Introduction to Optimization Modeling in Python

Alba Carrero, Juan Javaloyes & Daniel Vázquez

October 2018

Contents

https://github.com/CAChemE/pyomo

Install PYOMO and Solvers

PYOMO

- `conda install -c conda-forge pyomo`
- `conda install -c conda-forge pyomo.extras`

http://www.pyomo.org/installation/

SOLVERS

- glpk [LP, MILP] >> `conda install -c conda-forge glpk`

  http://ftp.gnu.org/gnu/glpk/

- gurobi [LP, MILP] >> download and install. Free university license

  http://www.gurobi.com/

- IPOPT [NLP] >> `conda install -c conda-forge ipopt`

  https://www.coin-or.org/download/binary/

- SCIP [MINLP] >> download and add the solver installation to the path environment variable

  http://scip.zib.de/#download

PYOMO Sources

Homepage:

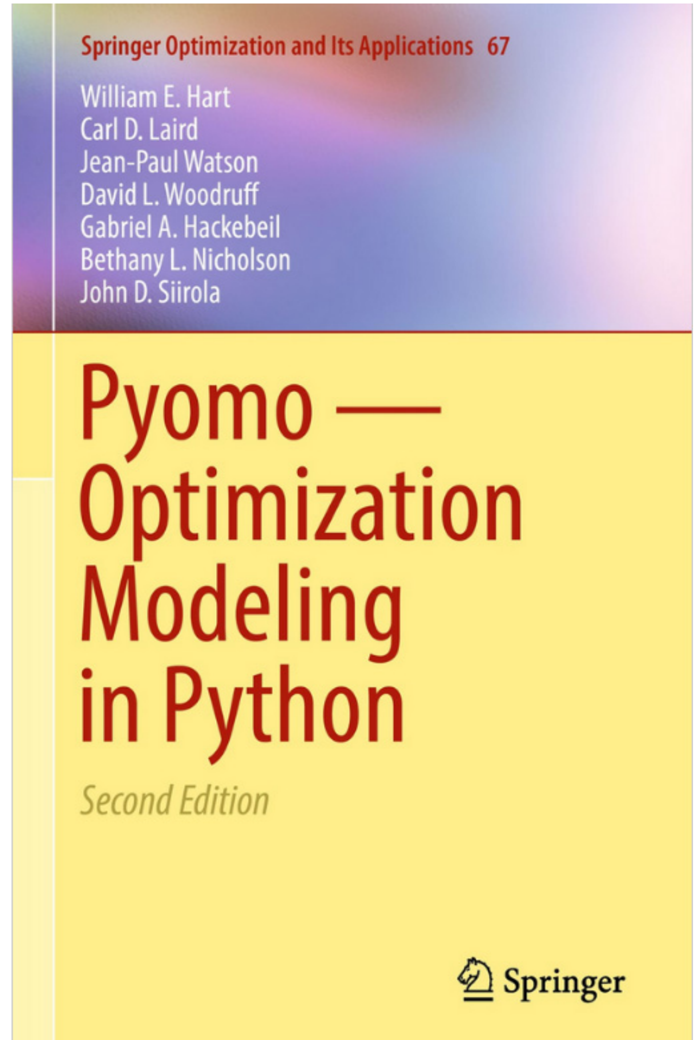
PYOMO
HOME / ABOUT / DOWNLOAD / DOCUMENTATION / BLOG

http://www.pyomo.org/

Sandia National Laboratories PYOMO

https://software.sandia.gov/trac/pyomo

Book Reference



PYOMO Sources

GitHub

https://github.com/Pyomo

Help Forums

https://groups.google.com/forum/#!forum/pyomo-forum

Stack Overflow

https://stackoverflow.com/questions/tagged/pyomo

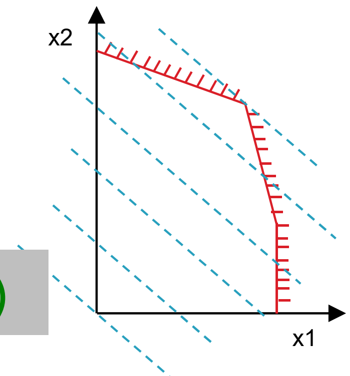
Mathematical Programming Background

Mathematical Programming

Classes of Optimization Problems

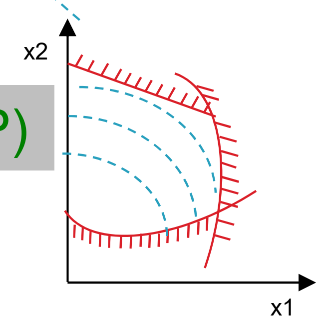
$$
\begin{aligned}
\min \quad & f_0(x) && \leftarrow \text{Objective Function} \\
s.t \quad & f_i(x) \le b_i \qquad i = 1,\dots,m. && \leftarrow \text{Constraints} \\
& x \in \mathbb{R}^n
\end{aligned}
$$

$$f_0(x) : \mathbb{R}^n \to \mathbb{R},\ f_i(x) : \mathbb{R}^n \to \mathbb{R}$$



$$f_i(\alpha x_1 + \beta x_2) = x f_i(x_1) + \beta f_i(x_2) \quad x_1, x_2 \in \mathbb{R}^n \quad \alpha,\beta \in \mathbb{R}$$

← Linear Program (LP)



$$f_i(\alpha x_1 + \beta x_2) \ne x f_i(x_1) + \beta f_i(x_2) \quad x_1, x_2 \in \mathbb{R}^n \quad \alpha,\beta \in \mathbb{R}$$

← Nonlinear Program (NLP)

$$f_i(\alpha x_1 + \beta x_2) \le x f_i(x_1) + \beta f_i(x_2) \quad x_1, x_2 \in \mathbb{R}^n \quad \alpha,\beta \in \mathbb{R} \text{ with } \alpha + \beta = 1,\ \alpha, \beta \ge 0$$

← Convex Optimization

Mathematical Programming

Classes of Optimization Problems

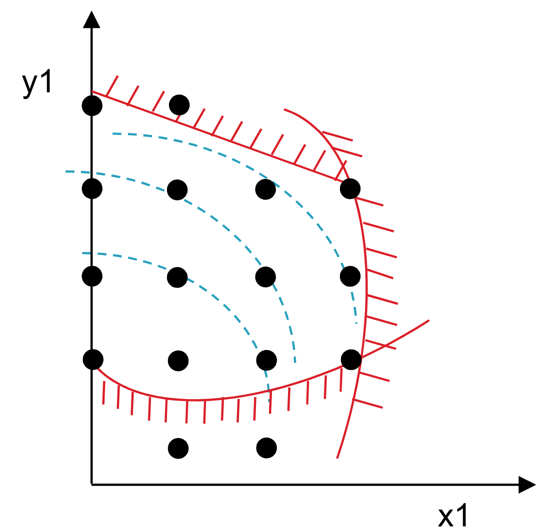
$$\begin{aligned} \min \quad & f_0(x,y) \\ s.t \quad & f_i(x,y) \le b_i \qquad i = 1,\ldots,m. \\ & x \in R^n,\ y \in \{0,1\}^q \end{aligned}$$

← Objective Function

← Constraints

← Mixed-Integer (Non) Linear Programming (MI(N)LP)

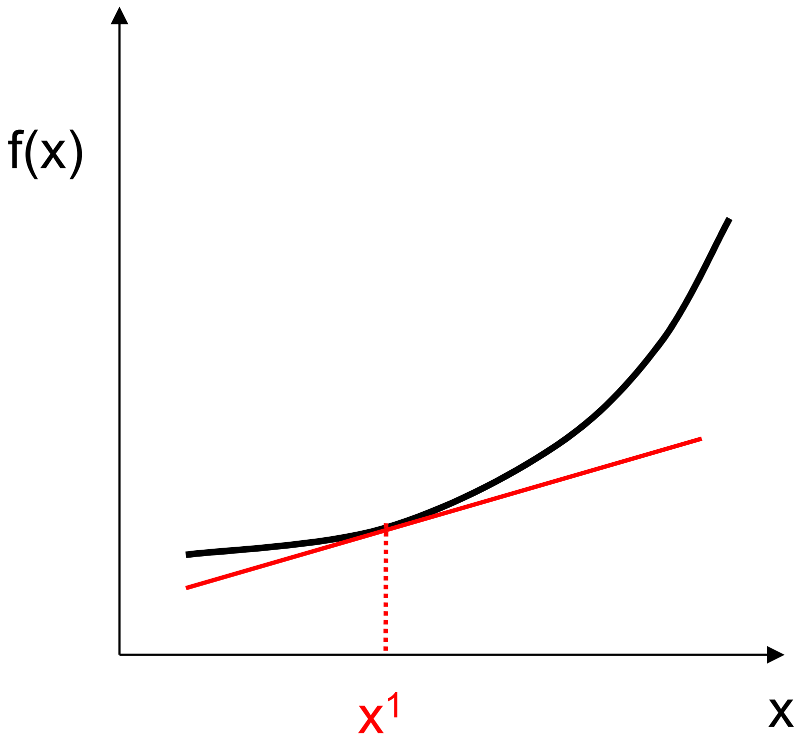
$$f_0(x,y) : R^n \to R,\ f_i(x,y) : R^n \to R$$



Mathematical Programming

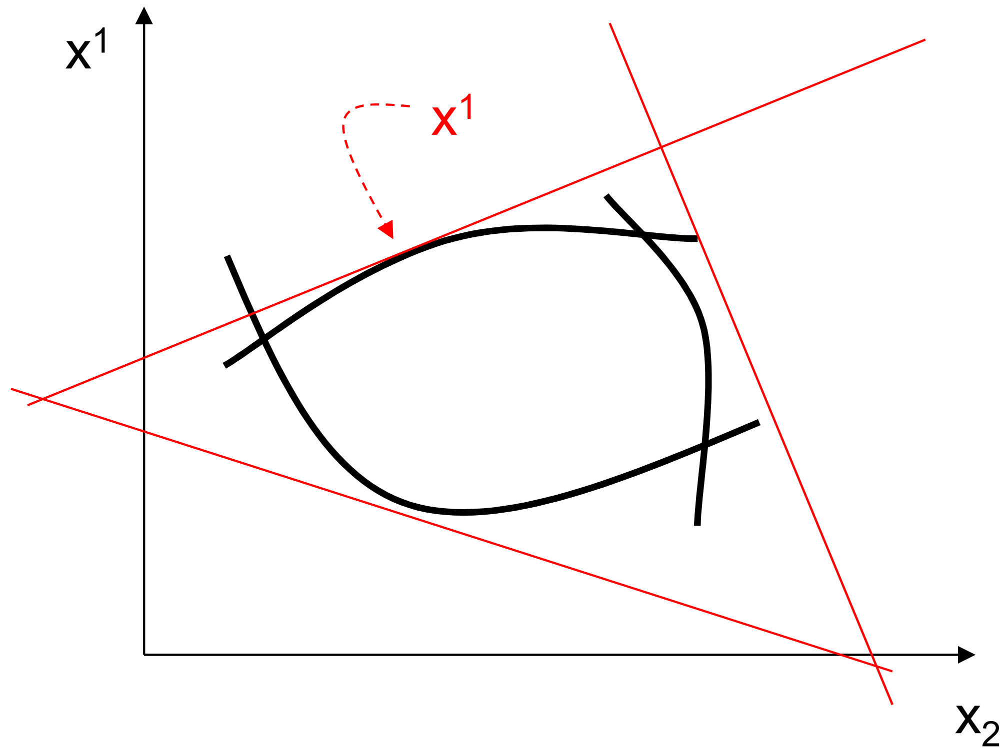
Classes of Optimization Problems – **Convex Problems**

Convex objective function



Sub-estimation of the Objective function

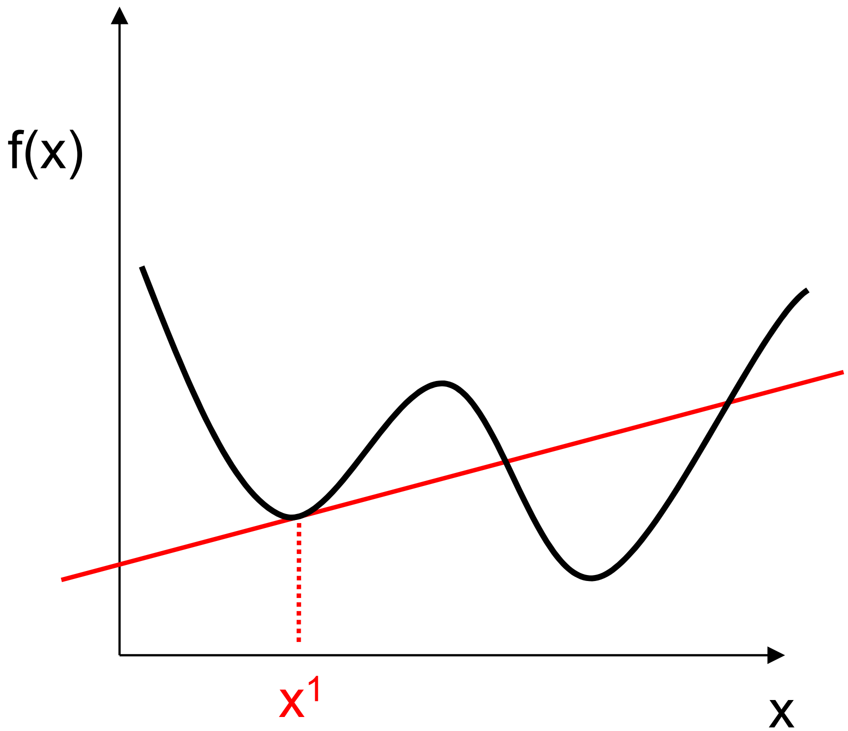
Convex feasible region



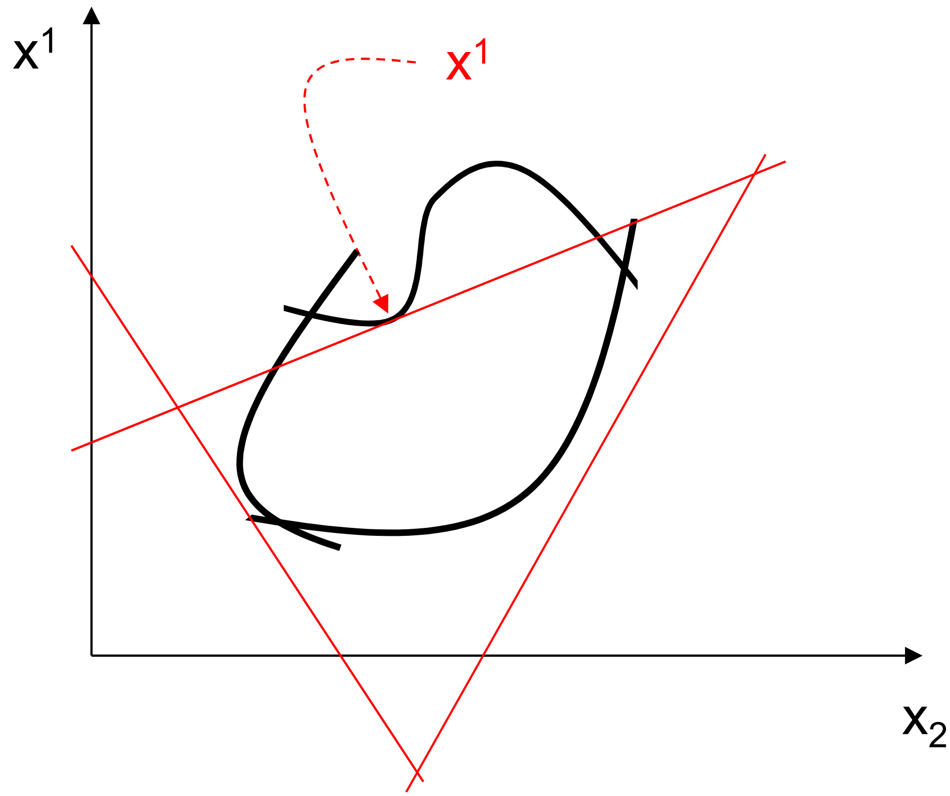
Overestimation of the feasible region

Classes of Optimization Problems – Non-Convex Problems

Non-Convex objective function

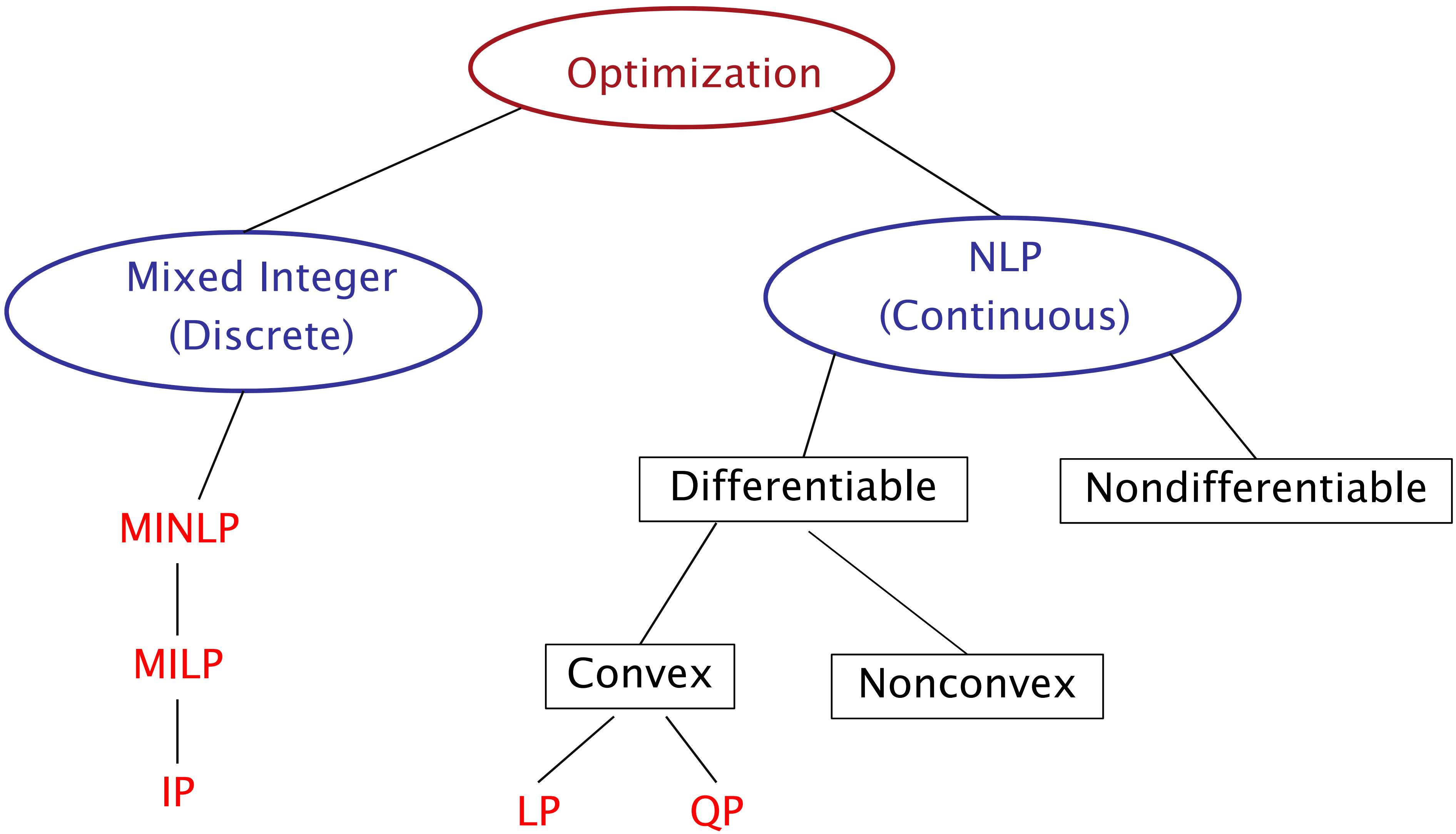


Non-Convex feasible region



Mathematical Programming

Classes of Optimization Problems



Mathematical Programming

Classes of Optimization Problems - Solvers

	Problem Class			
Solver	LP	MILP	NLP	MINLP
ALPHAECP				X
ANTIGONE			X	X
BARON	X	X	X	X
CONOPT4	X		X	
CPLEX	X	X		
DICOPT				X
GUROBI	X	X		
GLPK	X	X		
IPOPT	X		X	
SCIP		X	X	X

CAChemE

Mathematical Programming

Modeling of Discrete-Continuous Optimization Problems

Motivation Example (Grossmann & Trespalacios (2013), doi.org/10.1002/aic.14088)

"A company has to decide whether to produce either product A or product B in order to maximize its profit. The profit of product A is 3, and the profit of product B is 2. The limit on production of A is 4, and the limit in production of B is 5."

CAChemE

Mathematical Programming

Modeling of Discrete-Continuous Optimization Problems

Motivation Example (Grossmann & Trespalacios (2013), doi.org/10.1002/aic.14088)

$$
\begin{aligned}
\max \quad & 3A + 2B \\
s.t \quad & A\, y_2 = 0 \\
& B\, y_1 = 0 \\
& y_1\, y_2 = 0 \\
& y_1 + y_2 = 1 \\
& 0 \le A \le 4 \\
& 0 \le B \le 5 \\
& 0 \le y_1, y_2 \le 1 \\
& A, B, y_1, y_2 \in \mathbb{R}^n
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & 3A + 2B \\
s.t \quad & A \le 10\,(1 - y_2) \\
& \mathrm{B} \le 10\,(1 - y_1) \\
& y_1 + y_2 = 1 \\
& 0 \le A \le 4 \\
& 0 \le B \le 5 \\
& A, B \in \mathbb{R} \\
& y_1, y_2 \in \{0,1\}
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & 3A + 2B \\
s.t \quad & 0 \le A \le 4\, y_1 \\
& 0 \le B \le 5\, y_2 \\
& y_1 + y_2 = 1 \\
& A, B \in \mathbb{R} \\
& y_1, y_2 \in \{0,1\}
\end{aligned}
$$

CAChemE

Mathematical Programming

Generalized Disjunctive Programming (GDP) Formulation

$$
\begin{aligned}
\min:\ & z = f(x) \\
s.t.\quad & g(x) \le 0 \\
& h(x) = 0 \\
& \bigvee_{i \in D_k} \begin{bmatrix} Y_{k,i} \\ r_{i,k}(x) \le 0 \\ s_{i,k}(x) = 0 \end{bmatrix} \qquad k \in K \\
& \underline{\bigvee}_{i \in D_k} Y_{k,i} \qquad k \in K \\
& \Omega(Y) = True \\
& x^{lo} \le x \le x^{up} \\
& x \in \mathbb{R}^n; \\
& Y \in \{True,\ False\} \qquad k \in K, i \in D_k
\end{aligned}
$$

Objective function and global constraints

Disjunctions

Logic propositions

Balas (1979)

Raman and Grossmann (1994)

CAChemE

Mathematical Programming

Generalized Disjunctive Programming (GDP) Formulation

GDP Reformulation (Grossmann & Trespalacios (2013), doi.org/10.1002/aic.14088)

GDP

$$\min: \; z = f(x)$$
$$s.t. \quad g(x) \le 0$$
$$h(x) = 0$$

$$\bigvee_{i \in D_k} \begin{bmatrix} Y_{k,i} \\ r_{i,k}(x) \le 0 \\ s_{i,k}(x) = 0 \end{bmatrix} \quad k \in K$$
$$\underline{\bigvee}_{i \in D_k} Y_{k,i} \quad k \in K$$

$$\Omega(Y) = True$$
$$x^{lo} \le x \le x^{up}, \quad x \in \mathbb{R}^n$$
$$Y \in \{True, \; False\} \quad k \in K, i \in D_k$$

Big-M (BM)

$$\min: \; z = f(x)$$
$$s.t. \quad g(x) \le 0$$
$$h(x) = 0$$

$$r_{ki}(x) \le M^{ki}(1 - y_{ki}) \quad k \in K, i \in D_k$$
$$\sum_{i \in D_k} y_{ki} = 1 \quad k \in K$$

$$Hx \ge h$$
$$x^{lo} \le x \le x^{up}, \quad x \in \mathbb{R}^n$$
$$y_{ki} \in \{0,1\} \quad k \in K, i \in D_k$$

Hull Reformulation (HR) [Linear]

$$\min: \; z = f(x)$$
$$s.t. \quad g(x) \le 0$$
$$h(x) = 0$$

$$x = \sum_{i \in D_k} v^{ki} \quad k \in K$$
$$y_{ki} \, r_{ki}\left(v^{ki} / y_{ki}\right) \le 0 \quad k \in K, i \in D_k$$
$$x^{lo} \, y_{ki} \le v^{ki} \le x^{up} \, y_{ki} \quad k \in K, i \in D_k$$
$$\sum_{i \in D_k} y_{ki} = 1 \quad k \in K$$

$$Hx \ge h$$
$$x \in \mathbb{R}^n$$
$$y_{ki} \in \{0,1\} \quad k \in K, i \in D_k$$

PYOMO Components

PYOMO Components

Example: Machinery Problem

A company manufacture four types of machinery. The factory is divided in three sections. The first section has available 960 h/week, the second 1110 h/week and the third 400 h/week. Each machinery unit requires the following time at each section

Plants	hours per machinery			Profit [units/machinery]
	Machining	Painting	Assembly	
Machinery 1	6	3	2	12
Machinery 2	4	3	1	8
Machinery 3	4	6	2	12
Machinery 4	8	9	1	17

Determine the number of units of machinery for each type that should be manufacture per week to maximize the profit.

CACheme

PYOMO Components

Example: Machinery Problem

Nomenclature

m ➔ machinery type (set)

s ➔ factory section (set)

$profit_m$ ➔ profit per machinery type (parameter)

b_s ➔ time availability in each section per week (parameter)

$T_{m,s}$ ➔ time required for each machinery type in each section(parameter)

x_m ➔ number of units of machinery for each type (variable)

$$\min_x : \sum_m profit_m \, x_m$$

Objective function

$$s.t. \quad \sum_m T_{m,s} \, x_m \leq b_i \qquad \forall s$$

Factory section time limit

$$x_m \in \mathbb{Z}$$

Model Structure

```
from pyomo.environ import *

m = ConcreteModel()

M = m.M = Set(initialize =  ['m1', 'm2', 'm3', 'm4'])
S = m.S = Set(initialize =  ['s1', 's2', 's3'])

profit   = {'m1':12, 'm2':8, 'm3':12, 'm4':17}
max_time = {'s1': 960, 's2': 1110, 's3': 400}
time_x_section = {
('m1','s1'): 6 , ('m1','s2'): 3 , ('m1','s3'): 2,
('m2','s1'): 4 , ('m2','s2'): 3 , ('m2','s3'): 1,
('m3','s1'): 4 , ('m3','s2'): 6 , ('m3','s3'): 2,
('m4','s1'): 8 , ('m4','s2'): 9 , ('m4','s3'): 1}

x = m.x = Var( M, within = PositiveIntegers)

m.value = Objective(
expr = sum( profit[i] * m.x[i] for i in M),
            sense = maximize )

def constraint_rule(m, j):
    return sum(time_x_section[i,j] * x[i] for i in M)
                        <= max_time[j]
m.constraint = Constraint(S, rule = constraint_rule)

opt = SolverFactory('glpk’).solve(m)
```

Model Structure

```
from pyomo.environ import *
```
Import packages

```
m = ConcreteModel()
```
Create model object

```
M = m.M = Set(initialize =  ['m1', 'm2', 'm3', 'm4'])
S = m.S = Set(initialize =  ['s1', 's2', 's3'])
```
Sets declarations

```
profit   = {'m1':12, 'm2':8, 'm3':12, 'm4':17}
max_time = {'s1': 960, 's2': 1110, 's3': 400}
time_x_section = {
('m1','s1'): 6 , ('m1','s2'): 3 , ('m1','s3'): 2,
('m2','s1'): 4 , ('m2','s2'): 3 , ('m2','s3'): 1,
('m3','s1'): 4 , ('m3','s2'): 6 , ('m3','s3'): 2,
('m4','s1'): 8 , ('m4','s2'): 9 , ('m4','s3'): 1}
```
Specify/import problem data

```
x = m.x = Var( M, within = PositiveIntegers)
```
Variable declarations

```
m.value = Objective(
expr = sum( profit[i] * m.x[i] for i in M),
            sense = maximize )
```
Objective function declaration

```
def constraint_rule(m, j):
    return sum(time_x_section[i,j] * x[i] for i in M)
                <= max_time[j]
m.constraint = Constraint(S, rule = constraint_rule)
```
Constraint functions declaration

```
opt = SolverFactory('glpk’).solve(m)
```
Solver call

Case Studies

Assignment Problem

In this problem, we have a number of people "p" and a number of tasks "t". Each person has a suitability coefficient "SC", which represents how effectively can a person "p" perform a task "t". The objective is to maximize the total suitability of the system. For this example, the following data is presented:

PEOPLE: Pedro, Marta, Laura
TASKS: Accountant, Sell Manager, Human Resources
SUITABILITY COEFFICIENTS:

Person	SC Accountant	SC Sell Manager	SC Human Resources
Pedro	11	5	2
Marta	15	12	8
Laura	3	1	10

Assignment Problem

Mathematical Model

$$\max\left(\sum_{p,t} C_{p,t}\, y_{p,t}\right)$$

Maximize the suitability of the assignment

$$s.t.$$

$$\sum_{t} y_{p,t} = 1 \qquad \forall p$$

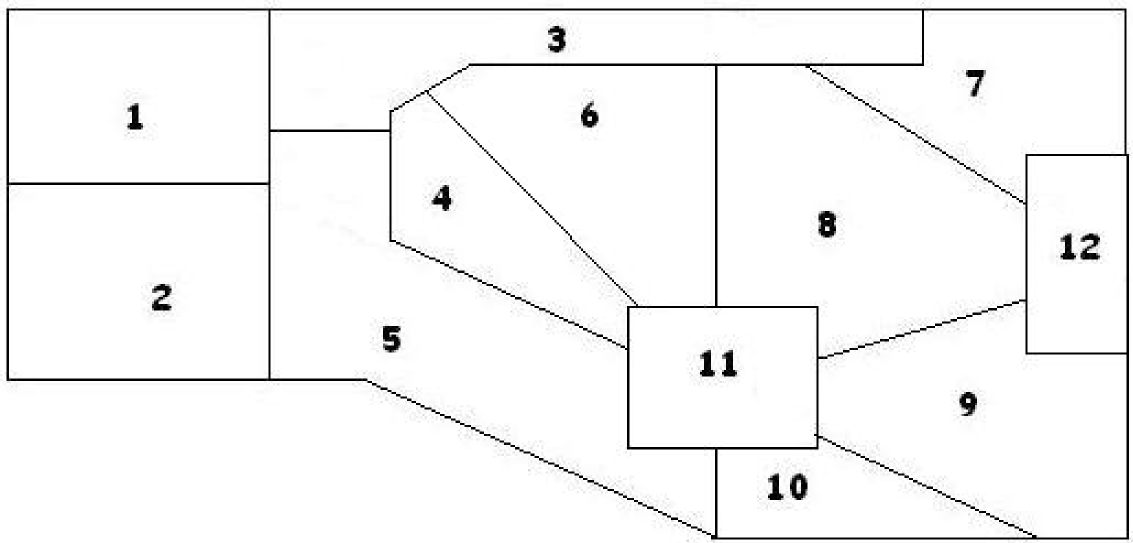
Each person can only perform one job

$$\sum_{p} y_{p,t} = 1 \qquad \forall t$$

Each job must be performed by one and only one person

Set Covering Problem

In this problem, we have a number of zones in which we may or may not install awesome new firefighter stations. However, the mayor of the city is a bit stingy, and wants us to install the absolute minimum number of stations possible. Having the following map of the zone:



And considering that a single station can only provide service to the zones in its immediate neighborhood, what stations should be built?

Set Covering Problem

Mathematical Model

$$\min(\sum_i y_i)$$

Minimize the number of stations

$$s.t.$$

$$\sum_i C_{i',i} y_i \geq 1 \qquad \forall i'$$

Service constraint

Knap-Sack Problem

In this problem, we are adventurous thieves. We want to loot all of the treasures that we can before the guards arrive. Since it will not be possible to come back to loot whatever we leave behind, we must ensure that we maximize the benefit of what we steal. Our horse can handle up until 2500 g of weight (It's a tiny pony) and a volume of 2000 cm3. Considering the loot table, what should we carry out there to sell?

Item	Market Price	Volume (cm3)	Unit Weight (g)	Units available
Chest	50	1000	2000	1
Ring	5	2	20	10
Necklace	3	10	300	1
Mirror	20	500	1000	1
Bracelet	16	15	300	15
Ruby	5	3	75	1
Parfum	1	100	100	1
Diamond	30	5	50	1
Gold goblet	12	250	500	1
Spice	40	100	100	1

Knap-Sack Problem

Mathematical Model

$$\max\left(\sum_i MP_i n_i\right)$$ Maximize Profit

$$s.t.$$

$$\sum_i V_i n_i \leq 2000$$ Volume constraint

$$\sum_i W_i n_i \leq 2500$$ Weight constraint

$$n_i \leq N_i \qquad \forall i$$ Amount constraint

Sudoku problem

Nomenclature

r ➔ rows | **c** ➔ columns | **k** ➔ value

$y_{r,c,k}$ ➔ binary variable. $y_{r,c,k}$ = 1 means cell [r. c] is assigned number k

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	5	3			7				
r2	6			1	9	5			
r3		9	8					6	
r4	8				6				3
r5	4			8					1
r6	7				2				6
r7		6					2	8	
r8				4	1				5
r9					8			7	9

Every position in the Sudoku is filled

$$\sum_k y_{r,c,k} = 1 \quad \forall r, c$$

Cells in the same column must be assigned distinct numbers

$$\sum_r y_{r,c,k} = 1 \quad \forall c, k$$

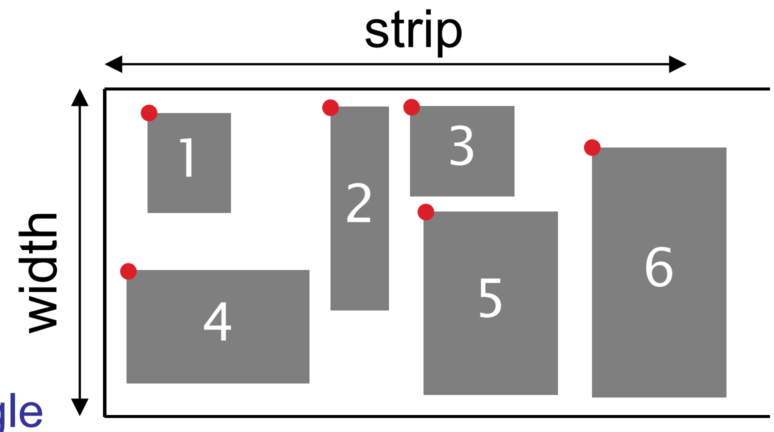
Cells in the same row must be assigned distinct numbers

$$\sum_c y_{r,c,k} = 1 \quad \forall r, k$$

Cells in the same 3x3 grid must be assigned distinct numbers

$$\sum_{r=3p-2}^{3p} \sum_{c=3q-2}^{3q} y_{r,c,k} = 1 \quad \forall k,\ p, q = \{1, 2, 3\}$$

Strip packing 2D problem



Nomenclature

- **i** ➔ rectangles, i = {1,2,…,n}
- **lt** ➔ length of the strip
- **(x_i,y_i)** ➔ rectangle coordinates
- L_i, H_i ➔ Length and height of rectangle i
- W ➔ Width of the strip
- UB_i ➔ Upper bound for the x-coordinate of every rectangle

Source: Sawaya & Grossmann (2005), https://doi.org/10.1016/j.compchemeng.2005.04.004

GDP Formulation

$$\min \ lt$$

$$s.t \quad lt \geq x_i + L_i \qquad \forall i \in N$$

$$\begin{bmatrix} Y^1_{i,j} \\ x_i + L_i \leq x_j \end{bmatrix} \vee \begin{bmatrix} Y^2_{i,j} \\ x_j + L_j \leq x_i \end{bmatrix} \vee \begin{bmatrix} Y^3_{i,j} \\ y_i - H_i \geq y_j \end{bmatrix} \vee \begin{bmatrix} Y^4_{i,j} \\ y_j - H_j \geq y_i \end{bmatrix} \qquad \forall i,j \in N, i < j$$

$$x_i \leq UB_i - L_i \qquad \forall i \in N$$

$$H_i \leq y_i \leq W \qquad \forall i \in N$$

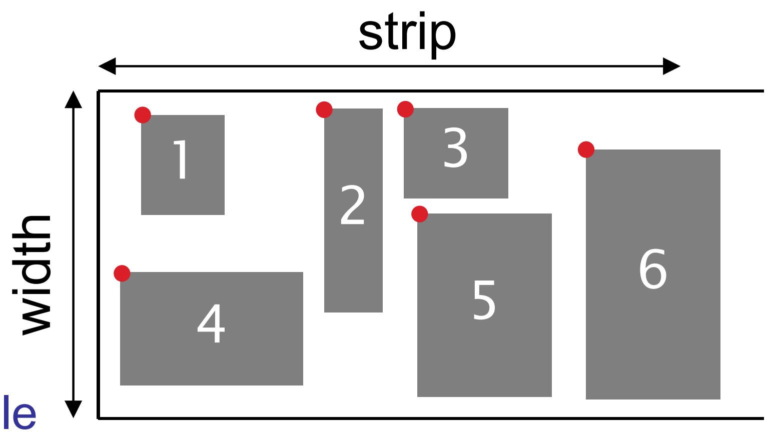
$$lt, x_i, y_i \in \mathbb{R}$$

$$Y_{i,j} \in \{Ture, False\}$$

Strip packing 2D problem

Nomenclature

i ➔ rectangles, i = {1,2,…,n}
lt ➔ length of the strip
(x_i,y_i) ➔ rectangle coordinates
L_i, H_i ➔ Length and height of rectangle i
W ➔ Width of the strip
UB_i ➔ Upper bound for the x-coordinate of every rectangle



MILP Formulation [Big-M]

$$
\begin{aligned}
&\min\ lt \\
&s.t\quad lt \geq x_i + L_i \qquad \forall i \in N \\
&\qquad x_i + L_i \leq x_j + M_{ij}^1\left(1 - w_{ij}^1\right) \qquad \forall i,j \in N, i < j \\
&\qquad x_j + L_j \leq x_i + M_{ij}^2\left(1 - w_{ij}^2\right) \qquad \forall i,j \in N, i < j \\
&\qquad \mathrm{y}_i - H_i \geq y_j - M_{ij}^3\left(1 - w_{ij}^3\right) \qquad \forall i,j \in N, i < j \\
&\qquad \mathrm{y}_j - H_j \geq y_i - M_{ij}^4\left(1 - w_{ij}^4\right) \qquad \forall i,j \in N, i < j \\
&\qquad \sum_{d \in D} w_{ij}^d = 1 \qquad \forall i,j \in N, i < j \\
&\qquad x_i \leq UB_i - L_i \qquad \forall i \in N \\
&\qquad H_i \leq y_i \leq W \qquad \forall i \in N \\
&\qquad lt, x_i, y_i \in \mathbb{R}, \qquad w_{i,j} \in \{0,1\}
\end{aligned}
$$

Introduction to Optimization Modeling in Python

Alba Carrero, Juan Javaloyes & Daniel Vázquez

javaloyes.juan@gmail.com

danielvazquez150791@gmail.com

October 2018