

AI505/AI801, Optimization – Exercise Sheet 09

2026-04-22

Exercises with the symbol $+$ are to be done at home before the class. Exercises with the symbol $*$ will be tackled in class. The remaining exercises are left for self training after the exercise class. Some exercises are from the text book and the number is reported. They have the solution at the end of the book.

Exercise 1 $+$ (11.1)

Suppose you do not know any optimization algorithm for solving a linear program. You decide to evaluate all the vertices and determine, by inspection, which one minimizes the objective function. Give a loose upper bound on the number of possible minimizers you will examine. Furthermore, does this method properly handle all linear constrained optimization problems?

Exercise 2 $+$ (11.2)

If the program in Example 11.1 of [KW] is bounded below, argue that the simplex method must converge.

Exercise 3 $+$ (11.3)

Suppose we want to solve:

$$\begin{aligned} &\text{minimize } 6x_1 + 5x_2 \\ &\text{subject to } 3x_1 - 2x_2 \geq 5. \end{aligned}$$

How would we translate this problem into a linear program in equality form with the same minimizer?

Exercise 4

Consider the following problem:

$$\begin{aligned} &\text{minimize } 5x_1 + 4x_2 \\ &\text{s.t. } 2x_1 + 3x_2 \leq 5 \\ &\quad 4x_1 + x_2 \leq 11 \\ &\quad x_1, x_2 \geq 0 \end{aligned}$$

Solve the problem numerically implementing the simplex algorithm.

Exercise 5 + (11.4)

Suppose your optimization algorithm has found a search direction \mathbf{d} and you want to conduct a line search. However, you know that there is a linear constraint $\mathbf{w}^T \mathbf{x} \geq 0$. How would you modify the line search to take this constraint into account? You can assume that your current design point is feasible.

Exercise 6 + (11.5)

Reformulate the linear program

$$\begin{aligned} & \text{minimize } \mathbf{c}^T \mathbf{x} \\ & \text{s.t. } A\mathbf{x} \geq 0 \end{aligned}$$

into an unconstrained optimization problem with a log barrier penalty.

Exercise 7

Reformulate the the problem of computing the analytic center of the set of linear inequalities

$$\mathbf{a}_i^T \mathbf{x} \leq 1, i = 1, \dots, m, \quad |x_i| \leq 1, i = 1, \dots, n.$$

Note that we can choose $\mathbf{x}(0) = 0$ as our initial point. You can generate instances of this problem by choosing a_i from some distribution on \mathbb{R}^n and then solving the problem with `scipy.optimize.minimize`.

Exercise 8

Write the Lagrangian function for the following problem and derive the dual function D . Determine $\mathbf{x}(\boldsymbol{\lambda})$, the minimizer of D and the gradient of D .

$$\begin{aligned} & \text{minimize } (x_1 - 1)^2 + (x_2 - 1)^2 \\ & \text{subject to } x_1 + 2x_2 - 1 = 0 \\ & \quad \quad \quad 2x_1 + x_2 - 1 = 0 \end{aligned}$$

Exercise 9 *

A pharmaceutical company produces three types of drugs: A, B, and C. These drugs require raw materials, chemical processing time, and packaging units. The goal is to maximize profit, taking into account restrictions due to resource availability and production balance.

Drug A contributes 60 DKK per unit to the profit, drug B contributes 100 DKK per unit, drug C contributes 80 DKK per unit. The consumptions per units of drug of raw materials, chemical processing time, and packaging units are given in Table [tab] together with the quantities available. While consumptions of raw material and processing time cannot exceed the amount available, excess of packaging units is allowed. Extra packaging units can be bought but each extra unit reduces profit at 5 DKK per unit while packaging units left can be reused in the next production period and hence contribute positively to the profit with the same amount of DKK per unit. Due to contractual obligations, Drug B must be produced in exactly twice the amount of Drug A.

Formulate the problem in linear programming terms. Write first the *instantiated* model and then the abstract, *general* model separating model from data.

Table 1: Data from the pharmaceutical company

Drug	A	B	C	
raw material (Kg)	5	8	6	600
processing time (hours)	3	4	5	400
packaging units	2	3	1	200
profit	60	100	80	-5

Exercise 10 *

Consider the following linear programming problem:

$$\begin{aligned}
 & \text{maximize } Z = 60x_1 + 100x_2 + 80x_3 - 5x_4 \\
 & \text{s.t. } 5x_1 + 8x_2 + 6x_3 \leq 600 \\
 & \quad 3x_1 + 4x_2 + 5x_3 \leq 400 \\
 & \quad 2x_1 + 3x_2 + x_3 = 200 + x_4 \\
 & \quad x_2 = 2x_1 \\
 & \quad x_1, x_2, x_3 \geq 0 \\
 & \quad x_4 \in \mathfrak{R}
 \end{aligned}$$

Your tasks:

- Transform the problem in standard form
- Transform the problem in equality form
- Solve the problem numerically with `scipy.optimize.linprog`. Read this [tutorial](#).

Exercise 11

Two opponents, say Antigonus and Brasidas, are fighting a battle over three mountain passes. Each of them commands seven legions. The one who sends more legions to a pass occupies it, but when the same number of legions meet, there will be a draw. Finally, the one who occupies more passes than the other wins the battle, with a draw occurring if both occupy the same number of passes.

They must simultaneously decide how to allocate their legions over the passes without knowing what the other will do. The goal is to win the most passes (or maximize some payoff based on the passes won). It makes sense to randomize the allocations to avoid being predictable. By randomized allocations, we mean that Antigonus is making his choices at random according to some probability distribution. Of course, the same applies to Brasidas. The problem is that they must decide on the probability distributions for their allocations. The probability distributions are defined over the set of all possible allocations of legions to the three passes.

(You can rethink the problem in the context of election or marketing competitions, if you like.)

Your tasks are:

1. Determine how many different allocations each opponent has.
2. Build the payoff matrix for Antigonus' allocations on the rows and Brasidas' allocations on the columns. Each element of the matrix gets 1 if Antigonus' allocation wins, -1 if Brasidas' allocation wins and 0 otherwise.

3. Model the problem of finding the optimal strategy for Antigonous as a linear programming problem. Assume that Antigonous, when playing some randomized allocation, expects Brasidas to play a best response against this allocation. The goal of Antigonous is to maximize the worst-case expected payoff, say v . In this way, by adopting his optimal strategy, he can assure himself of winning v on average.
4. Solve the linear programming problem with the tools from the Python module `scipy.optimize` and highlight the optimal strategy found for Antigonous and the optimal value, that is, the expected payoff (expected win for Antigonous).
5. Try changing the number of legions. Instead of seven, try with other numbers, say five, six, eight, nine. Do you find a case in which the optimal allocation strategy is deterministic, that is, a pure unique allocation?
6. Model the problem from the point of view of Brasidas, assuming that he also uses a randomized strategy. Use the same payoff matrix constructed for Antigonous. What is the expected payoff for him? Does the result make sense?