ChE 597 Computational Optimization

Homework 8

March 22nd 11:59 pm

1. Given is the integer programming problem

$$\max Z = 1.2y_1 + y_2$$
$$\text{s.t. } y_1 + y_2 \leq 1$$
$$1.2y_1 + 0.5y_2 \leq 1$$
$$y_1, y_2 = 0, 1$$

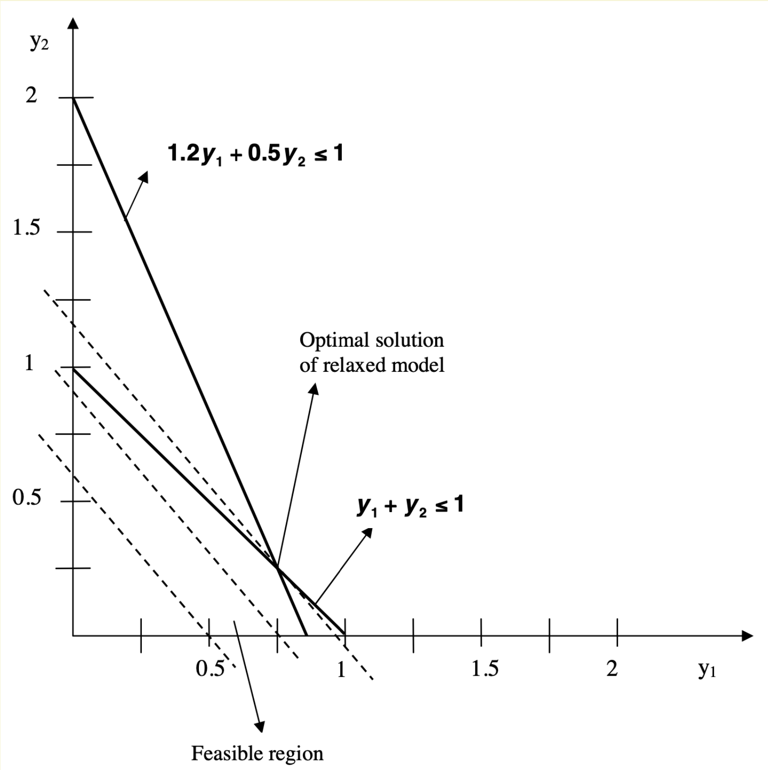
(a) Plot the contours of the objective and the feasible region for the case when the binary variables are relaxed as continuous variables $y_1, y_2 \in [0, 1]$.

(b) Determine from inspection the solution of the relaxed problem.

(c) Enumerate the four 0-1 combinations in your plot to find the optimal solution.

(d) Solve the relaxed LP problem by hand and derive Gomory mixed-integer cuts based on the LP relaxation (from the optimal simplex tableau) and verify that they cut-off the relaxed LP solution.

Solution: Part (a)



Part (b) By inspection, the solution os the relaxed problem is at the intersection of two lines in the figure, i.e., $y_1 = 5/7$ and $y_2 = 2/7$.

Part (c)

$$\begin{array}{llr} y_1 = 0 & y_2 = 0 & Z = 0 \\ y_1 = 0 & y_2 = 1 & Z = 1 \\ y_1 = 1 & y_2 = 0 & \text{infeasible} \\ y_1 = 1 & y_2 = 1 & \text{infeasible} \end{array}$$

Part (d)

Iter.1

	Z	y_1	y_2	s_1	s_2	RHS
	1	1.2	1	0	0	0
s_1	0	1	1	1	0	1
s_2	0	1.2	0.5	0	1	1

Iter.2

	Z	y_1	y_2	s_1	s_2	RHS
	1	0	$1/2$	0	-1	-1
s_1	0	0	$7/12$	1	$-5/6$	$1/6$
y_1	0	1	$5/12$	0	$5/6$	$5/6$

Iter.3

	Z	y_1	y_2	s_1	s_2	RHS
	1	0	0	$-6/7$	$-2/7$	$-8/7$
y_2	0	0	1	$12/7$	$-10/7$	$2/7$
y_1	0	1	0	$-5/7$	$10/7$	$5/7$

For the following two rows,

$$y_2 + 12/7 s_1 - 10/7 s_2 = 2/7$$
$$y_1 - 5/7 s_1 + 10/7 s_2 = 5/7$$

we can derive Gomory mixed integer cuts:

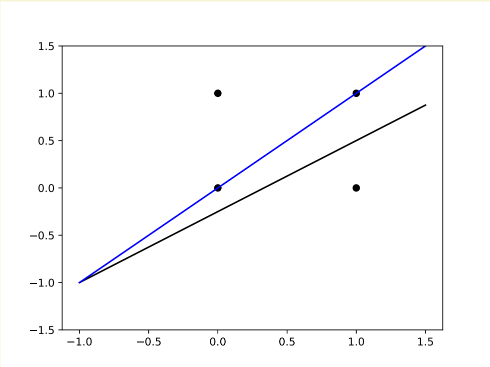
$$12/7 s_1 + 4/7 s_2 \geq 2/7$$
$$25/14 s_1 + 10/7 s_2 \geq 5/7$$

2. For each of the three sets below, find a missing valid inequality and verify graphically that its addition to the formulation gives $\mathrm{conv}(X)$.

   (a) $X = \left\{x \in \{0,1\}^2 : 3x_1 - 4x_2 \leq 1\right\}$
   (b) $X = \left\{(x,y) \in \{0,1\} \times \mathbb{R}^1_+ : y \leq 20x, y \leq 7\right\}$
   (c) $X = \left\{(x,y) \in \mathbb{Z}^1 \times \mathbb{R}^1_+ : y \leq 6x, y \leq 16\right\}$.

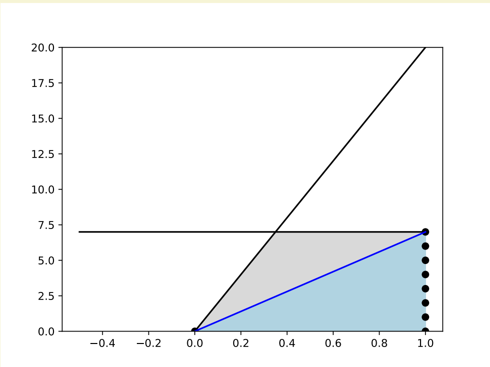
Solution: **(a)**

$$x_1 - x_2 \leq 0$$



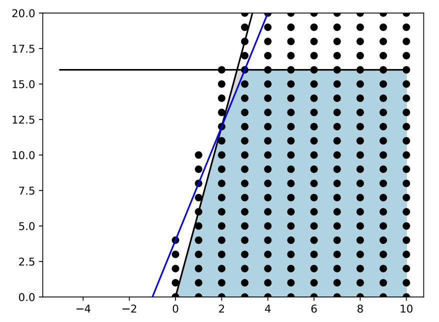
(b)

$$y \leq 7x$$



(c)

$$y \leq 4x + 4$$


20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−4
−2
0
2
4
6
8
10

3. Consider the Haverly's pooling problem (reference: `http://www.ii.uib.no/~lennart/drgrad/Adhya1999.pdf`) Formulate this problem using the P-formulation, Q-formulation, and PQ-formulation in pyomo and solve them using Gurobi.

Table 1: Summary

Category	**Quality**	**Unit Cost**	
Pool Sources	1: 3% sulfur	\$6	
	2: 1% sulfur	\$16	
Direct Supply	3: 2% sulfur	\$10	
Category	**Max Quality**	**Unit Price**	**Max Demand**
Products	1: 2.5% sulfur	\$9	100
	2: 1.5% sulfur	\$15	200

You don't need to consider the availability of raw materials and the pool capacity.

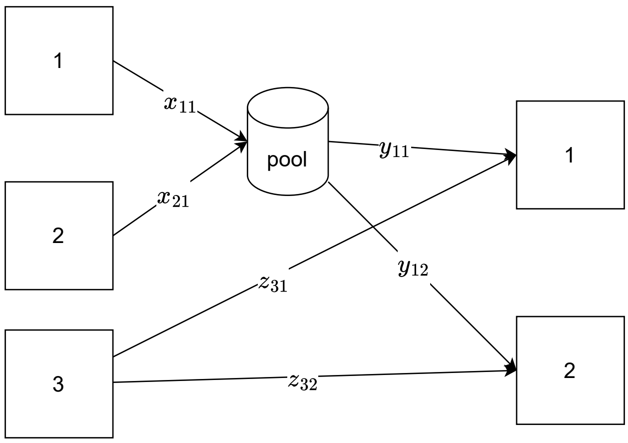


Figure 1: Haverly's pooling problem

Solution: Code available at:

This is a simplified case where qualities set is $\{1\}$ and pools set is $\{1\}$ because we only need to consider sulfur and a single pool. It is worth noting that only the raw material 3 is directly supplied to products. Therefore, there will not be z_{1j} and z_{2j} as well as x_{3j} when creating variables.

P-formulation:

Parameters

- c_i - unit cost of the ith raw material
- d_j - price of jth product
- C_{ik} - kth quality of raw material i
- D_j - demand of jth product
- P^U_{jk} - upper bound on kth quality of jth product

Variables

- p_{lk} - kth quality of pool l from pooling of raw materials
- x_{il} - flow of ith raw material into pool l
- y_{jk} - total flow from pool j to product k
- z_{ij} - direct flow of raw material i to product j

Either let $l \in \{1\}$ and $k \in \{1\}$ or eliminate them when formulating the model (e.g., $p_{lk} \to p_k$).

Objective: minimize the total cost of raw materials minus the total revenue from products.

Consider the following constraints:

- Conservation of flow into and out of the pools
- Sales of each product cannot exceed their demands
- Maintains the quality of the product within upper bounds
- Balances the quality

Q-formulation:

Replace the $x_{il} = q_{il} \sum_j y_{lj}$ terms in the P-formulation.

Add one more constraint: $\sum_i q_{il} = 1$.

PQ-formulation:

Add one more redundant constraint: $\sum_i q_{il} y_{lj} = y_{lj}$.

For all three formulation, the global optimum is -400.

4. Consider a k-means clustering problem. Each data point has dimension of 10. We have 20 data points, $k = 3$. Formulate the MIQCP and solve with gurobi

   The data set given in `https://github.com/li-group/ChE-597-Computational-Optimization/blob/main/HW%208/data_HW8_Q4.csv`

Solution: We define the following sets

- $\mathcal{N}$: set of data points
- $\mathcal{K}$: set of clusters

We normalize the dataset feature-wise (dimmension-wise). In the normalized dataset, p_{ij} denotes the jth coordinate of the ith data point.

The variables for the problem are as follows:

- y_{ik}: a binary decision variable taking value 1 if point i is assigned to cluster k and 0 otherwise
- d_i: distance of point i to its cluster center
- c_{kj}: jth coordinate of the kth cluster center.

The MIQCP formulation is as follows:

$$\begin{array}{lll}
\min_{\mathbf{c},\mathbf{d},\mathbf{y}} & \sum_{i\in\mathcal{N}} d_i & \\
\text{s.t.} & d_i \geq \sum_{j=1}^{D} \left(p_{ij} - c_{kj}\right)^2 - M_i(1 - y_{ik}) & \forall i \in \mathcal{N}, k \in \mathcal{K} \\
& \sum_{k\in\mathcal{K}} y_{ik} = 1 & \forall i \in \mathcal{N} \\
& \mathbf{c}_k \in \mathbb{R}^D & \forall k \in \mathcal{K} \\
& d_i \in \mathbb{R}_+ & \forall i \in \mathcal{N} \\
& y_{ik} \in \{0,1\} & \forall i \in \mathcal{N}, k \in \mathcal{K}
\end{array}$$

For 10 dimmensions, the optimization problem is very large and very difficult to solve. Restricting the solving time to 5 minutes, we get the following solution:

Point	cluster
1	1
2	1
3	2
4	2
5	1
6	1
7	2
8	2
9	1
10	3
11	3
12	1
13	1
14	3
15	3
16	3
17	2
18	2
19	2
20	3

Considering the first 3 dimmensions for the data, having an MIP gap of 1%, we get the following solution:

Point	cluster
1	2
2	3
3	1
4	1
5	2
6	3
7	2
8	1
9	3
10	3
11	3
12	3
13	3
14	1
15	3
16	2
17	2
18	1
19	2
20	1

5. Consider the following set of squares with lengths as shown below:

Square	Length
1	2
2	3
3	6
4	9
5	10
6	12

Try to pack these squares into a rectangle whose height and width are at least 10 and at most 25. Suppose we do not allow rotating the squares. What is optimum dimensions of the rectangle and how are the squares packed?

Solution: The variables for the problem are as follows:

- A: width of the container
- B : height of the container
- x_i: x-coordinate of the center of square i
- y_i: y-coordinate of the center of square i
- $d_{i,j}$: distance between the center of squares i, j

The parameters of the problem are defined as follows:

- A^{low}: lower limit of the width of the container
- A^{up}: upper limit of the width of the container
- B^{low}: lower limit of the height of the container
- B^{up}: upper limit of the height of the container
- r_i: half of the side length of square i

An optimization model for a feasible packing defined as follows:

$$
\begin{aligned}
\min \quad & AB \\
\text{s.t.} \quad & x_i + r_i - \tfrac{1}{2}A \leqslant 0, \quad i = 1,\dots,N, \\
& r_i - x_i - \tfrac{1}{2}A \leqslant 0, \quad i = 1,\dots,N, \\
& y_i + r_i - \tfrac{1}{2}B \leqslant 0, \quad i = 1,\dots,N, \\
& r_i - y_i - \tfrac{1}{2}B \leqslant 0, \quad i = 1,\dots,N, \\
& (r_i + r_j)\sqrt{2} \leqslant d_{ij}, \quad i,j = 1,\dots,N, i < j, \\
& d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad i,j = 1,\dots,N, i < j, \\
& A^{\text{low}} \leqslant A \leqslant A^{\text{up}} \\
& B^{\text{low}} \leqslant B \leqslant B^{\text{up}},
\end{aligned}
$$

However, this is not optimal. For an optimal packing, with squares placed such that their sides align with the x-axis and y-axis, the following optimization model can be used. The logic behind the model is to ensure that a corner point of 1 square is not inside a another square

The variables for the model are as follows:

- A: width of the container

- B : height of the container
- x_i: x-coordinate of the center of square i
- y_i: y-coordinate of the center of square i
- x_i^{max}: maximum x-coordinate of the square i
- x_i^{min}: minimum x-coordinate of the square i
- y_i^{max}: maximum y-coordinate of the square i
- y_i^{min}: minimum y-coordinate of the square i
- $x_{i,j,k}^{in1}$: binary variable to see if the x coordinate of the k^{th} corner point of point i is greater than the x_j^{min}
- $x_{i,j,k}^{in2}$: binary variable to see if the x coordinate of the k^{th} corner point of point i is lesser than the x_j^{max}
- $y_{i,j,k}^{in1}$: binary variable to see if the y coordinate of the k^{th} corner point of point i is greater than the y_j^{min}
- $y_{i,j,k}^{in1}$: binary variable to see if the y coordinate of the k^{th} corner point of point i is lesser than the y_j^{max}

The parameters of the problem are defined as follows:

- A^{low}: lower limit of the width of the container
- A^{up}: upper limit of the width of the container
- B^{low}: lower limit of the height of the container
- B^{up}: upper limit of the height of the container
- r_i: half of the side length of square i

An optimization model is defined as follows:

$$
\begin{aligned}
\min \quad & AB \\
\text{s.t.} \quad & x_i + r_i - \tfrac{1}{2}A \leqslant 0, \quad i = 1,\dots,N, \\
& r_i - x_i - \tfrac{1}{2}A \leqslant 0, \quad i = 1,\dots,N, \\
& y_i + r_i - \tfrac{1}{2}B \leqslant 0, \quad i = 1,\dots,N, \\
& r_i - y_i - \tfrac{1}{2}B \leqslant 0, \quad i = 1,\dots,N, \\
& x_i^{max} = x_i + r_i \quad i = 1,\dots,N, \\
& x_i^{min} = x_i - r_i \quad i = 1,\dots,N, \\
& y_i^{max} = y_i + r_i \quad i = 1,\dots,N, \\
& y_i^{min} = y_i - r_i \quad i = 1,\dots,N, \\
& x_i^{min} \geq x_j^{min} - M(1 - x_{i,j,1}^{in1}) \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{min} \leq x_j^{max} + M(1 - x_{i,j,1}^{in2}) \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{min} \geq y_j^{min} - M(1 - y_{i,j,1}^{in1}) \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{min} \leq y_j^{max} + M(1 - y_{i,j,1}^{in2}) \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{min} \leq x_j^{min} + M x_{i,j,1}^{in1} \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{min} \geq x_j^{max} - M x_{i,j,1}^{in2} \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{min} \leq y_j^{min} + M y_{i,j,1}^{in1} \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{min} \geq y_j^{max} - M y_{i,j,1}^{in2} \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{min} \geq x_j^{min} - M(1 - x_{i,j,2}^{in1}) \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{min} \leq x_j^{max} + M(1 - x_{i,j,2}^{in2}) \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{max} \geq y_j^{min} - M(1 - y_{i,j,2}^{in1}) \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{max} \leq y_j^{max} + M(1 - y_{i,j,2}^{in2}) \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{min} \leq x_j^{min} + M x_{i,j,2}^{in1} \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{min} \geq x_j^{max} - M x_{i,j,2}^{in2} \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{max} \leq y_j^{min} + M y_{i,j,2}^{in1} \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{max} \geq y_j^{max} - M y_{i,j,2}^{in2} \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{max} \geq x_j^{min} - M(1 - x_{i,j,3}^{in1}) \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{max} \leq x_j^{max} + M(1 - x_{i,j,3}^{in2}) \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{min} \geq y_j^{min} - M(1 - y_{i,j,3}^{in1}) \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{min} \leq y_j^{max} + M(1 - y_{i,j,3}^{in2}) \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{max} \leq x_j^{min} + M x_{i,j,3}^{in1} \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{max} \geq x_j^{max} - M x_{i,j,3}^{in2} \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{min} \leq y_j^{min} + M y_{i,j,3}^{in1} \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{min} \geq y_j^{max} - M y_{i,j,3}^{in2} \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{max} \geq x_j^{min} - M(1 - x_{i,j,4}^{in1}) \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{max} \leq x_j^{max} + M(1 - x_{i,j,4}^{in2}) \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{max} \geq y_j^{min} - M(1 - y_{i,j,4}^{in1}) \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{max} \leq y_j^{max} + M(1 - y_{i,j,4}^{in2}) \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{max} \leq x_j^{min} + M x_{i,j,4}^{in1} \quad i,j = 1,\dots,N, i \neq j \\
& x_i^{max} \geq x_j^{max} - M x_{i,j,4}^{in2} \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{max} \leq y_j^{min} + M y_{i,j,4}^{in1} \quad i,j = 1,\dots,N, i \neq j \\
& y_i^{max} \geq y_j^{max} - M y_{i,j,4}^{in2} \quad i,j = 1,\dots,N, i \neq j \\
& x_{i,j,k}^{in1} + x_{i,j,k}^{in2} + y_{i,j,k}^{in1} + y_{i,j,k}^{in2} \leq 3 \quad i,j = 1,\dots,N, k = 1,2,3,4, i \neq j \\
& A^{\text{low}} \leqslant A \leqslant A^{\text{up}} \\
& B^{\text{low}} \leqslant B \leqslant B^{\text{up}} \,,
\end{aligned}
$$

The optimum packing for the question is shown below

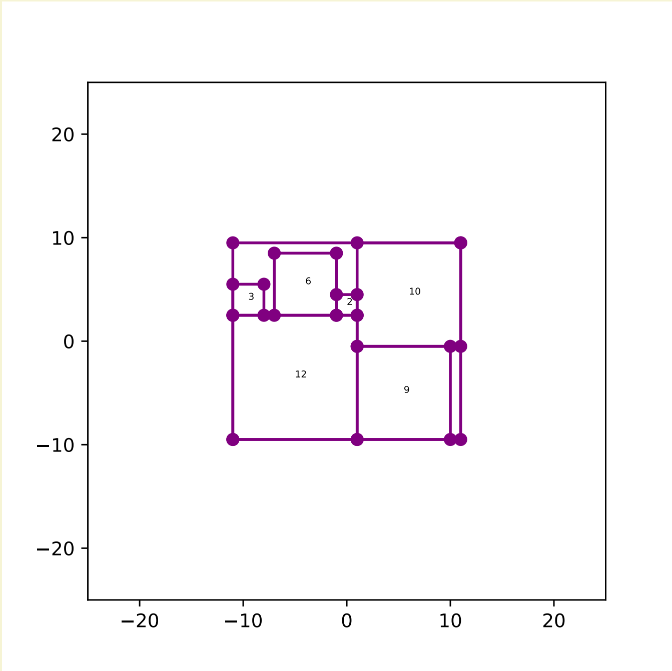


Figure 2: Haverly's pooling problem

We get $A = 22, B = 19$.