Mathematical Programming II

Chapter 2: Integer Programming

Outline

Outline

Motivation

- Every day we face Operations Research problems, such as:
  - Determine the fastest route to ISEG.
  - Decide where to do the weekly shopping (distance versus prices).

- To formulate these problems, we often need integer variables.

Examples of Operations Research applications

European Journal of Operational Research
Volume 240, Issue 3, 1 February 2015, Pages 718-733

Decision Support

An optimization framework for the development of efficient one-way car-sharing systems

Burak Boyacı [a b], Konstantinos G. Zografos [b], Nikolas Geroliminis [a]

(a) 2017

European Journal of Operational Research
Volume 257, Issue 3, 16 March 2017, Pages 992-1004

Innovative Applications of O.R.

Inventory rebalancing and vehicle routing in bike sharing systems

J. Schuijbroek [a], R.C. Hampshire [1 b], W.-J. van Hoeve [c]

(b) 2019

European Journal of Operational Research
Volume 271, Issue 3, 16 December 2018, Pages 1085-1099

Innovative Applications of O.R.

Scheduling last-mile deliveries with truck-based autonomous robots

Nils Boysen [1 a], Stefan Schwerdfeger [b], Felix Weidinger [1 a]

(c) 2020

European Journal of Operational Research
Volume 292, Issue 1, 1 July 2021, Pages 250-275

Innovative Applications of O.R.

Building disaster preparedness and response capacity in humanitarian supply chains using the Social Vulnerability Index

Douglas Alem [a], Hector F. Bonilla-Londono [b], Ana Paula Barbosa-Povoa [c], Susana Relvas [c], Deisemara Ferreira [d], Alfredo Moreno [e]

(d) 2023

Figure 1: Awards for innovative OR applications published in the scientific journal *European Journal of Operations Research*.

Outline

Introduction

A mixed-integer linear programming problem (MILP) is a linear programming problem in which some variables are required to take integer values.
$\Longrightarrow$ Loss of the divisibility property of LP.

- In an integer linear programming problem (ILP), all variables are required to take integer values.
- In a binary linear programming problem, all variables are required to be binary, i.e., to take value 0 or 1.

A combinatorial optimization problem (COP) is a MILP whose feasible region (FR) is a finite set.

- The optimal solution belongs to a finite set – the FR.

Introduction

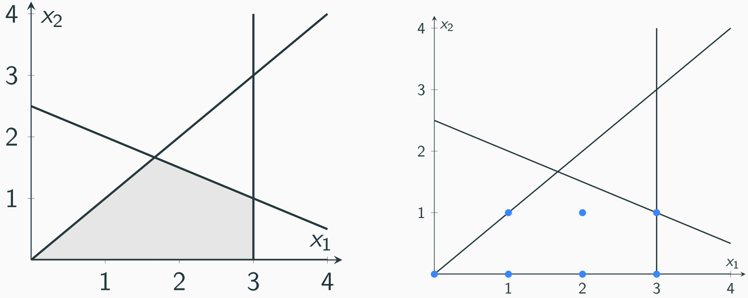


Figure 2: FR of an LP versus FR of a COP.

Introduction

Examples of combinatorial optimization problems include:

- the assignment problem;
- the transportation problem;
- the knapsack problem;
- the *lot-sizing* problem;
- the traveling salesman problem; or
- the vehicle routing problem.

Introduction

How can we solve a combinatorial optimization problem?

- Since the FR is a finite set, we can enumerate all feasible solutions (FS) and determine the best one. $\Longrightarrow$ Enumeration!

- Enumerating the FR of a COP can be very time-consuming.

Which algorithms can we use to determine the optimal solution of a COP?

- *Branch-and-bound* algorithm
- Gomory cutting-plane algorithm

Outline

Knapsack Problem

Given:

- a set of items $\mathcal{I}$, where item $i \in \mathcal{I}$ has weight p_i and value v_i; and
- a knapsack with capacity P;

the Knapsack Problem consists of determining which items should be included in the knapsack so as to maximize the total value without exceeding its capacity.

Knapsack Problem: an instance

Consider a knapsack with capacity 10 and a set of six items with the following characteristics:

Item (i)	1	2	3	4	5	6
Weight (p_i)	7	2	2	4	6	8
Value (v_i)	10	3	5	5	8	12

Can you give me examples of feasible solutions?

Knapsack Problem: formulation

- Variables:

$$x_{ij} = \begin{cases} 1, & \text{se o item } i \in \mathcal{I} \text{ é escolhido} \\ 0, & \text{otherwise} \end{cases}$$

- Formulation:

$$\begin{aligned} \max \sum_{i \in \mathcal{I}} v_i x_i & \\ \text{subject to: } \sum_{i \in \mathcal{I}} p_i x_i \leq P & \\ x_i \in \{0, 1\} & \qquad \forall i \in \mathcal{I} \end{aligned}$$

Knapsack Problem: examples of logical constraints

- Either item 2 or item 3 must be selected
  - $x_2 + x_3 \geq 1$
- At most 3 items may be selected
  - $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$
- If item 2 is selected, then item 4 must also be selected
  - $x_2 \leq x_4$
- If item 1 is selected, then item 3 cannot be selected
  - $x_1 + x_3 \leq 1$
- If item 1 is selected, at least one additional item must be selected
  - $x_1 \leq x_2 + x_3 + x_4 + x_5 + x_6$
- If item 1 is selected, items 3 and 4 must also be selected
  - $x_1 \leq x_3$ and $x_1 \leq x_4$
- If item 2 is selected, neither item 3 nor item 4 may be selected
  - $x_2 + x_3 \leq 1$ and $x_2 + x_4 \leq 1$

Lot-Sizing Problem

Given:

- a production line for a single product;
- a planning horizon consisting of a set T of periods;
- the demand d_t for the product in period $t \in T$;
- the unit production cost p_t in period $t \in T$;
- the fixed production cost f_t in period $t \in T$; and
- the cost h_t of holding one unit of the product in inventory during period $t \in T$;

the Lot-Sizing Problem consists of determining how many units to produce in each period to satisfy demand, taking into account that demand may be met by units produced in the current period or held in inventory, so as to minimize total production and inventory costs.

Lot-Sizing Problem: an instance

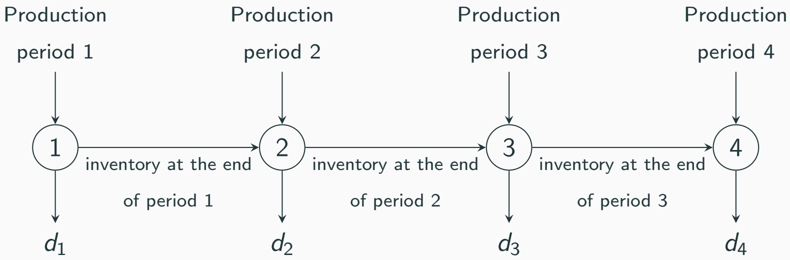
Consider a planning horizon with $T = \{1, 2, 3, 4\}$ periods. In each period, the unit production cost is 1 m.u., the fixed production cost is 10 m.u., and the inventory holding cost is 1 m.u. per unit per period, i.e., $p_t = h_t = 1$ and $f_t = 10, \forall t \in T$. Demand in each period is shown in the following table:

Period (t)	1	2	3	4
Demand (d_t)	7	6	8	4

Can you give me examples of feasible solutions?

Lot-Sizing Problem: an instance

- Schematic representation:



Lot-Sizing Problem: formulation

- Variables:

x_t : quantity produced in period $t \in T$

s_t : inventory at the end of period $t \in T \cup \{0\}$

$$y_t = \begin{cases} 1, & \text{if production takes place in period } t \in T, \\ 0, & \text{otherwise.} \end{cases}$$

- Formulation:

$$\begin{aligned} \min \quad & \sum_{t=1}^{T} (f_t y_t + p_t x_t + h_t s_t) & \\ & s_{t-1} + x_t = d_t + s_t & t \in T \\ & s_0 = 0 & \\ & x_t \leq M_t y_t & t \in T \\ & x_t \geq 0,\ s_t \geq 0,\ y_t \in \{0, 1\} & t \in T \end{aligned}$$

- M_t is a sufficiently large constant

Lot-Sizing Problem: examples of variable-bound constraints

- If production takes place in period t, the production line must be operating
  - $x_t \leq M_t y_t$ $\quad$ M_t is an upper bound on the quantity produced in period t
- If the production capacity in period t is C_t
  - $x_t \leq C_t y_t$
- If production takes place in period t, at least L_t units must be produced
  - $x_t \geq L_t y_t$ $\quad$ e $\quad$ $x_t \leq M_t y_t$

Traveling Salesman Problem

Given:

- a set of cities N; and
- travel costs between each pair of cities c_{ij}, $i, j \in N : i \neq j$,

the Traveling Salesman Problem consists of determining the minimum-cost tour that visits each city exactly once.

William Cook lecture on the traveling salesman problem:
`https://www.youtube.com/watch?v=5VjphFYQKj8&t=55s` [Euro 2019, Dublin]
Website with a traveling salesman problem *solver*:
`https://www.math.uwaterloo.ca/tsp/index.html`

Traveling Salesman Problem: an instance

Consider a set of five cities $\{1, 2, 3, 4, 5\}$ with the following travel costs between them:

	1	2	3	4	5
1	-	3	5	19	2
2	3	-	14	15	11
3	5	14	-	20	1
4	19	15	20	-	9
5	2	11	1	9	-

Can you give me examples of feasible solutions?

Traveling Salesman Problem: formulation

- Variables:

$$x_{ij} = \begin{cases} 1, & \text{if arc } (i,j) \in A \text{ is traversed by the traveling salesman,} \\ 0, & \text{otherwise.} \end{cases}$$

u_i - position of node $i \in N \setminus \{1\}$ in the tour.

- Formulation:

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

subject to:

$$\sum_{i\in N} x_{ji} = 1 \qquad \forall j \in N \qquad \text{(one arc leaves j)}$$

$$\sum_{i\in N} x_{ij} = 1 \qquad \forall j \in N \qquad \text{(one arc enters j)}$$

$$u_i + (|N|-1)x_{ij} \leq u_j + |N| - 2 \qquad \forall (i,j) \in A : i,j \neq 1$$

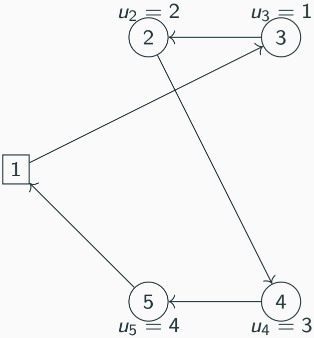
(if arc (i,j) is used, the position of j follows the position of i)

$$1 \leq u_i \leq |N| - 1 \qquad \forall i \in N \setminus \{1\} \qquad \text{(bounds on variables u)}$$

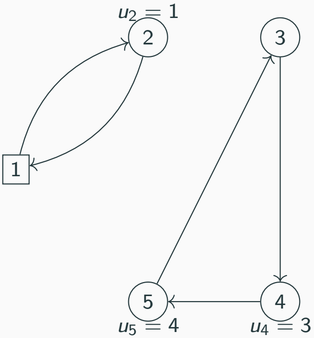
$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

Traveling Salesman Problem: formulation

u_3 cannot be simultaneously 5 and 2



(a) Feasible TSP solution.



(b) Infeasible TSP solution.

Disjunction of constraints

- We have already seen how to formulate some specific conditions:
  - Logical constraints
    - Examples: incompatible items and a limit on the number of selected items.
  - Quantity bounds
    - Examples: quantidades mínimas e quantidades máximas.

We now consider how to formulate a disjunction of constraints.

Disjunction of constraints: example

Consider the following constraints:

- A: $3x_1 + 4x_2 \leq 5$
- B: $5x_1 + 2x_2 \leq 2$

One of these constraints must be satisfied (either constraint A or constraint B is satisfied).

■ Variables:

$$y = \begin{cases} 1 & \text{if constraint A is satisfied} \\ 0 & \text{if constraint B is satisfied} \end{cases}$$

■ Formulation:

$$3x_1 + 4x_2 \leq 5 + M(1 - y)$$
$$5x_1 + 2x_2 \leq 2 + My$$
$$y \in \{0, 1\}$$

- M is a sufficiently large constant

Disjunction of constraints: general case

Consider the following constraints:

- A^i: $a^i x \leq b_i$, for $i = 1, \ldots, m$

At least k of these constraints must be satisfied.

- Variables:

$$y_i = \begin{cases} 1 & \text{if constraint } A^i \text{ is satisfied, for i = 1, ..., m} \\ 0 & \text{otherwise} \end{cases}$$

- Formulation:

$$\begin{aligned} & a^i x \leq b_i + M(1 - y_i), && i = 1, \ldots, m \\ & \sum_{i=1}^{m} y_i \geq k \\ & y_i \in \{0, 1\}, && i = 1, \ldots, m \end{aligned}$$

- M is a sufficiently large constant

Outline

Basic concepts: relaxation

Definition: Relaxation

A given problem

$$(RP) \qquad z^R = \min\{f(x) : x \in T \subseteq \mathbb{R}^n\}$$

is a *relaxation* of

$$(IP) \qquad z = \min\{c(x) : x \in X \subseteq \mathbb{R}^n\}$$

if:

(i) $X \subseteq T$; e

(ii) $f(x) \leq c(x), \forall x \in X$.

Basic concepts: relaxation

Proposition

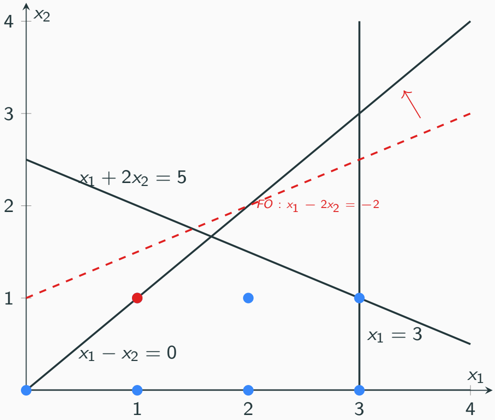
If RP is a relaxation of IP, then $z^R \leq z$.

Proof.

If x^* is an optimal solution of IP, then $x^* \in X$ and $z = c(x^*)$. Since RP is a relaxation of IP, we know that $X \subseteq T$ and $f(x^*) \leq c(x^*)$. Therefore, $z^R \leq f(x^*) \leq c(x^*) \leq z$. $\square$

Example

$$(IP) \equiv \min z = x_1 - 2x_2$$
$$s.t. : x_1 - x_2 \geq 0$$
$$x_1 + 2x_2 \leq 5$$
$$x_1 \leq 3$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

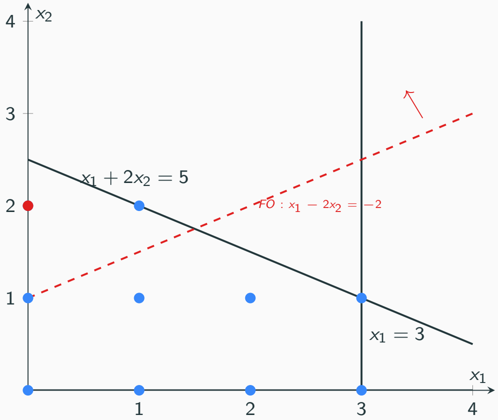


$$v(IP) = c(1,1) = 1 \times 1 - 2 \times 1 = -1$$

Example

$$(RL_1) \equiv \min z = x_1 - 2x_2$$
$$s.t. : x_1 + 2x_2 \leq 5$$
$$x_1 \leq 3$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

Constraint removed: $x_1 - x_2 \geq 0$.



$$v(RL_1) = c(0, 2) = 1 \times 0 - 2 \times 2 = -4$$

Basic concepts: relaxation

Property

(i) If relaxation RP is infeasible, then the original problem IP is infeasible.

(ii) Let x^* be an optimal solution of relaxation RP and let $f(x) = c(x), \forall x \in X$. If $x^* \in X$, then x^* is an optimal solution of IP.

Proof.

(i) If RP is infeasible, then $X = \emptyset$. Since $X \subseteq T$, the original problem IP is also infeasible.

(ii) Since x^* is an optimal solution of RP and $x^* \in X$, we have $z \leq c(x^*) = f(x^*) = z^R$. Since RP is a relaxation of IP, $z^R \leq z$. Thus, $z = z^R$, and x^* is an optimal solution of IP because $z = c(x^*)$.

□

Basic concepts: linear relaxation

The linear relaxation of a MILP is obtained by removing the integrality constraints.

Definition: Relaxation linear

Given the problem

$$(IP) \qquad z = \min\{c(x) : x \in X \subseteq \mathbb{Z}^n\},$$

its *linear relaxation* is

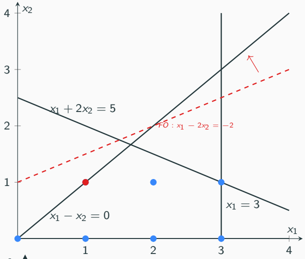
$$(LPrelaxation) \qquad z^{LR} = \min\{c(x) : x \in X \subseteq \mathbb{R}^n\}.$$

To assess the quality of the linear-relaxation bound, we can use its gap, defined by

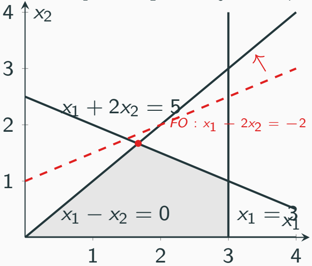
$$gap = \frac{z - z^{LR}}{z}.$$

Example

$$(IP) \equiv z = \min x_1 - 2x_2$$
$$s.t. : x_1 - x_2 \geq 0$$
$$x_1 + 2x_2 \leq 5$$
$$x_1 \leq 3$$
$$x_1, x_2 \geq 0 \text{ and integer}$$



$$(RL_2) \equiv z = \min x_1 - 2x_2$$
$$s.t. : x_1 - x_2 \geq 0$$
$$x_1 + 2x_2 \leq 5$$
$$x_1 \leq 3$$
$$x_1, x_2 \geq 0 \text{andinteger}$$



$gap = \frac{-1-(-5/3)}{|-1|}$[1] $= \frac{2}{3} \implies 66.67\%$.

[1]We use the absolute value in the denominator when the optimal value is negative.

Enumeration algorithms

Let

$$(ILP) \equiv z = \min\{c(x) : x \in S\}.$$

General idea: decompose S into "smaller" sets and solve the ILP over those sets.

Proposition

Let $S = S_1 \cup S_2 \cup \ldots \cup S_{|K|}$ a decomposition of S, and let $z^k = \min\{c(x) : x \in S_k\}, k \in K$. Then,

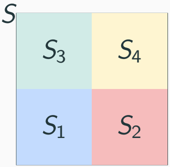
$$z = \min_{k \in K}\{z^k\}.$$

Enumeration algorithms

- If S_k is not easier to solve, it can itself be decomposed, i.e.,

$$S_k = S_{k_1} \cup S_{k_2} \cup \dots .$$

- This process can be repeated until all subproblems are solved. $\Longrightarrow$ Tree search.
- How can S be decomposed into subsets $S_1, \dots, S_k$?
  - The sets $S_1, \dots, S_k$ must form a partition of S.



  - The simplest approach is to partition S into two subsets, S_1 and S_2.

Enumeration algorithms

Binary variables

Let x_i be a binary variable, i.e., $x_i \in \{0, 1\}$.

The set S can be partitioned into:

- $S_1 = \{x \in S : x_i = 0\}$
- $S_2 = \{x \in S : x_i = 1\}$

Integer variables

Let x_i be a variable with a fractional value, i.e., $x_i = \alpha \notin \mathbb{Z}$.

The set S can be partitioned into:

- $S_1 = \{x \in S : x_i \leq \lfloor \alpha \rfloor\}$
- $S_2 = \{x \in S : x_i \geq \lceil \alpha \rceil\}$

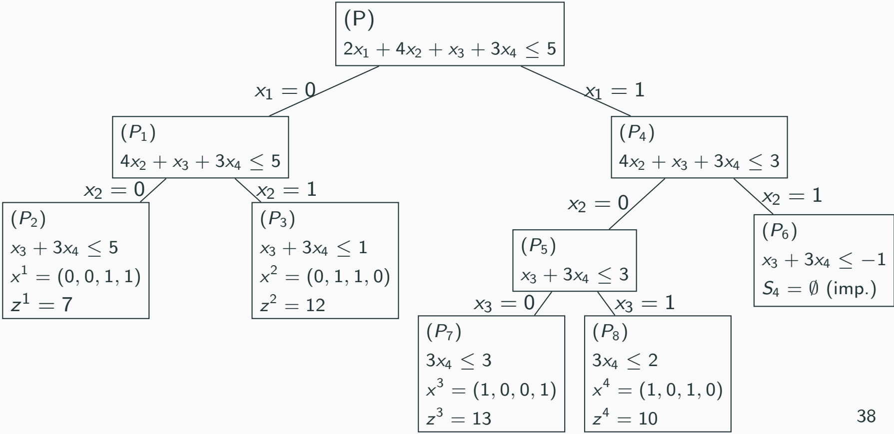
The sets can be successively decomposed. $\Longrightarrow$ Enumeration tree.

Example

$$(P) \equiv \max\ 8x_1 + 10x_2 + 2x_3 + 5x_4 \qquad \text{Knapsack problem}$$

$$\text{s.t.: } 2x_1 + 4x_2 + x_3 + 3x_4 \le 5$$

$$x_1, x_2, x_3, x_4 \in \{0, 1\}$$



Enumeration algorithms

- Each node k is associated with a problem P_k and a set S_k, which is the feasible region of P_k.
  - If P_k is obtained from P_j by branching, k is said to be a child of j.
  - If P_k is obtained from P_j through a sequence of branching operations, k is said to be a descendant of j.
  - We have $v(P_j) \leq v(P_k)$ for every descendant k of j.
    - This relation is valid for minimization problems.
  - We have $v(P_j) \geq v(P_k)$ for every descendant k of j.
    - This relation is valid for maximization problems.

Note: Each time we branch, we add constraints and, consequently, worsen the optimal value of the subproblems.

- How is z related to the bounds on the values of z^k?

Proposition

Let:

- $S = S_1 \cup S_2 \cup \ldots \cup S_K$ a decomposition of S;
- $z^k = \min\{c(x) : x \in S_k\}, k = 1, \ldots, K$;
- $\overline{z}^k$ an upper bound on z^k; and
- $\underline{z}^k$ a lower bound on z^k.

Then, $\overline{z} = \min_{k=1,\ldots,K}\{\overline{z}^k\}$ is an upper bound on z, and $\underline{z} = \min_{k=1,\ldots,K}\{\underline{z}^k\}$ is a lower bound on z.

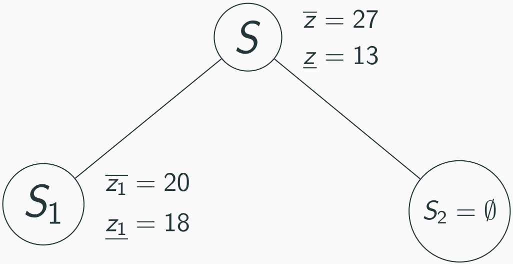
Enumeration algorithms

- We can use the previous proposition to reduce the size of the enumeration tree.

- A node k can be pruned (not branched) in three situations:
  1. the problem is infeasible, $S_k = \emptyset$;
  2. the solution obtained is feasible for the original problem P; or
  3. it can be proved that P_k does not contain the optimal solution (or solutions better than the best known solution for P).

Enumeration algorithms

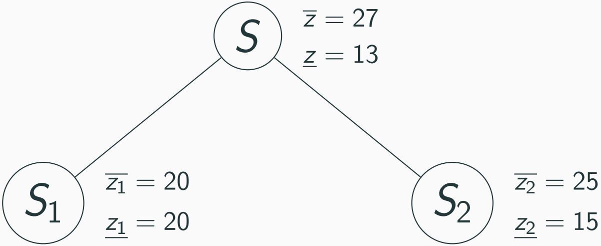
Minimization problem



Problem P_2 is infeasible, so there is no need to branch on it. $\Longrightarrow$ Pruning by infeasibility.

Enumeration algorithms

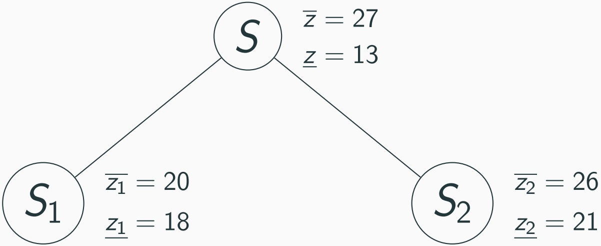
Minimization problem



Since the upper and lower bounds of P_1 are equal ($\overline{z}_1 = \underline{z}_1 = 20$), its optimal value has been obtained and there is no need to branch on it.
$\Longrightarrow$ Pruning by optimality.

Enumeration algorithm

Minimization problem



The optimal value of P_2 is at least 21, whereas the optimal value over S_1 is at most 20. Therefore, all solutions obtained from P_2 will be worse than those from P_1, so P_2 need not be explored further. $\Longrightarrow$ Pruning by bound.

Branch-and-bound algorithm

- It is an enumeration algorithm!
- How should the subproblems be solved?
- Its linear relaxation is solved.
- Notação:

$(P) \equiv z = \min\{c(x) : x \in X\}$

$(P_k) \equiv z_k = \min\{c(x) : x \in X_k\}$

L List of problems to be solved

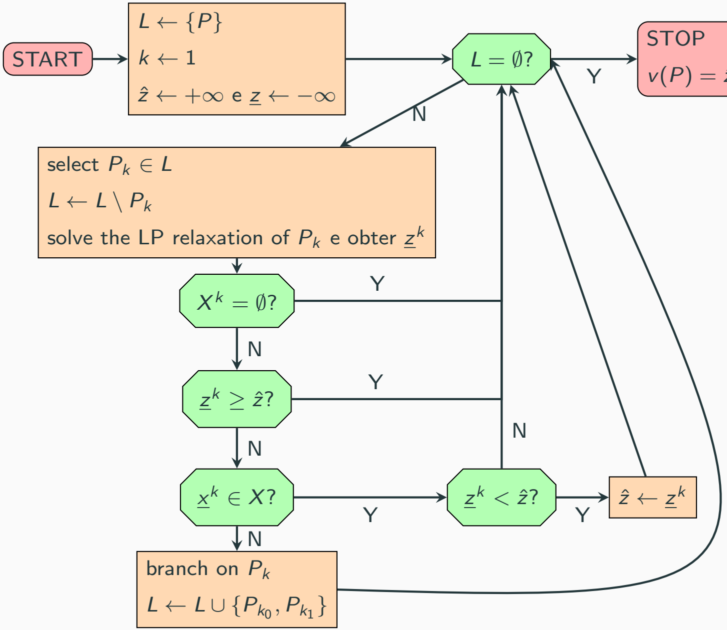
$\hat{z}$ value of the best known feasible solution – the incumbent.

If no feasible solution has been found, $\hat{z} = +\infty$.

$\underline{z}^k$ lower bound on z^k.

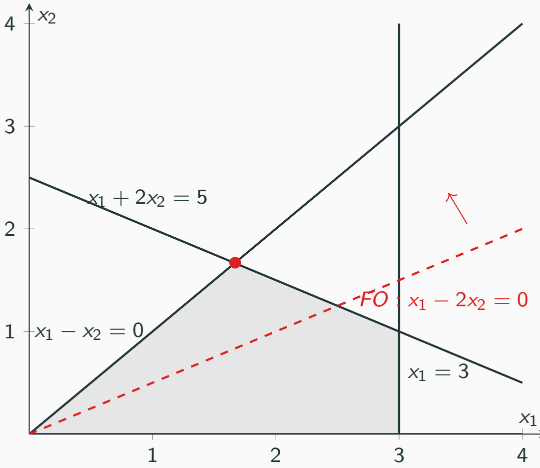
We set $\underline{z}^k = +\infty$ if P_k is infeasible and $\underline{z}^k = z^k$ otherwise.

Branch-and-bound algorithm



Example

$$(P) \equiv \min z = x_1 - 2x_2$$
$$s.t. : x_1 - x_2 \geq 0$$
$$x_1 + 2x_2 \leq 5$$
$$x_1 \leq 3$$
$$x_1, x_2 \geq 0 \text{ and integer}$$



Solving the linear relaxation:

$x_{RL} = (\frac{5}{3}, \frac{5}{3})$ e $\underline{z} = \frac{5}{3} - 2 \times \frac{5}{3} = -\frac{5}{3}$

Example

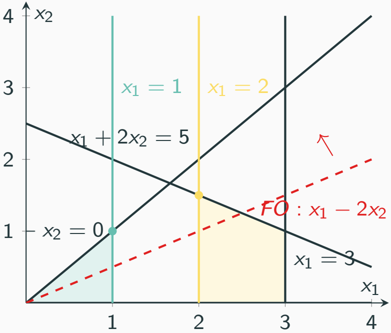
(P)
$\underline{z} = -\frac{5}{3}$
$\hat{z} = +\infty = -1$

$x_1 \leq \lfloor \frac{5}{3} \rfloor = 1$

(P_1)
$\underline{z_1} = -1$
$\hat{z}_1 = -1$

$x_1 \geq \lceil \frac{5}{3} \rceil = 2$

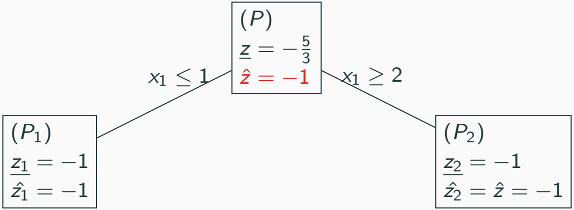
(P_2)
$\underline{z_2} = -1$
$\hat{z}_2 = \hat{z} = -1$



- $(P_1) : x_1^* = (1, 1)$ e $\underline{z_1} = \overline{z_1} = -1$
- $(P_2) : x_2^* = (2, \frac{3}{2})$ e $\underline{z_2} = -1$

■ There is no need to branch further from P_1, since we have already found a feasible solution. We can update the upper bound.

Example



Since the lower bound on P_2 equals $\hat{z}$, the node can be pruned because it cannot contain a better optimal solution. All solutions in descendants of P_2 will have an objective value greater than or equal to -1.

Branch-and-bound algorithm

- **Branching and search strategies**
  - How to branch?
    - Choose the fractional variable with value α closest to $\lfloor \alpha \rfloor + \frac{1}{2}$ in the linear relaxation (the "most fractional" variable).
    - Choose the fractional variable with the smallest index.
  - How to select the subproblem?
    - Depth-first search.
      Choose one of the most recently created problems. $\Longrightarrow$ Move quickly down the search tree in order to find a feasible solution early.
    - Best-bound node selection.
      Choose the open problem with the smallest value of $\underline{z}^k$.

Branch-and-bound algorithm

- Strategies for reducing the number of nodes in the tree
  - Use heuristics to obtain feasible solutions (upper bounds).
  - If the objective-function coefficients are integer, z takes only integer values for feasible solutions. In this case, the lower bounds $\underline{z}^k$ can be replaced by $\lceil \underline{z}^k \rceil$.
  - Try to improve the quality of the lower bounds.

■ The branch-and-bound algorithm can be used as a heuristic. $\Longrightarrow$ For example, by stopping the algorithm when a feasible solution is found.

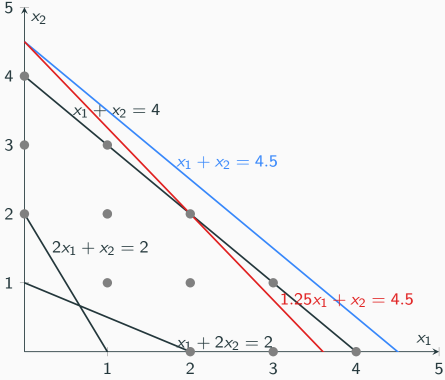
Gomory cutting-plane algorithm

Definition

Let $X \subseteq \mathbb{R}^n$. A *valid inequality* for X is an inequality $\pi_1 x_1 + \ldots + \pi_n x_n \leq \pi_0$ $(\pi x \leq \pi_0)$ that is satisfied by all points in X.

Example

Let $X = \{(x_1, x_2) \in \mathbb{Z}^2 : x_1 + x_2 \leq 4, x_1 + 2x_2 \geq 2, 2x_1 + x_2 \geq 2$ and $x_1, x_2 \geq 0\}$. $\Longrightarrow$ Gray points in the graph.



The inequality $x_1 + x_2 \leq 4.5$ is a valid inequality for X.

The inequality $1.25x_1 + x_2 \leq 4.5$ is not a valid inequality for X. $\Longrightarrow$ The points $(3, 1)$ and $(4, 0)$ belong to X and do not satisfy the inequality.

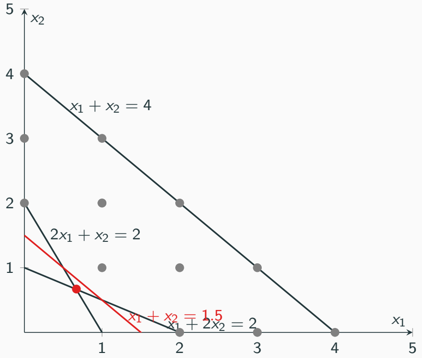
Gomory cutting-plane algorithm

Definition

Let $X \subseteq \mathbb{R}^n$ and $x^* \notin X$. A valid inequality for X that is violated by x^* is called a *cutting plane*.

Example

Let $X = \{(x_1, x_2) \in \mathbb{Z}^2 : x_1 + x_2 \leq 4, x_1 + 2x_2 \geq 2, 2x_1 + x_2 \geq 2$ and $x_1, x_2 \geq 0\}$. $\Longrightarrow$ Gray points in the graph.



The inequality $x_1 + x_2 \geq 1.5$ is a cutting plane for $x^* = (\frac{2}{3}, \frac{2}{3})$, which is the optimal solution of the linear relaxation of the problem

$$\min\{3x_1 + 2x_2 : (x_1, x_2) \in X\}.$$

- This is the general idea of the Gomory cutting-plane algorithm.

Gomory cutting-plane algorithm

Let

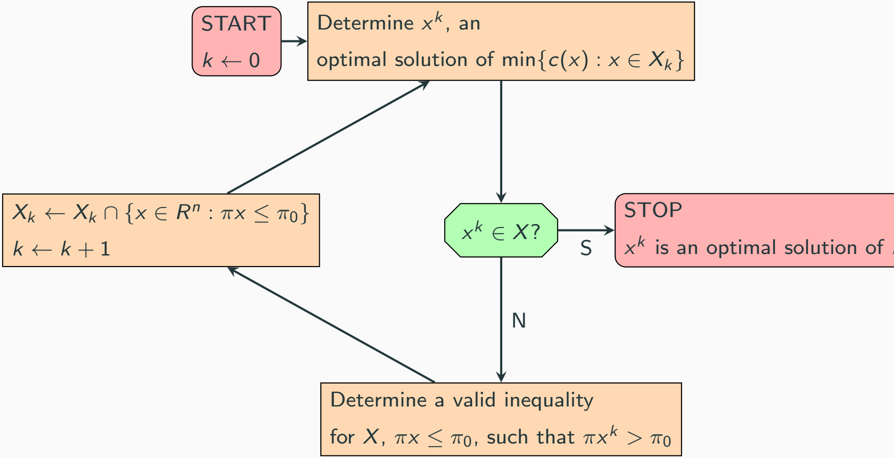
$$(P) \equiv \min\{c(x) : x \in X\},$$

where $X = \{x \in R^n : Ax \geq b \text{ and } x_j \text{ integer, for } j \in J\}$ and J is a subset of $\{1, ..., n\}$.

General idea: solve a sequence of linear relaxations until the solution obtained belongs to X. Each linear relaxation is obtained from the previous one by adding cutting planes.

Gomory cutting-plane algorithm

Let $\overline{X}_0 = \{x \in R^n : Ax \geq b\}$ (X without the integrality constraints).



Gomory cutting-plane algorithm

How can cutting planes for the optimal solution of the linear relaxation be determined?

- Gomory cut-generation procedure, which is valid for any MILP

Gomory cut-generation procedure

Let

$$(P) \equiv z = \min\{c(x) : Ax \geq b,\ x \geq 0 \text{ and integer}\}.$$

Step 1: Solve the linear relaxation of P.

Step 2: Write the system $Ax \geq b$ in augmented form.

Step 3: Rewrite the system in terms of the basic variables of the optimal solution, i.e., $x_i + \sum_{j \in VNB} \bar{a}_{ij} x_j = \bar{b}_i$ for every basic variable x_i.

Step 4: The Gomory cut is obtained using the following procedure:

$$x_i + \sum_{j \in VNB} \bar{a}_{ij} x_j = \bar{b}_i \implies x_i + \sum_{j \in VNB} \lfloor \bar{a}_{ij} \rfloor x_j \leq \bar{b}_i \implies x_i + \sum_{j \in VNB} \lfloor \bar{a}_{ij} \rfloor x_j \leq \lfloor \bar{b}_i \rfloor$$

$$\implies \sum_{j \in VNB} (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \geq \bar{b}_i - \lfloor \bar{b}_i \rfloor$$

Gomory cutting-plane algorithm

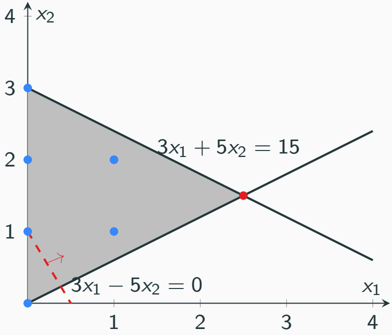
Definition

The *Gomory cut* is $\sum_{j \in VNB} (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \geq \bar{b}_i - \lfloor \bar{b}_i \rfloor$.

Example

$$(P) \equiv \max z = 2x_1 + x_2$$
$$\text{s.t.: } 3x_1 - 5x_2 \le 0$$
$$3x_1 + 5x_2 \le 15$$
$$x_1, x_2 \ge 0 \text{ and integer}$$

- Solving the linear relaxation of P we obtain $x^0 = (\frac{5}{2}, \frac{3}{2})$.



Example

- Writing the system in augmented form, we obtain:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \end{cases}$$

- The solution in augmented form is $x^0 = (\frac{5}{2}, \frac{3}{2}, 0, 0)$ and its basic variables are x_1 e x_2.

Example

- Rewriting the system in terms of the basic variables, we obtain:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \end{cases} \iff \begin{cases} x_1 = \frac{5}{3}x_2 - \frac{1}{3}x_3 \\ 3(\frac{5}{3}x_2 - \frac{1}{3}x_3) + 5x_2 + x_4 = 15 \end{cases}$$

$$\iff \begin{cases} - \\ 5x_2 - x_3 + 5x_2 + x_4 = 15 \end{cases} \iff \begin{cases} x_1 = \frac{5}{3}(\frac{15}{10} + \frac{1}{10}x_3 - \frac{1}{10}x_4) - \frac{1}{3}x_3 \\ x_2 - \frac{1}{10}x_3 + \frac{1}{10}x_4 = \frac{15}{10} \end{cases}$$

$$\iff \begin{cases} x_1 = \frac{5}{2} - \frac{1}{6}x_3 - \frac{1}{6}x_4 \\ x_2 - \frac{1}{10}x_3 + \frac{1}{10}x_4 = \frac{15}{10} \end{cases} \iff \begin{cases} x_1 + \frac{1}{6}x_3 + \frac{1}{6}x_4 = \frac{5}{2} \\ x_2 - \frac{1}{10}x_3 + \frac{1}{10}x_4 = \frac{3}{2} \end{cases}$$

Since both $\bar{b}_i$ are fractional, either constraint can be chosen to generate a Gomory cut.
$\Longrightarrow$ We choose the row corresponding to x_1.

$$(\frac{1}{6} - 0)x_3 + (\frac{1}{6} - 0)x_4 \geq (\frac{5}{2} - 2) \iff \frac{1}{6}x_3 + \frac{1}{6}x_4 \geq \frac{1}{2}$$

Example

- We need to write the cut $\frac{1}{6}x_3 + \frac{1}{6}x_4 \geq \frac{1}{2}$ in terms of x_1 and x_2 so that it can be represented graphically.
- From the augmented form of P we obtain:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \iff x_3 = -3x_1 + 5x_2 \\ 3x_1 + 5x_2 + x_4 = 15 \iff x_4 = 15 - 3x_1 - 5x_2 \end{cases}$$

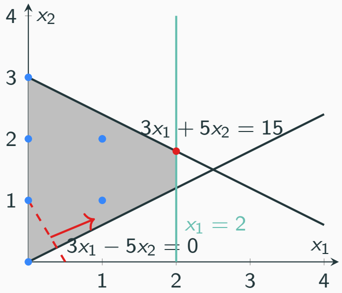
- Therefore,

$$\frac{1}{6}x_3 + \frac{1}{6}x_4 \geq \frac{1}{2} \iff \frac{1}{6}(-3x_1 + 5x_2) + \frac{1}{6}(15 - 3x_1 - 5x_2) \geq \frac{1}{2}$$
$$\iff -\frac{3}{6}x_1 + \frac{5}{6}x_2 + \frac{15}{6} - \frac{3}{6}x_1 - \frac{5}{6}x_2 \geq \frac{1}{2} \iff -x_1 \geq -\frac{12}{6} \iff x_1 \leq 2$$

Example

$$
\begin{aligned}
(P^1) \equiv \max z = 2x_1 + x_2 & \\
\text{s.t.:}\ & 3x_1 - 5x_2 \leq 0 \\
& 3x_1 + 5x_2 \leq 15 \\
& \color{red}{x1 \leq 2} \\
& x_1, x_2 \geq 0 \text{ and integer}
\end{aligned}
$$

- Solving the linear relaxation of P^1 we obtain $x^1 = (2, \frac{9}{5})$.



Example

- Writing the system in augmented form, we obtain:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \\ x_1 + x_5 = 2 \end{cases}$$

- The solution in augmented form is $x^1 = (2, \frac{9}{5}, 3, 0, 0)$ and its basic variables are x_1, x_2 e x_3.

Example

- Rewriting the system in terms of the basic variables, we obtain:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \\ x_1 + x_5 = 2 \end{cases} \iff \begin{cases} - \\ x_2 = 3 - \frac{3}{5}x_1 - \frac{1}{5}x_4 \\ x_1 = 2 - x_5 \end{cases} \iff \begin{cases} - \\ x_2 = \frac{9}{5} + \frac{3}{5}x_5 - \frac{1}{5}x_4 \\ - \end{cases}$$

$$\iff \begin{cases} 3(2 - x_5) - 5(\frac{9}{5} + \frac{3}{5}x_5 - \frac{1}{5}x_4) + x_3 = 0 \\ - \\ - \end{cases} \iff \begin{cases} x_3 + x_4 - 6x_5 = 3 \\ x_2 + \frac{1}{5}x_4 - \frac{3}{5}x_5 = \frac{9}{5} \\ x_1 + x_5 = 2 \end{cases}$$

We choose the constraint associated with x_2 to generate a Gomory cut because it is the only one with a fractional $\bar{b}_i$.

$$(\frac{1}{5} - 0)x_4 + (-\frac{3}{5} - (-1))x_5 \geq (\frac{9}{5} - 1) \iff \frac{1}{5}x_4 + \frac{2}{5}x_5 \geq \frac{4}{5}$$

Example

- We need to write the cut $\frac{1}{5}x_4 + \frac{2}{5}x_5 \geq \frac{4}{5}$ in terms of x_1 and x_2 so that it can be represented graphically.
- From the augmented form of P^1 we obtain:

$$\begin{cases} x_1 + 5x_2 + x_4 = 15 \iff x_4 = 15 - 3x_1 - 5x_2 \\ x_1 + x_5 = 2 \iff x_5 = 2 - x_1 \end{cases}$$

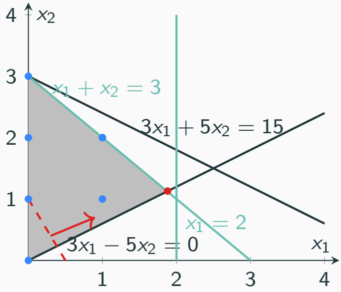
- Therefore,

$$\frac{1}{5}x_4 + \frac{2}{5}x_5 \geq \frac{4}{5} \iff \frac{1}{5}(15 - 3x_1 - 5x_2) + \frac{2}{5}(2 - x_1) \geq \frac{4}{5}$$
$$\iff x_1 + x_2 \leq 3$$

Example

$$
\begin{aligned}
(P^2) \equiv \max z = 2x_1 + x_2 & \\
\text{s.t.:}\ & 3x_1 - 5x_2 \leq 0 \\
& 3x_1 + 5x_2 \leq 15 \\
& x1 \leq 2 \\
& {\color{red} x_1 + x_2 \leq 3} \\
& x_1, x_2 \geq 0 \text{ and integer}
\end{aligned}
$$

- Solving the linear relaxation of P^1 we obtain $x^2 = (\frac{15}{8}, \frac{9}{8})$.



Example

- Writing the system in augmented form, we obtain:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \\ x_1 + x_5 = 2 \\ x_1 + x_2 + x_6 = 3 \end{cases}$$

- The solution in augmented form is $x^2 = (\frac{15}{8}, \frac{9}{8}, 0, \frac{30}{8}, \frac{1}{8}, 0)$ and its basic variables are x_1, x_2, x_4 e x_5.

Example

- Rewriting the system in terms of the basic variables, we obtain:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \\ x_1 + x_5 = 2 \\ x_1 + x_2 + x_6 = 3 \end{cases} \iff \dots \iff \begin{cases} x_1 + \frac{1}{8}x_3 + \frac{5}{8}x_6 = \frac{15}{8} \\ \frac{2}{8}x_3 + x_4 - \frac{30}{8}x_6 = \frac{30}{8} \\ -\frac{1}{8}x_3 + x_5 - \frac{5}{8}x_6 = \frac{1}{8} \\ x_2 - \frac{1}{8}x_3 + \frac{3}{8}x_6 = \frac{9}{8} \end{cases}$$

Since all $\bar{b}_i$ are fractional, either constraint can be chosen to generate a Gomory cut.
$\Longrightarrow$ We choose the row corresponding to x_1.

$$(\frac{1}{8} - 0)x_3 + (\frac{5}{8} - 0)x_6 \geq (\frac{15}{8} - 1) \iff \frac{1}{8}x_3 + \frac{5}{8}x_6 \geq \frac{7}{8}$$

Example

- We need to write the cut $\frac{1}{8}x_3 + \frac{5}{8}x_6 \geq \frac{7}{8}$ in terms of x_1 and x_2 so that it can be represented graphically.
- From the augmented form of P^2 we obtain:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \iff x_3 = -3x_1 + 5x_2 \\ x_1 + +x_2 + x_6 = 3 \iff x_6 = 3 - x_1 - x_2 \end{cases}$$

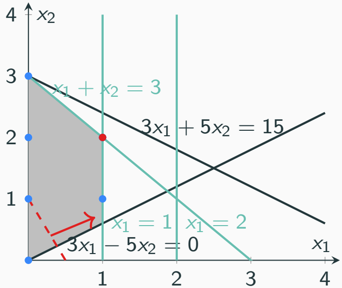
- Therefore,

$$\frac{1}{8}x_3 + \frac{5}{8}x_6 \geq \frac{7}{8} \iff \frac{1}{8}(-3x_1 + 5x_2) + \frac{5}{8}(3 - x_1 - x_2) \geq \frac{7}{8}$$
$$\iff x_1 \leq 1$$

Example

$$
\begin{aligned}
(P^3) \equiv \max z = 2x_1 + x_2 & \\
\text{s.t.:}\ & 3x_1 - 5x_2 \leq 0 \\
& 3x_1 + 5x_2 \leq 15 \\
& x1 \leq 2,\ x_1 + x_2 \leq 3 \\
& x_1 \leq 1 \\
& x_1, x_2 \geq 0 \text{ and integer}
\end{aligned}
$$

- Solving the linear relaxation of P^3 we obtain $x^3 = (1, 2)$. Since x^3 is integer, it is an optimal solution of P. Therefore, $z = 2 \times 1 + 2 = 4$. $\Longrightarrow$ *END*.



Outline

Valid inequalities

- One way to make the algorithms presented more efficient is bringing the feasible region of the LP relaxation closer to the convex hull of the original problem. $\implies$ This is achieved by adding valid inequalities that cut off fractional solutions.

- There are several ways to obtain valid inequalities for a problem.
  - Algorithms for generating valid inequalities. $\implies$ Chvátal–Gomory algorithm.
  - Valid inequalities based on problem-specific structure. $\implies$ Families of valid inequalities.
    - Example: Cover inequalities for the knapsack problem.

Chvátal–Gomory algorithm

Chvátal–Gomory procedure for constructing valid inequalities

Let $X = P \cap \mathbb{Z}^n$, where $P = \{x \in \mathbb{R}^n : Ax \le b, x \ge 0\}$, A is an $m \times n$ matrix with columns $\{a_1, \ldots, a_n$, e $u \in \mathbb{R}^m_+$:

(i) The inequality

$$\sum_{j=1}^{n} u_{a_j} x_j \le ub$$

is valid for X ($u \ge 0$ e $\sum_{j=1}^{n} a_j x_j \le b$).

(ii) The inequality

$$\sum_{j=1}^{n} \lfloor u_{a_j} \rfloor x_j \le ub$$

is valid for X ($x \ge 0$).

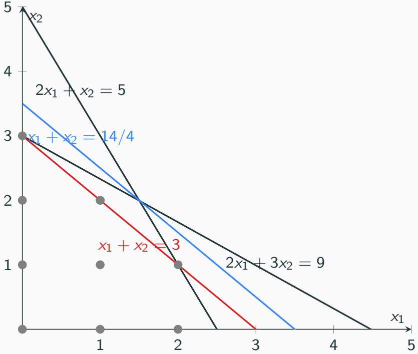
(iii) The inequality

$$\sum_{j=1}^{n} \lfloor u_{a_j} \rfloor x_j \le \lfloor ub \rfloor$$

is valid for X (x is integer and, consequently, $\sum_{j=1}^{n} \lfloor u_{a_j} \rfloor x_j$ is integer).

Example

Let $X = \{(x_1, x_2) \in \mathbb{Z}^2 : 2x_1 + 3x_2 \leq 9, 2x_1 + x_2 \leq 5 \text{ and } x_1, x_2 \geq 0\}$.



Consider the following vector of coefficients $u = (\frac{1}{4}, \frac{1}{4})$.

$$\begin{array}{lll} (1) & 7x_1 + 3x_2 \leq 9 & (\times \frac{1}{4}) \\ (2) & x_2 \leq 3 & (\times \frac{1}{4}) \quad + \\ \hline & x_1 + x_2 \leq \frac{14}{4} & \end{array}$$

Applying the Chvátal–Gomory procedure, we obtain
$x_1 + x_2 \leq \lfloor \frac{14}{4} \rfloor \iff x_1 + x_2 \leq 3.$

Valid inequalities

Theorem

Any valid inequality for X can be obtained by successively applying the Chvátal–Gomory procedure (starting from the system $Ax \leq b$, $x \geq 0$ and adding the inequalities obtained along the way).

Inequalities based on problem-specific structure

Consider the general knapsack problem

$$\max\{c(x) : x \in X\},$$

with

$$X = \{x \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \leq b\}.$$

Definition

A *cover* is a subset of items $S \subseteq N$ that cannot all fit in the knapsack, i.e.,

$$\sum_{i \in S} a_i > b.$$

Definition

If S is a cover, a valid inequality – a *cover inequality* – for the knapsack problem is

$$\sum_{i \in S} x_i \leq |S| - 1.$$

Example

$$\max z = 32x_1 + 20x_2 + 19x_3 + 13x_4 + 11x_5 + 8x_6$$
$$\text{s.t.: } 12x_1 + 9x_2 + 7x_3 + 5x_4 + 5x_5 + 3x_6 \leq 14$$
$$x_i \in \{0, 1\}, \; i = 1, \ldots, 6$$

- Solving the linear relaxation of the problem (with Solver), we obtain $z^{LPrelaxation} = 37.(6)$. $\Longrightarrow$ For binary variables, the constraints $x_i \leq 1$ must be added to the linear relaxation.

- Examples of cover inequalities for this instance are:

1. $x_1 + x_2 \leq 1$
2. $x_1 + x_3 \leq 1$
3. $x_1 + x_6 \leq 1$
4. $x_2 + x_3 + x_4 \leq 2$
5. $x_2 + x_3 \leq 1$ $\Longrightarrow$ Dominates inequality 4.

Adding these valid inequalities to the linear relaxation, we obtain $z^{LPrelaxation^+} = 37.4$.

References

Hillier, F. S. & Lieberman, G. J. (2021). *Introduction to Operations Research* Mc Graw-Hill, New York, Ufeasible solution, 11th edition.

Wolsey, L. A. (1998). *Integer programming*. John Wiley & Sons