Lecture 17 Nonconvex Optimization Applications

Can Li

ChE 597: Computational Optimization
Purdue University

Nonconvex continuous optimization

$$\begin{array}{ll} \min_x & f(x) \\ \text{subject to} & h_i(x) \le 0, i = 1, \ldots, m \\ & \ell_j(x) = 0, j = 1, \ldots, r \end{array}$$

- $x = (x_1, \cdots, x_n) \in \mathbb{R}^n$: optimization variables
- $f : \mathbb{R}^n \to \mathbb{R}$: objective (or cost) function
- $h_i : \mathbb{R}^n \to \mathbb{R}$: inequality constraints
- $\ell_j : \mathbb{R}^n \to \mathbb{R}$: equality constraints
- feasible region:

$$X = \{x \mid h_i(x) \le 0, i = 1, \cdots, m,\ \ell_j(x) = 0, i = 1, \cdots, r\}$$

where X can be a **nonconvex** set when any of h_i is nonconvex or ℓ_j is nonlinear

MILP$\subseteq$ QCQP

Continuous nonconvex NLP is a very general class of problems and are in general more difficult to solve to global optimality than MILP.
To see this, consider any binary variable $y \in \{0, 1\}$. It is equivalent to adding the following nonconvex quadratic constraint

$$y(1 - y) = 0$$

This shows MILP$\subseteq$ QCQP.

Nonconvex MINLP

$$
\begin{aligned}
\min_{x,y} \quad & f(x,y) \\
\text{s.t.} \quad & g(x,y) \leq 0 \\
& h(x,y) = 0 \\
& x \in \mathbb{R}^{n^x}, y \in \{0,1\}^{n^y}
\end{aligned}
$$

- x are continuous variables with dimension n^x.
- y are binary variables with dimension n^y.
- if f, g are convex, h is linear, it is called **convex MINLP**, i.e., the continuous relaxation is convex. Otherwise, it is called **nonconvex MINLP**.

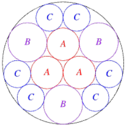
Applications

- Packing problem
- Continuous facility location
- k-means clustering
- Pooling problem
- Molecular structure prediction
- Design of multiproduct plant with single product campaign

Packing problem

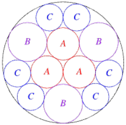
- The goal is to pack objects together into containers or a designated space as densely as possible.
- Application areas
  - Manufacturing and Material Cutting: In industries such as garment, metalworking, and woodworking, the problem involves cutting raw materials into pieces of predefined shapes and sizes while minimizing waste.
  - Logistics and Shipping: Optimizing the placement of goods into containers or trucks to maximize the amount of cargo transported while adhering to weight and space limitations.
  - Computer Science and Data Storage: Allocating memory in computer storage in a manner that maximizes efficiency and minimizes wasted space.

Circle packing problem in a circle



- **Goal:** given small circles, find the smallest circle that can pack all the small circles.
- **Parameters:** Given the the radiuses of all the small circles r_i
- **Variables:** Suppose the center of the big circle is the origin. The coordinates of the small circles are (x_i, y_i).

Circle packing problem in a circle



$$
\begin{aligned}
\min \quad & r_0 \\
\text{s.t.} \quad & \sqrt{x_i^2 + y_i^2} + r_i \leqslant r_0, \quad i = 1, \ldots, N \\
& r_i + r_j \leqslant \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad i, j = 1, \ldots, N; i < j.
\end{aligned}
$$

r_i is the radius of circle i, r_0 is the radius of the big circle to pack all the small circles.

- The first constraint specify each circle is within the big circle.
- The second constraint specify that the circles do not overlap.
- It can be represented as nonconvex QCQP (square term on the rhs): $x_i^2 + y_i^2 \leq (r_0 - r_i)^2$.

Circle packing in a rectangle

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

where A and B and width and height of the container. The origin is at the center of the rectangle. A^{low} and A^{up} are the width requirements (with $A^{\text{low}} \leqslant A^{\text{up}}$) and B^{low} and B^{up} are the height requirements (with $B^{\text{low}} \leqslant B^{\text{up}}$) of the rectangular container.

Continuous facility location

- Recall that in the Uncapacitated Facility Location, the locations of the depots are pre-selected. We had a binary variable to decide whether a depot is installed or not.
- In practice, the locations of the depots can be anywhere on 2D map.
- The continuous facility location problem allows the flexibility in selecting the depot locations.

Continuous facility location

Considers a set N of potential depots and a set M of clients, with associated fixed costs f_j for depots and unit variable transportation costs c from depot j to client i. The location of the depot can be selected anywhere on a 2D plane.

Parameters

- (x_i, y_i) coordinate of the ith client.

Variables

- $z_j = 1$ if depot j is used, $z_j = 0$ otherwise.
- w_{ij} represents the fraction of client i's demand satisfied from depot j.
- (x_j, y_j) coordinate of the jth depot.

Continuous facility location

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} w_{ij} d_{ij} + \sum_{j=1}^{n} f_j z_j$$

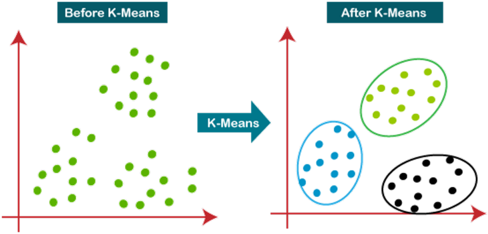
$$\sum_{j=1}^{n} w_{ij} = 1 \text{ for } i \in M \quad \text{demand satisfaction}$$

$$w_{ij} \leq z_j \text{ for } i \in M, j \in N \quad \text{force } w_{ij} \text{ to zero if depot not built}$$

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad \text{distance calculation}$$

$$w_{ij} \geq 0 \text{ for } i \in M, j \in N, \quad z_j \in \{0, 1\} \text{ for } j \in N.$$

k-means clustering



K-means clustering is a partitioning method that divides a dataset into K distinct, non-overlapping subsets (clusters) by minimizing the squared Euclidean distance between data points and the centroid of their assigned cluster.

k-means clustering

Sets

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

Parameters

Without loss of generality, we assume that all points $\mathbf{p}_1, \ldots, \mathbf{p}_N$ have been normalized to reside in a D dimensional hypercube. $0 \leq p_{ij} \leq 1$ where p_{ij} denotes the jth coordinate of the ith data point.

Variables

- 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.

k-means clustering

Big-M MIQCP formulation

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

Pooling problem

Wide applications in petrochemical refining, wastewater treatment, natural gas network design.

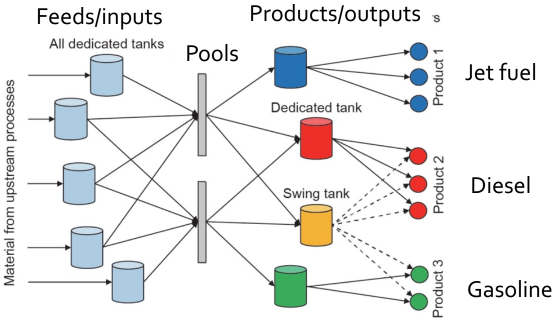


Figure: ref: Castillo et al. (2018)

- Network flow problem on a tripartite directed graph, with three type of node: Input Nodes (I), Pool Nodes (L), Output Nodes (J).
- Send flow from input nodes via pool nodes to output nodes.

Pooling problem

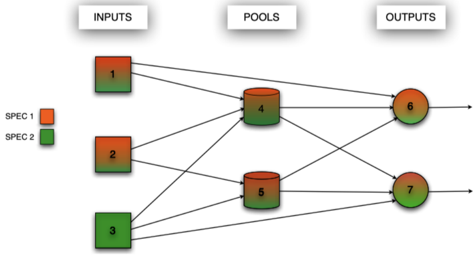


Figure: ref: Dey, 2020

- Raw material has specifications (like sulphur, octane number, etc.).
- Raw material gets mixed at the pool producing new specification level at pools.
- The material gets further mixed at the output nodes.
- The output node has required levels for each specification.

P formulation

Indices

- i - raw materials, $i = 1, \ldots, I$
- j - products, $j = 1, \ldots, J$
- k - qualities, $k = 1, \ldots, K$
- l - pools, $l = 1, \ldots, L$

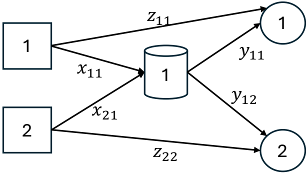
Parameters

- c_i - unit cost of the ith raw material
- d_j - price of jth product
- A_i - availability of ith raw material
- C_{ik} - kth quality of raw material i
- D_j - demand of jth product
- P_{jk}^U - upper bound on kth quality of jth product
- S_l - lth pool capacity

P formulation

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



Note: The connections (pipes) are sparse, i.e., not all the x_{il}, y_{jk}, z_{ij} exist. For example, z_{12}, z_{21} do not exist in the figure. We can treat these variables as fixed at zero in the formulation to simplify notations.

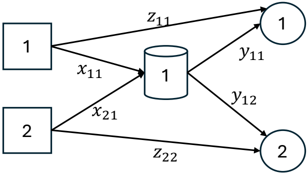
P formulation

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

$$\min \sum_{i=1}^{I}\sum_{l=1}^{L} c_i x_{il} - \sum_{j=1}^{J}\sum_{l=1}^{L} d_j y_{jl} - \sum_{i=1}^{I}\sum_{j=1}^{J}(d_j - c_i) z_{ij}$$

- Ensures the availability of raw materials is not exceeded.

$$\sum_{l=1}^{L} x_{il} + \sum_{j=1}^{J} z_{ij} \leq A_i \quad \forall i$$



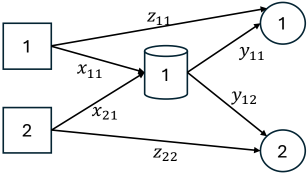
P formulation

- Enforces the conservation of flow into and out of the pools.

$$\sum_{i=1}^{I} x_{il} - \sum_{j=1}^{J} y_{jl} = 0 \quad \forall l$$

- Limits the flow into the pools by their capacity.

$$\sum_{i=1}^{I} x_{il} \leq S_l \quad \forall l$$



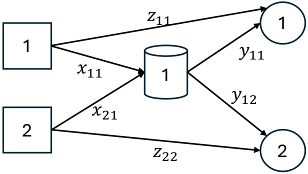
P formulation

- Sales of each product cannot exceed their demands.

$$\sum_{l=1}^{L} y_{jl} + \sum_{i=1}^{I} z_{ij} \leq D_j \quad \forall j$$

- Maintains the quality of the product within upper bounds.

$$\sum_{l=1}^{L} p_{lk} y_{jl} + \sum_{i=1}^{I} C_{ik} z_{ij} \leq p_{jk}^{U} \Big(\sum_{l=1}^{L} y_{jl} + \sum_{i=1}^{I} z_{ij} \Big) \quad \forall j, k$$



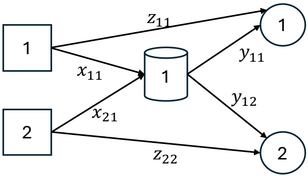
P formulation

- Balances the quality of the streams at the pools.

$$\sum_{i=1}^{I} C_{ik} x_{il} - p_{lk} \sum_{j=1}^{J} y_{jl} = 0 \quad \forall l, k$$

- Variable bounds

$$x_{il} \geq 0, \forall (i,l); \quad y_{lj} \geq 0, \forall (l,j); \quad z_{ij} \geq 0, \forall (i,j); \quad p_{lk} \geq 0, \forall (l,k)$$



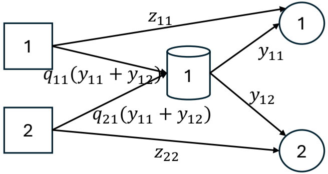
Q formulation

Replace the x_{il} variables (flow from inputs to pools) with variables q_{il} which represents the fraction of the inputs to pool l that comes from input i

$$x_{il} = q_{il} \sum_{j=1}^{J} y_{lj}$$

Since q_{il} represents the fraction, we have

$$\sum_{i=1}^{I} q_{il} = 1$$



Q formulation

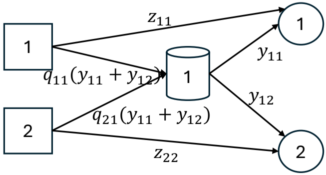
$$
\begin{aligned}
\min \quad & \sum_{j=1}^{J} \left(\sum_{l=1}^{L} y_{lj} \sum_{i=1}^{I} c_i q_{il} - d_j \sum_{l=1}^{L} y_{lj} + \sum_{i=1}^{I} c_i z_{ij} - \sum_{i=1}^{I} d_j z_{ij} \right) \\
\text{s.t.} \quad & \sum_{l=1}^{L} \sum_{j=1}^{J} q_{il} y_{jl} + \sum_{j=1}^{J} z_{ij} \leq A_i, \quad \forall i \\
& \sum_{j=1}^{J} y_{jl} \leq S_l, \quad \forall l \\
& \sum_{l=1}^{L} y_{jl} + \sum_{i=1}^{I} z_{ij} \leq D_j, \quad \forall j \\
& \sum_{l=1}^{L} \left(\sum_{i=1}^{I} C_{ik} q_{il} - P_{jk}^{U} \right) y_{jl} + \sum_{i=1}^{I} \left(C_{ik} - P_{jk}^{U} \right) z_{ij} \leq 0, \quad \forall j, k, l \\
& \sum_{i=1}^{I} q_{il} = 1, \quad \forall l \\
& q_{il} \geq 0, \quad \forall i, l; \quad y_{jl} \geq 0, \quad \forall j, l; \quad z_{ij} \geq 0, \quad \forall i, j.
\end{aligned}
$$

PQ formulation

Add the following constraints to the Q formulation

$$\sum_{i=1}^{I} q_{il} y_{lj} = y_{lj} \quad l = 1, \ldots, L; j = 1 \ldots, J$$

It is a redundant constraint. However, this tightens the convex relaxation, which will discussed in the next lecture.



Molecular structure prediction

Finding the molecular structure with lowest Gibbs free energy.

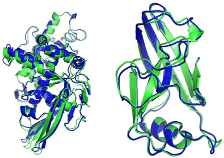


Figure: protein structure prediction

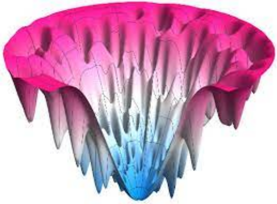
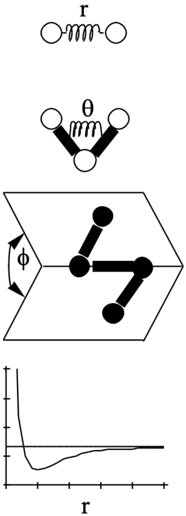


Figure: energy function has multiple local minimums

Molecular structure prediction

The energy function involves (1) bond stretching term. (2) angle bending term. (3) torsion (dihedral) angle term. (4) non-bonded van der Waals interaction term, e.g., Lennard-Jones potential. (5) Coulombic interaction term.

$$
\begin{aligned}
E = & \sum_{(ij)\in B} \left\{ k^b_{ij} \left(r_{ij} - r^0_{ij} \right)^2 \right\} \\
& + \sum_{(ijk)\in\Theta} \left\{ k^\theta_{ijk} \left(\theta_{ijk} - \theta^0_{ijk} \right)^2 \right\} \\
& + \sum_{(ijkl)\in\Phi} \left\{ \left| k^\phi_{ijkl} \right| - k^\phi_{ijkl} \cos\left(n\phi_{ijkl} \right) \right\} \\
& + \sum_{(ij)\in NB} \left\{ \frac{A_{ij}}{r^{12}_{ij}} - \frac{B_{ij}}{r^6_{ij}} \right\} \\
& + \sum_{ij)\in NB} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}
\end{aligned}
$$



Multiproduct plant with single product campaign (SPC) and zero wait policy

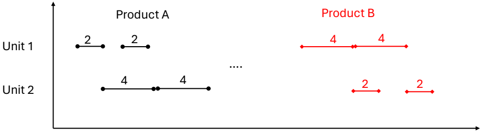


Figure: Single product campaign: produce one product after another. The cycle time is determined by the maximum processing time over each stage. Cycle time for both A and B are 4. The cycle time at for product i is $TL_i = \max_{j \in M} t_{ij}$; M is the set of stages

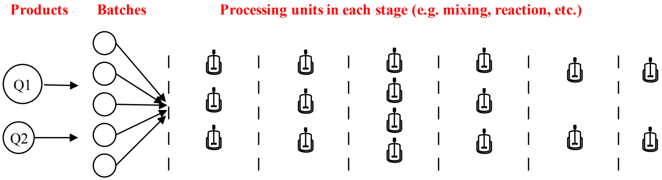


Figure: Each product can be split into indentical batches to be processed by parallel units at each stage. The cycle time $TL_i = \max_{j \in M} \frac{t_{ij}}{N_j}$ where N_j is the number of parallel units in stage j.

Design SPC parallel units

Goal: minimize the investment cost of multi-product SPC plants while satisfying the demand within a given planning horizon.

Sets

- $i \in N$ products
- $j \in M$ stages

Parameters

- t_{ij} processing times of product i in stage j (hours)
- S_{ij} Size factor for product i in processing stage j (L/kg)
- Q_i product demand (kg)
- α_j, β_j investment cost coefficient and exponent for purchasing vessel at stage j, i.e., $\texttt{cost} = \alpha_j V_j^{\beta_j}$ where $0 < \beta_j < 1$ (economy of scale)
- H time horizon. All the demands must be satisfied within the time horizon.

Design SPC parallel units

Variables

- V_j volume/size of unit j
- $N_j \in \mathbb{Z}^+$ number of units parallel stage j.
- B_i batch size of product i
- TL_i cycle time of product i

Objective

Minimize the investment costs of all the vessels

$$\min C = \sum_{j=1}^{M} N_j \alpha_j V_j^{\beta_j}$$

This is clearly a nonconvex objective.

Noncvonex MINLP with posynomial relaxation

Constraints

- Vessel size constraints (linear).

$$V_j \geq S_{ij} B_i, \quad i = 1, 2, \ldots, N, j = 1, \ldots, M$$

- Average cycle time is the maximum over all stages $TL_i = \max_{j \in M} \frac{t_{ij}}{N_j}$.

$$TL_i \geq \frac{t_{ij}}{N_j}, \quad i = 1, 2, \ldots, N, j = 1, \ldots, M \quad \text{(convex)}$$

- All the demand satisfied with the planning horizon

$$\sum_{i=1}^{N} \frac{Q_i}{B_i} TL_i \leq H \quad \text{(nonconvex)}$$

- Variable bounds

$$V_j^L \leq V_j \leq V_j^U, \quad N_j \in \mathbb{Z}^+$$

Take the log transformation of each variable

Consider a new set of variables

$$V_j = e^{v_j}, N_j = e^{n_j}, B_i = e^{b_i}, TL_i = e^{tl_i}$$

$$N_j = \sum_{k=1}^{\bar{N}_j} k y_{jk}, \ \sum_k y_{jk} = 1, y_{jk} = \{0,1\} \ \forall j$$

$$n_j = \sum_{k=1}^{\bar{N}_j} \ln(k) y_{jk}, \ \sum_k y_{jk} = 1, y_{jk} = \{0,1\} \ \forall j$$

where $\bar{N}_j$ denotes the maximum number of units in stage j.
Objective function:

$$N_j \alpha_j V_j^{\beta_j} = \alpha_j \cdot e^{n_j} \cdot e^{\beta_j v_j} = \alpha_j \cdot e^{n_j + \beta_j v_j}$$

It transforms the objective from nonconvex to convex.

Log transformation of constraints

constraints

$$V_j \geq S_{ij} B_j \Rightarrow v_j \geq \ln S_{ij} + b_i \quad \text{linear}$$

$$T_{Li} \geq \frac{t_{ij}}{N_j} \Rightarrow tl_i \geq \ln t_{ij} - n_j \quad \text{linear}$$

$$\sum_i \frac{Q_i}{B_i} TL_i = \sum_i Q_i e^{tl_i - b_i} \quad \text{convex}$$

Advantage: can solve the problem as a convex MINLP (continuous relaxation is convex). It is easier to solve than nonconvex MINLP.

convex MINLP formulation

$$\min C = \sum_{j=1}^{M} \alpha_j e^{n_j + \beta_j v_j} \quad \text{(convex)}$$

$$\text{s.t.} \quad v_j \geq \ln S_{ij} + b_i, \quad \forall i, j \quad \text{(linear)}$$

$$\sum_i Q_i e^{tl_i - b_i} \leq H \quad \text{(convex)}$$

$$tl_i \geq \ln t_{ij} - n_j, \forall i, j \quad \text{(linear)}$$

$$n_j = \sum_{k=1}^{\bar{N}_j} \ln(k) y_{jk}, \quad \forall j \quad \text{(linear)}$$

$$\sum y_{jk} = 1, \quad \forall j \quad \text{(linear)}$$

$$\ln V_j^L \leq v_j \leq \ln V_j^U, \quad \forall j \quad 0 \leq n_j \leq \ln \bar{N}_j, \quad \forall j$$

$$\ln \left[\max_j \left\{ t_{ij} / \bar{N}_j \right\} \right] \leq tl_i \leq \ln \left[\max_j \left\{ t_{ij} \right\} \right], \quad \forall i$$

$$\ln \left[\max_j \{ \frac{Q_i}{\bar{N}_j} \} \right] \leq b_i \leq \ln \left[\max_j \left\{ \frac{V_j^U}{S_{ij}} \right\} \right], \quad \forall i$$

References

- Tawarmalani, M., & Sahinidis, N. V. (2013). Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications (Vol. 65). Springer Science & Business Media.
- Horst, R., & Tuy, H. (2013). Global optimization: Deterministic approaches. Springer Science & Business Media.
- Papageorgiou, D. J., & Trespalacios, F. (2018). Pseudo basic steps: bound improvement guarantees from Lagrangian decomposition in convex disjunctive programming. EURO Journal on Computational Optimization, 6, 55-83.
- Castillo, I., Kampas, F. J., & Pintér, J. D. (2008). Solving circle packing problems by global optimization: numerical results and industrial applications. European Journal of Operational Research, 191(3), 786-802.
- Dey, S, Convexification in global optimization. IPCO summer school (2020)
- Grossmann, I. E., & Sargent, R. W. (1979). Optimum design of multipurpose chemical plants. Industrial & Engineering Chemistry Process Design and Development, 18(2), 343-348.