不等式约束的凸二次规划问题, 该方法可以有效避免增广目标函数变得病态的问题。

不过增广拉格朗日法相比于下面的原始-对偶路径跟随法并不占优势, 部分原因是原始-对偶路径跟随法可以运用数值计算软件库中的稀疏矩阵库函数, 而且算法复杂度可以通过同伦函数理论得到证明, 因此它在求解凸二次规划问题时运用更为广泛。

4.2.3 原始-对偶路径跟随法的原理

由于原始-对偶路径跟随法一般是用于求解不等式约束问题, 故同样将凸二次规划问题的约束条件限制为不等式形式, 具体如下所示:

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T Q x + c^T x, \\ \text{s.t. } Ax &\geq b, \end{aligned} \quad (4.18)$$

其中 Q 为对称半正定矩阵。根据 (4.18) 中的二次规划问题, 可得出 KKT 条件为:

$$\begin{aligned}
 Qx - A^T \lambda + c &= 0, \\
 Ax - b &\geq 0, \\
 (Ax - b)_i^T \lambda_i &= 0, \\
 \lambda &\geq 0.
 \end{aligned} \tag{4.19}$$

通过引入松弛变量 $y \geq 0$ ，可以进一步将上述 KKT 条件转化为如下等式形式：

$$\begin{aligned}
 Qx - A^T \lambda + c &= 0, \\
 Ax - b - y &= 0, \\
 y_i^T \lambda_i &= 0, \\
 (y, \lambda) &\geq 0.
 \end{aligned} \tag{4.20}$$

因为 Q 为对称半正定矩阵，所以求解 (4.20) 中的松弛 KKT 条件得到的局部最优解即为问题 (4.18) 的最优解。

下面介绍求解问题 (4.20) 的方法，引入对偶度量参数 $\mu = \frac{y^T \lambda}{m}$ 来刻画对偶间隙，于是可通过不断缩小对偶间隙来间接求出原问题的最优解，即求解问题 (4.18) 转化为求解中心路径：

$$F(x, y, \lambda; \sigma\mu) = \begin{pmatrix} Qx - A^T \lambda + c \\ Ax - b - y \\ Y\Lambda e - \sigma\mu e \end{pmatrix} = 0, \tag{4.21}$$

其中中心化参数 $\sigma \in (0, 1)$ 为缩减对偶度量的比例，矩阵 $Y = \text{diag}(y_1, y_2, \dots, y_m)$ ，矩阵 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$ ，向量 $e = (1, 1, \dots, 1)^T$ ，参数 σ 和 μ 在迭代中确定的数值序列构成了中心路径。当 $\sigma\mu$ 趋于零时，该中心路径趋近于原二次规划问题 (4.18) 的解。

下面介绍通过中心路径迭代求得最优解的迭代过程，通过固定参数 μ ，并对问题 (4.21) 运用 Newton 方法，将 (4.21) 中的变量 $(x, y, \lambda)^T$ 近似处理为

$$(x, y, \lambda)^T = (\tilde{x}, \tilde{y}, \tilde{\lambda})^T + (\Delta x, \Delta y, \Delta \lambda)^T,$$

其中 $(\tilde{x}, \tilde{y}, \tilde{\lambda})$ 相当于给定的初始点。于是可以将求解 (4.21) 转化为求解线性系统：

$$\begin{pmatrix} Q & 0 & -A^T \\ A & -E & 0 \\ 0 & \Lambda & Y \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} A^T \tilde{\lambda} - Q\tilde{x} - c \\ \tilde{y} + b - A\tilde{x} \\ \sigma\mu e - \Lambda Y e \end{pmatrix}, \tag{4.22}$$

其中 E 为单位矩阵。因为 (4.22) 的左侧矩阵是满秩的, 所以可利用矩阵分块消元可求出 $(\Delta x, \Delta y, \Delta \lambda)$ 。为了保证迭代点一直在可行域内, 采用线搜索中的回溯法来确定迭代方程^[56]:

$$(x_{k+1}, y_{k+1}, \lambda_{k+1})^T = (x_k, y_k, \lambda_k)^T + \alpha_k (\Delta x_k, \Delta y_k, \Delta \lambda_k)^T, \quad (4.23)$$

其中 $\alpha_{k+1} = \rho \alpha_k$, $\rho \in (0, 1)$ 为满足条件 $(y_{k+1}, \lambda_{k+1}) > 0$ 的步长, $(\Delta x_k, \Delta y_k, \Delta \lambda_k)$ 为由线性系统 (4.22) 确定的搜索方向, (x_k, y_k, λ_k) 为当前已知的迭代点。中心化参数 σ 和步长 α 的选取对于上述算法的效果有着重要影响, 直接或间接改进这些参数将得到具有不同性质的方法。

该方法可以使迭代点趋向中心路径, 而不那么靠近边界, 这样即使本次迭代对目标函数下降的贡献不大, 也可以在下次迭代时使目标函数取得较大降幅。

4.2.4 原始-对偶路径跟随算法步骤

原始-对偶路径跟随法一般在较大规模的问题上更具优势, 最复杂的情况在于每次迭代需要求解大型稀疏线性系统 (4.22), 而且这个系统还可能是稀疏的, 好在这里可以通过相应的函数库求解, 降低了编程难度。下面给出具体步骤:

- Step 0** 初始化。给定参数 $\sigma_0 \in (0, 1)$, 初始迭代次数 $k = 0$ 和初始点 (x_0, y_0, λ_0) , 其中 $(y_0, \lambda_0) > 0$, 设置最大误差 $0 < \varepsilon \ll 1$ 和最大迭代次数为 k_{\max} 。
- Step 1** 设置终止条件。计算参数 $\mu_k = y_k^T \lambda_k / m$, 当 $\sigma_k \mu_k < \varepsilon$, 停止迭代, 输出最优解 $x^* = x_k$ 和迭代次数等信息; 否则, 转 Step 2。
- Step 2** 求解线性系统 (4.22), 得到 $(\Delta x_k, \Delta y_k, \Delta \lambda_k)$, 选择参数 $\alpha \in (0, 1]$, 更新迭代点 $(x_{k+1}, y_{k+1}, \lambda_{k+1}) = (x_k, y_k, \lambda_k) + \alpha_k (\Delta x_k, \Delta y_k, \Delta \lambda_k)$, 其中 α 为满足条件 $(y_{k+1}, \lambda_{k+1}) > 0$ 的最大步长, 并更新参数 $\sigma_k \in (0, 1)$ 。
- Step 3** 更新迭代次数为 $k = k + 1$, 转 Step 1。

本章 4.2 部分介绍的两种内点法大多用于不等式约束, 对于含有等式约束的凸二次规划问题, 可以通过引入松弛变量, 将等式化解为不等式, 或者将一个等式约束化为两个不等式约束, 然后运用内点法, 但这样也将使得算法需要处理的约束条件增多。

4.3 梯度投影法

4.3.1 梯度投影法的原理

有效集方法虽然可以处理凸二次规划问题，但是它更新有效集的速度非常缓慢，可能需要大量迭代才可以求出一个大型问题的解，而内点法主要是针对不等式约束问题，因此能够克服以上缺点的梯度投影法成为了不错的选择。该方法最早由 Rosen^[16, 17]提出用于求解线性约束优化问题，之后 Goldfarb 和 Lapidus^[57]对该算法做出改进，并将其运用到非线性优化问题的求解，近些年该方法被 Naik^[35, 36]、Matija^[58]、Serena^[33, 34]等学者不断运用和持续改进，成为求解二次规划问题的重要方法。

为了便于解释该方法的基本原理，考虑一般约束的二次规划问题：

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T Q_1 x + c^T x, \\ \text{s.t. } A_1 x &\geq b_1, \\ A_2 x &= b_2. \end{aligned} \quad (4.24)$$

其中 Q_1 为对称矩阵。由于梯度投影法可以求解凸和非凸两类二次规划问题，因此不必限制矩阵 Q_1 为半正定^[1]，接下来说明 Rosen 梯度投影法的具体求解思想。

该算法主要是通过每次迭代中求解可行下降方向，来确保目标函数不断下降，最终获得最优解。其过程可以分为两个阶段：第一阶段，从当前可行域的内点 x 开始沿最速下降方向 $-\nabla f(x)$ 搜索，其中 $\nabla f(x) = Q_1 x + c$ ；第二阶段，每当遇到边界时，取 $-\nabla f(x)$ 在约束边界交集的投影作为搜索方向，以便保持方向可行，也因此在边界上的搜索方向是“曲折”的。该方法的一大关键在于求出可行下降方向，下面将就获得可行下降方向的过程进行介绍。

定理 4.3 设 \hat{x} 在问题 (4.24) 的可行域内，且满足 $A_{11}\hat{x} = b_{11}$ ， $A_{12}\hat{x} > b_{12}$ 这一条件，其中

$$A_1 = \begin{pmatrix} A_{11} \\ A_{12} \end{pmatrix}, \quad b_1 = \begin{pmatrix} b_{11} \\ b_{12} \end{pmatrix}.$$

设矩阵

$$M = \begin{pmatrix} A_{11} \\ A_2 \end{pmatrix}$$

是行满秩矩阵，矩阵 $P = E - M^T(MM^T)^{-1}M$ ，且满足条件 $P\nabla f(\hat{x}) \neq 0$ 。若令

$$d = -P\nabla f(\hat{x}),$$

则 d 是问题 (4.24) 的一个可行下降方向，其中 E 为单位矩阵。

证明：先证 d 为问题 (4.24) 的下降方向。因矩阵 $P = E - M^T(MM^T)^{-1}M$ 满足条件 $P = P^T$ 和 $P^2 = P$ ，故 P 为投影矩阵。根据 $P\nabla f(\hat{x}) \neq 0$ ， $d = -P\nabla f(\hat{x})$ 可得

$$\nabla f(\hat{x})^T d = -\|P\nabla f(\hat{x})\|^2 < 0,$$

所以 d 为下降方向。

下面证明 d 为问题 (4.24) 的可行方向。根据 $P = E - M^T(MM^T)^{-1}M$ 以及 $d = -P\nabla f(\hat{x})$ 可得

$$Md = (-M + M)\nabla f(\hat{x}) = 0,$$

即

$$Md = \begin{pmatrix} A_1 d \\ A_2 d \end{pmatrix} = 0,$$

所以 d 为可行方向。证毕。

由于定理 4.3 中只考虑了当 $P\nabla f(\hat{x}) \neq 0$ 时获得可行下降方向的方向，因此还需要进一步考虑当 $P\nabla f(\hat{x}) = 0$ 时的情况，具体如下列定理所示。

定理 4.4^[50] 设 \hat{x} 在问题 (4.24) 的可行域内，且满足 $A_1 \hat{x} = b_1$ ， $A_2 \hat{x} > b_2$ 这一条件，其中

$$A_1 = \begin{pmatrix} A_{11} \\ A_{12} \end{pmatrix}, \quad b_1 = \begin{pmatrix} b_{11} \\ b_{12} \end{pmatrix}.$$

设矩阵

$$M = \begin{pmatrix} A_{11} \\ A_2 \end{pmatrix}$$

是行满秩矩阵，矩阵 $P = E - M^T(MM^T)^{-1}M$ ，向量

$$\omega = (MM^T)^{-1}Mf(x) = \begin{pmatrix} v \\ \lambda \end{pmatrix},$$

其中向量 v , λ 为问题 (4.24) 等式约束和不等式约束对应的乘子。若 $P\nabla f(\hat{x}) = 0$, 则

(1) 当 $\lambda \geq 0$ 时, 那么 \hat{x} 为问题 (4.24) 的 KKT 点;

(2) 当 $\lambda \not\geq 0$ 时, 那么向量 λ 中存在至少一个分量 $\lambda_j < 0$, 从矩阵 A_{11} 中去除掉 λ_j 对应的行, 获得新矩阵 \tilde{A}_{11} , 并设

$$\tilde{M} = \begin{pmatrix} \tilde{A}_{11} \\ A_2 \end{pmatrix}, \quad \tilde{P} = E - \tilde{M}^T (\tilde{M} \tilde{M}^T)^{-1} \tilde{M}, \quad d = -\tilde{P} \nabla f(\hat{x}),$$

则 d 是问题 (4.24) 的一个可行下降方向, 其中 E 为单位矩阵。

定理 4.4 的证明思路和定理 4.3 类似, 在此不再赘述。结合定理 4.4 和定理 4.3 可以得到可行下降方向, 接下来介绍 Rosen 梯度投影法的算法步骤。

4.3.2 梯度投影算法步骤

Rosen 梯度投影法也是一种可行方向法^[59], 相比于有效集法和内点法, 该方法避免了求解大型线性系统, 而是通过确定负梯度方向或负梯度投影方向来搜索最优解, 其具体步骤如下所示:

Step 0 初始化。选定问题 (4.24) 的初始可行解 x_0 , 初始迭代次数 $k = 0$, 并设置最大容许误差为 $0 < \varepsilon \ll 1$, 最大迭代次数为 k_{\max} 。

Step 1 构造矩阵 A_{11} 和 A_{12} 。确定 x_k 的有效约束集合 $A_{11}x_k = b_{11}$ 和非有效约束集合 $A_{12}\hat{x} > b_{12}$, 其中

$$A_1 = \begin{pmatrix} A_{11} \\ A_{12} \end{pmatrix}, \quad b_1 = \begin{pmatrix} b_{11} \\ b_{12} \end{pmatrix}.$$

Step 2 构造矩阵 M 和 P 。令 $M = (A_{11}, A_2)^T$, 如果 M 为空, 则说明迭代点在可行域内部, 令投影矩阵 $P = E$; 否则说明迭代点在可行域的边界上, 令 $P = E - M^T (MM^T)^{-1} M$, 其中 E 为单位矩阵。

Step 3 计算下降可行方向。计算 $d_k = -P\nabla f(x_k) = -P(Q_1 x_k + c)$, 如果 $\|d_k\| < \varepsilon$, 则转 Step 4; 否则, 转 Step 5。

Step 4 设置终止条件。计算向量

$$\omega = (MM^T)^{-1} M f(x_k) = \begin{pmatrix} \lambda \\ v \end{pmatrix},$$

其中乘子 λ 和 v 分别对应矩阵 A_{11} 和 A_2 。如果 $\lambda \geq 0$, 输出 x_k 为问题 (4.24) 的 KKT 点; 否则选取 λ 中小于 0 的分量序列 λ_{\min} , 从矩阵 A_{11} 中去除掉 λ_{\min} 对应的行, 获得新矩阵 \tilde{A}_{11} , 转 Step 2。

Step 5 运用黄金分割法求解线搜索问题，找出合适的步长 α_k 。具体问题如下：

$$\begin{aligned} \min_{\alpha_k} f(x_k + \alpha_k d_k) &= \frac{(x_k + a_k d_k)^T Q_1 (x_k + a_k d_k)}{2} + c^T (x_k + a_k d_k), \\ \text{s.t. } 0 &\leq \alpha_k \leq \tilde{\alpha}, \end{aligned}$$

其中 $\tilde{\alpha}$ 由下列表达式确定：

$$\tilde{\alpha} = \begin{cases} +\infty, & A_{12} d_k \geq 0, \\ \min \left\{ \frac{(b_{12} - A_{12} x_k)_i}{(A_{12} d_k)_i} \mid (A_{12} d_k)_i < 0 \right\}, & A_{12} d_k \not\geq 0, \end{cases}$$

更新迭代点 $x_{k+1} = x_k + \alpha_k d_k$ 。

Step 6 更新迭代次数，令 $k = k + 1$ ，转 Step 1。

不过为了高效地投影，梯度投影法一般需要约束条件足够简单，而且计算精度容易受到线搜索问题求解结果的影响。

第 5 章 基于 CQP 图象特性的梯度投影法

5.1 改进算法的思路和步骤

根据第 4 章对非等式约束凸二次规划求解算法的介绍,能发现目前绝大多数常用算法基本上只利用了目标函数和约束条件的凸性,通过在可行域内运用不同的迭代方式,来逼近凸二次规划问题的最优解。特别的,当有效集方法在迭代过程中,出现子问题的解在可行域外,将会让其重新投影到可行域内,而后再更新迭代点。像有效集法、内点法、Rosen 梯度投影法等求解算法,虽然可以确保每次迭代点都被限制在可行域内,但也提高了计算量。同时,这些方法只利用了目标函数和约束条件的凸性,而忽略了目标函数最优解的图象特点。

有鉴于此,本文从凸二次规划问题的图象特性(简称为 CQP 图象特性)出发,设计出一种基于 CQP 图象特性的梯度投影算法,用于求解非等式约束的凸二次规划问题,相比于以往的常用方法和当前的主要改进方法,它额外利用了目标函数最优解的图象信息。考虑如下的凸二次规划问题:

$$\min f(x) = \frac{1}{2} x^T Q x + c^T x, \quad (5.1a)$$

$$s.t. \quad A_1 x \geq b_1, \quad (5.1b)$$

$$A_2 x = b_2, \quad (5.1c)$$

其中 Q 为半正定矩阵,并记约束条件形成的可行域为 D 。

该算法的大致思路是:先去除非等式约束凸二次规划问题的约束条件 (5.1b) 和 (5.1c),在目标函数 (5.1a) 可微的前提下,利用一阶最优性条件求出目标函数最优解 \tilde{x}^* ;接着判断 \tilde{x}^* 是否满足约束条件 (5.1b) 和 (5.1c),如果满足,则说明问题 (5.1) 的最优解 $x^* = \tilde{x}^*$;否则利用 CQP 图像特性,可知问题 (5.1) 的最优解不在由约束条件 (5.1b) 和 (5.1c) 构成的多面体内部,而是应该在边界上。下面选取一个凸二次规划问题的三维图象来展示其图象特征,如图 5.1 所示。

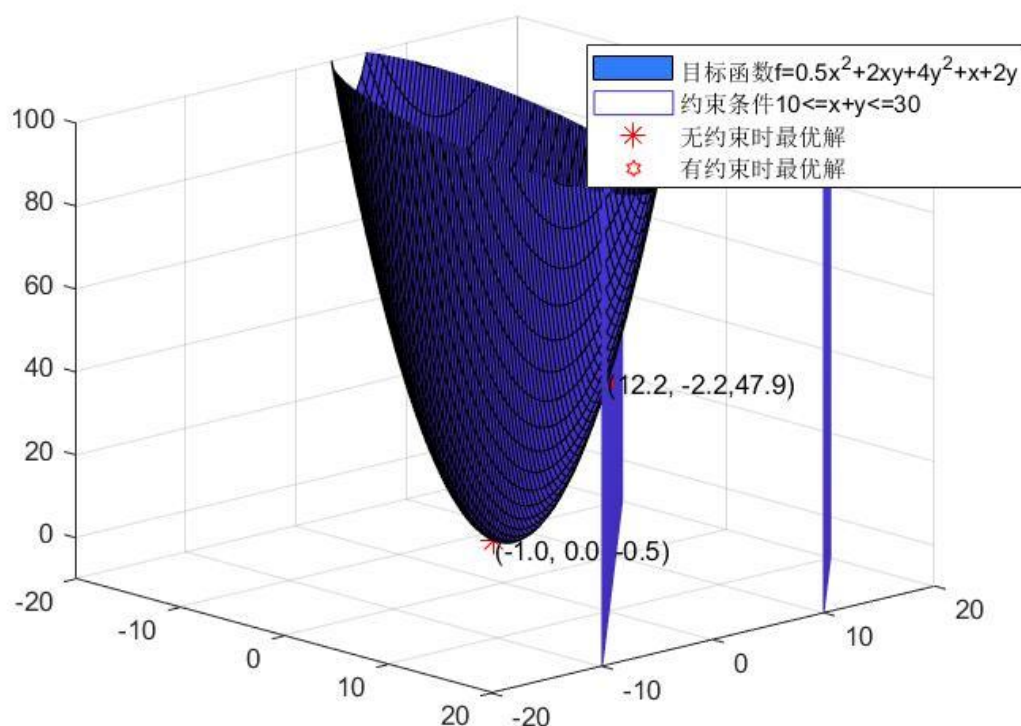


图5.1 目标函数最优解在可行域外时的三维图象

由图 5.1 可以发现, 当去除约束条件后求得的目标函数最优解 \tilde{x}^* , 不在可行域 D 的内部时, 凸二次规划问题 (5.1) 的最优解 x^* 必定在由可行域 D 确定的多面体边界上, 并且是更靠近目标函数最优解 \tilde{x}^* 的边界, 图 5.2 进行了更细致的展示。

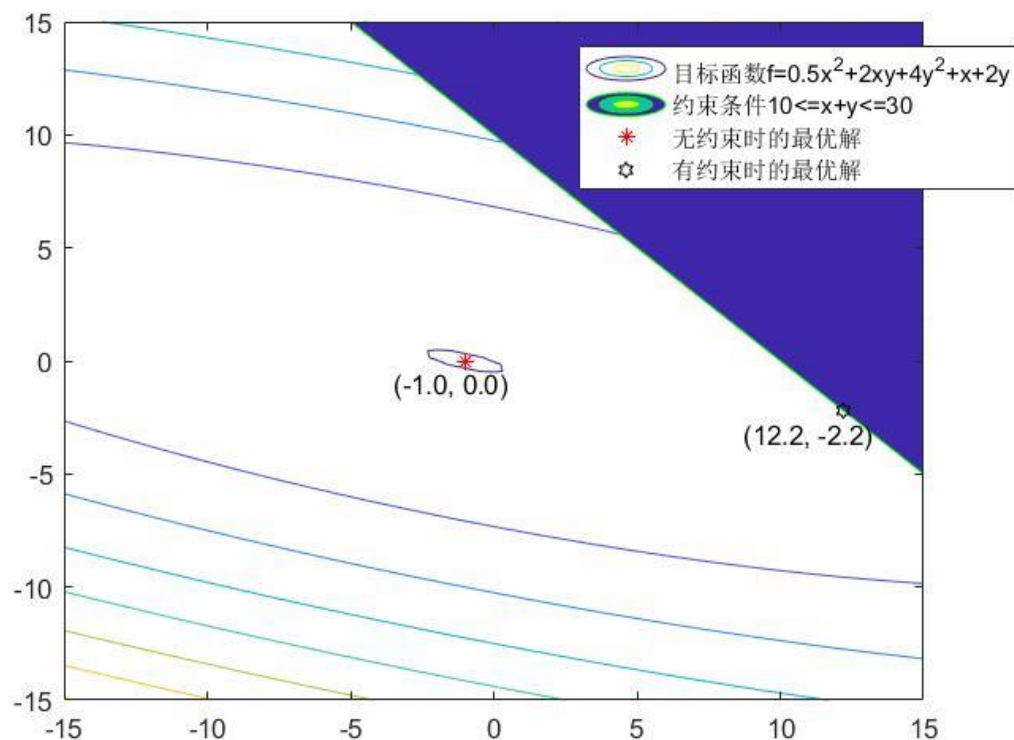


图5.2 目标函数最优解在可行域外时的等高线图象

于是可将 \tilde{x}^* 投影到线性约束构成的边界附近的可行域内, 以此来找到一个比较合适的初始点。本文选择了一种简易方法, 选择某一可行点 x_0^* , 令 $x_0 = \tilde{x}^* + \alpha_0 d_0$, 使得 x_0 为可行解, 其中 d_0 和 α_0 满足条件:

$$\begin{aligned} d_0 &= \tilde{x}^* - x_0^*, \\ A_1(\tilde{x}^* + \alpha_0 d_0) &\geq b_1, \\ A_2(\tilde{x}^* + \alpha_0 d_0) &= b_2, \end{aligned} \quad (5.2)$$

且 α_0 是满足 (5.2) 式的最小值。

经过上述的投影, 可以让点 x_0 更为靠近凸二次规划问题 (5.1) 的最优解。接下来可以运用本文 4.3 节的 Rosen 梯度投影法, 由可行点 $x_k (k=0, 1, 2, \dots)$ 确定有效约束和非有效约束, 进而构造投影矩阵 P 和下降可行方向 $d_k (k=0, 1, 2, \dots)$, 并在边界附近的可行域内搜索最优解, 这样便可降低投影次数和迭代次数。以上即为该算法的主要思路, 下面将进一步说明该算法的大致步骤:

Step 0 求出无约束凸二次规划问题最优解。设置初始迭代次数 $k=0$ 和可行点 x_0^* , 利用一阶无约束优化问题最优性条件, 在去除约束的前提下求出目标函数 (5.1a) 的最优解 \tilde{x}^* 。

Step 1 判断该可行解是否可行。将解 \tilde{x}^* 代入到约束条件 (5.1b) 和 (5.1c) 中, 如果满足条件, 则输出问题 (5.1) 的最优解 $x^* = \tilde{x}^*$, 终止计算; 否则, 利用可行点 x_0^* 将 \tilde{x}^* 投影至边界附近的可行域内, 令 $x_0 = \tilde{x}^* + \alpha_0 d_0$, 其中 d_0 和 α_0 如 (5.2) 所示, 转 Step 2。

Step 2 以 x_0 作为初始点, 运用 Rosen 梯度投影法对凸二次规划问题进行求解。

该算法的最大难点在于确定一个合适的可行点 x_0^* 和参数 α_0 , 使得 \tilde{x}^* 可以被投影在可行域内, 并且投影后的点 x_0 离点 \tilde{x}^* 附近的边界不至于太远。而对于去除约束可能出现目标函数值无下界的情况, 届时仅需选择一个能使目标函数值足够小的点作为去除约束后的最优解 \tilde{x}^* 即可。

5.2 算法的适用性和收敛性说明

由于该改进算法主要利用了凸二次规划问题中目标函数和线性约束条件的凸性, 通过单独求解目标函数 (5.1a), 来判断原问题的最优解是否在线性约束条件 (5.1b) 和 (5.1c) 构成的多面体内部, 而后分情况处理, 因此它主要适用于求解线性约束的凸二次规划问题。

相比于有效集法, 该方法对提前终止具有鲁棒性, 即中间迭代能以有意义的方式逼近最优解, 而且能更快速地更新有效集, 更合适运用于约束条件较为复杂或者规模较大的问题。同时, 该方法相比于内点法, 可以高效求解带有等式约束的凸二次规划问题, 具备较好的适应性。

此外，由于该算法充分利用了凸二次规划问题的图象特性，理论上可以更高效地构造下降可行方向，降低 Rosen 梯度投影方法的迭代次数。特别是当目标函数最优解刚好在约束条件构成的可行域时，该方法可以一步求出最优解。同时，由于该方法主要的改动之处在于选取更高效的初始点，而算法整体框架并没有变动，因此可以和梯度投影法一样做到收敛，不过改进方法能用更少的迭代次数收敛到原问题的最优解。

第 6 章 数值实验

6.1 实验目的及设置

本文在 Windows 11 系统上运用 Matlab R2021a 软件进行了数值实验，实验的主要目的是对比求解非等式约束的凸二次规划的常用算法计算效率，验证改进算法的可行性和有效性。为此，本章将主要选择 Maros 和 Mészáros 凸二次规划测试问题集中的 Time 问题和相关问题为测试对象，通过对比计算时间、迭代次数和误差等信息，综合评价文中涉及的部分非等式约束求解算法和改进算法。

为了实现上述实验目的，本文选取了有效集法、内点法中的原始-对偶路径跟随法、Rosen 梯度投影法这三种方法进行对比实验，具体步骤如下：首先用三种方法求解 Maros 和 Mészáros 凸二次规划测试问题集中的 Time 问题，验证这些算法的可行性；其次，通过在同一问题下随机选择不同的可行初始点，在一定程度上依次验证三种方法的全局收敛性；接下来通过拓展 Time 问题的变量维度和矩阵维度，用三种方法依次求解较高维度的变形问题，观察这三种方法在问题维度增大时的计算性能如何变化。

此外，还将选取其他类型的问题，通过用 Reson 梯度投影法和基于 CQP 图象特性的梯度投影法进行求解，来分析实验结果，测试改进算法的可行性，并初步评价两种算法的性能。

在数值实验中，为保证不被其他因素干扰，将确保同一测试问题中初始可行点等信息保持一致，并统一设置基于变量的最大容许误差 $\varepsilon = 10^{-6}$ ，最大迭代次数 $k_{\max} = 10^3$ 。同时在运用原始-对偶路径跟随算法求解时，将会把凸二次规划问题中的等式约束等价转化为两个不等式约束处理，并且设置里面的中心化参数 $\sigma = 0.9$ ，步长参数的更新方式为 $\alpha_{k+1} = 0.9\alpha_k$ ($k = 0, 1, 2, \dots$)。

6.2 实验结果及分析

例 1：求解 Maros 和 Mészáros 凸二次规划测试问题集中的 Time 问题

$$\begin{aligned} \min f_1(x) &= \frac{1}{2}(x_1, x_2) \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \\ \text{s.t. } x_1 + x_2 &= 1, \\ x_1 &\geq 0, x_2 \geq 0. \end{aligned} \quad (6.1)$$

记目标函数 $f_1(x)$ 中的矩阵为 Q_{f_1} ，通过用三种不同求解方法可得，该凸二次规划问题的最优解 x^* 为 $(0.50, 0.50)^T$ ，在可行域的边界上，其对应的最优值为 0，具体情况如下表 6.1 所示。

表 6.1 各算法求解 TIME 问题的结果

凸二次规划算法	初始点	计算时间/s	迭代次数	函数值误差
有效集法 ^①	$(0, 1)^T$	0.0027774	2	3.0815e-33
	$(1, 0)^T$	0.0035594	2	3.0815e-33
	$(0.2, 0.8)^T$	0.0043158	1	3.0815e-33
	$(0.6, 0.4)^T$	0.0008273	1	3.0815e-33
内点法 ^②	$(0, 1)^T$	0.000773	5	7.8886e-31
	$(1, 0)^T$	0.0005764	5	9.1508e-29
	$(0.2, 0.8)^T$	0.0005757	5	6.4095e-31
	$(0.6, 0.4)^T$	0.0006284	5	2.6032e-29
梯度投影法 ^③	$(0, 1)^T$	0.005005	2	4.8223e-22
	$(1, 0)^T$	0.0035702	2	4.8223e-22
	$(0.2, 0.8)^T$	0.0036797	1	1.0751e-24
	$(0.6, 0.4)^T$	0.0032713	1	5.3747e-24

从表 6.1 中可以初步验证, 有效集法、原始对偶路径跟随法和 Rosen 的梯度投影法在求解凸二次规划问题时都是可行的。通过观察同一方法在不同初值点获得的求解结果, 可从某种程度上验证出三种方法的全局收敛性。

通过比较不同方法在同一初值点的计算时间、迭代次数、误差可以发现, 有效集法在迭代次数和计算误差上较小, 在某种程度上可以认为, 有效集法在处理维度较小的问题时效果更优。而通过比较梯度投影法在不同初始点的迭代次数, 可以初步发现, 选择更优的初始点确实可以减少迭代次数。

此外, 在重复试验时, 初步发现计算时间呈不规则变动, 表 6.1 只取了其中某一次实验的时间, 初步认为算法的计算时间除了受到不同算法的性能影响, 还可能受到计算机不同时期性能波动的影响。

下面将增大问题维度到 4 维, 进一步测试这三种方法, 并观察这些方法的计算性能随着问题维度的增大将会如何改变。

^① 特指本文 4.1 部分的有效集法

^② 特指本文 4.2 部分的原始-对偶路径跟随法

^③ 特指本文 4.3 部分的 Rosen 梯度投影法

例 2: 求解 TIME 问题在拓展到 4 维时的凸二次规划问题

$$\min f_2(x) = \frac{1}{2}(x_1, x_2, x_3, x_4) \begin{pmatrix} 2 & -2 & -2 & 2 \\ -2 & 2 & 2 & -2 \\ -2 & 2 & 2 & -2 \\ 2 & -2 & -2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}, \quad (6.2)$$

$$s.t. \quad x_1 + x_2 + x_3 + x_4 = 1,$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0.$$

记目标函数 $f_2(x)$ 中的矩阵为 Q_{f_2} ，通过用三种不同求解方法可得，该凸二次规划问题的最优解 x^* 为 $(0.25, 0.25, 0.25, 0.25)^T$ ，在可行域的边界上，其对应的最优值为 0，具体情况如下表 6.2 所示。

表 6.2 各算法求解 4 维变形问题的结果

凸二次规划算法	初始点	计算时间/s	迭代次数	函数值误差
有效集法	$(0, 0, 0, 1)^T$	0.011336	7	3.0815e-33
	$(0.2, 0.3, 0.2, 0.3)^T$	0.0054371	1	3.0815e-33
	$(0.1, 0.2, 0.3, 0.4)^T$	0.0006047	1	3.0815e-33
	$(0.3, 0.3, 0.3, 0.1)^T$	0.0006607	1	3.0815e-33
内点法	$(0, 0, 0, 1)^T$	0.020257	5	2.5907e-24
	$(0.2, 0.3, 0.2, 0.3)^T$	0.0005501	5	1.6746e-24
	$(0.1, 0.2, 0.3, 0.4)^T$	0.0008765	5	1.8333e-24
	$(0.3, 0.3, 0.3, 0.1)^T$	0.0007822	5	1.3996e-24
梯度投影法	$(0, 0, 0, 1)^T$	0.011074	3	4.8219e-22
	$(0.2, 0.3, 0.2, 0.3)^T$	0.0027267	1	0
	$(0.1, 0.2, 0.3, 0.4)^T$	0.0024298	1	7.7037e-34
	$(0.3, 0.3, 0.3, 0.1)^T$	0.00231	1	3.4613e-22

结合表 6.1 和表 6.2 可以进一步验证三种算法在求解线性约束凸二次规划问题的可行性和全局收敛性。同时，由表 6.2 初步观察可得，当问题维度增大时，内点法的求解迭代次数较为稳定，而有效集法和梯度投影法出现不少迭代一次就获得最优解的情况，经排查发现这主要是由于算法计算效率以及初始点比较靠近最优解造成的，也侧面验证了梯度投影法求解凸二次规划问题的迭代次数，会受到初始点的影响。

此外，针对例 6.2 在同一初始点，不同时间运用同一方法求解，获得的计算时间仍会波动变化，有鉴于此，接下来将主要通过迭代次数来评判算法的计算效率，而计算时间将仅供参考。

下面继续增大维度到 8 维，再次运用这三种方法求解，由于这些方法的全局收敛性已经得到了一定程度的验证，所以接下来的表格不再列出多组初始点，仅随机选取一组初始点代入三种方法中。

例 3: 求解 TIME 问题在拓展到 8 维时的凸二次规划问题

$$\min f_3(x) = \frac{1}{2}(x_1, x_2, \dots, x_8) \begin{pmatrix} Q_{f_1} & -Q_{f_1} & Q_{f_1} & -Q_{f_1} \\ -Q_{f_1} & Q_{f_1} & -Q_{f_1} & Q_{f_1} \\ Q_{f_1} & -Q_{f_1} & Q_{f_1} & -Q_{f_1} \\ -Q_{f_1} & Q_{f_1} & -Q_{f_1} & Q_{f_1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{pmatrix}, \quad (6.3)$$

s.t. $x_1 + x_2 + \dots + x_8 = 1,$
 $x_1 \geq 0, x_2 \geq 0, \dots, x_8 \geq 0.$

记目标函数 $f_3(x)$ 中的矩阵为 Q_{f_3} , 通过用三种不同求解方法可得, 该凸二次规划问题的最优解 x^* 为 $(\frac{1}{8}, \frac{1}{8}, \dots, \frac{1}{8})^T$, 在可行域的边界上, 其对应的最优值为 0, 不妨选择初始可行点 $(0.50, 0, 0, 0, 0, 0, 0, 0.50)^T$, 具体情况如下表 6.3 所示。

表 6.3 各算法求解 8 维变形问题的结果

凸二次规划算法	计算时间/s	迭代次数	函数值误差
有效集法	0.01138	18	0
内点法	0.0065335	5	8.4139e-22
梯度投影法	0.0029081	1	0

根据表 6.1 至表 6.3 中的数据对比, 可知当凸二次规划问题维度增大时, 内点法和梯度投影法的迭代次数并没有明显变化, 而有效集法的迭代次数却有所上升, 一定程度上显示内点法和梯度投影法在处理高维度问题时更加稳定, 但有效集法的精度却要更优。不过最终也发现梯度投影算法求解出的最优值却依旧为初始可行点, 检验后发现该初始可行点对应的最优值确实为 0, 这可能是由于矩阵 Q_{f_3} 仅是半正定矩阵造成的。不过仍有必要继续探究原因, 并据此对高维情况的梯度投影程序和测试问题做进一步修改和补充。

为了进一步探讨基于 CQP 图象特性的梯度投影法的可行性, 以及该方法相比于 Rosen 梯度投影法在降低迭代次数的优势, 可用两种方法分别求解其他类型的凸二次规划问题, 着重对比求解所需的迭代次数。因此实验结果可以进行适当简化, 具体如下所示。

例 4: 求解如下的凸二次规划问题

$$\min f(x) = \frac{1}{2}(x_1, x_2) \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (6, 0) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad (6.4)$$

s.t. $2x_1 + x_2 \geq 4,$
 $x_1 \geq 0, x_2 \geq 0.$

通过用 Rosen 梯度投影法和改进算法分别求解例 4 可得, 该凸二次规划问题的最优解 x^* 约为 $(1.00, 2.00)^T$, 在可行域的边界上, 其对应的最优值为 11.0000, 不妨选

择初始可行点 $(2.00, 0)^T$ 。当去除约束时，目标函数最优解 \tilde{x}^* 为 $(-3.00, 0)^T$ ，不在可行域内，因此可选取可行点 x_0^* 来将点 \tilde{x}^* 投影到最近的约束条件 $2x_1 + x_2 \geq 4$ 形成的边界上，而后运用梯度投影法，具体情况如下表 6.4 所示。

表 6.4 各算法求解问题例 4 的结果

凸二次规划算法	迭代次数	函数值误差
梯度投影法	2	0
改进方法 ^④	1	0

根据表 6.4 中的数据结果，可以初步认为改进算法是可行的，而且相比于梯度投影法，能降低迭代次数。不过在实验的过程中也发现可行点 x_0^* 会在某种程度上对迭代次数产生影响，但不管取哪一个点 x_0^* ，改进方法求解例 4 所需的迭代次数都小于或等于 Rosen 梯度投影法所需的迭代次数，初步达到了改进算法的设计目标，后续还需选取更多不同类型的凸二次规划问题进行测试。

^④ 特指本文 5.1 部分基于 CQP 图象特性的梯度投影法

第7章 总结与展望

凸二次规划问题自上世纪五十年代开始,就一直得到了各领域学者的关注,他们相继提出了多种求解方法,其中的一些方法已经融入到相关数值求解软件的工具箱中,并发挥着巨大的作用。本文基于前人的理论,主要做了如下工作:

(1) 对凸二次规划的相关基础理论和模型进行了简要说明;

(2) 将线性约束的凸二次规划问题分为了等式约束和非等式约束两类,依次介绍了求解等式约束的变量消去法、KKT 法、交替方向乘子法,以及求解非等式约束的有效集法、内点法、梯度投影法和其他方法;

(3) 提出了基于 CQP 图象特性的梯度投影法,并说明了相关理论和算法步骤;

(4) 通过数值实验,验证了有效集法、原始-对偶路径跟随法和 Rosen 梯度投影法在求解凸二次规划问题的可行性、对不同维度问题的适用性和在可行域内的收敛性,以及改进算法的可行性和有效性。

本文虽然完成了以上几项工作,但是在下列几各方面仍有较大提升空间:

(1) 从数值实验的完整性来看,本文限于篇幅和计算机性能,没有将 Maros 和 Mészáros 凸二次规划测试问题集中的所有问题都进行数值实验,而且选取的问题维度不是很大;

(2) 从改进方法的适用性来看,本文利用了凸二次规划问题的凸性和目标函数最优解的图象信息改进了原方法,在理论上改进后的算法优势主要在于求解约束条件较为复杂的大规模线性约束凸二次规划问题,而对于约束条件较为单一的小规模问题,计算效率并没有太大的提升。

回顾整个凸二次规划求解算法的发展过程,可以了解到这是一个不断继承发展、推陈出新的过程。展望未来,二次规划的模型及求解算法将在更广阔的领域得到运用,其模型将会更多的涉及到非凸领域,而求解算法必将朝着大规模、高效率 and 强适应性的方向发展。同时,二次规划求解方法的创新不仅仅需要代数方面的知识,也需要几何等其他领域的支持。

参考文献

- [1] Wright S, Nocedal J. Numerical Optimization[M]. New York: Springer, 2006: 448-491.
- [2] Boyd S, Vandenberghe L. Convex Optimization[M]. Cambridge: Cambridge University Press, 2004: 21-272.
- [3] Donald G, Iyengar G. Robust Portfolio Selection Problems[J]. Mathematics of Operations Research, 2003, 1(28): 1-38.
- [4] Boyd S, Mueller M T, O'Donoghue B. Performance Bounds and Suboptimal Policies for Multi-Period Investment[J]. Foundations and Trends® In Optimization, 2014, 1(1): 1-72.
- [5] Shahzad A, Eric C K, George A. A Stable and Efficient Method for Solving a Convex Quadratic Program with Application to Optimal Control[J]. SIAM Journal on Optimization, 2012, 4(22): 1369-1393.
- [6] Raghunathan A U, Stefano Di C. Alternating Direction Method of Multipliers for Strictly Convex Quadratic Programs: Optimal Parameter Selection: 2014 American Control Conference. IEEE, 2014[C].
- [7] Luo Z, Yu W. An Introduction to Convex Optimization for Communications and Signal Processing[J]. IEEE Journal on Selected Areas in Communications, 2006, 8(24): 1426-1438.
- [8] Ban G, Nouredine El K, Andrew EB L. Machine Learning and Portfolio Optimization[J]. Management Science, 2018, 3(64): 1136-1154.
- [9] Frank M, Wolfe P. An Algorithm for Quadratic Programming[J]. Naval Research Logistics Quarterly, 1956, 1-2(3): 95-110.
- [10] Moré J, Toraldo G. Algorithms for Bound Constrained Quadratic Programming Problems[J]. Numerische Mathematik, 1989, 55(4): 377-400.
- [11] Boland N L. A Dual-Active-Set Algorithm for Positive Semi-Definite Quadratic Programming[J]. Mathematical Programming, 1996, 1(78): 1-27.
- [12] Karmarkar N. A New Polynomial-Time Algorithm for Linear Programming: Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing, 1984[C].
- [13] Monteiro R D, Adler I. Interior Path Following Primal-Dual Algorithms. Part II Convex Quadratic Programming[J]. Mathematical Programming, 1989, 1(44): 43-66.
- [14] Boggs P T, Domich P D, Rogers J E. An Interior Point Method for General Large-Scale Quadratic Programming Problems[J]. Annals of Operations Research, 1996, 62(1): 419-437.
- [15] Melman A, Polyak R. The Newton Modified Barrier Method for QP Problems[J]. Annals of Operations Research, 1996, 62(1): 465-519.
- [16] Rosen J. The Gradient Projection Method for Nonlinear Programming. Part I. Linear Constraints[J]. Journal of the Society for Industrial and Applied Mathematics, 1960, 1(8): 181-217.
- [17] Rosen J. The Gradient Projection Method for Nonlinear Programming. Part II. Nonlinear Constraints[J]. Journal of the Society for Industrial and Applied Mathematics, 1961, 4(9): 514-532.
- [18] Boggs P T, Tolle J W. Sequential Quadratic Programming[J]. Acta Numerica, 1995(4): 1-51.
- [19] Fletcher R, Leyffer S. Numerical Experience with Lower Bounds for MIQP Branch-And-Bound[J]. SIAM Journal on Optimization, 1998, 2(8): 604-616.
- [20] Philip W. The Simplex Method for Quadratic Programming[J]. Econometrica: Journal of the Econometric Society, 1959: 382-398.
- [21] De Marchi A. On A Primal-Dual Newton Proximal Method for Convex Quadratic Programs[J]. Computational Optimization and Applications, 2022, 81(2): 369-395.
- [22] Liao-Mcpherson D, Kolmanovsky I. Fbstab: A Proximally Stabilized Semismooth Algorithm for Convex Quadratic Programming[J]. Automatica, 2020, 113: 108801.
- [23] Bartlett R A, Biegler L T. Qpschur: A Dual, Active-Set, Schur-Complement Method for Large-Scale and Structured Convex Quadratic Programming[J]. Optimization and Engineering, 2006, 7(1): 5-32.
- [24] Forsgren A, Gill P E, Wong E. Primal and Dual Active-Set Methods for Convex Quadratic Programming[J]. Mathematical Programming, 2016, 159(1-2): 469-508.
- [25] Fält M, Giselsson P. QPDAS: Dual Active Set Solver for Mixed Constraint Quadratic Programming[A]. In: 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 2019. IEEE, 2019: 4891-4897.
- [26] Gondzio J. Interior Point Methods 25 Years Later[J]. European Journal of Operational

- Research,2012,3(218): 587-601.
- [27] MÉSZÁROS C. The BPMPD Interior Point Solver for Convex Quadratic Problems[J]. Optimization Methods & Software,1999,11(1-4): 431-449.
- [28] Potra F, Wright S. Interior-Point Methods[J]. Journal of Computational and Applied Mathematics,2000,1-2(124): 281-302.
- [29] Vanderbei R J. LOQO:An Interior Point Code for Quadratic Programming[J]. Optimization Methods & Software,1999,11(1-4): 451-484.
- [30] Absil P A, Tits A L. Newton-KKT Interior-Point Methods for Indefinite Quadratic Programming[J]. Computational Optimization and Applications,2007,36(1): 5-41.
- [31] Friedlander M P, Orban D. A Primal-Dual Regularized Interior-Point Method for Convex Quadratic Programs[J]. Mathematical Programming Computation,2012,4(1): 71-107.
- [32] Pougkakiotis S, Gondzio J. Dynamic Non-Diagonal Regularization in Interior Point Methods for Linear and Convex Quadratic Programming[J]. Journal of Optimization Theory and Applications,2019,181(3): 905-945.
- [33] Crisci S, Ruggiero V, Zanni L. Steplength Selection in Gradient Projection Methods for Box-Constrained Quadratic Programs[J]. Applied Mathematics and Computation,2019,356: 312-327.
- [34] Crisci S, Kružík J, Pecha M, Et Al. Comparison of Active-Set and Gradient Projection-Based Algorithms for Box-Constrained Quadratic Programming[J]. Soft Computing,2020,24(23): 17761-17770.
- [35] Naik V V, Bemporad A. Embedded Mixed-Integer Quadratic Optimization Using Accelerated Dual Gradient Projection[J]. IFAC-Papersonline,2017,50(1): 10723-10728.
- [36] Naik V V, Bemporad A. Exact and Heuristic Methods with Warm-Start for Embedded Mixed-Integer Quadratic Programming Based on Accelerated Dual Gradient Projection[J]. Arxiv Preprint Arxiv:2101.09264,2021.
- [37] Graves R L. A Principal Pivoting Simplex Algorithm for Linear and Quadratic Programming[J]. Operations Research,1967,3(15): 482-494.
- [38] Bland R G, Goldfarb D, Todd M J. The Ellipsoid Method a Survey[J]. Operations Research,1981,6(29): 1039-1091.
- [39] Hestenes M R. Multiplier and Gradient Methods[J]. Journal of Optimization Theory and Applications,1969,5(4): 303-320.
- [40] Powell M J D. A Method for Nonlinear Constraints in Minimization Problems[J]. Optimization,1969: 283-298.
- [41] Rockafellar R T. The Multiplier Method of Hestenes and Powell Applied to Convex Programming[J]. Journal of Optimization Theory and Applications,1973,6(12): 555-562.
- [42] O'Leary D P. A Generalized Conjugate Gradient Algorithm for Solving a Class of Quadratic Programming Problems[J]. Linear Algebra and its Applications,1980(34): 371-399.
- [43] Durazzi C, Ruggiero V. Indefinitely Preconditioned Conjugate Gradient Method for Large Sparse Equality and Inequality Constrained Quadratic Problems[J]. Numerical Linear Algebra with Applications,2003,10(8): 673-688.
- [44] Bueno L F, Haeser G, Santos L. Towards an Efficient Augmented Lagrangian Method for Convex Quadratic Programming[J]. Computational Optimization and Applications,2020,76(3): 767-800.
- [45] Necoara I, Kvamme S. Duquad: A Toolbox for Solving Convex Quadratic Programs Using Dual (Augmented) First Order Algorithms[A]. In: 2015. IEEE, 2015: 2043-2048.
- [46] Necoara I, Nesterov Y, Glineur F. Linear Convergence of First Order Methods for Non-Strongly Convex Optimization[J]. Mathematical Programming,2019,175(1-2): 69-107.
- [47] Karush W. Minima of Functions of Several Variables with Inequalities as Side Conditions[D]. Chicago, Illinois: Univ. of Chicagodept. of Mathematics, 1939.
- [48] Kuhn H W, Tucker A W. Nonlinear Programming[A]. In: Berkeley Symposium on Mathematical Statistics and Probability, 1951: 481-492.
- [49] Slater M. Lagrange Multipliers Revisited[A]. In: Giorgi G, Kjeldsen T H. Traces and Emergence of Nonlinear Programming[M]. Basel: Springer Basel, 2014: 293-306.
- [50] 马昌凤. 最优化方法及其 Matlab 程序设计[M]. 北京: 科学出版社, 2010: 1-173.
- [51] Glowinski R, Marroco A. Sur L'approximation, Par ÉLÉMENTS Finis D'ordre Un, Et La RÉSolution, Par PÉNalisation-DualITÉ D'une Classe De ProblÈMes De Dirichlet Non LinÉaires [J]. ESAIM: Mathematical Modelling and Numerical Analysis-ModÉlisation MathÉmatique Et Analyse NumÉrique,1975,R2(9): 41-76.
- [52] Gabay D, Mercier B. A Dual Algorithm for the Solution of Nonlinear Variational Problems via

- Finite Element Approximation[J]. Computers & Mathematics with Applications,1976,1(2): 17-40.
- [53] Boyd S, Parikh N, Chu E, Et Al. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers[J]. Foundations and Trends® In Machine Learning,2011,1(3): 1-122.
- [54] Ferreau H J, Kirches C, Potschka A, Et Al. Qpoases: A Parametric Active-Set Algorithm for Quadratic Programming[J]. Mathematical Programming Computation,2014,6(4): 327-363.
- [55] Dikin I I. Iterative Solution of the Problems of Linear and Quadratic Programming[J]. Soviet Mathematics Doklady,1967(8): 674-675.
- [56] 刘浩洋, 卢将, 李勇锋, 等. 最优化: 建模、算法和理论[M]. 北京: 高等教育出版社, 2020: 297-338.
- [57] Goldfarb D, Lapidus L. Conjugate Gradient Method for Nonlinear Programming Problems with Linear Constraints[J]. Industrial & Engineering Chemistry Fundamentals,1968,1(7): 142-151.
- [58] Perne M, Gerkešič S, Pregelj B. Soft Inequality Constraints in Gradient Method and Fast Gradient Method for Quadratic Programming[J]. Optimization and Engineering,2019,20(3): 749-767.
- [59] 陈宝林. 最优化理论与算法[M]. 北京: 清华大学出版社, 2005: 360-391.

致 谢

经过在校期间的学习研究，本论文得以顺利完成。回顾过去在武大短短的三年，大概是我迄今为止经历最为丰富的一段时光。我参加了一些感兴趣的社团活动和实习实践工作，并在周围同学和老师的帮助下，走出了内心的困境，找到了新的方向。在此向自强不息的自己，以及一路相助的所有人表示感谢。

过去几年的校园生活，虽因疫情等因素限制重重，不过我也因此有了更多与同学和老师接触的机会。在武大的校园里，我遇见了一批认真负责的学长学姐和志同道合的朋友，尤其是19级基础数学班的几位同学，他们给我的学业和工作提供了宝贵的支持，并使我的生活变得丰富多彩。

除了身边的同学，我也要多谢洪桂祥老师对我的包容和帮助，他一丝不苟的科研态度令人钦佩。同时，我要特别感谢黄安云老师、宋智祺辅导员、杨乔华班主任、张继伟老师和黄正华老师，他们在我最困难的时候，给予了我关心，并帮助我解决了一些实际问题。我还想致敬一位特别的老师——互联网，它使我在疫情期间依旧能学习到各种知识与技能，并拓展了我的社交和生活空间。

此刻的我，最想感谢的依旧是自己的家人。我从未像最近这两年一样感受到他们对自己的重要性，平凡的他们一直在尽力为我提供良好的家庭环境和物质条件，让我在多年学习后，能获得撰写这篇毕业论文的机会。今后的生活也许依旧会充满风雨，但只要想到他们，我便有了动力。

最后，由衷地感谢在百忙之中抽空评审论文，以及参加答辩的各位老师，你们辛苦了！

陈文标

2022年5月于数学与统计学院