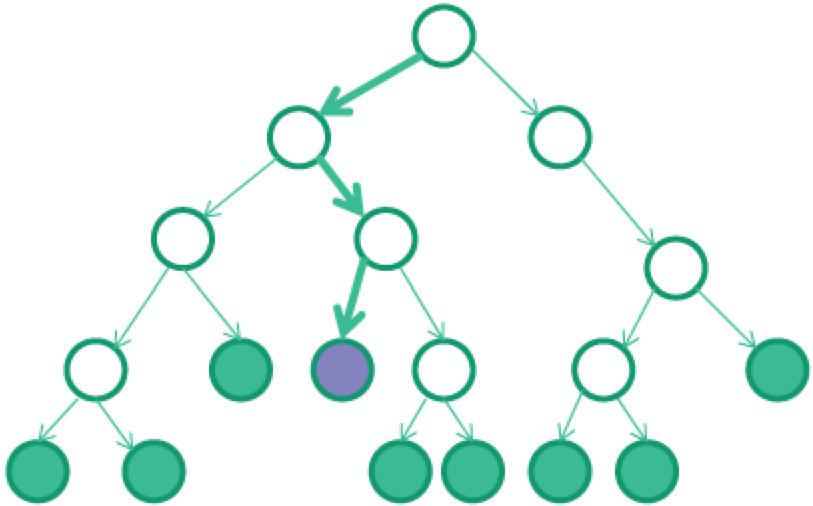
NTNU – Trondheim
Norwegian University of
Science and Technology

Introduction to Mixed Integer Linear Programming



Overview of module:

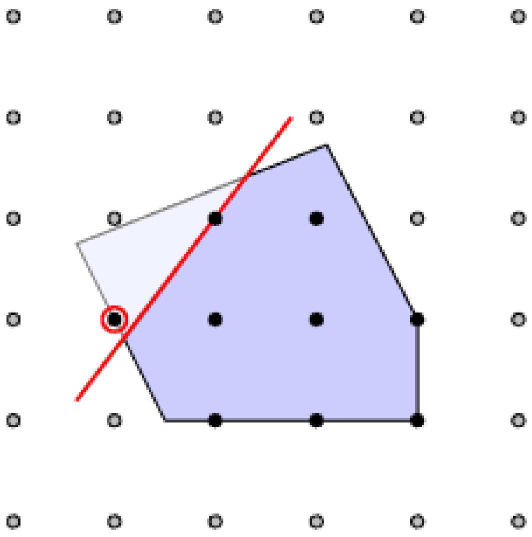
- Introduction and motivation.
- Fundamentals concepts and mathematics in mixed integer linear programming.
- The basic algorithms:
  - Branch-and-bound
  - Branch-and-cut
- Introduction to decomposition approaches in large-scale MILP:
  - Petroleum production optimization.
  - Unit commitment in electric power production.
- Software.

Learning outcome of course module

1. Basic understanding of mixed integer linear programming.
2. Know the basic differences between integer and continuous optimization.
3. Be able to formulate a MIP model based on a problem with discrete decision variables.
4. Knowledge of applications of MIP in control engineering, energy systems and economics.

What is fundamentaly different from continuous optimization?

- The feasible region consists of a set of disconnected integer points.
- Gradient-based algorithms cannot be *directly* applied.
- No conditions similar to the KKT conditions to prove first order optimality.



Source: http://www.zib.de/en/optimization/mip/projects-long/exactip.html

Motivation

- Mixed integer programming is used to solve optimization problems with discrete decisions in a wide range of disciplines:
  - Operations research (production planning, management science, finance, logistics)
  - Electric power production
  - Chemical engineering
  - Petroleum production
  - Control engineering
- The next slides contain particular examples of mixed integer programming applications in these disciplines.

Operations research

Source: Wikipedia

- Designing airline crew schedules:
  - Pair (assign duty periods) airline crews that cover every flight leg at the least cost.
  - Must satisfy legal rules such as limited total flying time, minimum rest time, etc.

- Train scheduling:
  - Find a feasible train schedule that secures sufficient transit time for passengers with connections, assigning trains to single tracks such that train collisions are avoided (hard constraint!), and minimize excessive wait time for trains.

- Production planning:
  - Given a set of X products to be produced in Y factories, with final shipment to Z sales areas.
  - Products are produced in batches, with both fixed and marginal costs.
  - Maximize profit/ minimize cost with respect to seasonal demands.

- Allocating lecture halls at NTNU:

Electric power production

Source: powerop.co.uk

The hydro-thermal unit-commitment (UC) dispatch problem:

Given a set of electric-power generating units with different characteristics:

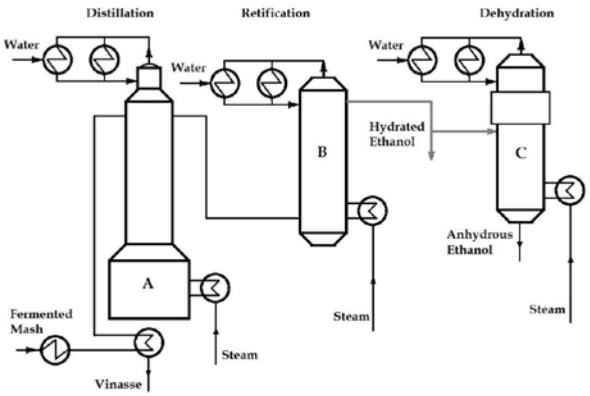
- Maximum output power (e.g. 400 MW).
- Efficiency curves.
- Start-up cost, start-up time and minimum up/down times.
- Emission level constraints.

Given a certain planning horizon (e.g. 24 hours): Select units such that

- The power demand d_t is satisfied for all time periods t.
- Fuel costs or emissions are minimized, or profit is maximized.
- The generating units have a certain excess reserve capacity r_t due to demand uncertainty.
- The unit schedule must satisfy a certain security level.

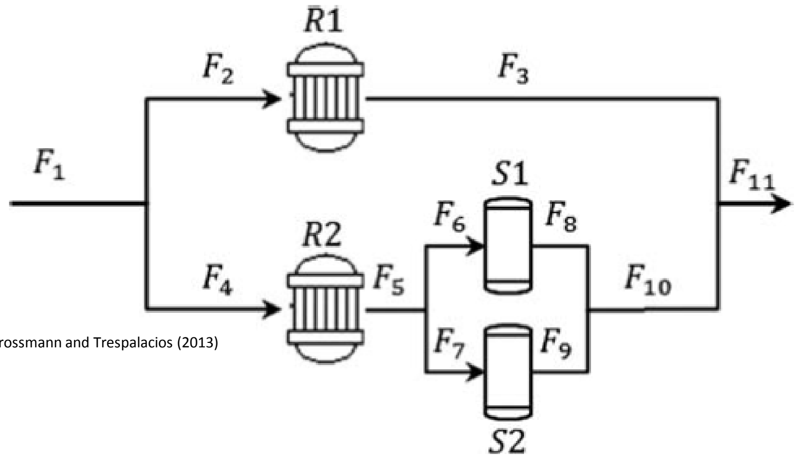
Chemical engineering

- Optimal design of distillation columns: Separation of components in a mixture passed through distillation units. Decision variables can be selecting the number of trays and feed locations, and the location of output streams (products).



Source: intechopen.com

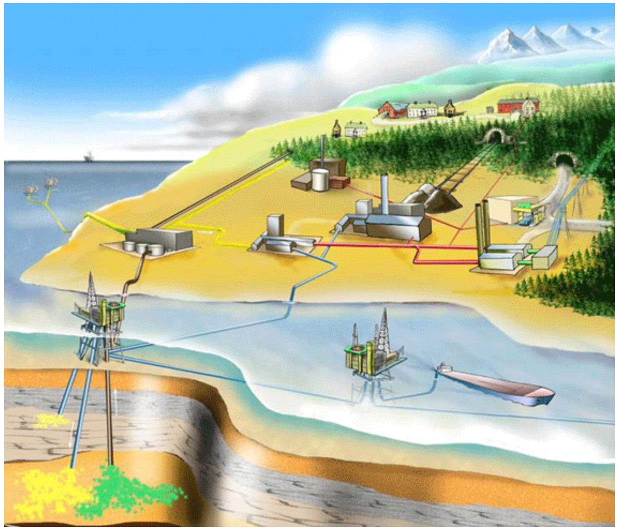
- Used extensively in process design and synthesis, e.g. Optimal reactor selection and configuration:



Source: Grossmann and Trespalacios (2013)

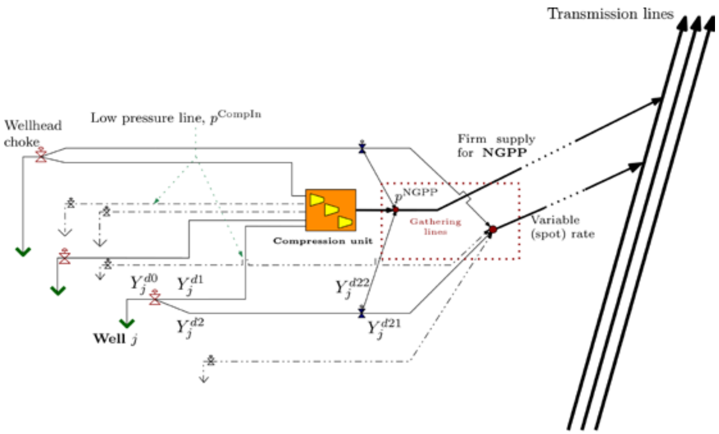
Petroleum production optimization

- Optimization of gas flow and routing in the natural-gas value chain: Meet seasonal varying gas demands, contractual obligations, minimize fuel consumption of compressors, etc.



Source: sintef.no

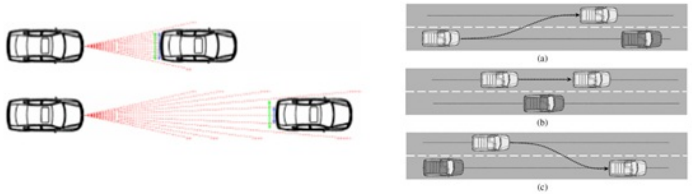
- Maximize revenues of oil and gas subject to constraints in the reservoir and wells, and the gathering system, for instance the capacity of separators and compressors.



Source: Knudsen et. al 2014

Motion control and hybrid systems

- Collision avoidance in trajectory-planning for aircrafts, UAVs and vehicles:
  - Avoid multiple vehicles colliding.
  - Obstacle avoidance for single vehicles.



Source: staff.blog.utm.my/vse/?p=276

- General hybrid *predictive* control: MPC with discrete variables. Numerous applications in chemical, mechanical and electrical engineering. See: http://cse.lab.imtlucca.it/~bemporad/teaching/mpc/imt/6-hybrid-examples.pdf

Definitions of problems with discrete variables

Comments on notation:

$\mathbb{R}^n_+$ is the n-dimensional space of all *non-negative* real numbers:

$$\mathbb{R}^n_+ = \{x \in \mathbb{R}^n : x \geq 0\}$$

$\mathbb{Z}^p_+$ is the p-dimensional space of all *non-negative* integers:

$$\mathbb{Z}^n_+ = \{y \in \mathbb{Z}^p : y \geq 0\}$$

$\mathbb{B}^q$ is the q-dimensional space of all binary variabls:

$$\mathbb{B}^q = \{y : y \in \{0, 1\}^q\}$$

- The expression mixed integer *program* and mixed integer *problem* is used interchangeably, both referring to a mathematical problem with continuous and discrete variables.

Definitions:

General MILP

$$J^* = \min_{(x,y)} \quad c^T x + d^T y$$

$$\text{s.t.}$$

$$Ax + By \geq b$$

$$(x,y) \in \mathbb{R}_+^n \times \mathbb{Z}_+^p$$

- A is an $m \times n$ matrix
- B is an $m \times p$ matrix
- b is an m-dimensional vector
- c is and n-dimensional vector
- d is an p-dimensional vector

We define X as the set of feasible solutions:

$$X = \left\{(x,y) \in \mathbb{R}_+^n \times \mathbb{Z}_+^p : Ax + By \geq b\right\}$$

Mixed binary linear program:

$$
\begin{aligned}
J^* = \min_{(x,y)} \quad & c^T x + d^T y \\
\text{s.t.} \quad & \\
& Ax + By \geq b \\
& x \in \mathbb{R}_+^n \\
& \boxed{y \in \{0,1\}^p}
\end{aligned}
$$

(Linear) Integer program (IP):

$$
\begin{aligned}
J^* = \min_{y} \quad & d^T y \\
\text{s.t.} \quad & By \geq b \\
& y \in \mathbb{Z}_+^p
\end{aligned}
$$

Examples on formulating integer programs (IPs):

The generalized assignment problem (GAP): Given n assignments/tasks and m agents/servers/vehicles to carry out the tasks:

$$
\begin{array}{lll}
i = 1 \dots n & : & \text{index of tasks} \\
j = 1 \dots m & : & \text{index of available agents} \\
d_{ij} & : & \text{cost of assigning task } i \text{ to agent } j \\
b_j & : & \text{resource available from agent } j \\
a_{ij} & : & \text{resource required by agent } j \text{ to do taks } i \\
y_{ij} & : & \text{a binary varible equal to 1 if agent } j \text{ is assigned to do task } i
\end{array}
$$

Problem can be formulated as the linear integer program (IP):

$$
\min_{y} \sum_{i=1}^{n} \sum_{j=1}^{m} d_{ij} y_{ij}
$$

s.t.

$$
\begin{array}{lll}
\displaystyle\sum_{j=1}^{m} y_{ij} = 1, & i = 1 \dots n & : \text{Each task is assigned to exactly one agent} \\
\displaystyle\sum_{i=1}^{n} a_{ij} y_{ij} \leq b_j, & j = 1 \dots m & : \text{Total assignment for agent } j \text{ cannot exceed its capacity} \\
y_{ij} \in \{0, 1\} & &
\end{array}
$$

Numerical example: GAP

Construct and solve the GAP with following specifications:

$n = 3$ tasks and $m = 2$ machines

$$\text{Available resources for machines } j : b_j = \begin{bmatrix} 13 \\ 11 \end{bmatrix}$$

$$\text{Costs} : d_{ij} = \left\{ \begin{array}{ccc} & j_1 & j_2 \\ i_1 & 9 & 2 \\ i_2 & 1 & 2 \\ i_3 & 3 & 8 \end{array} \right.$$

$$\text{Assigment costs for task } i \text{ to machine } j : a_{ij} = \left\{ \begin{array}{ccc} & j_1 & j_2 \\ i_1 & 6 & 8 \\ i_2 & 7 & 5 \\ i_3 & 9 & 6 \end{array} \right.$$

Minimize total costs, assigning each task to one machine.

The Uncapacitated facility location problem (MILP)

Suppose we have m clients, indexed by i, that are to be served by facilities than can be opened at n potential cites (locations), indexed by j. Supplying client i's demand from a facility at location j gives a profit c_{ij}, while there is a cost d_j to open a facility at location j.

Let $y_j = 1$ if facility j is opened, and $y_j = 0$ otherwise. Further, let x_{ij} be the *fraction* of client i's demand that is served by facility j. The problem consists of choosing optimal facility locations and assigning clients to these facilities.

$$\max_{x,y} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} - \sum_{j=1}^{n} d_j y_j$$

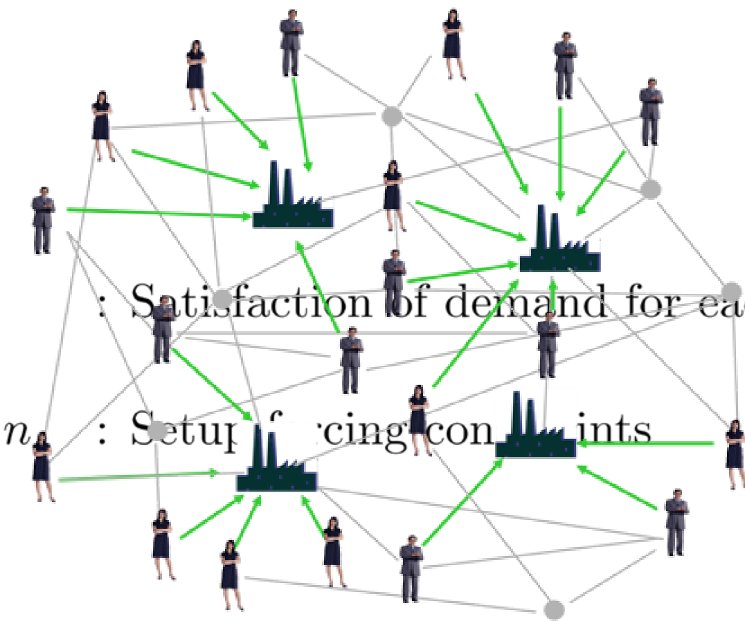
s.t.

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1 \ldots m \qquad : \text{Satisfaction of demand for each client } i$$

$$x_{ij} \le y_j, \qquad i = 1 \cdots m,\ j = 1 \ldots n \qquad : \text{Setup forcing constraints}$$

$$0 \le x_{ij} \le 1$$

$$y_j \in \{0, 1\}$$



Source: http://disopt.epfl.ch/page-31527-en.html

Numerical example (UFL)

Construct and solve the UFL with following specifications:

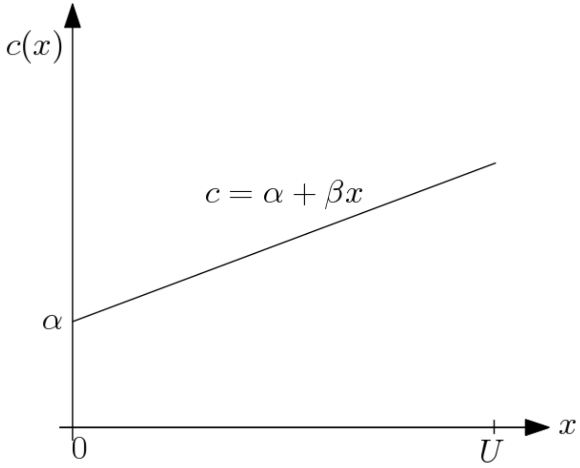
$m = 4$ clients and $n = 2$ facilities

$$\text{Profit coefficients : } c_{ij} = \left\{ \begin{array}{ccc} & j_1 & j_2 \\ i_1 & 9 & 1 \\ i_2 & 4 & 10 \\ i_3 & 3 & 1 \\ i_4 & 7 & 6 \end{array} \right.$$

$$\text{Costs for opening facility } j : d_j = \left\{ \begin{array}{cc} j_1 & 6 \\ j_2 & 8 \end{array} \right.$$

Maximize total profit

General model formulations requiring integer variables



Fixed costs:

A fixed cost α, only present if x>0.
The cost increases with βx:

$$c(x) = \begin{cases} \alpha + \beta x & \text{if } 0 < x \le U \\ 0 & \text{otherwise} \end{cases}$$

By introducing a binary $y \in \{0, 1\}$, we can model this as

$$c = \alpha y + \beta x$$
$$0 \le x \le Uy$$

Similarly for variables only defined in a certain range:

$$Ly \le x \le Uy$$
$$y \in \{0, 1\}$$

Implications and conditions:

Conditions and constraints given by a Boolean Y:

- Condition 1, modeled by $Y_1 = True$
- Condition 2, modeled by $Y_2 = True$

Given expression of the type:

$*$	$Y_1 \Rightarrow Y_2$	(if Y_1 then Y_2)
$**$	$Y_1 \vee Y_2$	(Y_1 or Y_2)
$***$	$Y_1 \Leftrightarrow Y_2$	(if and only if)

Replace Boolean Y with binary $y \in \{0, 1\}$. The given logical conditions can be defined by the constraints

$*$	$y_1 \leq y_2,$
$**$	$y_1 + y_2 \geq 1,$
$***$	$y_1 = y_2,$

respectively. General rules for reformulations logical conditions exist, see

Raman, R., & Grossmann, I. E. (1994). Modelling and computational techniques for logic based integer programming. *Computers and Chemical Engineering*, 18(7), 563–578.

Systematical derivation of linear inequalities from logic propositions

Goal: convert logical expressions to

$$Q_1 \wedge Q_2 \wedge \ldots \wedge Q_n \tag{1}$$

where each logical clause consists of expressions

$$Q_i \;:\; Y_1 \vee Y_2 \vee \ldots \vee Y_u \tag{2}$$

Steps:

1. Replace implication by disjunction:

$$Y_1 \Rightarrow Y_2 \quad \Leftrightarrow \quad \neg Y_1 \vee Y_2$$

2. If necessary, (particularly with several logical terms), apply DeMorgan's rules to move negation inward

$$\begin{aligned} \neg(Y_1 \vee Y_2) &\Leftrightarrow \neg Y_1 \wedge \neg Y_2 \\ \neg(Y_1 \wedge Y_2) &\Leftrightarrow \neg Y_1 \vee \neg Y_2 \end{aligned}$$

3. If more than two Booleans, recursively distribute OR operator over AND to get expressions of the form (1)–(2)

$$(Y_1 \wedge Y_2) \vee Y_3 \;\Leftrightarrow (Y_1 \vee Y_3) \wedge (Y_2 \vee Y_3) \tag{3}$$

4. Replace Booleans Y_i with binary y_i. Each clause with only OR operators defines linear inequalities. An AND operator as in (3) gives an additional inequality, i.e., (3) results in the constraints

$$\begin{aligned} y_1 + y_3 &\geq 1 \\ y_2 + y_3 &\geq 1 \end{aligned}$$

Example

If product A is chosen, product B cannot be chosen while product C have to be chosen. Define Booleans Y_i, for $i = A, B, C$.

1. Replace implication by disjunction:

$$Y_1 \Rightarrow Y_2 \quad \Leftrightarrow \quad \neg Y_1 \vee Y_2$$

2. If necessary, (particularly with several logical terms), apply DeMorgan's rules to move negation inward

$$\neg(Y_1 \vee Y_2) \ \Leftrightarrow \neg Y_1 \wedge \neg Y_2$$
$$\neg(Y_1 \wedge Y_2) \ \Leftrightarrow \neg Y_1 \vee \neg Y_2$$

3. If more than two Booleans, recursively distribute OR operator over AND to get expressions of the form (1)–(2)

$$(Y_1 \wedge Y_2) \vee Y_3 \ \Leftrightarrow (Y_1 \vee Y_3) \wedge (Y_2 \vee Y_3) \tag{1}$$

4. Replace Booleans Y_i with binary y_i. Each clause with only OR operators defines linear inequalities. An AND operator as in (3) gives an additional inequality, i.e., (3) results in the constraints

$$y_1 + y_3 \geq 1$$
$$y_2 + y_3 \geq 1$$

$$Y_A \Rightarrow \neg Y_B \wedge Y_C$$
$$\Updownarrow$$
$$\neg Y_A \vee (\neg Y_B \wedge Y_C)$$
$$\Updownarrow$$
$$(\neg Y_A \vee \neg Y_B) \wedge (\neg Y_A \vee Y_C)$$

Replace Y_i's with binaries and rewrite as linear constraints:

$$1 - y_A + 1 - y_B \geq 1$$
$$1 - y_A + y_C \geq 1$$

$$\boxed{\begin{aligned} 1 - y_A &\geq y_B \\ y_C &\geq y_A \end{aligned}}$$

Assignment: modeling logical conditions with binaries

(From Wolsey (1998)). Suppose you are interested in choosing a set of investments $\{1, \ldots, 7\}$. Model the following constraints:

1. You cannot invest in all of them.
2. You must choose at least one of them.
3. Investment 1 cannot be chosen if investment 3 is chosen.
4. Investment 4 can be chosen only if investment 2 is also chosen.
5. You must choose either both investments 1 and 5 or neither.

Disjunctive constraints:

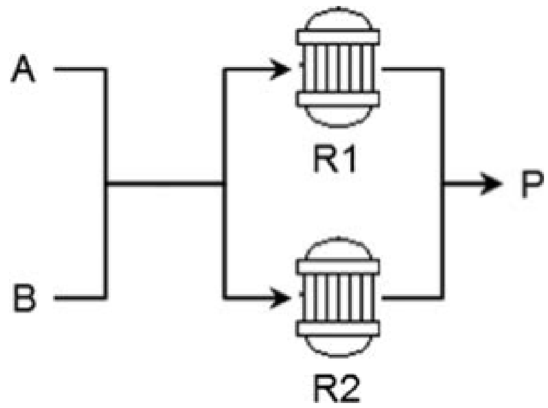
Given $x \in \mathbb{R}$ with lower and upper bound, $0 \leq x \leq U$, and two linear constraints:

$$\underbrace{[ax \leq b]}_{R1} \veebar \underbrace{[dx \leq e]}_{R2}$$

where only one must hold:

$$\begin{aligned} ax &\leq b + M(1 - y_1) \\ dx &\leq e + M(1 - y_2) \\ y_1 + y_2 &= 1 \\ y_1, y_2 &\in \{0, 1\} \end{aligned}$$

where M is a sufficiently large big-M parameter, $M \geq \max(b, e)$.



Source: Grossmann and Trespalacios (2013)

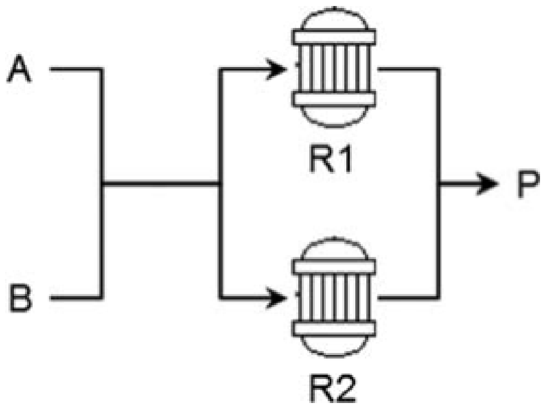
Alternatively, use the extended, but tighter, convex hull reformulation of linear disjuncitons (more on this later):

$$\begin{aligned} & x = z_1 + z_2, \\ & az_1 \leq by_1, \\ & dz_2 \leq ey_2, \\ & y_1 + y_2 = 1, \\ & 0 \leq z_i \leq Uy_i, \quad i = 1, 2 \\ & y_1, y_2 \in \{0, 1\} \end{aligned}$$

Example with disjunctive constraints:

Given the structure of a reactor and raw material selection with the following specifications:

- Objective: maximize profit of selling product P with price 10.
- To produce P, the options are:
  1. Buy reactor R1 with cost C = 5*F (flow), and with 90% conversion of material A and 70% of B.
  2. Buy reactor R2 with cost C = 4.6*F (flow), and with 85% conversion of material A and 80% of B.
- The cost of raw material A is 1.1, and available feed rate is 5.
- The cost of raw material B is 1, and available feed rate is 7.



Source: Grossmann and Trespalacios (2013)

Assignment:

1. Formulate the optimization problem using linear disjunctions.
2. Formulate the corresponding MILP using big-M reformulation.

How to prove optimality?

Given an integer program (IP)

$$
\begin{aligned}
J^* = \min_{y} \quad & d^T y \\
\text{s.t.} \quad & By \geq b \\
& y \in \mathbb{Z}_+^p
\end{aligned}
$$

- Total enumeration? In case y is binary, this corresponds to 2^p possible combinations of integers $\Rightarrow$ quickly becomes an infeasible approach.
- **Fundamental approach**: Iteratively generate a decreasing sequence of upper bounds $\bar{J}_i$ and a sequence of increasing lower bounds $\underline{J}_i$, and then stop when

$$\bar{J}_i - \underline{J}_i < \epsilon$$

for some small $\epsilon \geq 0$. Then, how to generate this sequences of points?

Relaxations

MILP:

$$J^* = \min_{(x,y)} \quad c^T x + d^T y$$

$$\text{s.t.}$$

$$Ax + By \geq b$$

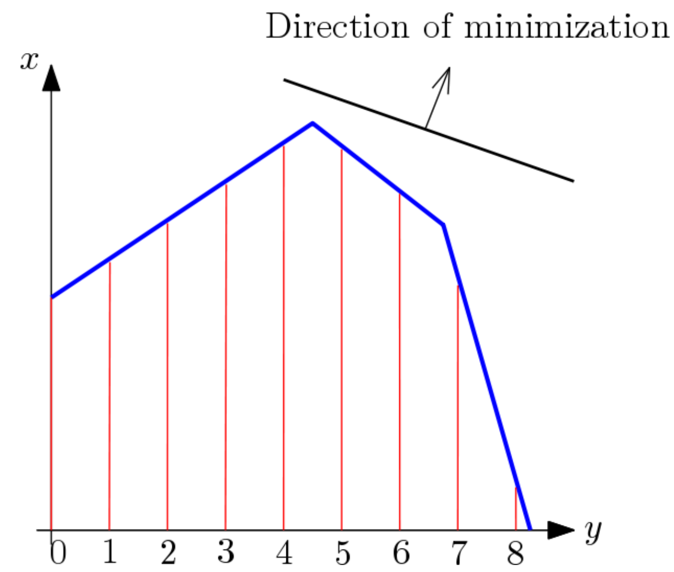
$$(x, y) \in \mathbb{R}_+^n \times \mathbb{Z}_+^p$$

The basic idea of a relaxation is to replace a "difficult" problem with a simpler optimization problem which provides a lower bound for *J* (minimization):

$$J_\mathrm{R} \leq J$$

Two possibilities:

- Enlarge the set of feasible solutions.
- Replace the objective function by a function which is guaranteed to be smaller over the entire set of feasible solutions.



LP relaxation of MILP

MILP:

$$J^* = \min_{(x,y)} \quad c^T x + d^T y$$
$$\text{s.t.} \quad (x,y) \in X$$

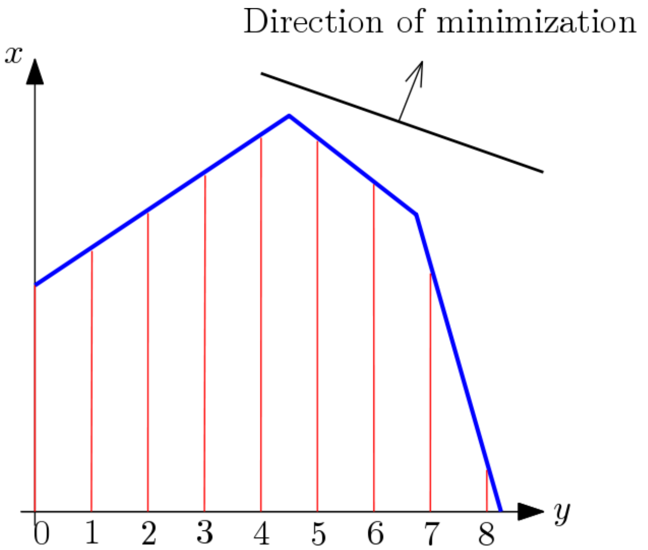
$$X = \left\{(x,y) \in \mathbb{R}^n_+ \times \mathbb{Z}^p_+ : Ax + By \geq b\right\}$$

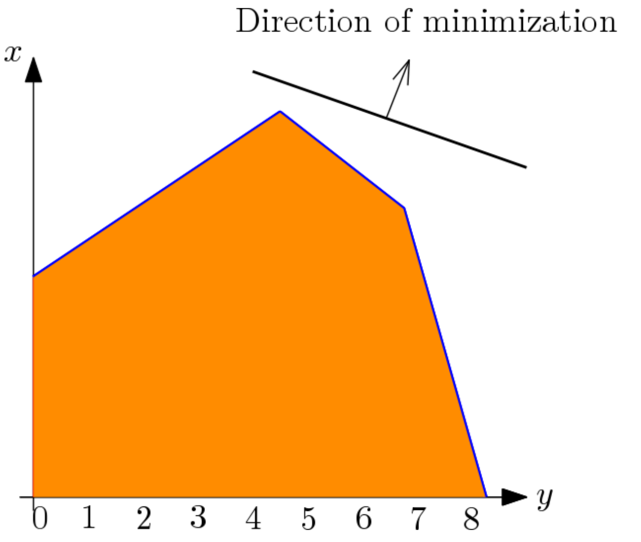
Relaxing integrality condition on y

LP:

$$J_{\mathrm{R}} = \min_{(x,y)} \quad c^T x + d^T y$$
$$\text{s.t.} \quad (x,y) \in P$$

$$P = \left\{(x,y) \in \mathbb{R}^n_+ \boxed{\times \mathbb{R}^p_+} : Ax + By \geq b\right\}$$





Corresponding LP relaxation of binaries

IP:

$$\begin{aligned} J^* = \min_{y} \quad & d^T y \\ \text{s.t.} \quad & By \geq b \\ & y \in \{0,1\}^p \end{aligned}$$

Relaxing 0/1 condition on y

$\longrightarrow$

LP:

$$\begin{aligned} J_{\mathrm{R}} = \min_{y} \quad & d^T y \\ \text{s.t.} \quad & By \geq b \\ & y \in [0,1]^p \end{aligned}$$

With the feasible set defined as

$$X = \{y \in \{0,1\}^p : By \geq b\}$$

The relaxed set

$$P = \{y \in [0,1]^p : By \geq b\}$$

is called a *formulation* for X, while

$$J_{\mathrm{R}} = \min\{d^T y : y \in P\}$$

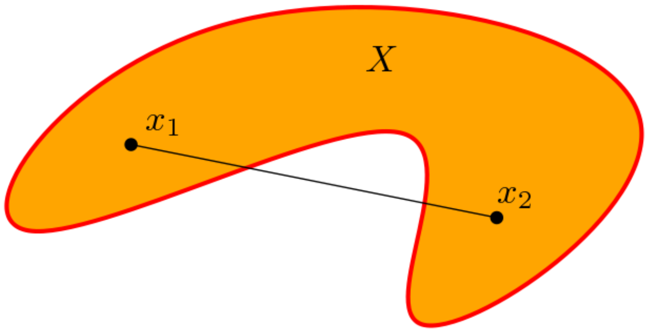
is defined as the linear programming relaxation of the IP with $J_{\mathrm{R}} \leq J$.

Basics of polyhedral theory (1/3)

A set X is *convex* if, for any two points $x_1, x_2 \in X$ and for any θ, $0 \leq \theta \leq 1$, where $X \subset \mathbb{R}^n$, the line segment connecting the two points lies entirely in X:

$$\theta x_1 + (1-\theta)x_2 \in X$$

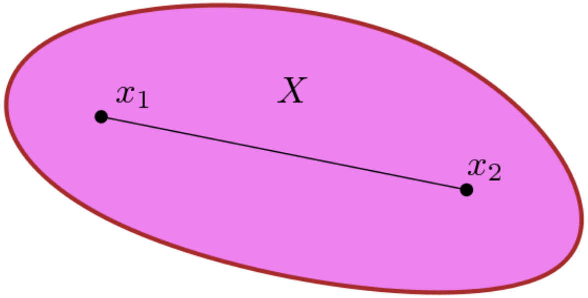
Nonconvex set



A *polyhedron* $P \subseteq \mathbb{R}^n$ is defined as the set of points that satisfies a finite number of linear inequalities :

$$P = \{x \in \mathbb{R}^n \;:\; Ax \leq b\}$$

Convex set



A polyhedron is called a *polytope* if it is bounded, that is, there is an $\epsilon \in [0, \infty)$ such that $P \subseteq \{x \in \mathbb{R}^n : -\epsilon \leq x_i \leq \epsilon,$ for $j = 1 \ldots n\}$

Basics of polyhedral theory (2/3)

A convex combination is a point $z = \theta_1 x_1 + \ldots \theta_n x_n$ with $\theta_1 + \ldots + \theta_n = 1$, $\theta_i \geq 0$ and $x_i \in X$ for all n. The *convex hull* of a set X is the set of all points that are convex combinations of points in X:

$$\text{conv}(X) = \left\{ z = \sum_{i=1}^{n} \theta_i x_i \; : \; x_i \in X, \; \sum_{i=1}^{n} \theta_i = 1, \theta_i \geq 0 \right\}$$

(i) The convex hull conv(X) is the smallest convex set that contains X.

(ii) conv(X) is a ***polyhedron***.

Given IP:

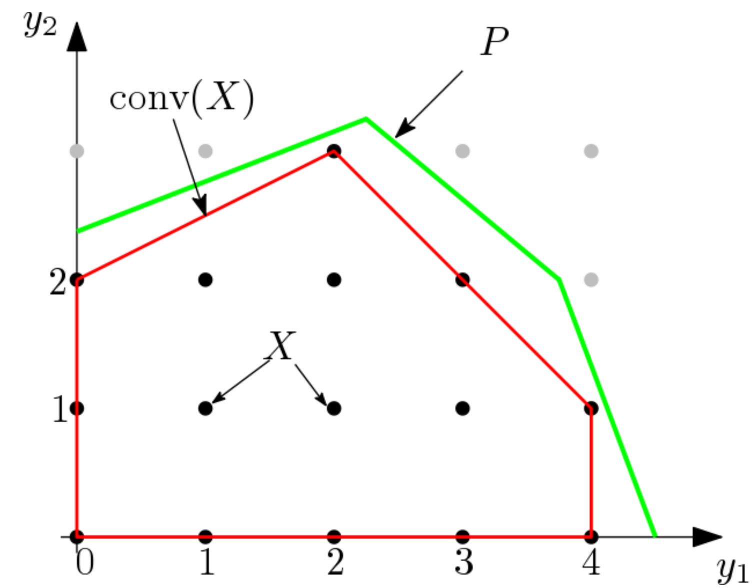
$$\begin{aligned} J^* = &\min_{y} \quad d^T y \\ &\text{s.t.} \quad y \in X \end{aligned}$$

$$X = \left\{ y \in \mathbb{Z}^p_+ : By \geq b \right\}$$
$$P = \left\{ y \in \mathbb{R}^p_+ : By \geq b \right\}$$
$$\text{conv}(X) = \left\{ y \in \mathbb{R}^p : \tilde{B} y \geq \tilde{b} \right\}$$

● points

green

red

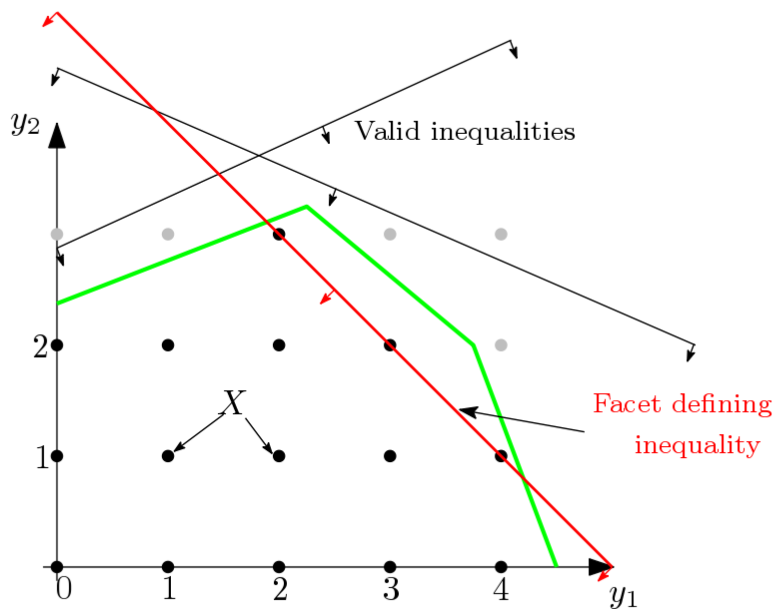


Basics of polyhedral theory (3/3)

- An inequality $\pi y \leq \pi_0$ is a valid inequality (constraint) for a set $X \subseteq \mathbb{R}^p$ if is satisfied by all points in X, i.e.

$$\pi y \leq \pi_0 \qquad \forall x \in X$$

- Only those valid inequalities are necessary for describing the polyhedron defining conv(X) is of real interest; these are called facet-defining inequalities.

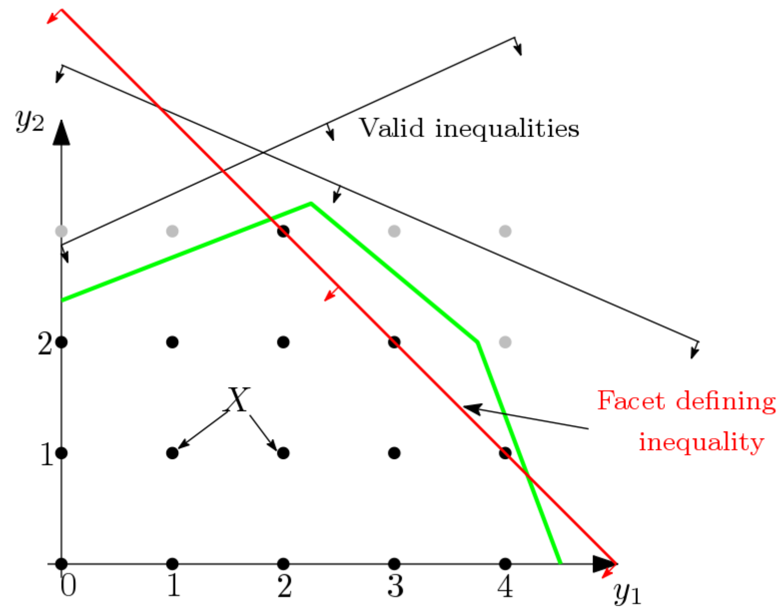


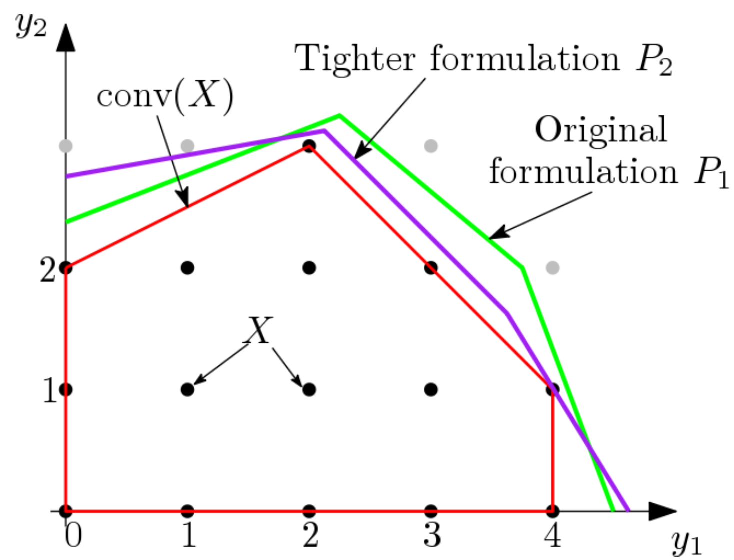
Valid inequalities and different formulations

- An inequality $\pi y \leq \pi_0$ is a valid inequality (constraint) for a set $X \subseteq \mathbb{R}^p$ if is satisfied by all points in X, i.e.
$$\pi y \leq \pi_0, \qquad \forall x \in X$$
- An inequality valid for a relaxation P of X is also valid for X.
- Valid inequalities are added to *strengthen* the formulation P_i of the integer feasible set X.

$$\Downarrow$$

- An IP(MILP) may have infinitely many formulations P_i,
$$P_i = \left\{ y \in \mathbb{R}^p_+ : By \geq b \right\}$$
rendering the same optimal integer solution y^*.
- We always seek a *tight formulation*: The tightest possible is conv(X).





Solving IP as LP:

If we know the complete description of conv(X), we can solve an IP as an LP:

$$\begin{aligned} J^* = \min_y \quad & d^T y \\ \text{s.t.} \quad & y \in X \\ & X = \{\boxed{y \in \mathbb{Z}^p} : By \geq b\} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} J^* = \min_y \quad & d^T y \\ \text{s.t.} \quad & y \in \text{conv}(X), \\ & \text{conv}(X) = \{\boxed{y \in \mathbb{R}^p_+} : \tilde{B}y \geq \tilde{b}\} \end{aligned}$$

Same equivalence holds for MILPs with rational matrices.

However, finding the complete polyhedral description of conv(X) is at least as difficult as solving the IP.

- Number of constraints of conv(X) can be exponential in the size of $By \geq b$.
- Solving IPs and MILPs are NP-hard.

The two basic approaches for solving MILP/IP

Solve LP relaxation. Fractional solutions must be eliminated:

- Iteratively decompose the feasible region and solve new LP relaxations.
- Add valid inequalities that cuts of the integer infeasible point.

Defines the basis for the branch-and-bound, cutting-plane and branch-and-cut algorithm.