

AI505/AI801, Optimization – Exercise Sheet 10

2026-05-01

i Solutions included.

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.

Enumeration tries all designs. Each component can either be true or false, thus resulting in 2^n possible designs in the worst case. This problem has $2^3 = 8$ possible designs.

The Boolean satisfiability problem merely seeks a valid solution. As such, we set c to zero. The constraints are more interesting. As with all integer linear programs, x is constrained to be nonnegative and integral. Furthermore, we let 1 correspond to true and 0 correspond to false and introduce the constraint $\mathbf{x} \leq 1$. This is equivalent to impose that $\mathbf{x} \in \mathbb{B}^n$.

...

Exercise 2 $*$

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

From the simplex

$$A_B \mathbf{x}_B + A_N \mathbf{x}_N = \mathbf{b}$$

The variables not in basis are set to zero while for a feasible basis the value of the variables in basis can be calculated as:

$$\mathbf{x}_B = A_B^{-1}\mathbf{b}$$

All coefficients in A_B and \mathbf{b} are integer. Moreover,

$$\mathbf{A}_B^{-1} = \frac{1}{\det(\mathbf{A}_B)} \text{adj}(\mathbf{A}_B).$$

where adj is the adjugate of A_B , which is the transpose of the cofactor matrix C of A_B . The cofactor matrix of A_B is the $n \times n$ matrix C whose (i, j) entry is the (i, j) cofactor of A_B , which is the (i, j) -minor $M_{i,j}$ times a sign factor:

$$\mathbf{C} = ((-1)^{i+j}\mathbf{M}_{ij})_{1 \leq i, j \leq n}.$$

The element M_{ij} , is the (i, j) -minor of A_B and corresponds to the determinant of the $(n - 1) \times (n - 1)$ matrix that results from deleting row i and column j of A_B .

Hence, the numerator of A_B^{-1} is a matrix whose absolute values of its elements are products and subtractions of integer numbers, the elements of A_B , and must therefore be integers.

Hence, if the denominator of A_B^{-1} , ie, $|\det(A_B)|$ is an integer number, the values of A_B^{-1} will be integer and x_B integer. This is the case if the matrix A_B has a determinant equal to ± 1 .

Hence if all square submatrices of the matrix A have determinant equal to 0, +1, -1 the value of the basis variables will be integer.

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.

We first visualize the situation in Figure 1.

- For $x_1 = 0$ we obtain $\frac{1}{2}x_2 \leq 1$, that is, $x_2 \leq 2$
- For $x_2 = 0$ we obtain $\frac{2}{3}x_1 \leq 1$, that is, $x_1 \leq \frac{3}{2}$

Hence, the constraints has to pass through $[0, 2]$ and $[3/2, 0]$.

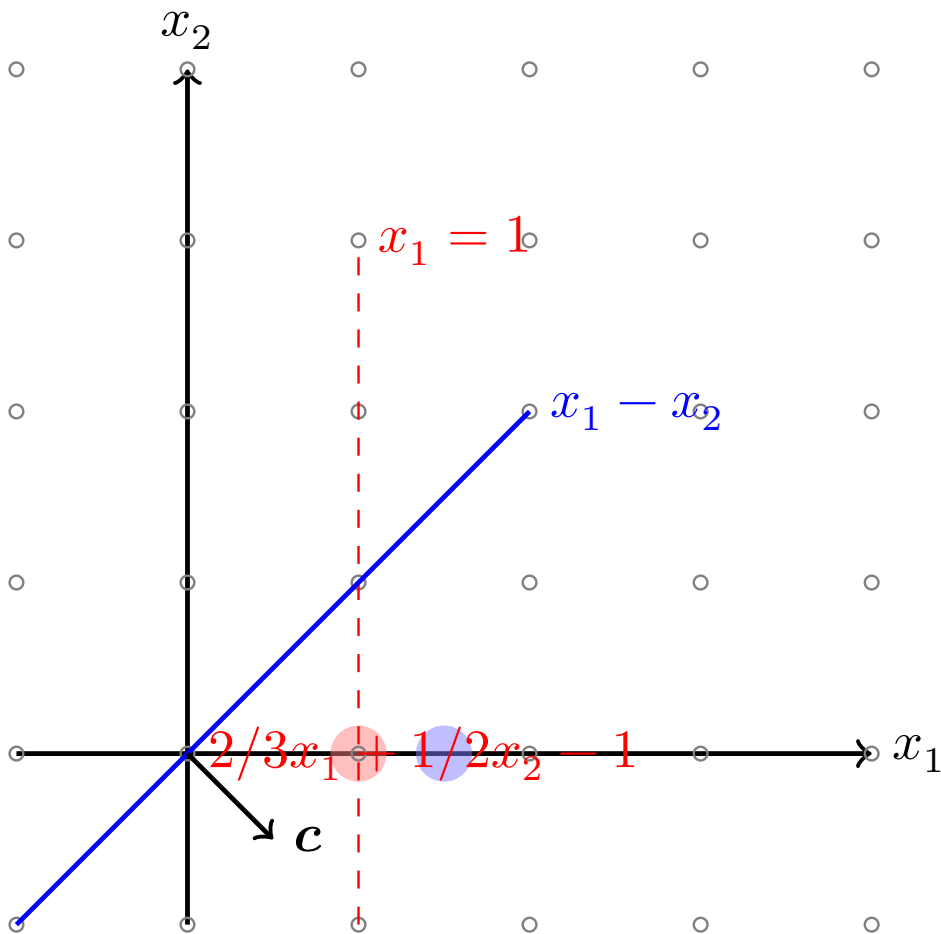


Figure 1: The situation in the exercise on the Gomory cut

One idea to derive a useful valid inequality (cut) is the following.

We first multiply the constraint by an arbitrary real number, say $3/2$:

$$\frac{3}{2} \left(\frac{2}{3}x_1 + \frac{1}{2}x_2 \right) \leq (1) \frac{3}{2} \quad \Rightarrow \quad x_1 + \frac{3}{4}x_2 \leq \frac{3}{2}$$

We can analyze and tighten the obtained constraint as follows

$$x_1 = \lfloor 1 \rfloor x_1 + \left\lfloor \frac{3}{4} \right\rfloor x_2 \leq x_1 + \frac{3}{4}x_2 \leq \frac{3}{2} = 1 + \frac{1}{2}$$

$$x_1 \leq 1 + \frac{1}{2} \text{ and } x_1 \text{ integer} \quad \Rightarrow \quad x_1 \leq 1$$

The constraint $x_1 \leq 1$ is a valid inequality because by the construction described it does not remove any feasible integer solution. It is also a useful inequality because it cuts away the current optimal solution of the linear relaxation. By chance then it will be the only inequality necessary to add to solve the problem since the linear relaxation of the original problem with this added constraint will give the optimal solution of the original problem, ie, $[1, 0]$.

A similar procedure underlays the development of the general formula for Chvatal-Gomory cuts given in the slides and repeated here:

$$x_b^* - \lfloor x_b^* \rfloor - \sum_{j \in N} (\bar{A}_{bj} - \lfloor \bar{A}_{bj} \rfloor) x_j \leq 0 \quad \bar{A} = A_B^{-1} A_N$$

To apply this formula we need to put the problem in equational standard form and apply the simplex algorithm.

$$\begin{aligned} & \max x_1 - x_2 \\ & \text{subject to: } \frac{2}{3}x_1 + \frac{1}{2}x_2 + x_3 = 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

$$A = \begin{bmatrix} \frac{2}{3} & \frac{1}{2} & 1 \end{bmatrix} \quad \mathbf{b} = 1$$

With hindsight, knowing from above that the optimal solution has value $[1.5, 0]$ and hence has x_1 in basis let's select as basis $B = [1]$. Applying the theory of the simplex we know that.

$$A_B = \begin{bmatrix} \frac{2}{3} \end{bmatrix} \quad A_N = \begin{bmatrix} \frac{1}{2} & 1 \end{bmatrix}$$

$$x_1^* = A_B^{-1}b = \frac{3}{2}1 = 1.5$$

and $x_2 = x_3 = 0$ because not in basis. Further,

$$\bar{A}_N = A_B^{-1}A_N = \begin{bmatrix} \frac{3}{4} & \frac{3}{2} \end{bmatrix}$$

and applying the Chvatal-Gomory formula given above:

$$\frac{3}{2} - \left\lfloor \frac{3}{2} \right\rfloor - \left[\left(\frac{3}{4} - \left\lfloor \frac{3}{4} \right\rfloor \right) x_2 + \left(\frac{3}{2} - \left\lfloor \frac{3}{2} \right\rfloor \right) x_3 \right] \leq 0$$

$$\frac{1}{2} - \frac{3}{4}x_2 - \frac{1}{2}x_3 \leq 0 \implies 3x_2 + 2x_3 \geq 2$$

We can put the inequality in terms of the original variables by substituting x_3 with $1 - \frac{2}{3}x_1 - \frac{1}{2}x_2$

$$3x_2 + 2 \left(1 - \frac{2}{3}x_1 - \frac{1}{2}x_2 \right) \geq 2$$

$$3x_2 + 2 - \frac{4}{3}x_1 - x_2 \geq 2 \implies 2x_2 - \frac{4}{3}x_1 \geq 0 \implies x_2 \geq \frac{2}{3}x_1$$

This is not a valid inequality. There must be an error in the calculations.

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.

This is an example to illustrate the use of the Highs library in scipy. The problem can be solved with `linprog` using the method “Highs” without the need for the integrality constraint. Why?