ORF307_HW8

April 24, 2023

ORF307 Homework 8

Due: Friday, May 5, 2023 9:00 pm ET

- Please export your code with output as pdf.
- If there is any additional answers, please combine them as **ONE** pdf file before submitting to the Gradescope.

Q1 A small integer programming problem

Consider the following integer programming problem:

$$\begin{array}{ll} \text{minimize} & -x_1 - 2x_2 \\ \text{subject to} & -3x_1 + 4x_2 \leq 4 \\ & 3x_1 + 2x_2 \leq 11 \\ & 2x_1 - x_2 \leq 5 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in \mathbf{Z}. \end{array}$$

Use a figure to answer the following questions:

1. What is the optimal cost of the linear programming relaxation? What is the optimal cost of the integer programming problem?
2. What is the convex hull of the set of all solutions to the integer programming problem?
3. Solve the problem by branch and bound. You can solve the linear programming relaxations graphically. Show the resulting tree.

Q2 Planning to move

It is the end of the semester and you might be planning to change accommodation. Moving is complicated and you want to use some integer programming tricks to make your life easier.

You rented a truck with capacity Q and you bought m boxes. Each box i has size b_i for $i = 1, \dots, m$. You have n items to move of size a_j for $j = 1, \dots, n$.

1. Formulate your packing problem as an integer programming problem to determine if your move is doable with the truck and the boxes you have.
2. Given the data below, is your move feasible? If so, what is the minimum number of boxes you need to use? Answer using a solver such as `GLPK_MI` or `GUROBI`.

   Note. If you find that `GLPK_MI` is taking too long to return a solution, please run it with the `tm_lim=30000` option which sets the time limit to 30 seconds: `problem.solve(solver=cp.GLPK_MI, tm_lim=30000)`. For this problem, the solution returned after 30 seconds is already optimal.

[ ]:
```
import numpy as np
n = 30 # Number of items
m = 10 # Number of boxes
Q = 100 # Truck capacity

# Item sizes
a = np.array([4, 3, 1, 2, 2,
              5, 4, 2, 4, 2,
              4, 1, 5, 2, 1,
              3, 5, 3, 3, 3,
              4, 5, 5, 2, 3,
              3, 3, 2, 2, 2])

# Box sizes
b = np.array([7, 19, 10, 14, 10, 12, 13, 10, 15, 14])
```