

# 线性矩阵不等式半定互补问题的数值求解方法

孔汕汕, 赵馨\*

东华大学理学院, 上海

收稿日期: 2021年12月18日; 录用日期: 2022年1月20日; 发布日期: 2022年1月27日

## 摘要

线性矩阵不等式半定互补问题是一类新颖的多项式优化问题, 是半定规划与互补问题的交叉研究内容。该问题可以转化为一系列带有线性矩阵不等式约束的多项式优化子问题, 进而可以采用标量型松弛方法或矩阵型松弛方法进行求解。更进一步, 当线性矩阵不等式半定互补问题的实数解个数有限时, 利用本文给出的算法可以计算出该问题的全部实数解。最后, 我们进行相关数值实验, 分别使用标量型松弛方法和矩阵型松弛方法求解该问题, 并将这两种方法的结果进行对比。

## 关键词

半定互补问题, 线性矩阵不等式, 标量型松弛方法, 矩阵型松弛方法

# Numerical Methods for the Semidefinite Complementarity Problem with Linear Matrix Inequalities

Shanshan Kong, Xin Zhao\*

College of Science, Donghua University, Shanghai

Received: Dec. 18<sup>th</sup>, 2021; accepted: Jan. 20<sup>th</sup>, 2022; published: Jan. 27<sup>th</sup>, 2022

## Abstract

The semidefinite complementarity problem with linear matrix inequalities is a novel polynomial optimization problem, which is the cross-study content of semidefinite programming and complementarity problems. The problem can be transformed into a series of polynomial optimization sub-problems with linear matrix inequality constraints. Then the problem can be solved by the scalar-type relaxation method or the matrix-type relaxation method. Furthermore, when the number of real solutions of the semidefinite complementarity problem with linear matrix inequalities is li-

\*通讯作者。

mitted, all real solutions of the problem can be calculated by using the algorithm given in this paper. Finally, we conduct related numerical experiments. We use the scalar-type relaxation method and the matrix-type relaxation method to solve the problem and compare the results of these two methods.

## Keywords

Semidefinite Complementarity Problem, Linear Matrix Inequalities, Scalar-Type Relaxation Method, Matrix-Type Relaxation Method

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

线性矩阵不等式(linear matrix inequalities, LMI)一般描述为:

$$F(\mathbf{x}) = F_0 + \sum_{i=1}^n x_i F_i \succeq \mathbf{0},$$

其中  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ ,  $F_i = F_i^T \in \mathbb{R}^{m \times m}$ ,  $i = 0, 1, \dots, n$  是对称矩阵, 矩阵  $F(\mathbf{x})$  是一个从  $n$  维实数空间到  $m \times m$  维实对称矩阵空间的映射, 矩阵中的每一个元素都是以  $\mathbf{x} \in \mathbb{R}^n$  为变量的多项式。矩阵  $F(\mathbf{x}) \succeq \mathbf{0}$  表示矩阵  $F(\mathbf{x})$  是正定的, 即对任意的  $\mathbf{u} \in \mathbb{R}^m$ , 有  $\mathbf{u}^T F(\mathbf{x}) \mathbf{u} \geq 0$ 。近年来, 线性矩阵不等式得到了越来越多的关注和重视, 被广泛地应用于解决控制中的问题, 已经成为系统与控制领域研究中的一大热点研究内容[1] [2]。

本文考虑带有线性矩阵不等式约束的半定互补问题: 寻求解  $\mathbf{x} \in \mathbb{R}^n$  满足  $F_1(\mathbf{x}) \succeq \mathbf{0}$ ,  $F_2(\mathbf{x}) \succeq \mathbf{0}$ ,  $\langle F_1(\mathbf{x}), F_2(\mathbf{x}) \rangle = 0$ , 其中  $F_i(\mathbf{x}) \succeq \mathbf{0}, i = 1, 2$  是线性矩阵不等式。线性矩阵不等式半定互补问题是一种新的半定互补问题, 目前很少有关于这个问题的相关文献和理论结果。线性矩阵不等式半定互补问题可以看作标量型多项式互补问题的推广, 这两个问题显著的区别是在线性矩阵不等式半定互补问题中涉及到矩阵型约束, 而不仅仅是标量型约束。这里所说的“矩阵型”是指多项式矩阵不等式约束, “标量型”是指单个多项式不等式约束。

求解带有  $F(\mathbf{x}) \succeq \mathbf{0}$  的半定约束问题的一类直观方法是将其转化为若干个标量的不等式约束。例如, 将线性矩阵不等式约束  $F(\mathbf{x}) \succeq \mathbf{0}$  转化为  $F(\mathbf{x})$  的顺序主子式大于等于 0, 进而将矩阵型约束转化为标量型约束, 继而可以采用传统的标量型松弛方法进行求解, 如采用经典的 Lasserre 松弛方法[3]进行求解。Lasserre 松弛方法是基于半定规划松弛的, 通过构造一系列半定规划松弛问题, 可以无限逼近原多项式优化问题的全局最优值。Lasserre 证明了当阿基米德条件成立时, 该松弛方法具有渐进收敛性。但是, 当  $F(\mathbf{x})$  的维数很大时, 此类方法可能会导致大量的高次约束, 导致松弛问题的规模过大, 从而使得不适合采用标量型松弛方法进行求解。因此, 许多学者致力于直接研究带有多项式矩阵约束的优化问题。例如, Hol 等[4] [5]和 Kojima [6]对多项式矩阵平方和分解进行了深入的研究。为了求解非凸多项式矩阵不等式优化问题, Henrion 等[7]利用 Hol 等[4] [5]和 Kojima [6]关于非负多项式矩阵平方和分解的矩理论的研究成果, 将标量型松弛方法推广到了求解非凸多项式矩阵不等式优化问题的矩阵型松弛方法。Henrion 等[7]指出当阿基米德条件成立时, 该矩阵型松弛方法具有渐进收敛性。

此外, Nie [8]给出了利用半定松弛方法来计算一个多项式的局部极小值的层次结构算法。下面我们在无约束情况下求一个多项式的局部极小值为例对该方法进行说明。具体方法如下: 任意给定一个多项式  $f(\mathbf{x})$ , 利用最优性条件  $\nabla f(\mathbf{x}) = 0$ ,  $\nabla^2 f(\mathbf{x}) \succeq 0$  构造半定松弛问题并定义  $\mathbf{H}$ -最小值点以及  $\mathbf{H}$ -最小值, 然后通过构造并求解一系列半定松弛问题, 可以将全部的  $\mathbf{H}$ -最小值点以及  $\mathbf{H}$ -最小值计算出来, 最后可以从这些值中找出  $f(\mathbf{x})$  的全体局部极小值, 且 Nie 证明了构造的这一系列松弛问题都具有有限收敛性。

利用线性矩阵不等式的性质[9] [10], 我们可以将线性矩阵不等式半定互补问题进行等价改写, 得到与该问题等价的矩阵形式的问题, 进而可以采用矩阵型松弛方法进行求解。进一步, 当线性矩阵不等式半定互补问题的实数解个数有限时, 利用层次结构算法的思想可以计算出该问题的全部实数解。

**符号说明:** 符号  $\mathbb{N}$  ( $\mathbb{R}$ )表示非负整数集(实数集)。符号  $\mathbb{N}^n$  表示  $n$  维非负整数向量集合。符号  $\mathbb{R}^n$  表示  $n$  维实向量集合。对于  $\mathbf{x} \in \mathbb{R}^n$ ,  $x_i$  表示  $\mathbf{x}$  中的第  $i$  个元素, 即  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 。对于  $\boldsymbol{\alpha} \in \mathbb{N}^n$ , 令  $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \dots + \alpha_n$ 。对于  $\mathbf{x} \in \mathbb{R}^n$  及  $\boldsymbol{\alpha} \in \mathbb{N}^n$ , 令  $\mathbf{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$ 。对于整数  $n > 0$ ,  $[n]$  表示集合  $\{1, 2, \dots, n\}$ 。对于任意整数  $t \in \mathbb{R}$ ,  $\lceil t \rceil$  表示不小于  $t$  的最小整数。上标  $\mathbf{T}$  表示矩阵或向量的转置。对于多项式  $f(x)$ ,  $\deg(f(x))$  表示它的次数。符号  $\mathbb{R}^{m \times m}$  表示  $m \times m$  维实对称矩阵空间。符号

$\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, x_2, \dots, x_n]$  表示在实数域上以  $\mathbf{x}$  为自变量的多项式环, 即任意  $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$  是一个以  $\mathbf{x}$  为自变量的多项式。令  $\mathbb{R}[\mathbf{x}]_k$  是指由次数至多为  $k$  的  $\mathbb{R}[\mathbf{x}]$  中的多项式构成的集合, 即

$\mathbb{R}[\mathbf{x}]_k = \{f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}] : \deg(f(\mathbf{x})) \leq k\}$ 。令  $\mathcal{S}[\mathbf{x}]^m$  表示  $m \times m$  维实对称多项式矩阵空间, 即任意

$\mathbf{F}(\mathbf{x}) \in \mathcal{S}[\mathbf{x}]^m$ ,  $\mathbf{F}(\mathbf{x}) : \mathbb{R}^n \mapsto \mathcal{S}[\mathbf{x}]^m$  是一个对称多项式矩阵且每一项  $F_{i,j}(\mathbf{x}) \in \mathbb{R}[\mathbf{x}] (1 \leq i, j \leq m)$  是一个以  $\mathbf{x}$  为自变量的多项式。如果对于任意非零向量  $\mathbf{u} \in \mathbb{R}^m$ , 二次型  $\mathbf{u}^T \mathbf{F}(\mathbf{x}) \mathbf{u}$  是非负的, 则称矩阵

$\mathbf{F}(\mathbf{x}) \in \mathcal{S}[\mathbf{x}]^m$  是正定的, 记作  $\mathbf{F}(\mathbf{x}) \succeq \mathbf{0}$ 。一个多项式矩阵  $\mathbf{F}(\mathbf{x})$  的次数  $\deg(\mathbf{F}(\mathbf{x}))$  是矩阵中全部元素所代表的多项式中的最大次数, 即  $\deg(\mathbf{F}(\mathbf{x})) = \max_{i,j} \deg(F_{i,j}(\mathbf{x}))$ 。令  $\langle \mathbf{A}, \mathbf{B} \rangle = \text{trace}(\mathbf{A}^T \mathbf{B})$  表示两个向量或矩阵的标准内积。

## 2. 线性矩阵不等式半定互补问题的等价改写

本节研究如何等价地改写线性矩阵不等式半定互补问题: 给定矩阵  $\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x}) \in \mathcal{S}[\mathbf{x}]^m$ , 寻求解  $\mathbf{x} \in \mathbb{R}^n$  满足  $\mathbf{F}_1(\mathbf{x}) \succeq \mathbf{0}$ ,  $\mathbf{F}_2(\mathbf{x}) \succeq \mathbf{0}$ ,  $\langle \mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x}) \rangle = 0$ , 其中  $\mathbf{F}_i(\mathbf{x}) \succeq \mathbf{0}, i = 1, 2$  是线性矩阵不等式。记该问题的全体实数解构成的集合为  $\mathcal{S}(\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x}))$ , 则

$$\mathcal{S}(\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x})) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{F}_1(\mathbf{x}) \succeq \mathbf{0}, \mathbf{F}_2(\mathbf{x}) \succeq \mathbf{0}, \text{trace}(\mathbf{F}_1(\mathbf{x}) \mathbf{F}_2(\mathbf{x})) = 0\}. \quad (1)$$

根据线性代数的知识可以得到: 对于两个半正定矩阵  $\mathbf{A}, \mathbf{B}$ ,  $\text{trace}(\mathbf{A}\mathbf{B}) = 0$  当且仅当  $\mathbf{A}\mathbf{B} = \mathbf{0}$ 。因此, 线性矩阵不等式半定互补问题中的互补条件等价于矩阵型等式约束, 即:

$$\mathcal{S}(\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x})) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{F}_1(\mathbf{x}) \succeq \mathbf{0}, \mathbf{F}_2(\mathbf{x}) \succeq \mathbf{0}, \mathbf{F}_1(\mathbf{x}) \mathbf{F}_2(\mathbf{x}) = \mathbf{0}\}. \quad (2)$$

令  $\mathbf{H}(\mathbf{x}) = \mathbf{F}_1(\mathbf{x}) \mathbf{F}_2(\mathbf{x})$  是  $m \times m$  维多项式矩阵, 其中  $F_i(\mathbf{x}) = [f_{st}^{(i)}(\mathbf{x})]_{s,t=1,2,\dots,m}, i = 1, 2$ , 则

$H_{ij}(\mathbf{x}) = \sum_{k=1}^m f_{ik}^{(1)}(\mathbf{x}) f_{kj}^{(2)}(\mathbf{x}), 1 \leq i, j \leq m$ 。令标量多项式  $h_{(i-1)m+j}(\mathbf{x}) = H_{ij}(\mathbf{x}), 1 \leq i, j \leq m$ , 则有多项式组

$\mathbf{h}(\mathbf{x}) = (H_{11}(\mathbf{x}), H_{12}(\mathbf{x}), \dots, H_{1m}(\mathbf{x}), H_{21}(\mathbf{x}), H_{22}(\mathbf{x}), \dots, H_{mm}(\mathbf{x})) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_m(\mathbf{x}))$ 。因此, 可以得到

$$S(F_1(\mathbf{x}), F_2(\mathbf{x})) = \{\mathbf{x} \in \mathbb{R}^n : F_1(\mathbf{x}) \succeq \mathbf{0}, F_2(\mathbf{x}) \succeq \mathbf{0}, h_z(\mathbf{x}) = 0, z = 1, 2, \dots, m^2\}. \quad (3)$$

由公式(3)可以看出, 集合  $S(F_1(\mathbf{x}), F_2(\mathbf{x}))$  的定义中涉及了线性矩阵不等式约束  $F_1(\mathbf{x}) \succeq \mathbf{0}$  和  $F_2(\mathbf{x}) \succeq \mathbf{0}$ 。此前, 对于这类约束的优化问题的求解较为困难。事实上, 一类处理方法是我们可以将线性矩阵不等式约束转化为标量型约束, 从而用一系列标量型不等式约束来描述集合  $S(F_1(\mathbf{x}), F_2(\mathbf{x}))$ 。例如: 当  $F(\mathbf{x})$  的维数大于 2 时,  $F(\mathbf{x}) \succeq \mathbf{0}$  当且仅当它的各阶顺序主子式大于等于零[11], 即: 对于  $i = 1, 2, \dots, m$ ,  $F_i(\mathbf{x}) \succeq \mathbf{0}$  等价于  $\{g_{ij}(\mathbf{x}) \geq 0, i = 1, 2, \dots, m; j = 1, 2, \dots, m\}$ , 其中  $g_{ij}(\mathbf{x})$  为  $F_i(\mathbf{x})$  的第  $j$  阶顺序主子式, 则可以得到  $S(F_1(\mathbf{x}), F_2(\mathbf{x}))$  的等价形式:

$$S(F_1(\mathbf{x}), F_2(\mathbf{x})) = \{\mathbf{x} \in \mathbb{R}^n : h_z(\mathbf{x}) = 0, g_{ij}(\mathbf{x}) \geq 0, z = 1, 2, \dots, m^2; i = 1, 2, \dots, m; j = 1, 2, \dots, m\}. \quad (4)$$

特别地, 当  $F(\mathbf{x})$  的维数等于 2 时,  $F(\mathbf{x}) \succeq \mathbf{0}$  当且仅当  $\text{trace}(F(\mathbf{x})) \geq 0$  且  $\det(F(\mathbf{x})) \geq 0$ , 则

$$S(F_1(\mathbf{x}), F_2(\mathbf{x})) = \{\mathbf{x} \in \mathbb{R}^n : \text{trace}(F_1(\mathbf{x})) \geq 0, \det(F_1(\mathbf{x})) \geq 0, \text{trace}(F_2(\mathbf{x})) \geq 0, \det(F_2(\mathbf{x})) \geq 0, h_z(\mathbf{x}) = 0, z = 1, 2, \dots, m^2\}.$$

由公式(4)中可以看出, 当原问题的规模较大时, 标量化改写方法会引入大量的高次约束, 导致松弛问题的规模过大, 从而变得难以求解。因此我们考虑保留线性矩阵不等式半定互补问题的矩阵型约束进行计算。

由文献[9]可知, 线性矩阵不等式约束序列  $F_1(\mathbf{x}) \succeq \mathbf{0}, F_2(\mathbf{x}) \succeq \mathbf{0}, \dots, F_k(\mathbf{x}) \succeq \mathbf{0}$  等价于一个线性矩阵不等式约束  $\text{diag}(F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_k(\mathbf{x})) \succeq \mathbf{0}$ 。

令  $Q(\mathbf{x}) = \text{diag}(F_1(\mathbf{x}), F_2(\mathbf{x}))$ , 则

$$S(F_1(\mathbf{x}), F_2(\mathbf{x})) = \{\mathbf{x} \in \mathbb{R}^n : Q(\mathbf{x}) \succeq \mathbf{0}, h_z(\mathbf{x}) = 0, z = 1, 2, \dots, m^2\}. \quad (5)$$

由公式(4)和(5)可知, 通过随机地选取一个恰当的多项式  $f(\mathbf{x})$  作为目标函数, 我们可以将原线性矩阵不等式半定互补问题的解集分别转化为下列两种形式的多项式优化问题的可行域, 其中标量型多项式优化问题为:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} & f(\mathbf{x}) \\ \text{s.t.} & g_{ij}(\mathbf{x}) \geq 0, i = 1, 2, \dots, m, \\ & h_z(\mathbf{x}) = 0, z = 1, 2, \dots, m^2, \end{aligned} \quad (6)$$

矩阵型多项式优化问题为:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} & f(\mathbf{x}) \\ \text{s.t.} & Q(\mathbf{x}) \succeq \mathbf{0}, \\ & h_z(\mathbf{x}) = 0, z = 1, 2, \dots, m^2. \end{aligned} \quad (7)$$

显然,  $\mathbf{x} \in S(F_1(\mathbf{x}), F_2(\mathbf{x}))$  当且仅当  $\mathbf{x}$  是问题(6)或(7)的一个可行解。

### 3. 求解全体实数解的层次结构方法

本节中, 我们将研究线性矩阵不等式半定互补问题的层次结构方法, 假设实数解的个数有限, 我们将给出求解该问题的全体实数解的方法。

首先, 我们考虑在问题(6)和问题(7)中选取目标函数  $f(\mathbf{x})$ 。事实上, 通过适当地选取, 多项式  $f(\mathbf{x})$

在  $S(\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x}))$  的不同的解处取得不同的函数值, 将这些值依次排列  $f_1 < f_2 < \dots < f_N$ , 称  $f_i$  是第  $i$  个可行值。

令第  $i$  个解集合为  $S_i = S(\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x})) \cap \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) = f_i\}$ , 则解集合  $S(\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x}))$  能被划分为有限多个互不相交的非空子集, 即  $S(\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x})) = S_1 \cup S_2 \cup \dots \cup S_N$ 。

若我们可以将解集合  $S_i, i=1, 2, \dots, N$  全部求解出来, 则可以计算出原问题的全部实数解。下面我们分别构造与每一个子集合  $S_i$  相对应的优化子问题, 进而求解出所有的  $S_i$ 。此处类比 Nie 在文献[8]中求解多项式的局部极小值的层次结构算法完成这项工作。

显然, 第一个解集合  $S_1$  是第 2 节中构造的两个多项式优化问题(6)和(7)的最优解的集合, 即, 标量型优化问题

$$\begin{aligned} f_1^{1*} &:= \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \\ \text{s.t.} \quad &g_{ij}(\mathbf{x}) \geq 0, i=1, 2; j=1, 2, \dots, m, \\ &h_z(\mathbf{x}) = 0, z=1, 2, \dots, m^2, \end{aligned} \quad (8)$$

和等价的矩阵型优化问题

$$\begin{aligned} f_1^{2*} &:= \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \\ \text{s.t.} \quad &\mathbf{Q}(\mathbf{x}) \succeq \mathbf{0}, \\ &h_z(\mathbf{x}) = 0, z=1, 2, \dots, m^2. \end{aligned} \quad (9)$$

如果问题(8)或问题(9)是可解的, 则可以得到问题的最优值, 在不需要对比两种方法时, 为了方便叙述都将其记为  $f_1$ , 称问题(8)或问题(9)是第一个子问题。

获得了可行值  $f_1$  和相应的可行集  $S_1$  之后, 进一步构造并求解下一个子问题以求解第二个可行集  $S_2$ 。依次进行下去, 每一步在求解出可行值  $f_t$  和相应的可行集  $S_t$  之后, 构造并求解下一个子问题以求解可行集  $S_{t+1}$ , 最终可以求解出全部可行集  $S_1, S_2, \dots, S_N$ 。具体的方法如下: 在获得了前  $t$  个可行值  $f_1, f_2, \dots, f_t$  和相应的可行集  $S_1, S_2, \dots, S_t$  之后, 构造第  $(t+1)$  个子问题, 其中第  $(t+1)$  个标量型子问题为:

$$\begin{aligned} f_{t+1}^{1*} &= \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \\ \text{s.t.} \quad &g_{ij}(\mathbf{x}) \geq 0, i=1, 2; j=1, 2, \dots, m, \\ &h_z(\mathbf{x}) = 0, z=1, 2, \dots, m^2, \\ &f(\mathbf{x}) \geq f_t + \varepsilon, \end{aligned} \quad (10)$$

第  $(t+1)$  个矩阵型子问题为:

$$\begin{aligned} f_{t+1}^{2*} &= \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \\ \text{s.t.} \quad &\mathbf{Q}(\mathbf{x}) \succeq \mathbf{0}, \\ &h_z(\mathbf{x}) = 0, z=1, 2, \dots, m^2, \\ &f(\mathbf{x}) \geq f_t + \varepsilon. \end{aligned} \quad (11)$$

其中问题(10)和问题(11)中使用的参数  $\varepsilon$  是满足

$$0 < \varepsilon < f_{t+1} - f_t, \quad (12)$$

的一个正数。当第  $(t+1)$  个子问题可解时, 无论标量型问题(10)还是等价的矩阵型问题(11), 在不需要对比两种方法时, 为了方便叙述都将其最优值记为  $f^>(f_t + \varepsilon)$ , 在需要对比两种方法时, 我们将这两个最优

值分别记为  $f_{t+1}^{1*}$  和  $f_{t+1}^{2*}$ 。

**引理 3.1**  $f^>(f_t + \epsilon) = f_{t+1}$  当且仅当参数  $\epsilon$  满足条件(12)。

**证明:** 如果  $\epsilon$  满足条件(12), 则  $f_{t+1} > f_t + \epsilon$ 。又因为  $f^>(f_t + \epsilon)$  是  $S(F_1(x), F_2(x))$  上大于或等于  $f_t + \epsilon$  的最小可行值, 则  $f^>(f_t + \epsilon) = f_{t+1}$ 。反之, 如果  $f^>(f_t + \epsilon) = f_{t+1}$ , 则表明在  $(f_t + \epsilon, +\infty)$  上存在可行值, 所以, 若  $f_{t+1}$  存在, 则必有  $f_{t+1} > f_t + \epsilon$ , 即参数  $\epsilon$  满足条件(12)。

但是, 考虑到  $f_{t+1}$  是否存在是未知的, 即便知道其存在, 也很难提前去确定它的值。在实际计算过程中, 参数  $\epsilon$  的取值不宜过小, 以免在数值计算过程中导致结果产生较大偏差。因此, 我们需要考虑如何找到合适的参数  $\epsilon$  来构造第  $(t+1)$  个子问题(10)或(11)。为了解决这一问题, 我们分别构造第  $(t+1)$  个判定优化子问题: 即标量型判定问题

$$\begin{aligned} \max_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_{ij}(x) \geq 0, i=1,2; j=1,2,\dots,m, \\ & h_z(x) = 0, z=1,2,\dots,m^2, \\ & f(x) \leq f_t + \epsilon, \end{aligned} \quad (13)$$

和矩阵型判定问题

$$\begin{aligned} \max_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & Q(x) \succeq 0, \\ & h_z(x) = 0, z=1,2,\dots,m^2, \\ & f(x) \leq f_t + \epsilon. \end{aligned} \quad (14)$$

明显地, 上述两类判定优化子问题都是可解的, 因为任意  $x \in S_1 \cup S_2 \cup \dots \cup S_t$  都是它的可行解, 我们将问题(13)和问题(14)的最优值记为  $f^<(f_t + \epsilon)$ , 则有下列结果成立。

**引理 3.2** 对于  $\epsilon > 0$ ,  $f^<(f_t + \epsilon) = f_t$  当且仅当  $\epsilon < f_{t+1} - f_t$ 。

**证明:** 如果  $\epsilon < f_{t+1} - f_t$ , 则  $f_{t+1} > f_t + \epsilon$ , 由问题(13)或(14)可知,  $f^<(f_t + \epsilon)$  是  $S(F_1(x), F_2(x))$  上小于或等于  $f_t + \epsilon$  的最大可行值, 则  $f^<(f_t + \epsilon) = f_t$ 。反之, 如果  $f^<(f_t + \epsilon) = f_t$ , 则表明在  $(f_t, f_t + \epsilon)$  上不存在可行值, 所以, 如果  $f_{t+1}$  存在, 则必有  $f_{t+1} > f_t + \epsilon$ , 即  $\epsilon < f_{t+1} - f_t$  成立。

事实上,  $\epsilon$  的值不能取太小, 例如 0.05 是符合计算上的要求的。在问题(13)或问题(14)中, 令  $\epsilon = 0.05$ 。如果  $f^<(f_t + \epsilon) > f_t$ , 我们将  $\epsilon$  减小至  $\epsilon/5$ , 再次求解问题(13)或问题(14), 重复这个过程直至  $f^<(f_t + \epsilon) = f_t$ , 我们即可得到合适的参数  $\epsilon$  的值。

利用引理 3.2 的结果, 我们可以确定适当的参数  $\epsilon$ , 利用该参数  $\epsilon$  和当前可行值  $f_t$ , 我们可以构造下一个子问题, 参考公式(10)和(11)。如果下一个子问题不可行, 则说明下一个可行值  $f_{t+1}$  不存在, 即  $f_t$  是线性矩阵不等式半定互补问题的所有的可行值中的最大值。否则, 若  $f_{t+1}$  存在且  $\epsilon$  满足条件(12), 通过求解第  $(t+1)$  个子问题可以得到下一个可行集  $S_{t+1}$ 。至此, 我们给出了线性矩阵不等式半定互补问题的层次结构方法。若我们能够求解出标量型子问题序列(10)和矩阵型多项式优化子问题序列(11), 以及它们对应的判定问题序列(13)和(14), 即可求解出原问题的全部实数解。

#### 4. 半定松弛算法及其收敛性

在前两节的内容中, 我们利用层次结构方法将线性矩阵不等式半定互补问题转化为标量型多项式优化子问题序列(8), (10), (13)或矩阵型多项式优化子问题序列(9), (11), (14)。本节中, 我们对这两类多



项式优化子问题分别给出半定松弛求解方法, 从而完成对原问题的求解。

#### 4.1. 求解标量型多项式优化子问题的半定松弛方法

对于标量型子问题序列(8), (10)和(13), 我们采用 Lasserre 在文献[3]中提出的经典半定松弛方法进行求解。

首先, 我们需要定义局部化矩阵。令  $\mathbf{v}(\mathbf{x}) = [1, x_1, \dots, x_n, x_1^2, x_1x_2, \dots, x_1^k, \dots, x_n^k, \dots]^T$  和  $\mathbf{v}_k(\mathbf{x}) = [1, x_1, \dots, x_n, x_1^2, x_1x_2, \dots, x_1^k, \dots, x_n^k]^T$  分别是  $\mathbb{R}[\mathbf{x}]$  和  $\mathbb{R}[\mathbf{x}]_k$  的一组基, 其中  $\mathbf{v}_k(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]^{s(k)}$ ,  $s(k) = \binom{n+k}{n}$  是它的维数。令  $\mathbf{y} = \{y_\alpha\}_{\alpha \in \mathbb{N}^n} \in \mathbb{R}^{\mathbb{N}^n}$  是一个矩序列,  $p(\mathbf{x})$  是任意一个多项式, 定义 Riesz 函数  $\mathcal{F}_y: \mathbb{R}[\mathbf{x}] \mapsto \mathbb{R}[\mathbf{x}]_k: \mathcal{F}_y(p) = \sum_{\alpha \in \mathbb{N}^n} p_\alpha y_\alpha$ 。如果给定一个截断矩序列  $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^{2k}}$ , 定义关于  $\mathbf{y}$  的  $k$  阶矩阵为  $\mathbf{M}_k(\mathbf{y}) = \mathcal{F}_y(\mathbf{v}_k(\mathbf{x})\mathbf{v}_k(\mathbf{x})^T)$ 。对于标量多项式  $l_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ , 定义由截断矩序列  $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^{2k}}$  生成的关于多项式  $l_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$  的  $k$  阶局部化矩阵为  $\mathbf{L}_{l_i(\mathbf{x})}^{(k)}(\mathbf{y}) = \mathcal{F}_y(l_i(\mathbf{x})\mathbf{v}_{d_1}(\mathbf{x})\mathbf{v}_{d_1}(\mathbf{x})^T)$ , 其中  $d_1 = k - \lfloor \deg(l_i)/2 \rfloor$ 。

如果一个截断矩序列  $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^{2k}}$  满足下面的条件:

$$\begin{aligned} \mathbf{M}_k(\mathbf{y}) \succeq 0, \mathbf{L}_{g_{ij}(\mathbf{x})}^{(k)}(\mathbf{y}) \succeq 0, \mathbf{L}_{h_z(\mathbf{x})}^{(k)}(\mathbf{y}) = 0, \mathbf{L}_{l_i(\mathbf{x})}^{(k)}(\mathbf{y}) \succeq 0, \\ s = \text{rank} \mathbf{M}_k(\mathbf{y}) = \text{rank} \mathbf{M}_{k-d}(\mathbf{y}), \end{aligned} \quad (15)$$

其中  $d = \max\{1, \lfloor \deg(g_{ij})/2 \rfloor, \lfloor \deg(h_z)/2 \rfloor, \lfloor \deg(l_i)/2 \rfloor\}$ , 则称  $\mathbf{y}$  满足平滑条件。若  $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^{2k}}$  满足平滑条件(15), 则  $\mathbf{y}$  在半代数集

$$\{\mathbf{x} \in \mathbb{R}^n : g_{ij}(\mathbf{x}) \geq 0, h_z(\mathbf{x}) = 0, l_i(\mathbf{x}) \geq 0, i = 1, 2; j = 1, 2, \dots, m; z = 1, 2, \dots, m^2\}$$

上存在唯一的一个  $s$ -原子表示测度。

接下来, 我们给出矩阵型子问题序列(8), (10)和(13)的求解方法。例如, 采用 Lasserre 松弛方法求解问题(10)时, 创建问题(10)的  $k$  阶松弛如下:

$$\begin{aligned} r_{i+1,k}^* &= \min \mathbf{L}_f(\mathbf{y}) \\ \text{s.t. } & \mathbf{M}_k(\mathbf{y}) \succeq 0, \\ & \mathbf{L}_{g_{ij}(\mathbf{x})}^{(k)}(\mathbf{y}) \succeq 0, i = 1, 2; j = 1, 2, \dots, m, \\ & \mathbf{L}_{h_z(\mathbf{x})}^{(k)}(\mathbf{y}) = 0, z = 1, 2, \dots, m^2, \\ & \mathbf{L}_{l_i(\mathbf{x})}^{(k)}(\mathbf{y}) \succeq 0, \\ & \mathbf{y}_0 = 1, \\ & \mathbf{y} \in \mathbb{R}^{\mathbb{N}^{2k}}, \end{aligned} \quad (16)$$

其中  $l_i(\mathbf{x}) = f(\mathbf{x}) - f_i - \epsilon$ ,  $k = k_0, k_0 + 1, \dots$ , 且

$$k_0 = \max\{1, \lfloor \deg(f)/2 \rfloor, \lfloor \deg(g_{ij})/2 \rfloor, \lfloor \deg(h_z)/2 \rfloor, \lfloor \deg(l_i)/2 \rfloor\}$$

是最低松弛阶数。

令  $I_{2k}(\mathbf{h})$  是由多项式组  $\mathbf{h}(\mathbf{x})$  生成的理想的第  $2k$  阶截断, 即

$$I_{2k}(\mathbf{h}) = h_1 \cdot \mathbb{R}[\mathbf{x}]_{2k-\deg(h_1)} + h_2 \cdot \mathbb{R}[\mathbf{x}]_{2k-\deg(h_2)} + \cdots + h_{m^2} \cdot \mathbb{R}[\mathbf{x}]_{2k-\deg(h_{m^2})},$$

$Q_k(g_{ij}, l_i)$  是由  $g_{ij}(\mathbf{x})$  和  $l_i(\mathbf{x})$  生成的二次模块的第  $k$  阶截断, 即

$$Q_k(g_{ij}, l_i) = \sum [\mathbf{x}]_{2k} + g_{ij}(x) \cdot \sum [\mathbf{x}]_{2k-\deg(g_{ij})} + l_i(x) \cdot \sum [\mathbf{x}]_{2k-\deg(l_i)}.$$

问题(16)的对偶是:

$$\begin{aligned} \lambda_{i+1,k}^{1*} &= \max_{\mathbf{x} \in \mathbb{R}^n} \gamma \\ \text{s.t.} \quad & f(\mathbf{x}) - \gamma \in I_{2k}(\mathbf{h}) + Q_k(g_{ij}, l_i). \end{aligned} \quad (17)$$

半定规划问题(16)和问题(17)是对偶问题。因此, 对于多项式优化问题(10), 可以采用原对偶内点方法[12]求解相应的松弛问题。通过求解一系列松弛问题(16), 可以得到序列  $\{r_{i+1,k}^{1*}\}$ , 随着松弛阶数增大,  $\{r_{i+1,k}^{1*}\}$  逐渐逼近于多项式优化问题(10)的最优值。问题(8)和(13)可以采用类似的方法进行求解。

## 4.2. 求解矩阵型多项式优化子问题的半定松弛方法

对于矩阵型子问题序列(9), (11)和(14), 我们采用矩阵型半定松弛方法进行求解。类似于标量型 Lasserre 半定松弛方法, 我们参考 Henrion 等在文献[7]中提出的矩阵型松弛方法。首先, 定义由矩阵  $Q(\mathbf{x})$  生成的局部化矩阵为  $L_{Q(\mathbf{x})}^{(k)}(\mathbf{y}) = \mathcal{F}_{\mathbf{y}}(\mathbf{v}_{d_2}(\mathbf{x})\mathbf{v}_{d_2}(\mathbf{x})^T \otimes Q(\mathbf{x}))$ , 其中  $d_2 = k - \lfloor \deg(Q)/2 \rfloor$ ,  $\otimes$  表示 Kronecker 积。

如果一个截断矩序列  $\mathbf{y} \in \mathbb{R}^{\mathbb{N}_{2k}^n}$  满足下面的条件:

$$\begin{aligned} \mathbf{M}_k(\mathbf{y}) \succeq 0, L_{Q(\mathbf{x})}^{(k)}(\mathbf{y}) \succeq 0, L_{h_z(\mathbf{x})}^{(k)}(\mathbf{y}) = 0, L_{l_i(\mathbf{x})}^{(k)}(\mathbf{y}) \succeq 0, \\ s = \text{rank} \mathbf{M}_k(\mathbf{y}) = \text{rank} \mathbf{M}_{k-d}(\mathbf{y}), \end{aligned} \quad (18)$$

其中  $d = \max\{1, \lfloor \deg(Q)/2 \rfloor, \lfloor \deg(h_z)/2 \rfloor, \lfloor \deg(l_i)/2 \rfloor\}$ , 则称  $\mathbf{y}$  满足平滑条件。若  $\mathbf{y} \in \mathbb{R}^{\mathbb{N}_{2k}^n}$  满足平滑条件(18), 则  $\mathbf{y}$  在半代数集

$$\{\mathbf{x} \in \mathbb{R}^n : Q(\mathbf{x}) \succeq 0, h_z(\mathbf{x}) = 0, l_i(\mathbf{x}) \geq 0, z = 1, 2, \dots, m^2\}$$

上存在唯一的一个  $s$ -原子表示测度。

接下来, 我们给出矩阵型子问题序列(9), (11)和(14)的求解方法。例如, 采用矩阵型松弛方法求解问题(11)时, 创建问题(11)的  $k$  阶松弛如下:

$$\begin{aligned} r_{i+1,k}^{2*} &= \min L_f(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{M}_k(\mathbf{y}) \succeq 0, \\ & L_{Q(\mathbf{x})}^{(k)}(\mathbf{y}) \succeq 0, \\ & L_{h_z(\mathbf{x})}^{(k)}(\mathbf{y}) = 0, z = 1, 2, \dots, m^2, \\ & L_{l_i(\mathbf{x})}^{(k)}(\mathbf{y}) \succeq 0, \\ & \mathbf{y}_0 = 1, \\ & \mathbf{y} \in \mathbb{R}^{\mathbb{N}_{2k}^n}, \end{aligned} \quad (19)$$

其中  $l_i(\mathbf{x}) = f(\mathbf{x}) - f_i - \epsilon$ ,  $k = k_0, k_0 + 1, \dots$ , 且

$$k_0 = \max\{1, \lfloor \deg(f)/2 \rfloor, \lfloor \deg(Q)/2 \rfloor, \lfloor \deg(h_z)/2 \rfloor, \lfloor \deg(l_i)/2 \rfloor\}$$



是最低松弛阶数。

令  $I_{2k}(\mathbf{h})$  是由多项式组  $\mathbf{h}(\mathbf{x})$  生成的理想的第  $2k$  阶截断, 即

$$I_{2k}(\mathbf{h}) = h_1 \cdot \mathbb{R}[\mathbf{x}]_{2k-\deg(h_1)} + h_2 \cdot \mathbb{R}[\mathbf{x}]_{2k-\deg(h_2)} + \cdots + h_{m^2} \cdot \mathbb{R}[\mathbf{x}]_{2k-\deg(h_{m^2})},$$

$Q_k(\mathbf{Q}, l_i)$  是由  $\mathbf{Q}(\mathbf{x})$  和  $l_i(\mathbf{x})$  生成的二次模块的第  $k$  阶截断, 即

$$Q_k(\mathbf{Q}, l_i) = \sum[\mathbf{x}]_{2k} + l_i(\mathbf{x}) \cdot \sum[\mathbf{x}]_{2k-\deg(l_i)} + \left\{ \langle \mathbf{R}, \mathbf{Q} \rangle : \mathbf{R} \in \sum[\mathbf{x}]^{m \times m}, \deg(\mathbf{R}_{(i,j)} \mathbf{Q}_{(i,j)}) \leq 2k, \forall i, j \in [m] \right\}.$$

问题(19)的对偶是:

$$\begin{aligned} \lambda_{i+1,k}^{2*} &= \max_{\mathbf{x} \in \mathbb{R}^n} \gamma \\ \text{s.t.} \quad & f(\mathbf{x}) - \gamma \in I_{2k}(\mathbf{h}) + Q_k(\mathbf{Q}, l_i). \end{aligned} \quad (20)$$

半定规划问题(19)和问题(20)是对偶问题。因此, 对于多项式优化问题(11), 可以采用原对偶内点方法[12]求解相应的松弛问题。通过求解一系列松弛问题(19), 可以得到序列  $\{r_{i+1,k}^{2*}\}$ , 随着松弛阶数增大,  $\{r_{i+1,k}^{2*}\}$  逐渐逼近于多项式优化问题(11)的最优值。问题(9)和(14)可以采用类似的方法进行求解。

### 4.3. 算法

下面, 我们给出求解线性矩阵不等式半定互补问题的全体实数解的半定松弛算法。

**算法 4.3.1:** 求解线性矩阵不等式半定互补问题的标量型算法

步骤 1. 随机地选取一个多项式  $f(\mathbf{x})$  作为目标多项式。令  $t := 0$ ,  $k := k_0$ 。符号  $\mathbf{S}(\mathbf{F}_1, \mathbf{F}_2)$  表示线性矩阵不等式半定互补问题的解集,  $\mathcal{F}$  表示可行值构成的集合。

步骤 2. 若问题(8)的松弛问题是不可行的, 则令  $\mathbf{S}(\mathbf{F}_1, \mathbf{F}_2) = \emptyset$ , 停止。否则, 求解问题(8)的松弛问题, 得到最优解  $\mathbf{y}_{1*}^{1,k}$  和最优值  $r_{1,k}^{1*}$ 。

步骤 3. 若存在某个  $s \in [k_0, k]$ , 截断矩阵序列  $\mathbf{y}_{1*}^{1,k} \big|_{2s}$  满足平滑条件(15), 则令  $\mathbf{S}(\mathbf{F}_1, \mathbf{F}_2) := \mathbf{S}_1$ , 其中  $\mathbf{S}_1$  是问题(8)的最优解的集合。令  $\mathcal{F} = \{r_{1,k}^{1*}\}$ ,  $k := k_0$ ,  $t := t+1$ , 转向步骤 4。若不存在这样的  $s$ , 则令  $k := k+1$ , 转向步骤 2。

步骤 4. 令  $\epsilon = 0.05$ , 计算问题(13)的最优值  $f^<(f_i + \epsilon)$ 。若  $f^<(f_i + \epsilon) > f_i$ , 则令  $\epsilon = \epsilon/5$ , 再次求解问题(13), 重复这个过程直至  $f^<(f_i + \epsilon) = f_i$ 。

步骤 5. 求解问题(10)的松弛问题。若它不可行, 则没有更多的实数解, 令  $\mathbf{S}(\mathbf{F}_1, \mathbf{F}_2) = \mathbf{S}$ , 其中  $\mathbf{S} = \mathbf{S}_1 \cup \mathbf{S}_2 \cup \cdots \cup \mathbf{S}_t$ , 停止; 否则, 求解问题(10)的松弛问题得到最优解  $\mathbf{y}_{1*}^{t+1,k}$  和最优值  $r_{1,k}^{t+1*}$ 。

步骤 6. 若存在某个  $s \in [k_0, k]$ , 截断矩阵序列  $\mathbf{y}_{1*}^{t+1,k} \big|_{2s}$  满足平滑条件(15), 则更新  $\mathbf{S} := \mathbf{S} \cup \mathbf{S}_{t+1}$ , 其中  $\mathbf{S}_{t+1}$  是问题(10)的最小值点的集合。令  $\mathcal{F} := \mathcal{F} \cup \{r_{1,k}^{t+1*}\}$ ,  $k := k_0$ ,  $t := t+1$ , 转向步骤 4。若不存在这样的  $s$ , 则令  $k := k+1$ , 转向步骤 5。

**算法 4.3.2:** 求解线性矩阵不等式半定互补问题的矩阵型算法

步骤 1. 随机地选取一个多项式  $f(\mathbf{x})$  作为目标多项式。令  $t := 0$ ,  $k := k_0$ 。符号  $\mathbf{S}(\mathbf{F}_1, \mathbf{F}_2)$  表示线性矩阵不等式半定互补问题的解集,  $\mathcal{F}$  表示可行值构成的集合。

步骤 2. 若问题(9)的松弛问题是不可行的, 则令  $\mathbf{S}(\mathbf{F}_1, \mathbf{F}_2) = \emptyset$ , 停止。否则, 求解问题(9)的松弛问题, 得到最优解  $\mathbf{y}_{2*}^{1,k}$  和最优值  $r_{1,k}^{2*}$ 。

步骤 3. 若存在某个  $s \in [k_0, k]$ , 截断矩阵序列  $\mathbf{y}_{2s}^{1,k} |_{2s}$  满足平滑条件(18), 则令  $\mathbf{S}(\mathbf{F}_1, \mathbf{F}_2) := \mathbf{S}_1$ , 其中  $\mathbf{S}_1$  是问题(9)的最优解的集合。令  $\mathcal{F} = \{r_{1,k}^{2*}\}$ ,  $k := k_0$ ,  $t := t+1$ , 转向步骤 4。若不存在这样的  $s$ , 则令  $k := k+1$ , 转向步骤 2。

步骤 4. 令  $\epsilon = 0.05$ , 计算问题(14)的最优值  $f^<(f_i + \epsilon)$ 。若  $f^<(f_i + \epsilon) > f_i$ , 则令  $\epsilon = \epsilon/5$ , 再次求解问题(14), 重复这个过程直至  $f^<(f_i + \epsilon) = f_i$ 。

步骤 5. 求解问题(11)的松弛问题。若它不可行, 则没有更多的实数解, 令  $\mathbf{S}(\mathbf{F}_1, \mathbf{F}_2) = \mathbf{S}$ , 其中  $\mathbf{S} = \mathbf{S}_1 \cup \mathbf{S}_2 \cup \dots \cup \mathbf{S}_t$ , 停止; 否则, 求解问题(11)的松弛问题得到最优解  $\mathbf{y}_{2s}^{t+1,k}$  和最优值  $r_{t+1,k}^{2*}$ 。

步骤 6. 若存在某个  $s \in [k_0, k]$ , 截断矩阵序列  $\mathbf{y}_{2s}^{t+1,k} |_{2s}$  满足平滑条件(18), 则更新  $\mathbf{S} := \mathbf{S} \cup \mathbf{S}_{t+1}$ , 其中  $\mathbf{S}_{t+1}$  是问题(11)的最小值点的集合。令  $\mathcal{F} := \mathcal{F} \cup \{r_{t+1,k}^{2*}\}$ ,  $k := k_0$ ,  $t := t+1$ , 转向步骤 4。若不存在这样的  $s$ , 则令  $k := k+1$ , 转向步骤 5。

对于 Lasserre 松弛方法[3], Lasserre 证明了在阿基米德条件成立时, 该松弛方法具有渐进收敛性。Henrion 等[7]利用非负多项式矩阵平方和分解的相关理论结果, 将标量型松弛方法进行推广, 得到了矩阵型松弛方法, 并证明了在阿基米德条件成立时, 该矩阵型松弛方法同样具有渐进收敛性。因此, 通过文献[3]和[7]的相关结论, 可以得到算法 4.3.1 和算法 4.3.2 的收敛性定理如下:

**定理 4.3.3** 假设线性矩阵不等式半定互补问题的实数解个数有限, 则当松弛阶数  $k \rightarrow \infty$  时,  $r_{t,k}^{1*} \rightarrow f_t^{1*}, \lambda_{t,k}^{1*} \rightarrow f_t^{1*}, r_{t,k}^{2*} \rightarrow f_t^{2*}, \lambda_{t,k}^{2*} \rightarrow f_t^{2*}$ 。

根据定理 4.3.3 可知: 当线性矩阵不等式半定互补问题的实数解个数有限时, 算法 4.3.1 和 4.3.2 具有渐进收敛性。

## 5. 数值实验

本节将给出求解线性矩阵不等式半定互补问题的数值实验结果。下表中的运行时间是每个维度 20 次的平均运行时间, 时间结果保留四位小数。我们分别用下面的两种方法求解线性矩阵不等式半定互补问题:

方法 1 (标量型方法): 采用直观的标量化方法, 将原问题中的半定约束转化为由顺序主子式组成的约束。将原问题等价转化为子问题序列(8)、(10)和(13), 再采用上一节的算法 4.3.1 进行求解。

方法 2 (矩阵型方法): 保留线性矩阵不等式半定互补问题的矩阵形式进行计算。将原问题等价转化为子问题序列(9)、(11)和(14), 再采用上一节的算法 4.3.2 进行求解。

在数值实验部分, 我们使用的是操作系统是 Windows10, 内存是 8.0 GB RAM, 处理器是 Inter(R)Core(TM)i58265U@1.80 GHz 的笔记本电脑。我们调用 Matlab R2014b 软件, 使用 YALMIP [13], SeDuMi [14]和 Gloptipoly 3 [15]求解线性矩阵不等式半定互补问题。

例 1. 给定多项式矩阵

$$\mathbf{F}_1(\mathbf{x}) = \begin{pmatrix} f_{11}^{(1)}(\mathbf{x}) & f_{12}^{(1)}(\mathbf{x}) \\ f_{21}^{(1)}(\mathbf{x}) & f_{22}^{(1)}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 1-4x_1 & x_1 \\ x_1 & 4-x_1-x_2 \end{pmatrix},$$

$$\mathbf{F}_2(\mathbf{x}) = \begin{pmatrix} f_{11}^{(2)}(\mathbf{x}) & f_{12}^{(2)}(\mathbf{x}) \\ f_{21}^{(2)}(\mathbf{x}) & f_{22}^{(2)}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 4-x_2 & x_1 \\ x_1 & x_2 \end{pmatrix}.$$

考虑线性矩阵不等式半定互补问题:  $\mathbf{F}_1(\mathbf{x}) \succeq \mathbf{0}$ ,  $\mathbf{F}_2(\mathbf{x}) \succeq \mathbf{0}$ ,  $\langle \mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x}) \rangle = 0$ 。设目标函数为  $f(\mathbf{x}) = x_2$ 。

记:  $P_i$ : 求解第  $i$  个子问题至达到收敛所需要的松弛次数;  $S_i$ : 第  $i$  个子问题的松弛问题中半定不等

式约束的规模。下表中的分号是用于分隔在不同松弛阶数的松弛问题中的数据。为了方便表示, 下表中松弛问题的规模  $S_i$  用数字表示, 例如数字 5 表示的是在松弛问题中有一个  $5 \times 5$  的矩阵。我们分别用方法 1 和方法 2 求解该问题, 求解结果见表 1 和表 2:

**Table 1.** Solution, calculation time of the problem and relaxation times of the  $i$ th subproblem

**表 1.** 问题的解、计算时间以及第  $i$  个子问题的松弛次数

方法	问题的解	求解子问题的个数	$P_1$	$P_2$	$P_3$	计算时间/s
1	[0.0000, 4.0000]	3	2	1	1	1.2576
2	[0.0000, 4.0000]	3	2	1	1	1.2315

**Table 2.** The scale of semidefinite inequality constraints in the relaxation problem of the  $i$ th subproblem

**表 2.** 第  $i$  个子问题的松弛问题中半定不等式约束的规模

方法	$S_1$	$S_2$	$S_3$
1	3, 1, 1, 1; 6, 3, 3, 3	3, 1, 1, 1, 1	3, 1, 1, 1, 1
2	3, 4; 6, 12	3, 1, 4	3, 1, 4

例 2. 给定多项式矩阵

$$\mathbf{F}_1(\mathbf{x}) = \begin{pmatrix} f_{11}^{(1)}(\mathbf{x}) & f_{12}^{(1)}(\mathbf{x}) \\ f_{21}^{(1)}(\mathbf{x}) & f_{22}^{(1)}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 1-x_1 & x_1 \\ x_1 & 1-x_2 \end{pmatrix},$$

$$\mathbf{F}_2(\mathbf{x}) = \begin{pmatrix} f_{11}^{(2)}(\mathbf{x}) & f_{12}^{(2)}(\mathbf{x}) \\ f_{21}^{(2)}(\mathbf{x}) & f_{22}^{(2)}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 1-x_2 & x_1 \\ x_1 & 0 \end{pmatrix}.$$

考虑线性矩阵不等式半定互补问题:  $\mathbf{F}_1(\mathbf{x}) \succeq \mathbf{0}$ ,  $\mathbf{F}_2(\mathbf{x}) \succeq \mathbf{0}$ ,  $\langle \mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x}) \rangle = 0$ 。设目标函数为  $f(\mathbf{x}) = x_2$ 。下表中的各个量以及符号所表示的含义与例 1 相同。我们分别用方法 1 和方法 2 求解该问题, 求解结果见表 3 和表 4:

**Table 3.** Solution, calculation time of the problem and relaxation times of the  $i$ th subproblem

**表 3.** 问题的解、计算时间以及第  $i$  个子问题的松弛次数

方法	问题的解	求解子问题的个数	$P_1$	$P_2$	$P_3$	计算时间/s
1	[0.0000, 1.0000]	3	2	2	1	1.6232
2	[0.0000, 1.0000]	3	2	2	1	1.5069

**Table 4.** The scale of semidefinite inequality constraints in the relaxation problem of the  $i$ th subproblem

**表 4.** 第  $i$  个子问题的松弛问题中半定不等式约束的规模

方法	$S_1$	$S_2$	$S_3$
1	3, 1, 1, 1, 1; 6, 3, 3, 3, 3	3, 1, 1, 1, 1, 1; 6, 3, 3, 3, 3, 3	3, 1, 1, 1, 1, 1
2	3, 4; 6, 12	3, 1, 4; 6, 3, 12	3, 1, 4

例 3. 给定多项式矩阵

$$F_1(\mathbf{x}) = \begin{pmatrix} f_{11}^{(1)}(\mathbf{x}) & f_{12}^{(1)}(\mathbf{x}) & f_{13}^{(1)}(\mathbf{x}) & f_{14}^{(1)}(\mathbf{x}) \\ f_{21}^{(1)}(\mathbf{x}) & f_{22}^{(1)}(\mathbf{x}) & f_{23}^{(1)}(\mathbf{x}) & f_{24}^{(1)}(\mathbf{x}) \\ f_{31}^{(1)}(\mathbf{x}) & f_{32}^{(1)}(\mathbf{x}) & f_{33}^{(1)}(\mathbf{x}) & f_{34}^{(1)}(\mathbf{x}) \\ f_{41}^{(1)}(\mathbf{x}) & f_{42}^{(1)}(\mathbf{x}) & f_{43}^{(1)}(\mathbf{x}) & f_{44}^{(1)}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 1-4x_1 & x_1 & x_1 & 0 \\ x_1 & 0 & 4x_1 & 0 \\ x_1 & 4x_1 & 1+x_1 & x_1 \\ 0 & 0 & x_1 & 5x_1 \end{pmatrix},$$

$$F_2(\mathbf{x}) = \begin{pmatrix} f_{11}^{(2)}(\mathbf{x}) & f_{12}^{(2)}(\mathbf{x}) & f_{13}^{(2)}(\mathbf{x}) & f_{14}^{(2)}(\mathbf{x}) \\ f_{21}^{(2)}(\mathbf{x}) & f_{22}^{(2)}(\mathbf{x}) & f_{23}^{(2)}(\mathbf{x}) & f_{24}^{(2)}(\mathbf{x}) \\ f_{31}^{(2)}(\mathbf{x}) & f_{32}^{(2)}(\mathbf{x}) & f_{33}^{(2)}(\mathbf{x}) & f_{34}^{(2)}(\mathbf{x}) \\ f_{41}^{(2)}(\mathbf{x}) & f_{42}^{(2)}(\mathbf{x}) & f_{43}^{(2)}(\mathbf{x}) & f_{44}^{(2)}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 4-x_2 & x_1 & x_1 & 0 \\ x_1 & x_2 & 4x_1 & x_1 \\ x_1 & 4x_1 & x_1 & x_1 \\ 0 & x_1 & x_1 & x_1+x_2 \end{pmatrix}.$$

考虑线性矩阵不等式半定互补问题： $F_1(\mathbf{x}) \succeq \mathbf{0}$ ， $F_2(\mathbf{x}) \succeq \mathbf{0}$ ， $\langle F_1(\mathbf{x}), F_2(\mathbf{x}) \rangle = 0$ 。设目标函数为  $f(\mathbf{x}) = x_2$ 。下表中的各个量以及符号所表示的含义与例 1 相同。我们分别用方法 1 和方法 2 求解该问题，求解结果见表 5 和表 6：

**Table 5.** Solution, calculation time of the problem and relaxation times of the  $i$ th subproblem

**表 5.** 问题的解、计算时间以及第  $i$  个子问题的松弛次数

方法	问题的解	求解子问题的个数	$P_1$	$P_2$	$P_3$	计算时间/s
1	[0.0000, 4.0000]	3	1	1	1	1.3597
2	[0.0000, 4.0000]	3	2	2	1	2.0091

**Table 6.** The scale of semidefinite inequality constraints in the relaxation problem of the  $i$ th subproblem

**表 6.** 第  $i$  个子问题的松弛问题中半定不等式约束的规模

方法	$S_1$	$S_2$	$S_3$
1	6, 3, 3, 3, 3, 1, 1, 1, 1	6, 3, 3, 3, 3, 3, 1, 1, 1, 1	6, 3, 3, 3, 3, 3, 1, 1, 1, 1
2	3, 8; 6, 24	3, 1, 8; 6, 3, 24	3, 1, 8

注：在例 3 中，方法 2 的第一和第二个子问题在最低阶松弛阶数时不收敛，因此我们需要求解 5 个半定问题，方法 1 的三个子问题都在最低阶松弛阶数时收敛，所以只需要求解 3 个半定问题。因为求解半定问题占据了大量的计算量，所以在例 3 中方法 2 的计算时间大于方法 1。

例 4. 给定多项式矩阵

$$F_1(\mathbf{x}) = \begin{pmatrix} f_{11}^{(1)}(\mathbf{x}) & f_{12}^{(1)}(\mathbf{x}) & f_{13}^{(1)}(\mathbf{x}) & f_{14}^{(1)}(\mathbf{x}) \\ f_{21}^{(1)}(\mathbf{x}) & f_{22}^{(1)}(\mathbf{x}) & f_{23}^{(1)}(\mathbf{x}) & f_{24}^{(1)}(\mathbf{x}) \\ f_{31}^{(1)}(\mathbf{x}) & f_{32}^{(1)}(\mathbf{x}) & f_{33}^{(1)}(\mathbf{x}) & f_{34}^{(1)}(\mathbf{x}) \\ f_{41}^{(1)}(\mathbf{x}) & f_{42}^{(1)}(\mathbf{x}) & f_{43}^{(1)}(\mathbf{x}) & f_{44}^{(1)}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 1-x_1 & x_1 & x_2 & 0 \\ x_1 & 0 & x_1 & 0 \\ x_2 & x_1 & x_2 & x_1 \\ 0 & 0 & x_1 & x_1 \end{pmatrix},$$

$$F_2(\mathbf{x}) = \begin{pmatrix} f_{11}^{(2)}(\mathbf{x}) & f_{12}^{(2)}(\mathbf{x}) & f_{13}^{(2)}(\mathbf{x}) & f_{14}^{(2)}(\mathbf{x}) \\ f_{21}^{(2)}(\mathbf{x}) & f_{22}^{(2)}(\mathbf{x}) & f_{23}^{(2)}(\mathbf{x}) & f_{24}^{(2)}(\mathbf{x}) \\ f_{31}^{(2)}(\mathbf{x}) & f_{32}^{(2)}(\mathbf{x}) & f_{33}^{(2)}(\mathbf{x}) & f_{34}^{(2)}(\mathbf{x}) \\ f_{41}^{(2)}(\mathbf{x}) & f_{42}^{(2)}(\mathbf{x}) & f_{43}^{(2)}(\mathbf{x}) & f_{44}^{(2)}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 1-x_2 & x_1 & x_1 & 0 \\ x_1 & x_2 & x_1 & x_1 \\ x_1 & x_1 & x_1 & x_1 \\ 0 & x_1 & x_1 & x_2 \end{pmatrix}.$$

考虑线性矩阵不等式半定互补问题:  $F_1(x) \succeq 0$ ,  $F_2(x) \succeq 0$ ,  $\langle F_1(x), F_2(x) \rangle = 0$ 。设目标函数为  $f(x) = x_1$ 。下表中的各个量以及符号所表示的含义与例 1 相同。我们分别用方法 1 和方法 2 求解该问题, 求解结果见表 7 和表 8:

**Table 7.** Solution, calculation time of the problem and relaxation times of the  $i$ th subproblem  
**表 7.** 问题的解、计算时间以及第  $i$  个子问题的松弛次数

方法	问题的解	求解子问题的个数	$P_1$	$P_2$	$P_3$	计算时间/s
1	[0.0000, 1.0000]	3	1	1	1	1.3331
2	[0.0000, 1.0000]	3	1	1	1	1.0233

**Table 8.** The scale of semidefinite inequality constraints in the relaxation problem of the  $i$ th subproblem  
**表 8.** 第  $i$  个子问题的松弛问题中半定不等式约束的规模

方法	$S_1$	$S_2$	$S_3$
1	6, 3, 3, 3, 3, 1, 1, 1, 1	6, 3, 3, 3, 3, 3, 1, 1, 1, 1	6, 3, 3, 3, 3, 3, 1, 1, 1, 1
2	3, 8	3, 1, 8	3, 1, 8

通过上面的数值实验的结果可以发现: 虽然这两种方法一般都可以求解出线性矩阵不等式半定互补问题的全部实数解, 但是这两种方法的求解效果却不同。下面给出这两种方法的数值实验结果的对比:

1) 一般情况下, 方法 2 比方法 1 消耗时间较少, 这是因为在用方法 1 时, 随着矩阵的维数增加, 松弛阶数和引入的不等式约束的个数会进一步增加; 但是也有特例, 如例 3, 这是因为在使用矩阵型方法求解线性矩阵不等式半定互补问题时, 有两个子问题在最低阶松弛阶数时不收敛, 所以需要求解更高阶数的半定问题。因为求解半定问题占据了大量的计算量, 所以在例 3 中方法 2 的计算时间大于方法 1;

2) 一般情况下, 随着矩阵维数的增加, 用标量型方法和矩阵型方法求解时的消耗时间的差别会更大, 这是因为当用标量型松弛方法求解时, 松弛阶数和引入的不等式约束的数量会随着矩阵的维数的增加而快速增加。所以, 在一般情况下, 当矩阵维数较大时, 使用矩阵型方法更有优势。

## 6. 结语

线性矩阵不等式半定互补问题是一种新型的半定互补问题, 是半定规划问题与互补问题的交叉研究内容。该问题的求解可以转化为一系列带有线性矩阵不等式约束的多项式优化子问题, 进而可以采用标量型松弛方法或矩阵型松弛方法进行求解。当这个问题的实数解个数有限时, 利用本文给出的算法 4.3.1 或 4.3.2 可以求出线性矩阵不等式半定互补问题的全部实数解。在数值实验部分, 将算法 4.3.1 和 4.3.2 进行了比较, 验证了这两类算法都可以对线性矩阵不等式半定互补问题进行求解, 但在一般情况下, 当矩阵的维数较大时, 矩阵型算法更有优势。此外, 我们还可以考虑利用本文给出的算法 4.3.1 和 4.3.2 求解更加一般的问题。例如, 给定  $F_1(x), F_2(x), \dots, F_k(x) \in \mathcal{S}[x]^m$ , 寻求解  $x \in \mathbb{R}^n$  满足  $F_i(x) \succeq 0$ ,  $F_j(x) \succeq 0$ ,  $\langle F_i(x), F_j(x) \rangle = 0$ ,  $1 \leq i < j \leq k$ , 其中  $F_i(x) \succeq 0$  是线性矩阵不等式。另外, 我们也可以考虑采用算法 4.3.1 和算法 4.3.2 求解问题:  $F_1(x) \succeq 0$ ,  $F_2(x) \succeq 0$ ,  $\langle F_1(x), F_2(x) \rangle = 0$ , 其中  $F_i(x) \succeq 0, i=1, 2$  是多项式矩阵不等式。

## 基金项目

中央高校基本科研业务费专项资金资助项目(19D110905)。

## 参考文献

- [1] 戴昊, 崔志文, 袁鹏, 等. 基于线性矩阵不等式的巡检机器人路径规划[J]. 机械制造与自动化, 2021, 50(4): 212-215.
- [2] Park, I.S., Park, C., Kwon, N.K., *et al.* (2021) Dynamic Output-Feedback Control for Singular Interval-Valued Fuzzy Systems: Linear Matrix Inequality Approach. *Information Sciences*, **576**, 393-406. <https://doi.org/10.1016/j.ins.2021.06.053>
- [3] Lasserre, J.B. (2001) Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, **11**, 796-817. <https://doi.org/10.1137/S1052623400366802>
- [4] Hol, C.W.J. and Scherer, C.W. (2004) Sum of Squares Relaxations for Polynomial Semi-Definite Programming. *Proceedings Symposium on Mathematical Theory of Networks and Systems (MTNS)*, Leuven, 5-9 July 2004.
- [5] Hol, C.W.J. and Scherer, C.W. (2005) A Sum-of-Squares Approach to Fixed-Order  $H_\infty$ -Synthesis. In: *Positive Polynomials in Control*, Springer, Berlin, 45-71. [https://doi.org/10.1007/10997703\\_3](https://doi.org/10.1007/10997703_3)
- [6] Kojima, M. (2003) Sums of Squares Relaxations of Polynomial Semi-Definite Programs. Institute of Technology.
- [7] Henrion, D. and Lasserre, J.B. (2006) Convergent Relaxations of Polynomial Matrix Inequalities and Static Output Feedback. *IEEE Transactions on Automatic Control*, **51**, 192-202. <https://doi.org/10.1109/TAC.2005.863494>
- [8] Nie, J.W. (2015) The Hierarchy of Local Minimums in Polynomial Optimization. *Mathematical Programming*, **151**, 555-583. <https://doi.org/10.1007/s10107-014-0845-2>
- [9] 张怡, 阎晓艳. 用线性矩阵不等式方法求解控制理论问题[J]. 机械管理开发, 2006, 28(1): 24-25.
- [10] Nesterov, Y. and Nemirovskii, A. (1994) Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics, Philadelphia. <https://doi.org/10.1137/1.9781611970791>
- [11] 王萼芳, 石生明. 高等代数[M]. 第四版. 北京: 高等教育出版社, 2013.
- [12] Griva, I., Shanno, D.F., Vanderbei, R.J., *et al.* (2008) Global Convergence of a Primal-Dual Interior-Point Method for Nonlinear Programming. *Algorithmic Operations Research*, **3**, 12-29.
- [13] Löfberg, J. (2004) YALMIP: A Toolbox for Modeling and Optimization in MATLAB. *2004 IEEE International Conference on Robotics and Automation*, Taipei, 2-4 September 2004, 284-289. <https://doi.org/10.1109/CACSD.2004.1393890>
- [14] Sturm, J.F. (1999) Using SeDuMi 1.02, a MATLAB Toolbox for Optimization over Symmetric Cones. *Optimization Methods and Software*, **11**, 625-653. <https://doi.org/10.1080/10556789908805766>
- [15] Henrion, D., Lasserre, J.B. and Löfberg, J. (2009) Gloptipoly3: Moments, Optimization and Semi-Definite Programming. *Optimization Methods and Software*, **24**, 761-779. <https://doi.org/10.1080/10556780802699201>