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.

$$y_2 + \text{floor}(\frac{12}{7})s_1 + \text{floor}(\frac{-10}{7})s_2 \leq \text{floor}(\frac{2}{7})$$
$$y_1 + \text{floor}(\frac{-5}{7})s_1 + \text{floor}(\frac{10}{7})s_2 \leq \text{floor}(\frac{5}{7})$$
$$s_3 + \text{floor}(\frac{5}{7})s_1 + \text{floor}(\frac{-10}{7})s_2 \leq \text{floor}(\frac{2}{7})$$
$$s_4 + \text{floor}(\frac{-12}{7})s_1 + \text{floor}(\frac{10}{7})s_2 \leq \text{floor}(\frac{5}{7})$$

These turn into the cuts

$$y_2 + s_1 - 2s_2 \leq 0$$
$$y_1 - s_1 + s_2 \leq 0$$
$$s_3 - 2s_2 \leq 0$$
$$s_4 - 2s_1 + s_2 \leq 0$$

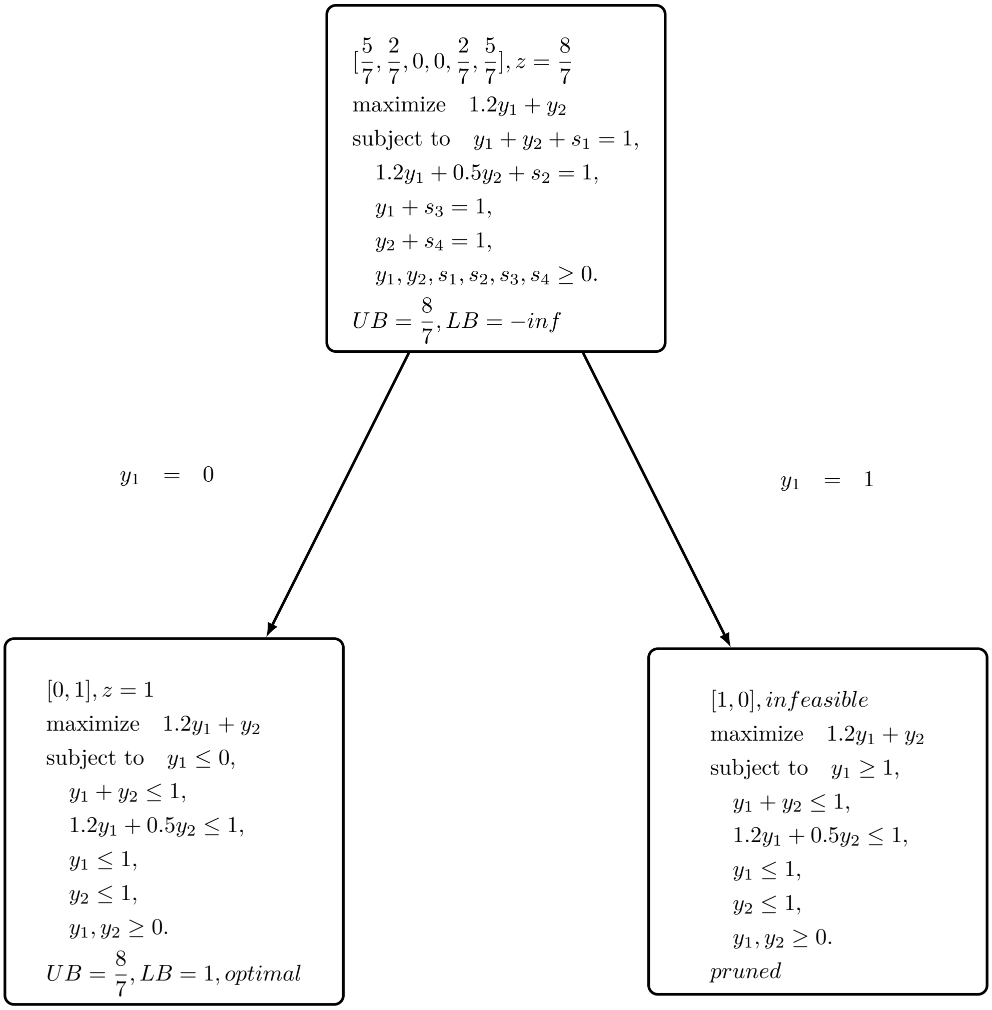
1.2 Part b

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 4.1

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

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

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:

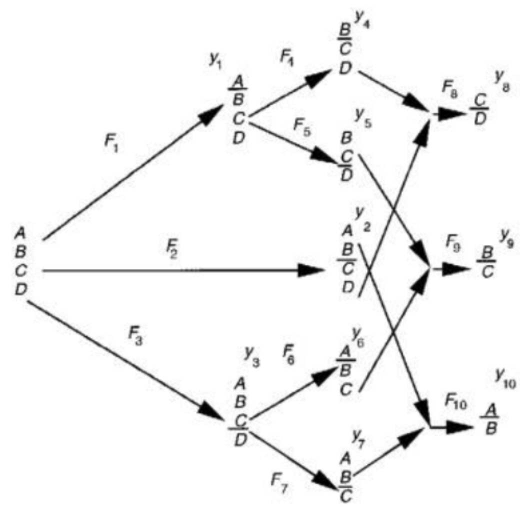


Figure 1: Problem 2 superstructure

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

where:

- 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:

2.1 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.
Solution: In order to solve this problem, I will first identify the variables and model the binary flow and decision logic. Then, I will attempt to solve the problem while ignoring the Hot/Cold variables since I am much less confident on implementing that than the other sections. After getting a solution working while ignoring the hot/cold components, I will then implement those γ, Q parameters into the problem.

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

2.1.1 Binary logic

To solve this, I will start by creating a series of statements that must be true for this flow to work. I will be using the same y variables for each distillation column that can be seen in the superstructure diagram 1.

- One of y_1, y_2, y_3 must be chosen.
- If y_1 is chosen, then either y_4 xor y_5 must be chosen.
- If y_2 is chosen, then both y_8 and y_{10} must be chosen.
- If y_3 is chosen, then either y_6 xor y_7 must be chosen.
- If y_4 is chosen, then y_8 must be chosen.
- If y_5 is chosen, then y_9 must be chosen.
- If y_6 is chosen, then y_9 must be chosen.
- If y_7 is chosen, then y_{10} must be chosen.

I believe this to be a sufficient set of logic statements to model the problem of choosing a distillation problem. Getting rid of the implications:

- One of y_1, y_2, y_3 must be chosen.
- $y_1 = 0$ or either y_4 xor y_5 must be chosen.
- $y_2 = 0$ or both y_8 and y_{10} must be chosen.
- $y_3 = 0$ or either y_6 xor y_7 must be chosen.
- $y_4 = 0$ or y_8 must be chosen.

- $y_5 = 0$ or y_9 must be chosen.
- $y_6 = 0$ or y_9 must be chosen.
- $y_7 = 0$ or y_{10} must be chosen.

We can then translate these second parts into different clauses

$$\begin{aligned}
\text{One of } y_1, y_2, y_3 \text{ must be chosen.} \quad &\rightarrow & y_1 + y_2 + y_3 = 1 \\
\text{Either } y_4 \text{ xor } y_5 \text{ must be chosen.} \quad &\rightarrow & y_4 + y_5 = 1 \\
\text{both } y_8 \text{ and } y_{10} \text{ must be chosen.} \quad &\rightarrow & y_8 = 1 \cap y_{10} = 1 \\
\text{Either } y_6 \text{ xor } y_7 \text{ must be chosen.} \quad &\rightarrow & y_6 + y_7 = 1 \\
y_8 \text{ must be chosen.} \quad &\rightarrow & y_8 = 1 \\
y_9 \text{ must be chosen.} \quad &\rightarrow & y_9 = 1 \\
y_9 \text{ must be chosen.} \quad &\rightarrow & y_9 = 1 \\
y_{10} \text{ must be chosen.} \quad &\rightarrow & y_{10} = 1
\end{aligned}$$

which can in turn be converted to

$$\begin{aligned}
y_1 + y_2 + y_3 = 1 \\
y_1 &= 0 \cup y_4 + y_5 = 1 \\
y_2 &= 0 \cup (y_8 = 1 \cap y_{10} = 1) \\
y_3 &= 0 \cup y_6 + y_7 = 1 \\
y_4 &= 0 \cup y_8 = 1 \\
y_5 &= 0 \cup y_9 = 1 \\
y_6 &= 0 \cup y_9 = 1 \\
y_7 &= 0 \cup y_{10} = 1
\end{aligned}$$

We can subtract each variable on the left from 1 so we can add them together and distribute the and operation out for the third.

$$\begin{aligned}
y_1 + y_2 + y_3 = 1 \\
(1 - y_1 = 1) \cup (((y_4 = 1) \cup (y_5 = 1)) \cap ((1 - y_4 = 1) \cup (1 - y_5 = 1))) \\
((1 - y_2 = 1) \cup (y_8 = 1)) \cap ((1 - y_2 = 1) \cup (y_{10} = 1)) \\
(1 - y_3 = 1) \cup (((y_6 = 1) \cup (y_7 = 1)) \cap ((1 - y_6 = 1) \cup (1 - y_7 = 1))) \\
1 - y_3 = 1 \cup y_6 + y_7 = 1 \\
1 - y_4 = 1 \cup y_8 = 1 \\
1 - y_5 = 1 \cup y_9 = 1 \\
1 - y_6 = 1 \cup y_9 = 1 \\
1 - y_7 = 1 \cup y_{10} = 1
\end{aligned}$$

We turn this into the equivalent equations

$$
\begin{aligned}
&y_1 + y_2 + y_3 = 1 \\
&\quad (1 - y_1 = 1) \cup ((y_4 = 1 \cup y_5 = 1)) \\
&\quad (1 - y_1 = 1) \cup ((1 - y_4 = 1) \cup (1 - y_5 = 1)) \\
&1 - y_2 + y_8 \geq 1 \\
&1 - y_2 + y_{10} \geq 1 \\
&\quad (1 - y_3 = 1) \cup (y_6 = 1 \cup y_7 = 1) \\
&\quad (1 - y_3 = 1) \cup ((1 - y_6 = 1) \cup (1 - y_7 = 1)) \\
&1 - y_4 + y_8 \geq 1 \\
&1 - y_5 + y_9 \geq 1 \\
&1 - y_6 + y_9 \geq 1 \\
&1 - y_7 + y_{10} \geq 1
\end{aligned}
$$

Finally, we can bring them all into pure math below

$$
\begin{aligned}
y_1 + y_2 + y_3 &= 1 \\
1 - y_1 + y_4 + y_5 &\geq 1 \\
1 - y_1 + 1 - y_4 + 1 - y_5 &\geq 1 \\
1 - y_2 + y_8 &\geq 1 \\
1 - y_2 + y_{10} &\geq 1 \\
1 - y_3 + y_6 + y_7 &\geq 1 \\
1 - y_3 + 1 - y_6 + 1 - y_7 &\geq 1 \\
1 - y_4 + y_8 &\geq 1 \\
1 - y_5 + y_9 &\geq 1 \\
1 - y_6 + y_9 &\geq 1 \\
1 - y_7 + y_{10} &\geq 1
\end{aligned}
$$

Note: These equations are not completely sufficient on themselves, they require that each distillation column has a positive cost component. Otherwise, it could be possible for y_8 and y_7 to be chosen, but it isn't necessary since the optimization should only yield a solution where distillation columns are used if they are needed to.

2.2 Model w/o HotCold

After having solved the binary logic problems above, I will start by denoting the parameters for the model. This model is actually not too bad (unless I am missing the importance of the feed composition asides from flow removal).

Parameters: This problem has a feed composition parameter set $f_A = 0.15, f_B, 0.30, f_C = 0.35, f_D = 0.2$. (Important note: I am using x for the flow, so f is a parameter for feed composition). There is the input supply $S = 1000$. Fixed capital cost α_k for each $k \in [1, \ldots, 10]$ distillation column and variable cost β_k.

Variables: I am using the continuous variables $x_i \in [1, \ldots, 13]$ to represent the flow from column to column. Below is a table that maps the variable index to each flow I am also using variables y_k, $k \in [1, \ldots, 10]$ to denote the binary choice of using distillation column k.

Index	Flow Description
1	Feed to Column 1
2	Feed to Column 2
3	Feed to Column 3
4	Output from Column 1 to Column 4
5	Output from Column 1 to Column 5
6	Output from Column 2 to Column 8
7	Output from Column 2 to Column 10
8	Output from Column 3 to Column 6
9	Output from Column 3 to Column 7
10	Output from Column 4 to Column 8
11	Output from Column 5 to Column 9
12	Output from Column 6 to 9
13	Output from Column 7 to 10

Table 1: Mapping of variable indices to flow descriptions.

Constraints: We can model our constraints as follows:

$$
\begin{aligned}
y_k & \quad \text{See above for distillation constraints} \\
x_1, x_4, x_5 \leq Sy_1 & \quad \text{BigM Column 1} \\
x_2, x_6, x_7 \leq Sy_2 & \quad \text{BigM Column 2} \\
x_3, x_8, x_9 \leq Sy_3 & \quad \text{BigM Column 3} \\
x_4, x_{10} \leq Sy_4 & \quad \text{BigM Column 4} \\
x_5, x_{11} \leq Sy_5 & \quad \text{BigM Column 5} \\
x_8, x_{12} \leq Sy_6 & \quad \text{BigM Column 6} \\
x_9, x_{13} \leq Sy_7 & \quad \text{BigM Column 7} \\
x_6, x_{10} \leq Sy_8 & \quad \text{BigM Column 8} \\
x_{11}, x_{12} \leq Sy_9 & \quad \text{BigM Column 9} \\
x_7, x_{13} \leq Sy_{10} & \quad \text{BigM Column 10} \\
x_{13} = \frac{1-f_c}{1-f_d} x_9 \text{Column 7 Flow} & \\
x_{12} = \frac{1-f_a}{1-f_d} x_8 \text{Column 6 Flow} & \\
x_{11} = \frac{1-f_d}{1-f_a} x_5 \text{Column 5 Flow} & \\
x_{10} = \frac{1-f_b}{1-f_a} x_4 \text{Column 4 Flow} & \\
x_9 + x_8 = (1-f_d) x_3 \text{Column 3 Flow} & \\
x_7 = (1-f_a-f_b) x_2 \text{Column 2 Lower Flow} & \\
x_6 = (1-f_c-f_d) x_2 \text{Column 2 Upper Flow} & \\
x_5 + x_4 = (1-f_a) x_1 \text{Column 1 Flow} & \\
x_1 + x_2 + x_3 = S \text{Initial Flow} &
\end{aligned}
$$

Objective: We can model the objective function via components. For brevity, I will just show the component for a general distillation column.

$$z_k = \alpha_k y_k + \beta_k * x_k + heat$$

where x_k is the entering flow.

2.2.1 Tightening constraints

This problem assumes that binary variables will be pushed to zero since they have a positive component in the objective. We can tighten this assumption by introducing constraints that bind parent flows to nested children flows. So, for example, $x_1 3$ will not be active if x_3 is not active, however we do not have a constraint that directly models that relationship. We can introduce these two logic-based constraints to tighten the problem:

$$1 - y_3 + y_{10} \leq 1$$
$$1 - y_2 + y_9 \geq 1$$

2.3 Part B: Solve Using GAMS

Solve the problem using GAMS.
Solution: A solution was found and summarized in table 2. Code is available in section 4.2.

2.4 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.

I add an integer cut $y_3 = 0$ and solve to get a worse optimal solution of:

Variable	**Value**
z	581.0000
x_1	0
x_2	0
x_3	1000.0000
x_4	0
x_5	0
x_6	0
x_7	0
x_8	800.0000
x_9	0
x_{10}	0
x_{11}	0
x_{12}	850.0000
x_{13}	0
y_1	0
y_2	0
y_3	1
y_4	0
y_5	0
y_6	0
y_7	0
y_8	1
y_9	0
y_{10}	0

Table 2: Optimal solution values for the variables.

Variable	**Value**
z	622.0000
x_1	0
x_2	1000.0000
x_3	0
x_4	0
x_5	0
x_6	450.0000
x_7	550.0000
x_8	0
x_9	0
x_{10}	0
x_{11}	0
x_{12}	0
x_{13}	0
y_1	0
y_2	1.0000
y_3	0
y_4	0
y_5	0
y_6	0
y_7	0
y_8	1.0000
y_9	0
y_{10}	1.0000

Table 3: Summary of problem with integer cut.

3 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 4: Reactor Data

3.1 Part A: Superstructure Design

Design a superstructure to represent this problem.
Solution: Not sure if I am oversimplifying this. This superstructure assumes that no A is recycled

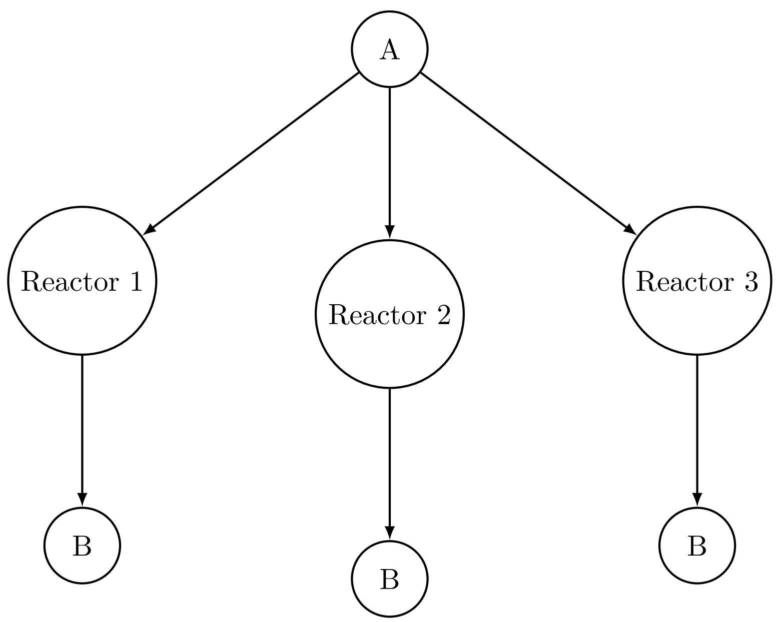


Figure 3: Superstructure for the reaction $A \rightarrow B$ with three reactors.

back into the reactor, and we only choose one reactor. This is generally a simple problem, we are effectively picking which reactor to use to minimize the cost.

3.2 Part B: MILP Formulation

Determine a MILP formulation.
Solution: In order to solve this MILP formulation, I will start thinking about this in a GAMS/modeling language way starting with parameters.
Parameters: We have parameters k_i as the $A \rightarrow B$ conversion of reactor i. We also have cost components d_i as the constant cost of using reactor i and c_i as the multiplicative cost of feeding A through reactor i. Our demand parameter D for the amount of B we would like to produce and

our supply S that we have available. We can also have the price of using a kmol of A as p. Below are the numerical representations of the problem parameters:

$$k_1 = \frac{4}{5}, k_2 = \frac{2}{3}, k_3 = \frac{5}{9}$$
$$d_1 = 8, d_2 = 5.4, d_3 = 2.7; c_1 = 1.5, c_2 = 1, c_3 = 0.5$$
$$D = 10, S = 15, p = 2$$

Variables: Our variables for this problem are as follows: We have x_i as the amount of A delivered to the i-th reactor. In order to model our decision to pick a reactor, we have variables y_i which are booleans that signify if reactor i has been chosen.
Constraints: We can model our constraints as follows:

$$\begin{aligned}
x_i \leq S, \quad \forall i \in [1,2,3] & \qquad \text{Supply Constraint} \\
\sum_{i=1}^{3} k_i x_i \geq D & \qquad \text{Demand Constraint} \\
\sum_{i=1}^{3} y_i = 1 & \qquad \text{One Reactor Constraint} \\
x_i \leq S y_i, \quad \forall i \in [1,2,3] & \qquad \text{Big-M One Reactor Flow Constraint} \\
x_i \geq 0, \quad \forall i \in [1,2,3] & \qquad \text{Non-negativity} \\
y_i \in \{0,1\} \quad \forall i \in [1,2,3] & \qquad \text{Binaries}
\end{aligned}$$

We can also get rid of the supply constraint since the Big-M naturally handles that. The GAMS solution below drops the supply constraint. *Objective:* Our objective in this problem is to minimize the cost while still meeting the demand. We can bunch these costs into different components

$$\begin{aligned}
p\sum_{i=1}^{3} x_i & \quad \text{Supply component} \\
\sum_{i=1}^{3} d_i y_i & \quad \text{Reactor constant component} \\
\sum_{i=1}^{3} c_i y_i & \quad \text{Reactor processing component}
\end{aligned}$$

3.3 Part C: Solve Using GAMS

Solve in GAMS.
Solution: The optimal solution was found to be $x_1 = 12.5, y_1 = 1, z = 34.5$, and the rest of the variables equal to 0. GAMS code available in section 4.3

4 Code

4.1 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.        , 0.285714
       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
```

4.2 Problem 2 Code

```
  GAMS 49.1.0  5c4d4ed6 Feb 15, 2025          DAX-DAC arm 64bit/macOS - 04/14/25 22:09:18 Page
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
C o m p i l a t i o n


* I know this is terribly verbose and I should be using sets
* but I haven’t been able to get that logic to work
Scalars
f_a /0.15/
f_b /0.30/
f_c /0.35/
f_d /0.20/
S /1000/
    a_1 / 145 /, b_1 / 0.42 /,
    a_2 / 52  /, b_2 / 0.12 /,
    a_3 / 76  /, b_3 / 0.25 /,
    a_4 / 125 /, b_4 / 0.78 /,
    a_5 / 44  /, b_5 / 0.11 /,
    a_6 / 38  /, b_6 / 0.14 /,
    a_7 / 66  /, b_7 / 0.21 /,
    a_8 / 112 /, b_8 / 0.39 /,
    a_9 / 37  /, b_9 / 0.08 /,
    a_10 / 58 /, b_10 / 0.19 /
;
Variables
z
;
Positive Variables
x_1
x_2
x_3
x_4
x_5
x_6
```

```
x_7
x_8
x_9
x_10
x_11
x_12
x_13
;
Binary Variables
y_1
y_2
y_3
y_4
y_5
y_6
y_7
y_8
y_9
y_10
;
Equations
    bin_constraint_1
    bin_constraint_2
    bin_constraint_3
    bin_constraint_4
    bin_constraint_5
    bin_constraint_6
    bin_constraint_7
    bin_constraint_8
    bin_constraint_9
    bin_constraint_10
    bin_constraint_11
    bigm_constraint_1_1
    bigm_constraint_1_4
    bigm_constraint_1_5
    bigm_constraint_2_2
    bigm_constraint_2_6
    bigm_constraint_2_7
    bigm_constraint_3_3
    bigm_constraint_3_8
    bigm_constraint_3_9
    bigm_constraint_4_4
    bigm_constraint_4_10
    bigm_constraint_5_5
    bigm_constraint_5_11
    bigm_constraint_6_8
    bigm_constraint_6_12
    bigm_constraint_7_9
```

```
    bigm_constraint_7_13
    bigm_constraint_8_6
    bigm_constraint_8_10
    bigm_constraint_9_11
    bigm_constraint_9_12
    bigm_constraint_10_7
    bigm_constraint_10_13
    flow_constraint_7
    flow_constraint_6
    flow_constraint_5
    flow_constraint_4
    flow_constraint_3
    flow_constraint_2_lower
    flow_constraint_2_upper
    flow_constraint_1
    supply_constraint
    objective_eq
;
bin_constraint_1..      y_1 + y_2 + y_3 =e= 1;
bin_constraint_2..      1 - y_1 + y_4 + y_5 =g= 1;
bin_constraint_3..      1 - y_1 + 1 - y_4 + 1 - y_5 =g= 1;
bin_constraint_4..      1 - y_2 + y_8 =g= 1;
bin_constraint_5..      1 - y_2 + y_10 =g= 1;
bin_constraint_6..      1 - y_3 + y_6 + y_7 =g= 1;
bin_constraint_7..      1 - y_3 + 1 - y_6 + 1 - y_7 =g= 1;
bin_constraint_8..      1 - y_4 + y_8 =g= 1;
bin_constraint_9..      1 - y_5 + y_9 =g= 1;
bin_constraint_10..     1 - y_6 + y_9 =g= 1;
bin_constraint_11..     1 - y_7 + y_10 =g= 1;

bigm_constraint_1_1..  x_1  =l= S * y_1;
bigm_constraint_1_4..  x_4  =l= S * y_1;
bigm_constraint_1_5..  x_5  =l= S * y_1;

bigm_constraint_2_2..  x_2  =l= S * y_2;
bigm_constraint_2_6..  x_6  =l= S * y_2;
bigm_constraint_2_7..  x_7  =l= S * y_2;

bigm_constraint_3_3..  x_3  =l= S * y_3;
bigm_constraint_3_8..  x_8  =l= S * y_3;
bigm_constraint_3_9..  x_9  =l= S * y_3;

bigm_constraint_4_4..  x_4  =l= S * y_4;
bigm_constraint_4_10.. x_10 =l= S * y_4;

bigm_constraint_5_5..  x_5  =l= S * y_5;
bigm_constraint_5_11.. x_11 =l= S * y_5;

```

```
bigm_constraint_6_8..  x_8  =l= S * y_6;
bigm_constraint_6_12.. x_12 =l= S * y_6;

bigm_constraint_7_9..  x_9  =l= S * y_7;
bigm_constraint_7_13.. x_13 =l= S * y_7;

bigm_constraint_8_6..  x_6  =l= S * y_8;
bigm_constraint_8_10.. x_10 =l= S * y_8;

bigm_constraint_9_11.. x_11 =l= S * y_9;
bigm_constraint_9_12.. x_12 =l= S * y_9;

bigm_constraint_10_7.. x_7  =l= S * y_10;
bigm_constraint_10_13.. x_13 =l= S * y_10;

flow_constraint_7..        x_13 =e= (1 - f_c) / (1 - f_d) * x_9;
flow_constraint_6..        x_12 =e= (1 - f_a) / (1 - f_d) * x_8;
flow_constraint_5..        x_11 =e= (1 - f_d) / (1 - f_a) * x_5;
flow_constraint_4..        x_10 =e= (1 - f_b) / (1 - f_a) * x_4;
flow_constraint_3..        x_9 + x_8 =e= (1 - f_d) * x_3;
flow_constraint_2_lower..  x_7 =e= (1 - f_a - f_b) * x_2;
flow_constraint_2_upper..  x_6 =e= (1 - f_c - f_d) * x_2;
flow_constraint_1..        x_5 + x_4 =e= (1 - f_a) * x_1;

supply_constraint.. x_1 + x_2 + x_3 =e= S;

objective_eq..
z =e=
    a_1*y_1 + b_1*x_1 +
    a_2*y_2 + b_2*x_2 +
    a_3*y_3 + b_3*x_3 +
    a_4*y_4 + b_4*x_4 +
    a_5*y_5 + b_5*x_5 +
    a_6*y_6 + b_6*x_8 +
    a_7*y_7 + b_7*x_9 +
    a_8*y_8 + b_8*(x_6+x_10) +
    a_9*y_9 + b_9*(x_11+x_12) +
    a_10*y_10 + b_10*(x_7+x_13);

Model Superstructure / all /;
solve Superstructure using MIP minimizing z;
```

```
Proven optimal solution
MIP Solution:          581.000000    (8 iterations, 0 nodes)
Final Solve:           581.000000    (0 iterations)
```

```
Best possible:         581.000000
Absolute gap:            0.000000
Relative gap:            0.000000


                          LOWER          LEVEL          UPPER         MARGINAL

---- EQU bin_const~        1.0000         1.0000         1.0000           .
---- EQU bin_const~         .              .             +INF             .
---- EQU bin_const~       -2.0000          .             +INF             .
---- EQU bin_const~         .              .             +INF             .
---- EQU bin_const~         .              .             +INF             .
---- EQU bin_const~         .         -8.88178E-16       +INF             .
---- EQU bin_const~       -2.0000        -2.0000         +INF             .
---- EQU bin_const~         .              .             +INF             .
---- EQU bin_const~         .             1.0000         +INF             .
---- EQU bin_const~         .              .             +INF             .
---- EQU bin_const~         .              .             +INF             .
---- EQU bigm_cons~       -INF             .              .            -0.0100
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF             .              .            -0.3100
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF         -200.0000          .               .
---- EQU bigm_cons~       -INF        -1000.0000          .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF         -200.0000          .               .
---- EQU bigm_cons~       -INF         -150.0000          .               .
---- EQU bigm_cons~       -INF             .              .            -0.0150
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF        -1000.0000          .               .
---- EQU bigm_cons~       -INF         -150.0000          .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU bigm_cons~       -INF             .              .               .
---- EQU flow_cons~         .              .              .               .
---- EQU flow_cons~         .              .              .             0.0800
---- EQU flow_cons~         .              .              .               .
---- EQU flow_cons~         .              .              .               .
---- EQU flow_cons~         .              .              .             0.2250
---- EQU flow_cons~         .              .              .               .
---- EQU flow_cons~         .              .              .               .
```

```
---- EQU flow_cons~          .              .              .              .
---- EQU supply_co~      1000.0000      1000.0000      1000.0000         0.4300
---- EQU objective~          .              .              .             1.0000

                          LOWER          LEVEL          UPPER         MARGINAL

---- VAR z                -INF          581.0000        +INF             .
---- VAR x_1                .              .            +INF             .
---- VAR x_2                .              .            +INF             .
---- VAR x_3                .          1000.0000        +INF             .
---- VAR x_4                .              .            +INF            0.7800
---- VAR x_5                .              .            +INF            0.1100
---- VAR x_6                .              .            +INF            0.3900
---- VAR x_7                .              .            +INF            0.1900
---- VAR x_8                .           800.0000        +INF             .
---- VAR x_9                .              .            +INF             .
---- VAR x_10               .              .            +INF            0.3900
---- VAR x_11               .              .            +INF            0.0800
---- VAR x_12               .           850.0000        +INF             .
---- VAR x_13               .              .            +INF            0.1900
---- VAR y_1                .              .             1.0000       135.0000
---- VAR y_2                .              .             1.0000      -258.0000
---- VAR y_3                .             1.0000         1.0000        76.0000
---- VAR y_4                .              .             1.0000       125.0000
---- VAR y_5                .              .             1.0000        44.0000
---- VAR y_6                .             1.0000         1.0000        38.0000
---- VAR y_7                .              .             1.0000        51.0000
---- VAR y_8                .              .             1.0000       112.0000
---- VAR y_9                .             1.0000         1.0000        37.0000
---- VAR y_10               .              .             1.0000        58.0000
```

4.3 Problem 3 Code

```
  GAMS 49.1.0  5c4d4ed6 Feb 15, 2025          DAX-DAC arm 64bit/macOS - 04/14/25 18:53:07 Page
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
C o m p i l a t i o n


SETS
i           'set of reactors'       /r1,r2,r3/

;
Parameters
k(i)    'conversion factors'    /r1 0.8, r2 0.666667, r3 0.555555/
d(i)    'Reactor constant costs'/r1 8,r2 5.4,r3 2.7/
c(i)    'Reactor coeff costs'   /r1 1.5,r2 1,r3 0.5/
```

```
   9  ;
  10  Scalars
  11  De 'Demand' /10/
  12  Su 'Supply' /15/
  13  pr 'Price'  /2/
  14  ;
  15  Variables
  16  z
  17  ;
  18  Positive Variables
  19  x(i)
  20  ;
  21  Binary Variables
  22  y(i)
  23  ;
  24  Equations
  25      demand_constraint
  26      one_reactor_constraint
  27      flow_constraints(i)
  28      objective_eq
  29  ;
  30  demand_constraint..         sum(i,k(i) * x(i)) =g= De;
  31  one_reactor_constraint..    sum(i,y(i)) =e= 1;
  32  flow_constraints(i)..       x(i) =l= Su*y(i);
  33  objective_eq..              z =e= sum(i,pr*x(i) + d(i) * y(i) + c(i) * y(i));
  34
  35  Model Superstructure / all /;
  36  solve Superstructure using MIP minimizing z;


COMPILATION TIME     =        0.000 SECONDS      3 MB  49.1.0 5c4d4ed6 DAX-DAC
GAMS 49.1.0  5c4d4ed6 Feb 15, 2025          DAX-DAC arm 64bit/macOS - 04/14/25 18:53:07 Page 2
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
Equation Listing    SOLVE Superstructure Using MIP From line 36


---- demand_constraint  =G=

demand_constraint..  0.8*x(r1) + 0.666667*x(r2) + 0.555555*x(r3) =G= 10 ; (LHS = 0, INFES = 10


---- one_reactor_constraint  =E=

one_reactor_constraint..  y(r1) + y(r2) + y(r3) =E= 1 ; (LHS = 0, INFES = 1 ****)


---- flow_constraints  =L=
```

```
flow_constraints(r1)..  x(r1) - 15*y(r1) =L= 0 ; (LHS = 0)

flow_constraints(r2)..  x(r2) - 15*y(r2) =L= 0 ; (LHS = 0)

flow_constraints(r3)..  x(r3) - 15*y(r3) =L= 0 ; (LHS = 0)


---- objective_eq  =E=

objective_eq..  z - 2*x(r1) - 2*x(r2) - 2*x(r3) - 9.5*y(r1) - 6.4*y(r2) - 3.2*y(r3) =E= 0 ; (LI
```



```
---- z

z
                (.LO, .L, .UP, .M = -INF, 0, +INF, 0)
        1       objective_eq


---- x

x(r1)
                (.LO, .L, .UP, .M = 0, 0, +INF, 0)
        0.8     demand_constraint
        1       flow_constraints(r1)
       -2       objective_eq

x(r2)
                (.LO, .L, .UP, .M = 0, 0, +INF, 0)
        0.6667  demand_constraint
        1       flow_constraints(r2)
       -2       objective_eq

x(r3)
                (.LO, .L, .UP, .M = 0, 0, +INF, 0)
        0.5556  demand_constraint
        1       flow_constraints(r3)
       -2       objective_eq


---- y

y(r1)
                (.LO, .L, .UP, .M = 0, 0, 1, 0)
```

```
        1        one_reactor_constraint
      -15        flow_constraints(r1)
       -9.5      objective_eq

y(r2)
                 (.LO, .L, .UP, .M = 0, 0, 1, 0)
        1        one_reactor_constraint
      -15        flow_constraints(r2)
       -6.4      objective_eq

y(r3)
                 (.LO, .L, .UP, .M = 0, 0, 1, 0)
        1        one_reactor_constraint
      -15        flow_constraints(r3)
       -3.2      objective_eq

Proven optimal solution
MIP Solution:          34.500000    (0 iterations, 0 nodes)
Final Solve:           34.500000    (0 iterations)

Best possible:         34.500000
Absolute gap:           0.000000
Relative gap:           0.000000


                          LOWER          LEVEL          UPPER         MARGINAL

---- EQU demand_co~       10.0000        10.0000        +INF            2.5000
---- EQU one_react~        1.0000         1.0000         1.0000          .

---- EQU flow_constraints

          LOWER          LEVEL          UPPER         MARGINAL

r1        -INF           -2.5000          .              .
r2        -INF             .              .              .
r3        -INF             .              .              .

                          LOWER          LEVEL          UPPER         MARGINAL

---- EQU objective~         .              .              .             1.0000

                          LOWER          LEVEL          UPPER         MARGINAL

---- VAR z                -INF           34.5000        +INF             .

---- VAR x
```

```
          LOWER          LEVEL          UPPER         MARGINAL

r1          .           12.5000        +INF             .
r2          .             .            +INF           0.3333
r3          .             .            +INF           0.6111

---- VAR y

          LOWER          LEVEL          UPPER         MARGINAL

r1          .            1.0000         1.0000         9.5000
r2          .             .             1.0000         6.4000
r3          .             .             1.0000         3.2000


**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED


EXECUTION TIME       =        0.050 SECONDS      4 MB  49.1.0 5c4d4ed6 DAX-DAC


USER: GAMS Demo, for EULA and demo limitations see   G250131/0001CB-GEN
      https://www.gams.com/latest/docs/UG%5FLicense.html         DC0000


**** FILE SUMMARY

Input      /Users/andresespinosa/Documents/GAMS/Studio/workspace/hw5.gms
Output     /Users/andresespinosa/Documents/GAMS/Studio/workspace/hw5.lst
```