2012 年度 大問 4

hari64boli64 (hari64boli64@gmail.com)

2025 年 4 月 23 日

1 問題

双対問題を用いた最適化

2 解答

(1),(2),(3)

略

(4)

(3) より、

$$\begin{array}{rl} \text{Minimize} & s \\ \text{subject to} & 3x_1^2 - 4x_1x_2 - 2x_2^2 + s \geq 0 \quad (\forall \boldsymbol{x} \in U) \end{array}$$

を解けばよい。
(2) より、

$$\begin{aligned} 3x_1^2 - 4x_1x_2 - 2x_2^2 + s \geq t_1(&-2x_1^2 - 2x_1x_2 - x_2^2 + 4) \\ &+t_2(x_1^2 - 1) \quad (\forall \boldsymbol{x} \in \mathbb{R}^n) \end{aligned}$$

を満たす t_1, t_2 が存在する範囲内で、s を最小化すればよい。
(1) より、

$$
\begin{pmatrix}
3+2t_1-t_2 & -2+t_1 & \\
-2+t_1 & -2+t_1 & \\
 & & s-4t_1+t_2
\end{pmatrix} \succeq 0
$$

の下で、s を最小化すればよい。

則ち、

$$
\begin{pmatrix}
3+2t_1-t_2 & -2+t_1 \\
-2+t_1 & -2+t_1
\end{pmatrix} \succeq 0
$$

の下で、$4t_1-t_2$ を最小化すればよい。

これは、固有値を求めた上で考えると、解の配置問題である。

尤も、直接解くのはかなり厳しい。

しかし、適当に考えると、$t_1=2, t_2=7$ の時、固有値は共に 0 で、目的関数の値は 1 になる。

そして、$x_1=1, x_2=-1+\sqrt{3}$ を代入すると、確かに元の最適化問題において、制約を満たした上で、これの目的関数も 1 になる。

以上より、$s=1$ が最適解である。

なお、後半 3 行は計算機パワーを用いたチートであり、試験の上では、(3),(2),(1) を用いた変形さえ出来ていれば、十分ではないかと思われる。

普通に小行列式に注目すると解ける。

3 おまけ

Listing 1 vis

```
import matplotlib.pyplot as plt
import random
import numpy as np
from tqdm import tqdm


def g_0(x1, x2):
    return -3 * x1 * x1 + 4 * x1 * x2 + 2 * x2 * x2


def g_1(x1, x2):
    return -2 * x1 * x1 - 2 * x1 * x2 - x2 * x2 + 4


def g_2(x1, x2):
```

```python
    return x1 * x1 - 1


def solve():
    ans = -1e9
    best_x1 = 0
    best_x2 = 0
    for _ in tqdm(range(int(1e5))):
        x1 = random.random() * 10 - 5
        x2 = random.random() * 10 - 5
        if g_1(x1, x2) >= 0 and g_2(x1, x2) >= 0:
            ans = max(ans, g_0(x1, x2))
            if ans == g_0(x1, x2):
                best_x1 = x1
                best_x2 = x2
    print(f"ans: {ans} x1: {best_x1} x2: {best_x2}")

    x1 = np.linspace(-5, 5, 1000)
    x2 = np.linspace(-5, 5, 1000)
    X1, X2 = np.meshgrid(x1, x2)

    Z = g_0(X1, X2)
    Z[g_1(X1, X2) < 0] -= 100
    Z[g_2(X1, X2) < 0] -= 100
    plt.imshow(Z, extent=[-5, 5, -5, 5], origin="lower")
    plt.title("visualization of problem")
    plt.savefig("4_vis.png")
    plt.close()

    Z[g_1(X1, X2) < 0] = np.nan
    Z[g_2(X1, X2) < 0] = np.nan
    plt.imshow(Z, extent=[-5, 5, -5, 5], origin="lower")
    plt.colorbar()
    plt.title("maximize g_0 s.t. g_1 >= 0, g_2 >= 0")
    plt.savefig("4_actual.png")
    plt.close()


def solve2():
    ans = +1e9
    best_t1 = 0
    best_t2 = 0
    for _ in tqdm(range(int(1e5))):
        t1 = random.random() * 20 - 10
        t2 = random.random() * 20 - 10
        A = np.array([[3 + 2 * t1 - t2, -2 + t1], [-2 + t1, -2 +
            t1]])
```

```python
        if np.all(np.linalg.eigvals(A) >= 0):
            ans = min(ans, 4 * t1 - t2)
            if ans == 4 * t1 - t2:
                best_t1 = t1
                best_t2 = t2
    print(f"ans: {ans} t1: {best_t1} t2: {best_t2}")


def main():
    solve()
    # solve2()


if __name__ == "__main__":
    main()
```

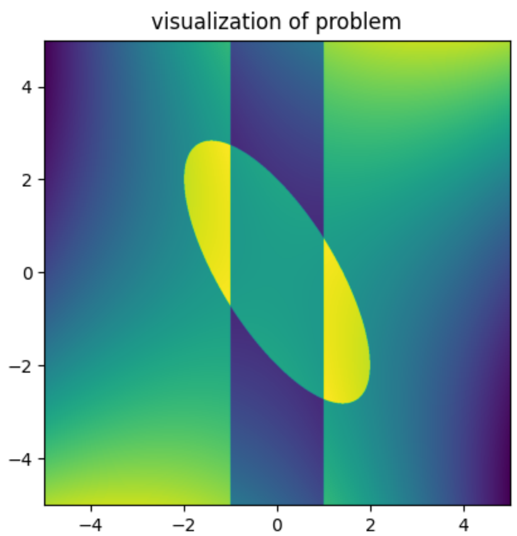

図 1　イメージ図

1 個目の条件式は、とある楕円形内部の点であることを表す。
2 個目の条件式は、$x_1 \leq -1$、または、$x_1 \geq 1$ を表す。
目的関数は右上と左下の値が高い、双曲線のような形をしている。
そのことに対応した図であることが読みとれるかと思われる。

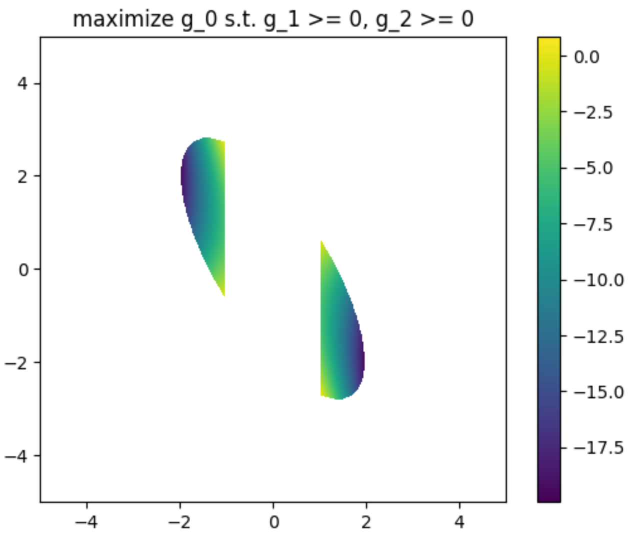

図 2　総当たりによる実際の図