Princeton University

ORF 363/COS 323
Final Exam, Fall 2021

December 15, 2021

Instructor:
A.A. Ahmadi

AIs:
Becker, Chaudhry, Dibek, Silin, Yu

1. Please write your name on the first page of your solutions. Next to it, please write out and sign the following pledge: "I pledge my honor that I have not violated the Honor Code or the rules specified by the instructor during this examination."

2. The exam is not to be discussed with *anyone* except possibly the instructors and the AIs. You can only ask *clarification questions*, and only as *public* (and preferably non-anonymous) questions on Ed Discussion. No emails.

3. You are allowed to consult the lecture notes, your own notes, the reference books of the course as indicated on the syllabus, the problem sets and their solutions (yours and ours), the midterm and its solutions (yours and ours), the practice midterm and final exams and their solutions, all Ed Discussion posts, all videos, but *nothing else.* You can only use the Internet in case you run into problems related to software.

4. You may refer to facts proven in the notes or problem sets without reproving them.

5. For all computational problems, include your code. The output that you present should come from your code.

6. Unless you have been granted an extension because of overlapping finals, the exam is to be turned in on Friday (December 17, 2021) at 10 AM EST on Gradescope as a single PDF file. You are free to write your solutions on paper or on a tablet, or to type them up. Only the latest version submitted before your deadline will be graded.

7. Each question has 25 points. You need to justify your answers to receive full credit.

Problem 1: Friends in High Places

You follow the path of Jeff Bezos and become a billionaire after graduation. Naturally, you feel compelled to send your friends to space. You would like to send as many friends as possible, but on a safety-critical mission like this, there is no room for disagreement on the spaceship. If you have n friends to choose from, you write down an $n \times n$ matrix describing the disagreement between any pair of friends. The entries of this matrix are real numbers between 0 and 1, where 0 means "they get along perfectly" and 1 means "they fight all the time". This matrix is symmetric and has zero diagonal. Here is an example of a disagreement matrix when there are five friends to choose from:

$$\begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 0.1 & 0.6 & 0.4 & 0.7 \\ 0.1 & 0 & 0.4 & 0.2 & 1.0 \\ 0.6 & 0.4 & 0 & 0.3 & 0.4 \\ 0.4 & 0.2 & 0.3 & 0 & 0.7 \\ 0.7 & 1.0 & 0.4 & 0.7 & 0 \end{bmatrix} \end{array}.$$

For a given subset of friends, the total disagreement is the sum of all disagreement values between pairs in the subset. For example, the set of friends $\{1, 3, 4\}$ incurs a total disagreement of $0.6 + 0.4 + 0.3 = 1.3$. Given your disagreement matrix M and a nonnegative scalar p, you want to find the maximum number of friends you can send to space with total disagreement at most p.[1] You formulate this problem as follows:

$$\begin{aligned} \eta^* \quad := \quad \max_{x \in \mathbb{R}^n} \quad & \sum_{i=1}^{n} x_i \\ \text{s.t.} \quad & \sum_{i,j=1}^{n} M_{ij} x_i x_j \leq 2p \\ & x_i(x_i - 1) = 0 \ \text{ for } i = 1, \ldots, n. \end{aligned} \tag{1}$$

1. Consider the following linear program[2]:

$$\begin{aligned} \eta^*_{LP} \quad := \quad \max_{x \in \mathbb{R}^n} \quad & \sum_{i=1}^{n} x_i \\ \text{s.t.} \quad & \sum_{i,j=1}^{n} M_{ij} \max\{x_i + x_j - 1, 0\} \leq 2p \\ & 0 \leq x_i \leq 1 \ \text{ for } i = 1, \ldots, n. \end{aligned} \tag{2}$$

   Show that $\eta^* \leq \eta^*_{LP}$.

[1] While you are not required to show this, the problem you are facing is NP-hard.

[2] Even though no points are assigned to this, you may want to convince yourself that this optimization problem can indeed be reformulated as a linear program.

2. Let S^{n+1} denote the set of symmetric $(n+1)\times(n+1)$ matices. Let $\hat{M} \in S^{n+1}$ denote the matrix obtained from M by adding a row and a column of zeros, i.e., M is the upper left $n \times n$ submatrix of $\hat{M}$ and the other entries of $\hat{M}$ are all zeros. For $i = 1,\ldots,n$, let $A^i \in S^{n+1}$ denote the matrix with $A^i_{ii} = 1$, $A^i_{i,n+1} = A^i_{n+1,i} = -0.5$, and zeros everywhere else. Consider the following semidefinite program:

$$\begin{aligned} \eta^*_{SDP} \quad := \quad \max_{X \in S^{n+1}} \quad & \mathrm{Tr}(X) - 1 \\ \text{s.t.} \quad & \mathrm{Tr}(\hat{M}X) \leq 2p \\ & \mathrm{Tr}(A^i X) = 0 \ \text{ for } i = 1,\ldots,n \\ & X_{n+1,n+1} = 1 \\ & X \succeq 0. \end{aligned} \tag{3}$$

Show that $\eta^* \leq \eta^*_{SDP}$.

3. Solve problems (2) and (3) for the 14×14 matrix M given in `disagreement_matrix.mat` and $p = 1.12$. You may use the following code to load the disagreement matrix:

```
load('disagreement_matrix.mat')
```

Report the two optimal values and specify in each case whether they are equal to η^*. Which of your 14 friends should you be sending to space?

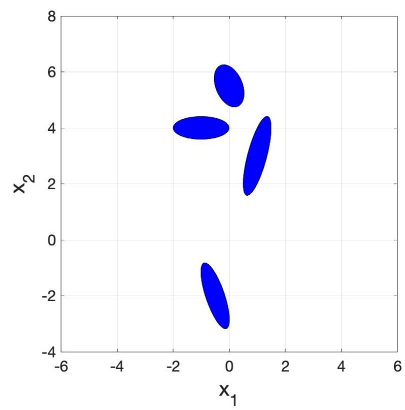
Problem 2: Winter Break Trip

For the winter break, you are planning to travel to a national park with 4 lakes. Each lake L_i has the shape of an ellipse with known center $c_i \in \mathbb{R}^2$ and shape matrix $A_i \in \mathbb{R}^{2\times 2}$, $A_i \succ 0$:

$$L_i = \left\{x \in \mathbb{R}^2 \mid (x - c_i)^\top A_i (x - c_i) \leq 1\right\}, \quad i = 1, \ldots, 4.$$

The lakes are shown in the figure below and correspond to the following data:

$$\begin{aligned} &A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 6 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 8 & -2 \\ -2 & 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 8 & 2.5 \\ 2.5 & 1.5 \end{bmatrix}, \\ &c_1 = [-1, 4]^\top, \quad c_2 = [1, 3]^\top, \qquad c_3 = [0, 5.5]^\top, \quad c_4 = [-0.5, -2]^\top. \end{aligned}$$



You want to visit all 4 lakes, one lake per day. You need to pick a permanent location $z \in \mathbb{R}^2$ for your tent where you will be staying overnight. On the i-th day, you will be hiking from your tent to the i-th lake and back (using the shortest route to the closest point of each lake). For simplicity, no restrictions are imposed on the permanent location z of your tent. Write down and solve two convex optimization problems which would respectively minimize:

(a) the sum of squared Euclidean distances to all the lakes;

(b) the maximum Euclidean distance to any of the lakes (i.e. on no day you want to hike too much).

Report the optimal solutions. You can optionally plot your optimal solutions on the MATLAB figure provided (no points are assigned to plotting). For instance, to plot the point $[2, 1]^\top$ on the map, you can use the following code:

```
openfig('lakes.fig')
plot(2, 1, 'k*', 'MarkerSize', 10, 'LineWidth', 2)
```

Problem 3: Dynamical Leontief Economy

Recall the Leontief input-output model from Lecture 10. Imagine an economy where each sector decides its production amount for tomorrow based on the demand today. Let n denote the number of sectors, $C \in \mathbb{R}^{n \times n}$ be the consumption matrix, $d \in \mathbb{R}^n$ be the demand vector of the open sector (e.g., the general public), and $x_t \in \mathbb{R}^n$ be the vector of production on day t. Then, x_t evolves according to the dynamics:

$$x_{t+1} = Cx_t + d.$$

As we have seen, the right amount of production in the Leontief model is $x^* = (I - C)^{-1}d$. If the initial production $x_0 \in \mathbb{R}^n$ is away from x^*, one might wonder if the economy can balance itself given enough time. More precisely, we want to see if $x_t \to x^*$ as $t \to \infty$. Consider the same Leontief economy as in Problem Set 6. There are four sectors and a consumption matrix C given by:

	Agriculture	Energy	Transportation	Manufacturing
Agriculture	0.1	0	0.15	0.15
Energy	0.15	0.12	0	0
Transportation	0.1	0.15	0.2	0.15
Manufacturing	0	0.1	0.1	0.7.

Prove that (no matter what the vector d is) for any initial $x_0 \in \mathbb{R}^4$, we have $x_t \to x^*$.
Hint: Show this by presenting an appropriate quadratic Lyapunov function. It may be helpful to perform a change of variables first.

Problem 4: Making The Most Out of an Optimal Value Blackbox

Consider the following optimization problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega, \end{aligned} \tag{4}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous function and $\Omega \subseteq \mathbb{R}^n$ is a nonempty, closed, and bounded set. Suppose you have a blackbox that can return the optimal value of any optimization problem with a continuous objective function and a nonempty, closed, and bounded feasible set. How can you use only this blackbox (potentially many times) to:

(a) find an optimal solution to (4),

(b) test if the optimal solution to (4) is unique.

In each case, how many calls to the blackbox does your approach require?