ECH4905 ChemE Optimization HW 4

Andres Espinosa

April 14, 2025

1 Problem 1

Consider the following integer programming problem:

$$\begin{aligned} \text{maximize} \quad & 1.2y_1 + y_2 \\ \text{subject to} \quad & y_1 + y_2 \leq 1 \\ & 1.2y_1 + 0.5y_2 \leq 1 \\ & y_1, y_2 \in \{0, 1\} \end{aligned}$$

1.1 Part a

Solve the first relaxed LP subproblem by hand using the simplex method and derive Gomory cuts based on the LP relaxation.

Solution: To tackle this problem we first relax the problem and then turn the problem above into the standard form so we can create a simplex tableau from it.

$$\begin{aligned} \text{maximize} \quad & 1.2y_1 + y_2 \\ \text{subject to} \quad & y_1 + y_2 + s_1 = 1 \\ & 1.2y_1 + 0.5y_2 + s_2 = 1 \\ & y_1 + s_3 = 1 \\ & y_2 + s_4 = 1 \\ & y_1, y_2, s_1, s_2, s_3, s_4 \geq 0 \end{aligned}$$

In matrix notation,

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \succeq 0 \end{aligned}$$

where

$$\mathbf{c} = \begin{bmatrix} -1.2 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1.2 & 0.5 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} y_1 \\ y_2 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$$

(with a flipped objective component)
The initial simplex tableau for the problem is as follows:

Basic Var	y_1	y_2	s_1	s_2	s_3	s_4	RHS	α
s_1	1	1	1	0	0	0	1	
s_2	1.2	0.5	0	1	0	0	1	
s_3	1	0	0	0	1	0	1	
s_4	0	1	0	0	0	1	1	
obj	-1.2	-1	0	0	0	0	$-$	$-$

we can define the slack variables equal to the right hand side, and this is in turn a basic feasible solution, so we can jump into phase 2.
We select the y_1 as the entering variable and calculate the alpha value for each basic variable

Basic Var	y_1	y_2	s_1	s_2	s_3	s_4	RHS	α
s_1	1	1	1	0	0	0	1	$\frac{1}{1}$
s_2	1.2	0.5	0	1	0	0	1	$\frac{1}{1.2}$
s_3	1	0	0	0	1	0	1	$\frac{1}{1}$
s_4	0	1	0	0	0	1	1	$\frac{1}{0}$
obj	-1.2	-1	0	0	0	0	$-$	$-$

We pivot this on the 1st column (y_1) and the 2nd row (s_2)

Basic Var	y_1	y_2	s_1	s_2	s_3	s_4	RHS	α
s_1	0	$\frac{7}{12}$	1	$-\frac{5}{6}$	0	0	$\frac{1}{6}$	
y_1	1	$\frac{5}{12}$	0	$\frac{5}{6}$	0	0	$\frac{5}{6}$	
s_3	0	$-\frac{5}{12}$	0	$-\frac{5}{6}$	1	0	$\frac{1}{6}$	
s_4	0	1	0	0	0	1	1	
obj	0	-0.5	$-$	$-$	$-$	$-$	$-$	$-$

With blands rule, we pick y_2 and calculate the alpha value for each basic variable.

Basic Var	y_1	y_2	s_1	s_2	s_3	s_4	RHS	α
s_1	0	$\frac{7}{12}$	1	$-\frac{5}{6}$	0	0	$\frac{1}{6}$	$\frac{2}{7}$
y_1	1	$\frac{5}{12}$	0	$\frac{5}{6}$	0	0	$\frac{5}{6}$	$\frac{2}{1}$
s_3	0	$-\frac{5}{12}$	0	$-\frac{5}{6}$	1	0	$\frac{1}{6}$	$-\frac{1}{5}$
s_4	0	1	0	0	0	1	1	$\frac{1}{1}$
obj	0	-0.5	$-$	$-$	$-$	$-$	$-$	$-$

We pivot on the 2nd column (y_2) and the 1st row (s_1).

Basic Var	y_1	y_2	s_1	s_2	s_3	s_4	RHS	α
y_2	0	1	$\frac{12}{7}$	$-\frac{10}{7}$	0	0	$\frac{2}{7}$	
y_1	1	0	$-\frac{5}{7}$	$\frac{60}{42}$	0	0	$\frac{30}{42}$	
s_3	0	0	$\frac{5}{7}$	$-\frac{60}{42}$	1	0	$\frac{12}{42}$	
s_4	0	0	$-\frac{12}{7}$	$\frac{60}{42}$	0	1	$\frac{30}{42}$	
obj	0	0	0.857	0.286	0	0	0	-1.143

This is the optimal solution to the LP relaxed problem. Now we will derive Gomory cuts from this LP relaxed problem. Since each constraint has a non-integer solution, we can generate a Gomory cut on each constraint.

$$
\begin{aligned}
y_2 + \text{floor}(\frac{12}{7})s_1 + \text{floor}(\frac{-10}{7})s_2 &\le \text{floor}(\frac{2}{7}) \\
y_1 + \text{floor}(\frac{-5}{7})s_1 + \text{floor}(\frac{10}{7})s_2 &\le \text{floor}(\frac{5}{7}) \\
s_3 + \text{floor}(\frac{5}{7})s_1 + \text{floor}(\frac{-10}{7})s_2 &\le \text{floor}(\frac{2}{7}) \\
s_4 + \text{floor}(\frac{-12}{7})s_1 + \text{floor}(\frac{10}{7})s_2 &\le \text{floor}(\frac{5}{7})
\end{aligned}
$$

These turn into the cuts

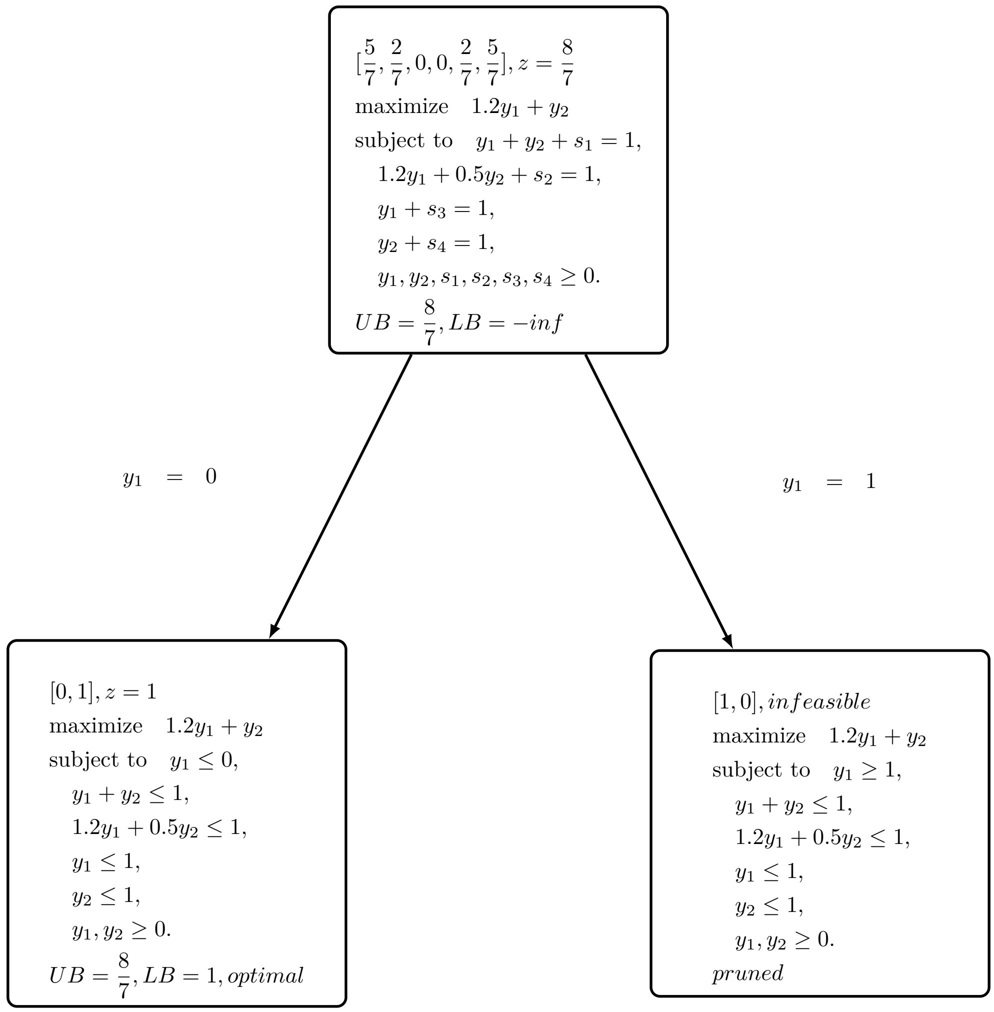
$$
\begin{aligned}
y_2 + s_1 - 2s_2 &\le 0 \\
y_1 - s_1 + s_2 &\le 0 \\
s_3 - 2s_2 &\le 0 \\
s_4 - 2s_1 + s_2 &\le 0
\end{aligned}
$$

1.2 Part b

Solve the above problem with the branch and bound method by enumerating nodes in the tree and solving the LP subproblems using GAMS.
Solution: The initial LP relaxed problem is solved in 1.1, so we can start with the parent node. An important note for this question, I will be solving the LPs in my custom `gatorpy` LP solver so that I can use them as verification tests. The code used will be available in section 6.1

$$
\boxed{
\begin{aligned}
&[\frac{5}{7}, \frac{2}{7}, 0, 0, \frac{2}{7}, \frac{5}{7}], z = \frac{8}{7} \\
&\text{maximize} \quad 1.2y_1 + y_2 \\
&\text{subject to} \quad y_1 + y_2 + s_1 = 1, \\
&\quad 1.2y_1 + 0.5y_2 + s_2 = 1, \\
&\quad y_1 + s_3 = 1, \\
&\quad y_2 + s_4 = 1, \\
&\quad y_1, y_2, s_1, s_2, s_3, s_4 \ge 0. \\
&UB = \frac{8}{7}, LB = -inf
\end{aligned}
}
$$

Since all variables are fractional, we can pick the first one y_1 to branch on



Via the branch and bound method, the optimal solution to the problem

$$\begin{aligned} \text{maximize} \quad & 1.2y_1 + y_2 \\ \text{subject to} \quad & y_1 + y_2 \leq 1 \\ & 1.2y_1 + 0.5y_2 \leq 1 \\ & y_1, y_2 \in \{0, 1\} \end{aligned}$$

is $y_1 = 0, y_2 = 1, z = 1$.

2 Problem 2

3 Problem 2

Consider the following superstructure for the separation of four chemical components using sharp distillation columns. The total cost of a distillation column is calculated as follows:

$$\text{cost}_k = \alpha_k + \beta_k F_k + \gamma_{\text{Hot}} Q_k^{\text{Hot}} + \gamma_{\text{Cold}} Q_k^{\text{Cold}}$$

where:

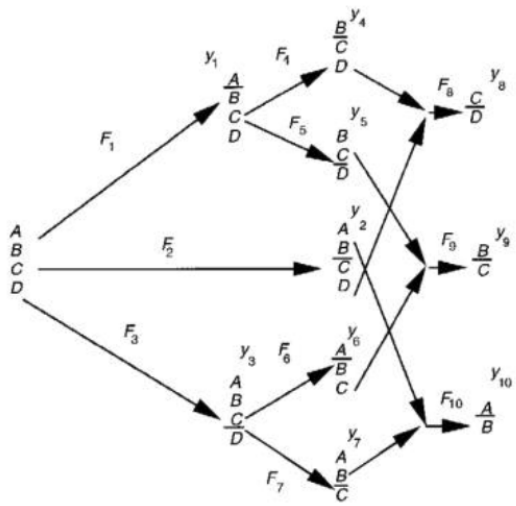


Figure 1: Problem 2 superstructure

- α_k represents a fixed capital cost,
- β_k represents the variable investment cost,
- $\gamma_{\text{Hot/Cold}}$ is the cost of hot/cold utilities, and
- $Q_k^{\text{Hot}}/Q_k^{\text{Cold}}$ is the total demand of hot and cold utilities (assumed to be equal).

Given:

- Initial feed: 1000 Kmol/h,
- Feed composition (mole fraction): A = 0.15, B = 0.3, C = 0.35, D = 0.2.

and the following data:

k	Separator	Investment cost: α_k, fixed (10^3\$/yr)	Investment cost: β_k, variable (10^3\$hr/kmol yr)	Heat duty coefficients, K_k, (10^6kJ /kgmol)
1	*A/BCD*	145	0.42	0.028
2	*AB/CD*	52	0.12	0.042
3	*ABC/D*	76	0.25	0.054
6	*A/BC*	125	0.78	0.024
7	*AB/C*	44	0.11	0.039
4	*B/CD*	38	0.14	0.040
5	*BC/D*	66	0.21	0.047
10	*A/B*	112	0.39	0.022
9	*B/C*	37	0.08	0.036
8	*C/D*	58	0.19	0.044

Cost of utilities:

Cooling water $C_C = 1.3$ (10^3\$/10^6kJyr)

Steam $C_H = 34$ (10^3\$/10^6kJyr)

Figure 2: Problem 2 Data

3.1 Part A: MILP Formulation

Formulate a Mixed-Integer Linear Programming (MILP) problem to find the optimal sequence of distillation columns. Identify at least 2 logicx based equations that can be formulated to tighten the problem formulation.

3.2 Part B: Solve Using GAMS

Solve the problem using GAMS.

3.3 Part C: Integer Cut

Once the solution is found:

- Identify the active binary variables in the optimal solution.
- Formulate an integer cut to exclude this solution from the feasible space.
- Solve the problem again to find the next best solution.

4 Problem 3

5 Problem 3

Given are three candidate reactors for the reaction $A \rightarrow B$, where we would like to produce 10 kmol/h of B. Up to 15 kmol/hr of reactant A are available at a price of \$2/kmol. The data on the three reactors is as follows:

Reactor	**Conversion**	**Cost**
1	0.8	$8 + 1.5 \cdot \text{Feed}$
2	0.667	$5.4 + \text{Feed}$
3	0.555	$2.7 + 0.5 \cdot \text{Feed}$

Table 1: Reactor Data

5.1 Part A: Superstructure Design

Design a superstructure to represent this problem.

5.2 Part B: MILP Formulation

Determine a MILP formulation.

5.3 Part C: Solve Using GAMS

Solve in GAMS.

6 Code

6.1 Problem 1 Code

```
# Parameters
A_arr = np.array([[1,1],[1.2,0.5]])
b_arr = np.array([1,1])
c_arr = np.array([1.2,1])
A = Parameter(A_arr)
b = Parameter(b_arr)
c = Parameter(c_arr)
# Variables
y = Variable(2)
# Problem
problem = Problem({
    'maximize': c.T @ y,
    'subject to': [
        A @ y <= b,
        y >= 0,
        y <= 1
    ]
})

# CVX Vars
y_cvx = cp.Variable(2)
# CVX Problem
objective = cp.Maximize(c_arr @ y_cvx)
constraints = [
    A_arr @ y_cvx <= b_arr,
    y_cvx >= 0,
    y_cvx <= 1
]
problem_cvx = cp.Problem(objective, constraints)

get_test_results(problem, problem_cvx, "HW5 Problem 1 initial LP relaxation")
# branch and bound arrays
I_y_1_arr = np.array([[1,0],[0,0]])
I_y_2_arr = np.array([[0,0],[0,1]])
I_y_1 = Parameter(I_y_1_arr)
I_y_2 = Parameter(I_y_2_arr)
# LEFT SPLIT
# Problem
problem = Problem({
    'maximize': c.T @ y,
    'subject to': [
        A @ y <= b,
        y >= 0,
        y <= 1,
```

```
        I_y_1 @ y <= 0,
    ]
})

# CVX Vars
y_cvx = cp.Variable(2)
# CVX Problem
objective = cp.Maximize(c_arr @ y_cvx)
constraints = [
    A_arr @ y_cvx <= b_arr,
    y_cvx >= 0,
    y_cvx <= 1,
    I_y_1_arr @ y_cvx <= 0
]
problem_cvx = cp.Problem(objective, constraints)

get_test_results(problem, problem_cvx, "HW5 Problem 1 Branch and Bound Left Split y_1 <= 0")

# RIGHT SPLIT
# Problem
problem = Problem({
    'maximize': c.T @ y,
    'subject to': [
        A @ y <= b,
        y >= 0,
        y <= 1,
        I_y_1 @ y >= 1,
    ]
})

# CVX Vars
y_cvx = cp.Variable(2)
# CVX Problem
objective = cp.Maximize(c_arr @ y_cvx)
constraints = [
    A_arr @ y_cvx <= b_arr,
    y_cvx >= 0,
    y_cvx <= 1,
    I_y_1_arr @ y_cvx >= 1
]
problem_cvx = cp.Problem(objective, constraints)

get_test_results(problem, problem_cvx, "HW5 Problem 1 Branch and Bound Right Split y_1 >=1")

Test ID: HW5 Problem 1 initial LP relaxation
CVX: (np.float64(1.14), array([[0.71, 0.29]]), True)
GatORPy: (array(1.14285714), array([0.71428571, 0.28571429, 0.        , 0.        , 0.2857142
       0.71428571]), True)
```

```
Test passed: False

Test ID: HW5 Problem 1 Branch and Bound Left Split y_1 <= 0
CVX: (np.float64(1.0), array([[0., 1.]]), True)
GatORPy: (array(1.), array([0. , 1. , 0. , 0.5, 1. , 0. , 0. , 0. ]), True)
Test passed: False

Test ID: HW5 Problem 1 Branch and Bound Right Split y_1 >=1
CVX: (None, None, False)
GatORPy: (array(0.8), array([ 1. , -0.4,  0.4,  0. ,  0. ,  1.4,  0. , -1. ]), True)
Test passed: False
```