

AI505/AI801, Optimization – Exercise Sheet 10

2026-04-27

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 $+$ (19.2)

A Boolean satisfiability problem, often abbreviated SAT, requires determining whether a Boolean design exists that causes a Boolean-valued objective function to output true. SAT problems were the first to be proven to belong to the difficult class of NP-complete problems. This means that SAT is at least as difficult as all other problems whose solutions can be verified in polynomial time.

Consider the Boolean objective function:

$$f(\mathbf{x}) = x_1 \wedge (x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2)$$

Formulate the problem as an integer linear program. Can any Boolean satisfiability problem be formulated as an integer linear program? Solve the problem with scipy.

Exercise 2 $*$

Consider $\max\{\mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{Z}^n\}$. Sometimes the solution of the linear relaxation is already integral. Can you find a sufficient condition for the matrix A for that to happen?

Exercise 3 $*$

Consider the following problem:

$$\begin{aligned} & \max x_1 - x_2 \\ & \text{subject to: } \frac{2}{3}x_1 + \frac{1}{2}x_2 \leq 1 \\ & \quad x_1, x_2 \geq 0 \\ & \quad x_1, x_2 \in \mathbb{Z} \end{aligned}$$

Derive a Chvatal-Gomory cut.

Exercise 4 *

The 0-1 knapsack problem is a combinatorial optimization problem that can be described as follows: Given a set of items, each with a size/weight and a value, the problem is to choose the items that maximize the total value under the condition that the total size/weight is below a certain threshold.

Design and apply by hand a dedicated branch and bound algorithm to the following instance of the 0-1 knapsack problem: values $v = [9, 4, 2, 3, 5, 3]$, weights $w = [7, 8, 4, 5, 9, 4]$ and capacity $W = 20$.

You will find useful the following observation that can be proven true: The relaxed knapsack problem can be efficiently solved with a greedy approach. Items are added one at a time by selecting the next item with the greatest ratio of value to weight. If there is enough remaining capacity, the item is fully assigned with $x_i = 1$. If not, a fractional value is assigned such that the remaining capacity is saturated and all remaining items have $x_i = 0$.

item:	1	2	3	4	5	6
value:	9	4	2	3	5	3
weight:	7	8	4	5	9	4
ratio:	9/7	4/8	2/4	3/5	5/9	3/4
ratio:	1.28	0.5	0.5	0.6	0.555	0.75

Exercise 5

Solve the following instance of the 0-1 knapsack problem: weights $w = [21, 11, 15, 9, 34, 25, 41, 52]$, values $v = [22, 12, 16, 10, 35, 26, 42, 53]$ and capacity $W = 60$.

Formulate the problem as a MILP and solve it in Python using `scipy.optimize.linprog`. Solve the problem both with real and integer variables and compare the results.

```
import numpy as np
from scipy import optimize
weights = np.array([21, 11, 15, 9, 34, 25, 41, 52])
values = np.array([22, 12, 16, 10, 35, 26, 42, 53])
```

Exercise 6 *

A medley relay is a team swimming event where each swimmer on the team swims a different stroke in a specific order: backstroke, breaststroke, butterfly, and freestyle. This event can vary in distance, typically ranging from 100 to 400 meters, depending on the competition. Consider the problem of selecting students for a swimming medley relay team. In Table 1 we show times for each swimming style of five students.

Student	backstroke	breaststroke	butterfly	freestyle
A	43.5	47.1	48.4	38.2
B	45.5	42.1	49.6	36.8
C	43.4	39.1	42.1	43.2
D	46.5	44.1	44.5	41.2
E	46.3	47.8	50.4	37.2

We need to choose a student for each of the four swimming styles such that the total relay time is minimized. Try first to do this task by hand. Then, formulate the problem as a MILP and solve it in Python. Finally, compare the solution with the one obtained by hand.