Discrete Optimization

Chapter 3: Discrete Optimization – Integer Linear Programming

Edoardo Amaldi

DEIB – Politecnico di Milano
edoardo.amaldi@polimi.it

Course material on Beep "[2019-20] - Optimization"

Academic year 2019-20

3.1 Integer Programming models

Huge variety of decision-making problems can be formulated/approximated as linear optimization problems with integer/discrete variables.

Generic discrete optimization problem:

$$\min_{\underline{x} \in X} c(\underline{x})$$

where X is a discrete set and $c : X \to \mathbb{R}$.

Example: $X \subseteq \{$ all subsets of a given finite set $\}$.

A natural and systematic way to tackle such problems is to express them as Integer Programming (Optimization) problems.

Definitions:

A **Mixed Integer Linear Programming** (MILP) problem is an optimization problem like

$$\begin{array}{ll} \min & \underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \\ s.t. & A_1 \underline{x} + A_2 \underline{y} \geq \underline{b} \\ & \underline{x} \geq \underline{0} \text{ integer}, \ \underline{y} \geq \underline{0} \end{array}$$

with $A_1 \in \mathbb{Z}^{m \times n_1}$, $A_2 \in \mathbb{Z}^{m \times n_2}$, $\underline{c}_1 \in \mathbb{Z}^{n_1}$, $\underline{c}_2 \in \mathbb{Z}^{n_2}$ and $\underline{b} \in \mathbb{Z}^m$.

If all variables are integer, we have an **Integer Linear Programming** (ILP) problem:

$$\begin{array}{lll} \min & \underline{c}^t \underline{x} & \\ s.t. & A\underline{x} \geq \underline{b} & \\ & \underline{x} \geq \underline{0} & \text{integer.} \end{array} \tag{1}$$

If in (1) all variables $x_i \in \{0, 1\}$, a **Binary Linear Programming** (0-1-ILP) problem.

W.l.o.g. only inequalities and all coefficients are integer.

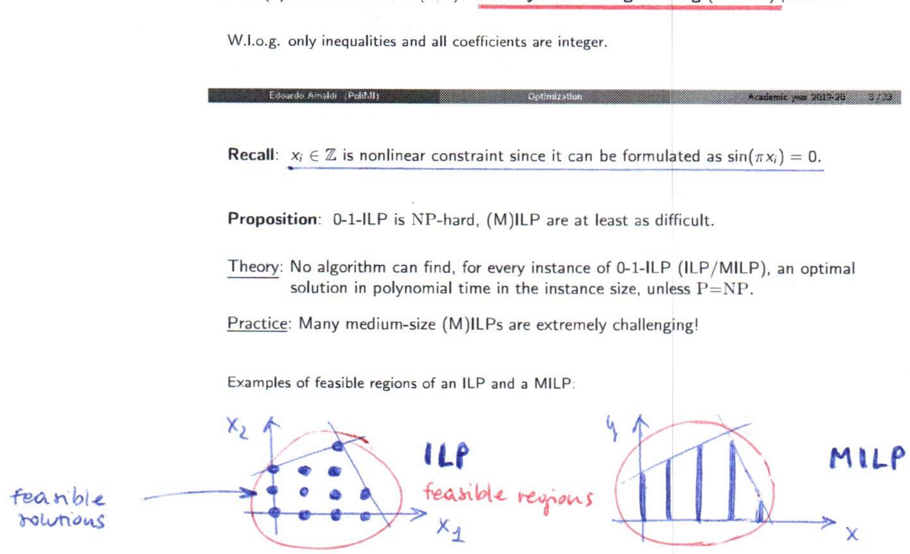
Recall: $x_i \in \mathbb{Z}$ is nonlinear constraint since it can be formulated as $\sin(\pi x_i) = 0$.

Proposition: 0-1-ILP is NP-hard, (M)ILP are at least as difficult.

Theory: No algorithm can find, for every instance of 0-1-ILP (ILP/MILP), an optimal solution in polynomial time in the instance size, unless P=NP.

Practice: Many medium-size (M)ILPs are extremely challenging!

Examples of feasible regions of an ILP and a MILP:



(M)ILP is a powerful and versatile modeling/solution framework.

3.1.1 Modeling techniques and examples

- binary choice
- association between entities
- forcing constraints
- piecewise linear cost functions
- modeling with exponentially many constraints
- disjunctive constraints
- linearizations

1) Binary choice

A binary variable allows to model a choice between two alternatives.

Example 1: Knapsack problem

Given

- n objects
- profit p_i and weight a_i for each object i, with $1 \leq i \leq n$
- knapsack capacity b

decide which objects to select so as to maximize total profit while respecting the capacity constraint.

ILP formulation

Variables: $x_i = 1$ if i-th object is selected and $x_i = 0$ otherwise, $1 \leq i \leq n$

$$\begin{array}{ll} \max & \sum_{i=1}^{n} p_i x_i \\ & \sum_{i=1}^{n} a_i x_i \leq b \\ & x_i \in \{0,1\} \quad \forall i \end{array}$$

Knapsack is $\mathcal{NP}$-hard.

Example 2: Set Covering/Packing/Partitioning problems

Given

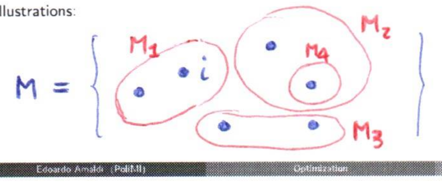
- groundset $M = \{1, 2, \ldots, m\}$ with $1 \leq i \leq m$,
- collection $\{M_1, \ldots, M_n\}$ of subsets indexed by $N = \{1, \ldots, n\}$ ($M_j \subseteq M$ for $j \in N$),
- a cost c_j for each M_j with $j \in N$,

a subset of indices $F \subseteq N$ defines a

- **cover** of M if $\cup_{j \in F} M_j = M$ (each i is covered at least once),
- **packing** of M if $M_{j_1} \cap M_{j_2} = \emptyset \ \forall j_1, j_2 \in F, j_1 \neq j_2$ (each i is covered at most once),
- **partition** of M if both a cover and a packing of M (each i is covered exactly once).

Total cost/weight of a subset indexed by $F \subseteq N$ is $\sum_{j \in F} c_j$.

Illustrations:



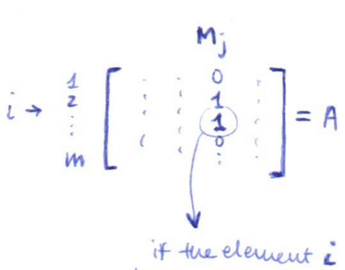
costs: c_1, c_2, c_3, c_4

$N = \{1, 2, 3, 4\}$

- $F = \{1, 2, 3\}$ is a cover
- $F = \{1, 3, 4\}$ is a packing
- $F = \{1, 2, 3\}$ is a partition

Set Covering problem:

Given $M = \{1, 2, \ldots, m\}$, $\{M_1, \ldots, M_n\}$ indexed by $N = \{1, \ldots, n\}$, and a cost c_j of M_j for each $j \in N$, find a cover of M with minimum total cost.



if the element i belongs to j

ILP formulation

Parameters: incidence matrix $A = [a_{ij}]$ with $a_{ij} = 1$ if $i \in M_j$ and $a_{ij} = 0$ otherwise

Variables: $x_j = 1$ if M_j is selected and $x_j = 0$ otherwise, $j \in N$

$$\begin{array}{lll} \min & \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} & \boxed{\sum_{j \in N} a_{ij} x_j \geq 1 \quad \forall i} & (2) \\ & x_j \in \{0,1\} \quad \forall j & \end{array}$$

we have to cover each element

(2) are the *covering constraints*.

Set Covering is $\mathcal{NP}$-hard.

Matrix notation:

cover each element at least once

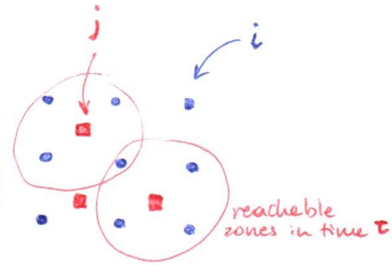
$$\min \left\{ \sum_{j=1}^{n} c_j x_j : A\underline{x} \geq \underline{1}, \ \underline{x} \in \{0,1\}^n \right\}$$

where $A = [a_{ij}]$ with $a_{ij} = 1$ if $i \in M_j$ and $a_{ij} = 0$ otherwise

Application: Emergency service location (ambulances or fire stations)

$M = \{$ areas to be covered $\}$
$N = \{$ candidate sites $\}$
$M_j = \{$ areas reachable in at most τ minutes from candidate site j $\}$

Decide where to locate ambulances so as to minimize the total cost, while guaranteeing that the next call is served within τ minutes.



we suppose the city to be divided in zones $\in M = \{\bullet\}$. we can put the sites in $N = \{$ ▪ $\}$

Set Packing problem:

cover each element at most once

$$\max \left\{ \sum_{j=1}^{n} c_j x_j : A\underline{x} \leq \underline{1}, \ \underline{x} \in \{0,1\}^n \right\}$$

where the c_j represent "profits" (so we try to select as many as we can)

Application: Combinatorial auctions (see introduction)

Determine the winner of each item so as to maximize total revenue:

$$\begin{array}{lll} \max & \sum_{S \subseteq M} b(S) x_S & \\ s.t. & \sum_{S \subseteq M \,:\, i \in S} x_S \leq 1 & \forall i \in M \\ & x_S \in \{0,1\} & \forall S \subset M. \end{array}$$

Set Packing is NP-hard.

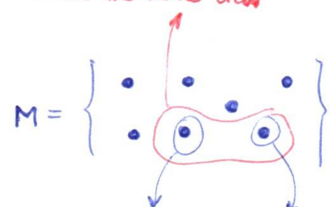
Set Partitioning problem:

cover each element exactly once

$$\min \ \text{ or } \ \max \left\{ \sum_{j=1}^{n} c_j x_j : A\underline{x} = \underline{1}, \ \underline{x} \in \{0,1\}^n \right\}$$

where c_js represent "costs" or "profits"

M_j = set of feasible flight legs with a cost c_j if it has the same crew



flight from X to Y starting at T_1, arriving at T_2

flight $Y \to Z$ starting at T_3, arriving at T_4

they're compatible ("feasible") if $T_3 > T_2$ sufficiently

Application: Airline crew scheduling (see Computer Lab 3)

Given planning horizon.

$M = \{$ flight legs $\}$ single takeoff-landing phases to be carried out within a predefined time window.

flight leg: flight that goes from X to Y, starting at T_1 and arriving at T_2

$M_j = \{$ feasible subsets of flight legs $\}$ can be carried out by same crew respecting all constraints (e.g., compatible flights, rest periods, total flight time,...).

Assign the crews to the flight legs so as to minimize total cost.

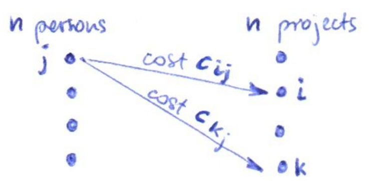
Other application: distribution planning (assign customers to routes)

Set Partitioning is NP-hard.

2) Association between entities

Binary variables allow to model associations between two (several) entities.

Example 3: Assignment problem



we want all the projects done at the minimum cost

Given

- n projects and n persons
- cost c_{ij} for assigning project i to person j, $\forall i, j \in \{1, \dots, n\}$

decide which project to assign to each person so as to minimize the total cost while completing all projects.

Assumption: every person can perform any project, and each person (project) must be assigned to a single project (person).

ILP formulation

Variables: $x_{ij} = 1$ if i-th project is assigned to j-th person and $x_{ij} = 0$ otherwise, $1 \leq i, j \leq n$

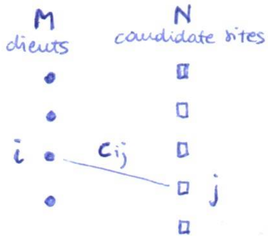
$$\begin{array}{lll} \min & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} & \\ s.t. & \sum_{i=1}^{n} x_{ij} = 1 & \forall j \\ & \sum_{j=1}^{n} x_{ij} = 1 & \forall i \\ & x_{ij} \in \{0,1\} & \forall i, \forall j \end{array}$$

→ all the persons have a project
→ all the projects are assigned

3) Forcing constraints

The fact that a decision X can be made only if a decision Y has also been made can be modeled via $x \leq y$.

Example 4: Uncapacitated Facility Location (UFL) (there is no capacity limit on the depots)



c_{ij} = cost if the whole demand of client i is served in j

Given

- $M = \{1, 2, \ldots, m\}$ clients, $i \in M$
- $N = \{1, 2, \ldots, n\}$ candidate sites where a depot (deposito) can be located, $j \in N$
- fixed cost f_j for opening a depot in candidate site j, $\forall j \in N$
- c_{ij} transportation cost if the whole demand of client i is served from depot j, $\forall i \in M$, $\forall j \in N$

decide where to locate the depots and how to serve the clients so as to minimize the total costs while satisfying all demands.

Illustration: ←

UFL is $\mathcal{NP}$-hard.

MILP formulation

Variables:

- x_{ij} = fraction of demand of client i served by depot j, with $1 \leq i \leq m$, $1 \leq j \leq n$
- $y_j = 1$ if depot j is opened and $y_j = 0$ otherwise, with $1 \leq j \leq n$

$$\begin{array}{lll} \min & \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} + \sum_{j \in N} f_j y_j & \text{service costs + opening costs} \\ s.t. & \sum_{j \in N} x_{ij} = 1 \quad \forall i \in M & \text{demand is satisfied} \\ & \boxed{\sum_{i \in M} x_{ij} \leq m y_j \quad \forall j \in N} & (3) \\ & y_j \in \{0, 1\} \quad \forall j \in N & \\ & 0 \leq x_{ij} \leq 1 \quad \forall i \in M, j \in N & \end{array}$$

x_{ij} and y_j must be linked:

$x_{ij} > 0 \Rightarrow y_j = 1$

$y_j = 0 \Rightarrow x_{ij} = 0$

we know that $\sum x_{ij} \leq m$ (since $\sum_j x_{ij} = 1$)

with n linking constraints (3).

Capacitated FL variant:

If d_i is the demand of client i and k_j the capacity of depot j, capacity constraints:

$$\sum_{i \in M} d_i x_{ij} \leq k_j y_j \quad \forall j \in N$$

(this too is NP-hard)

4) Piecewise linear cost functions

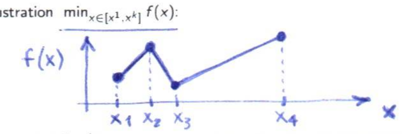
Continuous and binary variables allow to model nonconvex piecewise linear cost functions.

Example 5: Minimization of piecewise linear cost functions

Arbitrary piecewise linear function $f : [x^1, x^k] \to \mathbb{R}$.

Suppose $x^1 < x^2 < \ldots < x^k$ and $f(x)$ is specified by $(x^i, f(x^i))$, for $i = 1, \ldots, k$.

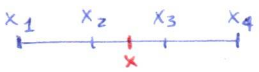
Illustration $\min_{x \in [x^1, x^k]} f(x)$:



Any $x \in [x^1, x^k]$ and corresponding $f(x)$ can be expressed as

$$\boxed{x = \sum_{i=1}^{k} \lambda_i x^i} \text{ and } \boxed{f(x) = \sum_{i=1}^{k} \lambda_i f(x^i)} \text{ with } \boxed{\sum_{i=1}^{k} \lambda_i = 1 \text{ and } \lambda_1, \ldots, \lambda_k \geq 0,}$$

Choice of λ_is is not unique.

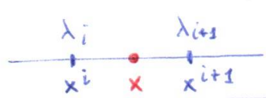


we can represent x either using x_1 and x_4 or maybe using x_2 and x_3 (or other combinations)

Unique if at most two consecutive λ_i can be nonzero.

Any $x \in [x^i, x^{i+1}]$ is then represented as

$$x = \lambda_i x^i + \lambda_{i+1} x^{i+1} \quad \text{with} \quad \lambda_i + \lambda_{i+1} = 1 \quad \text{and} \quad \lambda_i \geq 0, \ \lambda_{i+1} \geq 0.$$



$\boxed{x = \lambda_i x^i + \lambda_{i+1} x^{i+1}}$

we're representing $\forall x$ only using the boundaries of the smallest interval in which x's are in ⇒ in this way the representation is unique. To do this we need to identify in which subinterval x is in. We use y_i to do so.

By defining

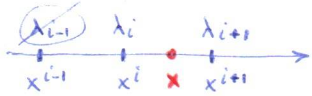
$$y_i = 1 \text{ if } x \in [x^i, x^{i+1}] \text{ and } y_i = 0 \text{ otherwise, for } i = 1, \ldots, k-1$$

$\min_{x \in [x^1, x^k]} f(x)$ can be formulated as

$$\begin{array}{lll} \min & \sum_{i=1}^{k} \lambda_i f(x^i) & \\ s.t. & \sum_{i=1}^{k} \lambda_i = 1, \boxed{\sum_{i=1}^{k-1} y_i = 1} & \\ & \lambda_1 \leq y_1, \lambda_k \leq y_{k-1} & \\ & \lambda_i \leq y_{i-1} + y_i & i = 2, \ldots, k-1 \\ & \lambda_i \geq 0, \quad y_i \in \{0, 1\} & i = 1, \ldots, k \end{array}$$

every point belongs to just 1 subinterval

Linking λ_i and y_i:



⇓

$\lambda_i \neq 0$ only if x is in one of the two positions:

$(y_{i-1} = 1) \vee (y_i = 1)$

⇓

$\lambda_i \neq 0$ if $\left\langle \begin{array}{l} y_{i-1} = 1 \\ y_i = 1 \end{array} \right.$

$\Rightarrow \boxed{\lambda_i \leq y_{i-1} + y_i}$

N.B.: If $y_j = 1$ then $\lambda_i = 0$ for all i, $1 \leq i \leq n$, different from j or $j + 1$.

5) Modeling with exponentially many constraints

Example 6: Asymmetric Traveling Salesman Problem (ATSP)

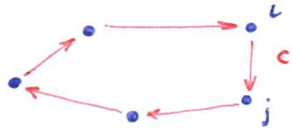
Given

- a complete directed graph $G = (V, A)$ with $n = |V|$ nodes
- a cost $c_{ij} \in \mathbb{R}$ for each arc $(i, j) \in A$ (in case $c_{ij} = \infty$)

V = subset of nodes, A = arcs

determine a Hamiltonian circuit (tour), i.e., a circuit that visits exactly once each node, of minimum total cost. (starting from one node and returning to it)

Illustration:



c_{ij} = cost for the connection (i,j)

(If there is no way to connect them then $c_{ij} = \infty$)

number of Hamiltonian circuits: $(n-1)!$

we start from one node. For the second we have $(n-1)$ choices, for the third $(n-2)$ and so on.

ATSP is $\mathcal{NP}$-hard.

Applications: logistics, microchip manufacturing, scheduling, (DNA) sequencing,...

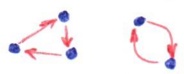
Also symmetric TSP version with undirected graph G.

Website devoted to TSP: `http://www.math.uwaterloo.ca/tsp/`

Many variants with

- time windows (earliest and latest arrival time)
- precedence constraints
- capacity constraint
- several vehicles ("Vehicle Routing Problem" – VRP)
- ...

The conditions (4), (5) are not enough for a characterization:



(*) respect (4), (5). (not an Hamiltonian circuits) We need to specify that ∄ unconnected subspaces.

We define the CUT $\delta(S)$:

$\delta^+(S) = \{(i,j) \in A : i \in S, j \notin S\}$.

To avoid (*) we consider always at least one element of $\delta^+(S)$ $(\forall S)$.

An ILP formulation

Variables: $x_{ij} = 1$ if (i, j) is included in the Hamiltonian circuit and $x_{ij} = 0$ otherwise, with $(i, j) \in A$

$$\begin{array}{lll} \min & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\ \text{s.t.} & \sum_{j\in V: j\neq i} x_{ij} = 1 \quad \forall i & (4) \\ & \sum_{i\in V: i\neq j} x_{ij} = 1 \quad \forall j & (5) \\ & \sum_{(i,j)\in\delta^+(S)} x_{ij} \geq 1 \quad \forall\, S \subset V, S \neq \emptyset & (6) \\ & x_{ij} \in \{0,1\} \quad \forall (i,j) \in A & \end{array}$$

where

(4) and (5) are the *assignment constraints*,

$\delta^+(S) = \{(i,j) \in A : i \in S, j \in V \setminus S\}$,

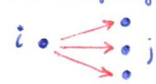
(6) are the *cut-set inequalities*.

CUT SETS INEQUALITIES (one inequality $\forall S \subset V$)

Observation: number of constraints (6) is exponential in n. $(n = |V|)$

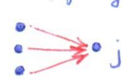
We need to characterize Hamiltonian circuits:

- for every node i we need just 1 outgoing arch:



$\sum x_{ij} = 1 \quad \forall i$ with $j \in V$ = set of possible successors (we can write $V = \{j : j \neq i\}$

- for every node j we need just 1 ingoing node:



$\sum x_{ij} = 1 \quad \forall j$ with $i \in V$ = set of possible precessors

Alternative ILP formulation

they're a lot of inequalities (order 2^n), maybe we can substitute them:

Substitute the cut-set inequalities

$$\sum_{(i,j)\in\delta^+(S)} x_{ij} \geq 1 \qquad \forall\, S \subset V, S \neq \emptyset$$

here we're imposing condition on $\delta^+(S)$, we've saying that we have to take at least 1 arc from every cut set.

with the subtour elimination inequalities:

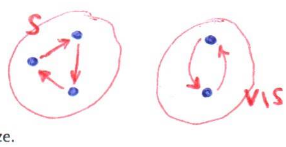
alternatives

instead of this we can try to put some conditions on the arcs that are already selected.

$$\sum_{(i,j)\in E(S)} x_{ij} \leq |S| - 1 \qquad \forall\, S \subseteq V, 2 \leq |S| \leq n-1 \qquad (7)$$

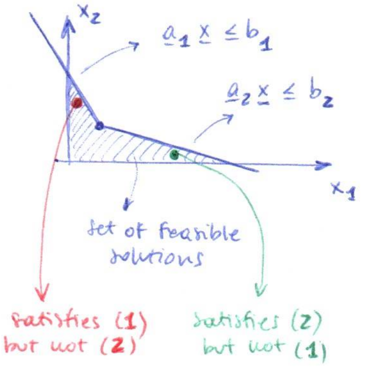
where $E(S) = \{(i,j) \in A : i \in S, j \in S\}$ for $S \subseteq V$.

Illustration:



If $S \neq V$ then the number of of arcs fully contained in S must be strictly smaller than the number of nodes in S. Here we have 3 nodes in S and 3 arcs in S $\Rightarrow$ the condition is violated.

Still exponential size.



6) Disjunctive constraints

Binary variables allow to impose disjunctive constraints such as:

$$\textit{either} \quad \underline{a}_1\underline{x} \le b_1 \quad \textit{or} \quad \underline{a}_2\underline{x} \le b_2$$

with $\underline{x} \in \mathbb{R}$ and $\underline{0} \le \underline{x} \le \underline{u}$, where $\underline{u}$ is an upper bound vector.

Illustration: ←

$$y_1 = \begin{cases} 1 & \text{if (1) is satisfied} \\ 0 & \text{if (1) is not satisfied} \end{cases}$$

Introduce $y_i \in \{0,1\}$ for each $\underline{a}_i\underline{x} \le b_i$, with $1 \le i \le 2$, and consider constraints:

$$\boxed{\underline{a}_i\underline{x} - b_i \le M(1 - y_i) \quad \text{for } i = 1,2} \ (*)$$
$$y_1 + y_2 = 1 \longrightarrow \text{either one inequality must hold}$$
$$y_i \in \{0,1\} \quad \text{for } i = 1,2$$
$$\underline{0} \le \underline{x} \le \underline{u}$$

where $M \ge \max_{1 \le i \le 2}\{\underline{a}_i\underline{x} - b_i \ : \ \underline{0} \le \underline{x} \le \underline{u}\}$.

(*) If $y_1 = 1$ then $\underline{x}$ satisfies $\underline{a}_1\underline{x} \le b_1$ while $\underline{a}_2\underline{x} \le b_2$ is inactive, and conversely if $y_2 = 1$.

Example 7: Scheduling problem

Given

- m machines and n products
- for each product j, deadline d_j and time p_{jk} needed to process j on machine k, for $\le k \le m$

determine an optimal schedule so as to minimize the time needed to complete all products, while satisfying all deadlines.

Assumptions:

- products cannot be processed simultaneously on the same machine
- the execution of a product on a machine cannot be interrupted (non-preemptive scheduling).

For simplicity also assume: each product must be processed on all the machines according to the order of the machine indices.

See Computer Lab 1

7) Linearization of products of variables

- Product of two (several) binary variables:

$z = y_1 \cdot y_2$, with $y_i \in \{0,1\}$ for $i = 1,2$ and $z \in \{0,1\}$, can be replaced by

$$z \le y_1$$
$$z \le y_2$$
$$\boxed{z \ge y_1 + y_2 - 1} \quad \equiv \quad z = 1 \text{ when both } y_1 \text{ and } y_2 \text{ are } 1$$

extendable to case with ≥ 3 binary variables.

- Product of a binary variable and a bounded continuous variable:

$z = x \cdot y$, with $x \in [0,u]$, $y \in \{0,1\}$ and $z \in [0,u]$, can be replaced by

$$0 \le z \le uy$$
$$z \le x$$
$$\boxed{z \ge x - (1-y)u} \quad \equiv \quad \text{If } y = 1 \text{ we want } z = x$$

Question: If y_1 and y_2 are continuous and bounded, can $y_1 \cdot y_2$ be exactly linearized?

3.2 Alternative, strong and ideal formulations

In linear optimization, good formulations contain a small number of variables n and constraints m because the complexity of algorithms grows polynomially in n and m.

The choice of the formulation does not critically affect the possibility of solving LPs.

For ILP and MILP problems, the choice of the formulation is crucial.

3.2.1 Alternative and strong formulations

Definition: Given any MILP

$$\begin{array}{rll} \underline{z_{MILP}} = \min & \underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \\ s.t. & A_1 \underline{x} + A_2 \underline{y} \geq \underline{b} \\ & \underline{x} \geq \underline{0} \\ & \boxed{\underline{y} \geq \underline{0} \text{ integer}} \end{array}$$

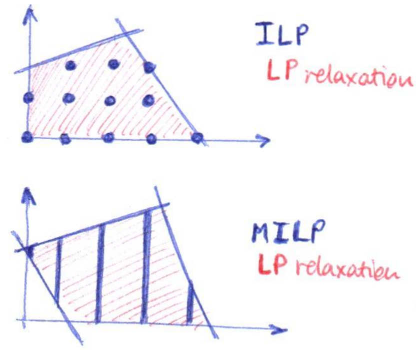
its **linear programming** (LP) **relaxation** is the linear program:

$$\begin{array}{rll} \underline{z_{LP}} = \min & \underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \\ s.t. & A_1 \underline{x} + A_2 \underline{y} \geq \underline{b} \\ & \underline{x} \geq \underline{0}, \boxed{\underline{y} \geq \underline{0},} \end{array}$$

where $y_j \in \mathbb{Z}$ are omitted for all j.

If $y_j \in \mathbb{Z}$ with $0 \leq y_j \leq u_j$, then in the linear relaxation $y_j \in [0, u_j]$.

Illustration (feasible region of LP relaxations for (M)ILP): →



Feasible region of the MILP:

$$\underline{X_{MILP}} = \{(\underline{x}, \underline{y}) \in \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2} \ : \ A_1 \underline{x} + A_2 \underline{y} \geq \underline{b}, \ \underline{x} \geq \underline{0}, \ \underline{y} \geq \underline{0}\}$$

and feasible region of its linear relaxation:

$$\underline{X_{LP}} = \{(\underline{x}, \underline{y}) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \ : \ A_1 \underline{x} + A_2 \underline{y} \geq \underline{b}, \ \underline{x} \geq \underline{0}, \ \underline{y} \geq \underline{0}\}$$

Obviously $\boxed{X_{MILP} \subseteq X_{LP}.}$

Proposition: For any *minimization* MILP, we have:

- $\mathbf{z_{LP}} \leq \mathbf{z_{MILP}}$, that is, z_{LP} is a lower bound for z_{MILP}, (we're minimizing over a larger feasible region)
- if optimal solution $\underline{x}^*_{LP}$ of LP relaxation is feasible for the original MILP, it is also optimal for MILP.

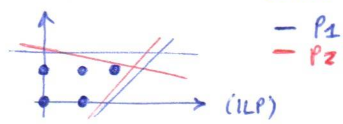
For *maximization* problems, $\mathbf{z_{MILP}} \leq \mathbf{z_{LP}}$. (we're maximizing over a larger region)

The same holds for the special case of ILPs.

Definition:

A polyhedron $P = \{(\underline{x}, \underline{y}) \in \mathbb{R}^{n_1+n_2} \ : \ A_1 \underline{x} + A_2 \underline{y} \geq \underline{b}, \ \underline{x} \geq \underline{0}, \ \underline{y} \geq \underline{0}\} \subseteq \mathbb{R}^{n_1+n_2}$ is a **formulation** for a mixed integer set $X \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ if and only if $X = P \cap (\mathbb{R}^{n_1} \times \mathbb{Z}^{n_2})$.

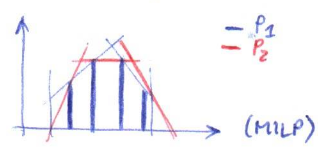
Illustration of (M)ILP formulations: (M)ILP has ∞ formulations:



$$P_1 \cap \mathbb{Z}^2 = P_2 \cap \mathbb{Z}^2 = X$$

N.B.: In modeling fixed costs, we considered $X \cup \{(0, 1)\}$ instead of $X = \{(0, 0), (x, 1) \ \text{with} \ 0 < x \leq 1\}$.

Alternatively:



Observation: Any MILP admits an infinite number of *alternative formulations.*
Equivalent from the IP point of view but different feasible regions of the LP relaxations.

Examples:

1) Two alternative formulations for TSP: with cut-set or subtour-elimination constraints.

2) Original formulation for UFL:

$$\begin{array}{lll} \min & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{j=1}^{n} f_j y_j & \\ s.t. & \sum_{j=1}^{n} x_{ij} = 1 & \forall i \in M \\ & \boxed{\sum_{i=1}^{m} x_{ij} \leq m y_j \qquad \forall j \in N} & (1) \\ & y_j \in \{0, 1\} & \forall j \in N \\ & 0 \leq x_{ij} \leq 1 & \forall i \in M, j \in N \end{array}$$

n constraints (1) link the corresponding x_{ij} and y_j.

Alternative formulation for UFL:

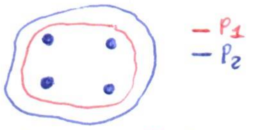
$$\begin{array}{lll} \min & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{j=1}^{n} f_j y_j & \\ s.t. & \sum_{j=1}^{n} x_{ij} = 1 & \forall i \in M \\ & \boxed{x_{ij} \leq y_j \qquad \forall i \in M, j \in N} & (2) \\ & y_j \in \{0, 1\} & \forall j \in N \\ & 0 \leq x_{ij} \leq 1 & \forall i \in M, j \in N \end{array}$$

alternative way of linking x_{ij} and y_j.

Note: this formulation has more constraints but this is not that important in terms of efficiency of the formulation

mn constraints (2) link the corresponding x_{ij} and y_j.

We want the smaller feasible region possible:



In this case P_1 is a stronger (better) formulation than P_2.

Definition:

Given a mixed integer set $X \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ and two formulations P_1 and P_2 for X, P_1 is **stronger** than P_2 if $P_1 \subset P_2$.

The lower bound provided by the LP relaxtion P_1 is not smaller (not weaker) than that provided by P_2:

$$\begin{aligned} z_{MILP} &= \min\{\underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \,:\, (\underline{x}, \underline{y}) \in X\} \\ &\geq \min\{\underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \,:\, (\underline{x}, \underline{y}) \in P_1\} \\ &\geq \min\{\underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \,:\, (\underline{x}, \underline{y}) \in P_2\}. \end{aligned}$$

Examples:

1) **Uncapacitated Facility Location** (UFL)

Proposition: The LP relaxation of the MILP formulation with constraints $\boxed{x_{ij} \leq y_j}$ is stronger than that with aggregated constraints $\boxed{\sum_{i=1}^m x_{ij} \leq m y_j}$.

Proof:

$P_1 = \left\{(\underline{x}, \underline{y}) \in \mathbb{R}^{mn+n} \,:\, \sum_{j=1}^n x_{ij} = 1,\ \forall i, \boxed{x_{ij} \leq y_j}\ \forall i, \forall j,\ 0 \leq x_{ij} \leq 1,\ \forall i, \forall j,\ 0 \leq y_j \leq 1,\ \forall j\right\}$

$P_2 =$
$\left\{(\underline{x}, \underline{y}) \in \mathbb{R}^{mn+n} \,:\, \sum_{j=1}^n x_{ij} = 1,\ \forall i, \boxed{\sum_{i=1}^m x_{ij} \leq m y_j}\ \forall j,\ 0 \leq x_{ij} \leq 1,\ \forall i, \forall j,\ 0 \leq y_j \leq 1,\ \forall j\right\}$

Obviously $P_1 \subseteq P_2$ (for any given j, summing $x_{ij} \leq y_j$ over i yields $\sum_{i=1}^m x_{ij} \leq m y_j$)

We exhibit a point $(\underline{x}, \underline{y})$ in $P_2 \setminus P_1$:

Suppose $m = kn$ for some integer $k \geq 2$, and let each depot serve k clients:
$x_{ij} = 1$ for $i = k(j-1)+1, \ldots, k(j-1)+k$, $j = 1, \ldots, n$, and $x_{ij} = 0$ otherwise,

$y_j = k/m$ for $j = 1, \ldots, n$.

Suppose: m = # clients, n = # depots, $m = kn$, $k \geq 2$

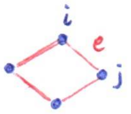
In particular, wlog: $m=6$, $n=3$, $k=2$

each depot serves k clients ($k=2$)

- $x_{ij} = 1$ for $i = k(j-1)+1, \ldots, k(j-1)+k$
- $y_j = \frac{k}{m}$ $\forall j$

$\Rightarrow$ $(\underline{x}, \underline{y})$ respect P_2 but not P_1 $\Rightarrow$ $\in P_2 \setminus P_1$

2) **Symmetric TSP (STSP)** (this variant is that the graph is not directed) $\Rightarrow$ we have edges and not arcs



$x_e = \begin{cases} 1 & e \text{ selected} \\ 0 & e \text{ not selected} \end{cases}$

STSP: Given undirected $G = (V, E)$ and cost c_e for every $e = \{i, j\} \in E$, determine a **Hamiltonian cycle** of G (i.e., a cycle visiting each $i \in V$ exactly once) of minimal total cost.

Two alternative formulations:

$$\begin{array}{lll} \min & \sum_{e \in E} c_e x_e & \\ \text{s.t.} & \sum_{e \in \delta(i)} x_e = 2 & i \in V \quad (DEG) \\ & \boxed{\sum_{e \in \delta(S)} x_e \geq 2 \quad S \subset V, S \neq \emptyset \quad (CUT)} & \\ & x_e \in \{0, 1\} & e \in E \end{array}$$

and

$$\begin{array}{lll} \min & \sum_{e \in E} c_e x_e & \\ \text{s.t.} & \sum_{e \in \delta(i)} x_e = 2 & i \in V \quad (DEG) \\ & \boxed{\sum_{e \in E(S)} x_e \leq |S| - 1 \quad S \subset V, S \neq \emptyset \quad (SEC)} & \\ & x_e \in \{0, 1\} & e \in E, \end{array}$$

They're equally strong

where $\delta(S) = \{\{i, j\} \in E \,:\, i \in S, j \in V \setminus S\}$, $\delta(i) = \delta(\{i\})$,
and $E(S) = \{\{i, j\} \in E \,:\, i \in S, j \in S\}$.

(DEG), (SEC) and (CUT) are, respectively, the *degree*, *subtour-elimination* and *cut-set* constraints.

In this case we're not distinguing enter/exit, so we have to say that for each node i we should select exactly 2 incident edges:

$\delta(i)$ = set of edges that touch in i

(DEG) constraint

Let P_{sec} and P_{cut} be the polyhedra (feasible regions) of the LP relaxations of these two formulations.

Proposition: The two formulations are equally strong ($P_{sec} = P_{cut}$). ($P_{sec} \subset P_{cut}$, $P_{sec} \supset P_{cut}$)

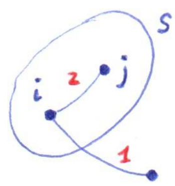
Proof:

We verify that: a) $P_{sec} \subseteq P_{cut}$ and b) $P_{cut} \subseteq P_{sec}$.

For any $S \subset V$, the sum of (DEG) with coefficients 1 over all $i \in S$ yields

$$\sum_{i \in S} \sum_{e \in \delta(i)} x_e = 2|S|. \qquad (3)$$

In $\sum_{i \in S} \sum_{e \in \delta(i)} x_e$ each $e = \{i, j\}$ with $i \in S$ and $j \in S$ occurs twice, while each $e = \{i, j\}$ with $i \in S, j \in V \setminus S$ occurs just once.



$\Downarrow$

Thus (3) can be rewritten as

$$2 \sum_{e \in E(S)} x_e + \sum_{e \in \delta(S)} x_e = 2|S|,$$

where $E(S) = \{\{i, j\} \in E \,:\, i \in S, j \in S\}$.

Since

$$2\sum_{e\in E(S)} x_e + \sum_{e\in\delta(S)} x_e = 2|S|$$

a) From (SEC) we have $2\sum_{e\in E(S)} x_e \le 2|S| - 2$. $\forall S$

Thus $2|S| - \sum_{e\in\delta(S)} x_e = 2\sum_{e\in E(S)} x_e \le 2|S| - 2$, that is

$$\sum_{e\in\delta(S)} x_e \ge 2. \qquad \text{(CUT)}$$

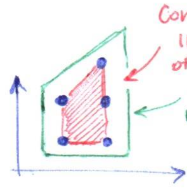
b) (CUT) amounts to $\sum_{e\in\delta(S)} x_e \ge 2$.

Thus $2|S| \ge 2\sum_{e\in E(S)} x_e + 2$, that is

$$\sum_{e\in E(S)} x_e \le |S| - 1. \qquad \text{(SEC)}$$

3.2.2 Ideal formulations

Theorem (Meyer): Let $X \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ be the mixed integer feasible set (feasible region) of an arbitrary MILP with rational coefficients, then $conv(X)$ is a rational polyhedron. Moreover, the extreme points of $conv(X)$ belong to X.



If we consider the subset given by conv(X) where $X = \{\bullet\}$ then this is always a RATIONAL POLYHEDRON (i.e. it can be described as the set of solution of a finite set of linear inequalities / equations) (rational → coefficients of the inequalities / equations are rational)

For bounded integer sets X, intuitive and no need for the rational coefficients assumption.

Consequence: $\boxed{\min\{\underline{c}^t\underline{x} \; : \; \underline{x} \in X\} = \min\{\underline{c}^t\underline{x} \; : \; \underline{x} \in conv(X)\}}$

(M)ILP → LP : We reduce the discrete problem to a linear problem. The fact is that conv(X) is very difficult to characterize explicitly.

If we knew $conv(X)$ explicitly, we could solve the left-hand-side (M)ILP by solving a *single* Linear Program!

Clearly the feasible region P of the LP relaxation of any formulation satisfies $X \subseteq conv(X) \subseteq P$.

Definition: Let $X \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ be any mixed integer feasible set, the **ideal** (perfect) **formulation** for X is the polyhedron $P \subseteq \mathbb{R}^{n_1+n_2}$ with $P = conv(X)$.

Since the ideal formulation is often of exponential size or difficult to determine (also for bounded X), we strive for formulations that closely approximate $conv(X)$.

Examples:

1) **Assignment problem**

Natural ILP formulation:

$$\begin{array}{lll} \min & \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij} & \\ s.t. & \sum_{i=1}^n x_{ij} = 1 & \forall j \\ & \sum_{j=1}^n x_{ij} = 1 & \forall i \\ & \boxed{x_{ij} \in \{0,1\}} & \forall i, \forall j \end{array}$$

Proposition: The polyhedron corresponding to its LP relaxation

$$P = \{\underline{x} \in \mathbb{R}^{n^2} \; : \; \sum_{i=1}^n x_{ij} = 1 \; \forall j, \; \sum_{j=1}^n x_{ij} = 1 \; \forall i, \boxed{0 \le x_{ij} \le 1} \; \forall i, j\}$$

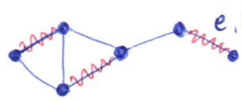
is an *ideal formulation* for the Assignment problem.

Proof later

2) **Perfect Matching problem (PM)**

PM: Given an undirected $G = (V, E)$ with $n = |V|$ even and a cost c_e for each $e = \{i, j\} \in E$, determine a **perfect matching** (i.e., a subset of edges without common nodes but incident to all the nodes) of minimum total cost.

Illustration:



e = edge
c_e = cost of the edge
{ ~~~ } = perfect match

Application: Nodes correspond to persons, and edges $e = \{i, j\} \in E$ to pairs i and j that can be matched.

A natural ILP formulation:

$$\begin{array}{lll} \min & \sum_{e\in E} c_e x_e & \\ s.t. & \boxed{\sum_{e\in\delta(i)} x_e = 1 \quad \forall i \in V} & \\ & x_e \in \{0,1\} & \forall e \in E, \end{array}$$

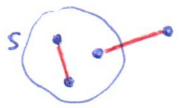
for every i we select just one edge:

$$x_e = \begin{cases} 1 & \text{if } e \text{ is selected} \\ 0 & \text{if } e \text{ is not selected} \end{cases}$$

where $x_e = 1$ if e is selected, and $x_e = 0$ otherwise.

If $|S| = m$ odd :

at most 2 nodes can be matched within S, but then 1 should necessary match with a node outside of S :



$\Longrightarrow$ we should peak at least one edge in the cut induced by the subset S $(\delta(S))$

Clearly all $\underline{x} \in \{0,1\}^{|E|}$ corresponding to matchings satisfy the class of inequalities:

$$\sum_{e \in \delta(S)} x_e \geq 1 \qquad \forall S \subset V \text{ with } |S| \text{ odd.}$$

for sure these inequalities are satisfied by the perfect matching $\Longrightarrow$ we add it at the formulation

Theorem (Edmonds):
$P_M = \{\underline{x} \in \mathbb{R}^{|E|} \ : \ \sum_{e \in \delta(i)} x_e = 1 \ \forall i \in V, \boxed{\sum_{e \in \delta(S)} x_e \geq 1 \ \forall S \subset V, \ |S| \text{ odd},}$
$0 \leq x_e \leq 1 \ \forall e \in E\}$

is an ideal formulation for the Perfect Matching problem.

3.2.3 Extended formulations

Alternative formulations can use additional and/or different variables.

Definition: The formulations including additional variables, are **extended formulations.**

Example: **Uncapacitated Lot-Sizing (ULS)**

Plan the production of a single type of product for the next n periods.

Assumption: the stock is empty at the beginning and empty at the end.

Given

- f_t fixed cost for producing during period t
- p_t unit production cost in period t
- h_t unit storage cost in period t
- d_t demand in period t

determine a production plan for the next n periods that minimizes the total (production and storage) costs, while satisfying the demand in each period.

MILP formulation

Variables:

- x_t = amount produced in period t, with $1 \leq t \leq n$
- s_t = amount in stock at the end of period t, with $0 \leq t \leq n$
- $y_t = 1$ if production occurs in period t and $y_t = 0$ otherwise, with $1 \leq t \leq n$

$$\begin{array}{lrll}
\min & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} f_t y_t & & \\
s.t. & s_t = s_{t-1} + x_t - d_t & \forall t \\
& x_t \leq M y_t & \forall t \\
& s_0 = 0, s_n = 0 & \forall t \\
& s_t, x_t \geq 0 & \forall t \\
& y_t \in \{0,1\} & \forall t
\end{array}$$

(production + storage + fixed) costs

where $M > 0$ is large enough, for instance, $M = \sum_{t=1}^{n} d_t + s_n - s_0$.

N.B.: Since $s_t = \sum_{i=1}^{t} x_i + s_0 - \sum_{i=1}^{t} d_i$, storage variables s_t can be deleted.

Extension with minimum lot sizes.

MILP extended formulation for ULS : we're willing to introduce more variables to get a stronger formulation

Variables:

- w_{it} = amount produced in period i to satisfy the demand in period t, with $1 \leq i \leq t \leq n+1$
- $y_t = 1$ if production occurs in period t and $y_t = 0$ otherwise, with $1 \leq t \leq n$

(at least a part of the demand in periodt t)

$$\begin{array}{lrll}
\min & \sum_{i=1}^{n} \sum_{t=i}^{n} c_{it} w_{it} + \sum_{t=1}^{n} f_t y_t & & \\
s.t. & \sum_{i=1}^{t} w_{it} = d_t & \forall t, 1 \leq t \leq n \\
& \sum_{i=1}^{n} w_{i,n+1} = 0 & \text{(stock at the end)} \\
& w_{it} \leq d_t y_i & \forall i, t. i \leq t \\
& w_{it} \geq 0 & \forall i, t. i \leq t \\
& y_t \in \{0,1\} & \forall t
\end{array}$$

with aggregate costs (production and storage) $c_{it} = p_i + h_i + \ldots + h_{t-1}$.

we're creating another variable: c_{it} = cost of producing something in i and taking it stored till period $t-1$ (production + storage)

Explicit extended formulation is obtained by adding, for each period i, constraints $x_i = \sum_{t=i}^{n} w_{it}$ and $s_i = \sum_{l=1}^{i} \sum_{t=i+1}^{n+1} w_{lt}$.

Comparison between formulations with different variables

For simplicity of notation, consider an ILP formulation

$$\min\{\underline{c}^t\underline{x} \; : \; \underline{x} \in P_1 \cap \mathbb{Z}^n\}$$

with $P_1 \subseteq \mathbb{R}^n$, and an extended formulation

$$\min\{\underline{c}^t(\underline{x},\underline{w}) \; : \; (\underline{x},\underline{w}) \in P_2 \cap (\mathbb{Z}^n \times \mathbb{R}^{n'})\}$$

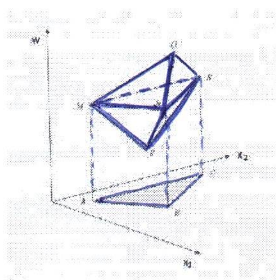
with $P_2 \subseteq \mathbb{R}^n \times \mathbb{R}^{n'}$.

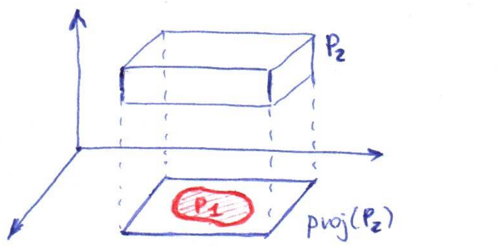
There is an issue: we want to compare two formulations that are associated to different variables, the two polyedrons are in different spaces ⟶ we need to **project** them on the same space

Definition: Given a polyhedron $P \subseteq \mathbb{R}^n \times \mathbb{R}^{n'}$, the orthogonal projection of P onto the x-subspace $\mathbb{R}^n$ is the polyhedron

$$\text{proj}_{\underline{x}}(P) = \{\underline{x} \in \mathbb{R}^n \; : \; \exists \, \underline{w} \in \mathbb{R}^{n'} \text{ s.t. } (\underline{x},\underline{w}) \in P\}.$$

Example: orthogonal projection of a 3-D polyhedron





If this is the situation then P_1 is better (stronger) than P_2.

To compare a formulation $P_1 \subseteq \mathbb{R}^n$ and an extended formulation $P_2 \subseteq \mathbb{R}^n \times \mathbb{R}^{n'}$, we compare P_1 and $\text{proj}_{\underline{x}}(P_2)$.

One way to determine the orthogonal projection of polyhedra onto subspaces:

Fourier-Motzkin elimination method (1827)

First procedure to find a feasible solution of systems of linear inequalities.

Idea: At each iteration one variable x_i is eliminated (an equivalent linear system without x_i is derived), the process ends when the resulting system contains a single variable.

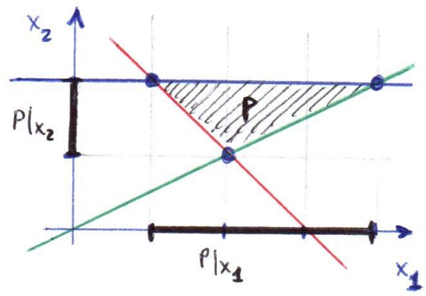
Given $A\underline{x} \geq \underline{b}$, suppose we wish to eliminate variable x_i.

The equivalent system without x_i includes

- all the inequalities of $A\underline{x} \geq \underline{b}$ in which x_i does not appear,
- the inequalities resulting from all the possibile combinations of the upper and lower bounds on x_i implied by all the inequalities of $A\underline{x} \geq \underline{b}$ containing x_i.

Example:

$$\begin{array}{rrl} x_1 & +x_2 & \geq 3 \qquad (4)\\ -\frac{1}{2}x_1 & +x_2 & \geq 0 \qquad (5)\\ & -x_2 & \geq -2 \qquad (6) \end{array}$$



- Eliminate x_2 (project polyhedron P defined by (4)-(6) onto the subspace of x_1):

we isolate one variable (x_2):

$$\begin{array}{l} 1. \; 3 - x_1 \leq x_2 \\ 2. \; \frac{1}{2}x_1 \leq x_2 \\ \qquad\qquad x_2 \leq 2 \; 3. \end{array}$$

and, considering all the pairs of inequalities, one obtains

we do the combination of all upper and lower bounds:
1. ≤ 3.
2. ≤ 3.

and we eliminate it (elimination method):

$$\boxed{\begin{array}{rcl} 3 - x_1 & \leq & 2 \\ \frac{1}{2}x_1 & \leq & 2, \end{array}}$$

namely $\boxed{1 \leq x_1 \leq 4,}$ hence the projection $[1,4]$.

- Eliminate x_1 (project polyhedron P onto the subspace of x_2): one obtains $1 \leq x_2 \leq 2$ and hence the projection $[1,2]$.

Complexity: since at each step an inequality is derived for each pair of upper-lower bounds on x_i, the number of constraints can grow exponentially in **n**.
(so it's not so efficient)

Comparing formulations for ULS:

$M > \sum_{t=1}^{n} d_t + s_n$

Consider the formulation P_1 defined by

$$\boxed{\begin{array}{rl} s_t = s_{t-1} + x_t - d_t & \forall t \\ x_t \le M y_t & \forall t \\ s_0 = 0, s_t \ge 0, x_t \ge 0, 0 \le y_t \le 1 & \forall t \end{array}} \quad P_1 \qquad (7)$$

relaxation

and $\text{proj}_{\underline{x},\underline{s},\underline{y}}(P_2)$, with P_2 defined by

$$\boxed{\begin{array}{rll} \sum_{i=1}^{t} w_{it} = d_t & \forall t & \\ w_{it} \le d_t y_i & \forall i, t. 1 \le i \le t & (8) \\ x_i = \sum_{t=i}^{n} w_{it} & \forall i & (9) \\ s_i = \sum_{l=1}^{i} \sum_{t=i+1}^{n+1} w_{lt} & \forall i & (10) \\ w_{it} \ge 0 & \forall i, t. 1 \le i \le t & \\ 0 \le y_t \le 1 & \forall t. & \end{array}} \quad P_2$$

to compare them we have to get rid of the w variable in P_2 $\Rightarrow \text{proj}_{\underline{x},\underline{s},\underline{y}}(P_2)$

We want to get rid of w_{it}:

$x_i = \sum_{t=i}^{n} w_{it} \le \sum_{t=i}^{n} d_t y_i$

$x_i \le \sum_{t=i}^{n} d_t y_i < M y_i$

$< M$

$\Rightarrow d_t < M\left(\frac{d_t}{M}\right)$ ↯

Easy to verify that $\boxed{\text{proj}_{\underline{x},\underline{s},\underline{y}}(P_2) \subset P_1}$ e.g., the point $\boxed{x_t = d_t}$ $\boxed{y_t = d_t/M}$ for all t is an extreme point of P_1 that does not belong to $\text{proj}_{\underline{x},\underline{s},\underline{y}}(P_2)$.

Proposition: P_2 is the ideal formulation of ULS.

3.2.4 Stronger extended formulations

A way to strengthen a formulation is to look for an extended formulation involving additional variables whose projection is a better approximation of the ideal formulation ($conv(X)$).

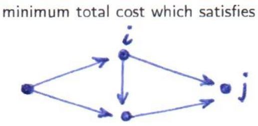
Example: **Fixed charge network flow problem (FCNF)**:

Given a directed $G = (V, A)$ with

- a positive fixed cost f_{ij}, a unit cost c_{ij} and a capacity u_{ij} for each $(i, j) \in A$,
- a demand b_i for each $i \in V$ ($b_i < 0$ for sources and $b_i > 0$ for destinations) with $\sum_{i \in V} b_i = 0$,

determine a feasible flow of minimum total cost which satisfies all demands and capacity constraints.

Example:



f_{ij} fixed cost
c_{ij} unit cost
u_{ij} capacity
b_i demand

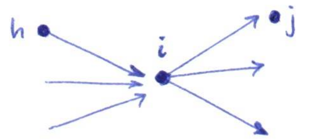
Proposition: FCNF is NP-hard.

1) Natural MILP formulation:

Variables:

- x_{ij} = amount of flow through (i, j), for all $(i, j) \in A$
- $y_{ij} = 1$ if (i, j) is used and $y_{ij} = 0$ otherwise, for all $(i, j) \in A$ ← always when we have fixed costs

How we characterize the flows? We need to link the x_{ij} variables: for each node i all that enter minus all that go out is equal to demand:



$$\begin{array}{lll} \min & \sum_{(i,j) \in A} (c_{ij} x_{ij} + f_{ij} y_{ij}) & \\ \text{s.t.} & \boxed{\sum_{(h,i) \in \delta^-(i)} x_{hi} - \sum_{(i,j) \in \delta^+(i)} x_{ij} = b_i \quad \forall i \in V} & (11) \\ & 0 \le x_{ij} \le u_{ij} y_{ij} \quad \forall (i,j) \in A & (12) \\ & y_{ij} \in \{0, 1\} \quad \forall (i,j) \in A & \end{array}$$

→ this makes the formulation weak since $x_{ij} \le u_{ij} y_{ij}$ and the maximum capacity u_{ij} may be a lot larger than x_{ij}

where $\delta^+(i) = \{(i, j) \in A : j \in V\}$ and $\delta^-(i) = \{(h. i) \in A : h \in V\}$

Its LP relaxation yields poor bounds because of the weak coupling between the x_{ij} and y_{ij}, imposed by (12).

2) Multi-commodity extended MILP formulation: (see Computer Lab 1)

Idea: Suppose w.l.o.g. $\exists$ a single source node (say node s, with $b_s = -\sum_{i \in V \setminus \{s\}} b_i$) and decompose the flows according to their destinations.

Let $K \subseteq V$ be the set of nodes with strictly positive demand.
Define one "commodity" for each $k \in K$, with the flow variables x_{ij}^k for all $(i, j) \in A$.
Define $d_i^k = -b_k$ if $i = s$, $d_i^k = b_k$ if $i = k$, and $d_i^k = 0$ otherwise.

Variables: ...

Significantly stronger (re)formulation of FCNF but with $|K|$ times more variables and constraints.

3.2.5 Remarks on the strength and size of formulations

Definition: A **compact formulation** for a given problem is a formulation with a number of variables and constraints that is polynomial w.r.t. the instance size.

Remark 1: A compact extended formulation can be much weaker than an alternative exponential-size formulation.

we should not rely on the size of the formulation

Example: Asymmetric TSP (ATSP)

To exclude subtours, instead of (SEC) one can add, for each $i \in V$, a variable t_i representing the "position" in which node i is visited in the tour and an appropriate set of constraints.

See Computer Lab 1

Remark 2: A compact extended formulation can have a projection into the space of the natural variables that is of exponential size.

Example: ATSP

3.3 "Easy" ILP problems and totally unimodular matrices

Using the concept of totally unimodular matrices we can find out if a formulation for an ILP is an IDEAL formulation (only ILP, not MILP).

Consider a generic ILP

$$\boxed{\min\{\underline{c}^t\underline{x} \ : \ A\underline{x} = \underline{b}, \ \underline{x} \in \mathbb{Z}^n_+\}} \tag{1}$$

we always assume integrality of the coefficients in ILP problems

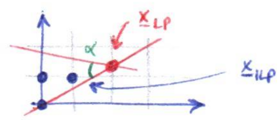
where $A \in \mathbb{Z}^{m \times n}$ with $n \geq m$, and $\underline{b} \in \mathbb{Z}^m$.

$P(\underline{b}) = \{\underline{x} \in \mathbb{R}^n \ : \ A\underline{x} = \underline{b}, \underline{x} \geq \underline{0}\}$ feasible region (polyhedron) of its LP relaxation.

Assumption: A is of full rank (since $n \geq m$, rank$(A)=m$), i.e, $\nexists$ redundant constraints. ≡ no constraints are redundant

In general, optimal solutions of LP relaxation far away from optimal solutions of ILP (1).

Illustration:



and if we "close" the angle α then x_{LP} goes away

If all extreme points of $P(\underline{b})$ are integer, the formulation is ideal and ILP (1) can be solved by just solving its LP relaxation.

According to **Linear Programming theory**:

- Any LP $\min\{\underline{c}^t\underline{x} \ : \ A\underline{x} = \underline{b}, \ \underline{x} \geq \underline{0}\}$ with a finite optimal solution has an optimal extreme point.
- To each extreme point of $P(\underline{b})$ corresponds (at least) one **basic feasible solution**

$$\underline{x} = (\underline{x}_B, \underline{x}_N) = (B^{-1}\underline{b}, \underline{0}),$$

where B is a **basis** of A, i.e., an $m \times m$ non-singular submatrix of A.

Consider any basis B.

By partitioning the columns of A into basic and non basic, $A\underline{x} = \underline{b}, \ \underline{x} \geq \underline{0}$ can be written as

$$B\underline{x}_B + N\underline{x}_N = \underline{b} \ \text{ with } \underline{x}_B \geq \underline{0} \text{ and } \underline{x}_N \geq \underline{0}.$$

and can be expressed in canonical form:

$$\underline{x}_B = B^{-1}\underline{b} - B^{-1}N\underline{x}_N \ \text{ with } \underline{x}_B \geq \underline{0} \text{ and } \underline{x}_N \geq \underline{0},$$

which emphasizes the basic feasible solution $(\underline{x}_B, \underline{x}_N) = (B^{-1}\underline{b}, \underline{0})$.

Observation: If an optimal basis B has $det(B) = \pm 1$, the optimal solution of LP relaxation of (1) is integer and hence also optimal for ILP (1).

Proof:

Recall that $\underline{x}_B = B^{-1}\underline{b}$ and

$$B^{-1} = \frac{1}{det(B)} C^t,$$

where C is the matrix of cofactors $\alpha_{ij} = (-1)^{i+j} det(B_{ij})$ with submatrix B_{ij} obtained from B by deleting the i-th row and the j-th column.

If all coefficients of B are integer, all cofactors α_{ij} are integer.

If $det(B) = \pm 1$, B^{-1} is integer and, since $\underline{b}$ is integer, also the basic feasible solution $(\underline{x}_B, \underline{x}_N) = (B^{-1}\underline{b}, \underline{0})$ is integer.

□

Since $B^{-1}\underline{b}$ is integer also if $det(B) = 2$ and all b_is are even integers, $det(B) = \pm 1$ is only a sufficient condition for the integrality of $(\underline{x}_B, \underline{x}_N) = (B^{-1}\underline{b}, \underline{0})$.

3.3.1 Totally unimodular matrices and optimal integer solutions

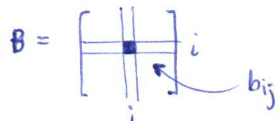
Definition: An $m \times n$ matrix A with integer coefficients is **totally unimodular** (TU) if every squared submatrix has a determinant -1, 0 or 1.

Clearly, if A is TU, $a_{ij} \in \{-1, 0, 1\}$ for all i and j.

Examples of TU and not TU matrices:

$$\text{TU} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \text{not TU} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Recall: For any $m \times m$ matrix B the *Laplace expansion* along any row i, with $1 \leq i \leq m$, is $det(B) = \sum_{j=1}^{m} b_{ij}\alpha_{ij}$, where $\alpha_{ij} = (-1)^{i+j} det(B_{ij})$ are the cofactors of B.
Analogously the expansion along any column j is $det(B) = \sum_{i=1}^{m} b_{ij}\alpha_{ij}$.



$$det(B) = \sum_{i=1}^{m} b_{ij}\alpha_{ij}$$

cofactors of **B**

Proposition:

- A is TU if and only if A^t is TU.
- A is TU if and only if $(A \,|\, I_m)$ is TU.
- A' obtained from A by permuting and/or changing the sign of some columns and/or rows is TU if and only if A is TU.

Theorem 1:
Consider an $m \times n$ TU matrix A and an integer $\underline{b}$ such that

$$P(\underline{b}) = \{\underline{x} \in \mathbb{R}^n \ : \ A\underline{x} = \underline{b}, \underline{x} \geq \underline{0}\} \neq \emptyset,$$

then all extreme points of $P(\underline{b})$ are integer.

Proof: As for the previous observation.

From the ILP point of view, if A is TU it suffices to solve the LP relaxation.

Corollary:
Consider an $m \times n$ TU matrix A and an integer $\underline{b}$ such that

$$P(\underline{b}) = \{\underline{x} \in \mathbb{R}^n \ : \ A\underline{x} \geq \underline{b}, \underline{x} \geq \underline{0}\} \neq \emptyset,$$

then all the extreme points (vertices) of the polyhedron $P(\underline{b})$ are integer.

Proof:
Let $\underline{\tilde{x}}$ be any extreme point of $P(\underline{b})$.

First we show that $(\underline{\tilde{x}}, \underline{\tilde{s}})$ with $\underline{\tilde{s}} := A\underline{\tilde{x}} - \underline{b}$ is an extreme of

$$P'(\underline{b}) := \{(\underline{x}, \underline{s}) \in \mathbb{R}^{n+m} \ : \ A\underline{x} - \underline{s} = \underline{b}, (\underline{x}, \underline{s}) \geq \underline{0}\}.$$

If not, there would exist two distinct $(\underline{x}_1, \underline{s}_1)$ and $(\underline{x}_2, \underline{s}_2)$ of $P'(\underline{b})$ such that $(\underline{\tilde{x}}, \underline{\tilde{s}}) = \alpha(\underline{x}_1, \underline{s}_1) + (1-\alpha)(\underline{x}_2, \underline{s}_2)$ for some α with $0 < \alpha < 1$.

Since $\underline{s}_1 = A\underline{x}_1 - \underline{b} \geq \underline{0}$ and $\underline{s}_2 = A\underline{x}_2 - \underline{b} \geq \underline{0}$, $\underline{x}_1$ and $\underline{x}_2$ belong to $P(\underline{b})$.

Moreover, $(\underline{x}_1, \underline{s}_1) \neq (\underline{x}_2, \underline{s}_2)$ would imply $\underline{x}_1 \neq \underline{x}_2$ and hence $\underline{\tilde{x}} = \alpha\underline{x}_1 + (1-\alpha)\underline{x}_2$ could not be a vertex of $P(\underline{b})$.

Since A is TU, also $(A \,|\, -I_m)$ is TU. According to Theorem 1 for $P'(\underline{b})$, $(\underline{\tilde{x}}, \underline{\tilde{s}})$ is integer, hence also $\underline{\tilde{x}}$ is integer. □

(*) TU satisfying the cond:

$$\begin{bmatrix} 1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$I = \{1,2,3,4\} = I_1$
$I_2 = \emptyset$

TU not satisfying the cond:

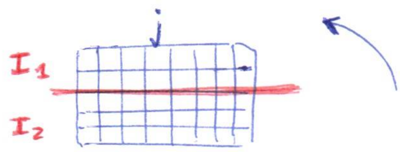
$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3 non-zero elements

Sufficient conditions for a matrix to be TU

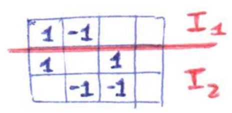
Proposition: An $m \times n$ matrix A with integer coefficients is TU if

i) $a_{ij} \in \{-1, 0, +1\}$ for all i and j,
ii) each column of A contains at most two nonzero coefficients,
iii) the set I of the indices of the rows of A can be partitioned into two subsets I_1 and I_2 ($I_1 \cup I_2 = I$ and $I_1 \cap I_2 = \emptyset$) such that, for each column j with two nonzero coefficients, we have $\sum_{i \in I_1} a_{ij} - \sum_{i \in I_2} a_{ij} = 0$.



The condition must be valid (remember: it's just sufficient) for a fixed partition and for $\forall j$ (column).

N.B.: If a column has two nonzero coefficients of the same (different) sign, their rows must belong to different subsets (same subset) of indices I_1 and I_2.



Proof:
Suppose A is not TU, and let Q be a smallest squared submatrix of A among those with $det(Q) \notin \{-1, 0, 1\}$.

Q cannot contain a column with a single nonzero coefficient, otherwise Q would not be minimal. Thus, each column of Q contains exactly two nonzero coefficients.

According to (iii), $\sum_{i \in I_1} a_{ij} - \sum_{i \in I_2} a_{ij} = 0$ and hence, since $a_{ij} = 0$ for every column j of Q and every row i not in Q, we would have $det(Q) = 0$, which is a contradiction. □

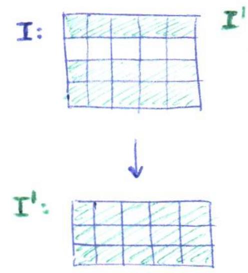
Examples of TU matrices (not) satisfying these conditions: (*)

Strong characterization but difficult to be applied

Characterization of TU matrices

Theorem 2: An integer $m \times n$ matrix A is TU if and only if each subset $I' \subseteq I = \{1, \dots, m\}$ of indices of the rows of A can be partitioned into two subsets I'_1 and I'_2 such that $\left(\sum_{i \in I'_1} a_{ij} - \sum_{i \in I'_2} a_{ij}\right) \in \{-1, 0, +1\}$ for every column j, with $1 \leq j \leq n$.

(every subset I', not only the full I)



the condition must be valid ($\forall j$)

If A is TU it suffices to solve the LP relaxation of ILP.

In some sense also the converse is true: linear relaxation

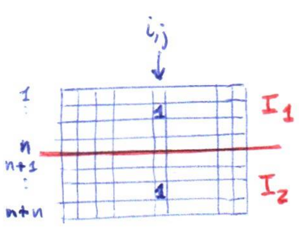
Proposition: $\min\{\underline{c}^t\underline{x} \ : \ A\underline{x} = \underline{b}, \underline{x} \in \mathbb{R}^n_+\}$ has an optimal integer solution for any integer $\underline{b}$ for which it admits a finite optimal solution if and only if A is TU.

! So for a matrix being TU is the right property that really explain when the optimal solution of the linear relaxation ($\forall \underline{c}$) has all integer components.

Given A and a basis B with $det(B) \neq \pm 1$, there always exists a LP $\min\{\underline{c}^t\underline{x} \ : \ A\underline{x} = \underline{b}, \underline{x} \in \mathbb{R}^n_+\}$, for appropriate $\underline{c}$ and $\underline{b}$, with a fractional optimal solution.

3.3.2 Some natural formulations that are ideal

i = jobs
j = machines



1) Assignment problem

Given n jobs and n machines with costs c_{ij} for all $i, j \in \{1, \dots, n\}$, decide which job to assign to which machine so as to minimize the total cost to complete all the jobs.

ILP formulation:

$$\begin{array}{llll} \min & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} & & \\ & \sum_{i=1}^n x_{ij} = 1 & \forall j & (2) \\ & \sum_{j=1}^n x_{ij} = 1 & \forall i & (3) \\ & \boxed{x_{ij} \in \{0, 1\} \quad \forall i, \forall j} & & \end{array}$$

where $x_{ij} = 1$ if i-th job is assigned to j-th machine, with $1 \leq i, j \leq n$.

linear programming relaxation:
$0 \leq x_{ij} \leq 1 \quad \forall i, j$
Actually $x_{ij} \leq 1$ is not necessary since it's implied by (2) and (3).

N.B.: In the LP relaxation, $x_{ij} \geq 0 \ \forall i, j$ suffice, because (2)-(3) imply $x_{ij} \leq 1 \ \forall i, j$.

Property: The matrix of constraints (2)-(3) is TU.

In the sufficient conditions for TU, just take $I_1 = \{1, \dots, n\}$ and $I_2 = \{n+1, \dots, 2n\}$.

Consequence: Each extreme point of the feasible region of the LP relaxation is integer, and the formulation is ideal.

In I_1 we put the jobs, in I_2 we put the machines: each column says which job $i \in \{1, .., n\}$ is being assigned to which machine $j \in \{1, .., n\}$.
In the example the 2^{nd} job is being assigned to the 3^{rd} machines.
Since the matrix is built like that $\Rightarrow$ A is TU.

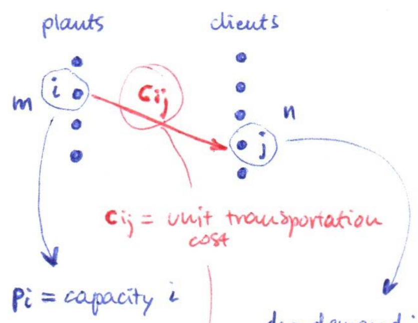
(special case of the uncapacitated facilities location problem)

2) Transportation problem

Suppose there is a single type of product.

Given

- m production plants ($1 \leq i \leq m$)
- n clients ($1 \leq j \leq n$)
- c_{ij} = transportation cost of one unit of product from plant i to client j
- p_i = maximum amount that can be produced (capacity) at plant i
- d_j = demand of client j
- q_{ij} = maximum amount that can be transported from plant i to client j



determine a transportation plan so as to minimize total transportation costs while satisfying all client demands and all the plant capacity constraints.

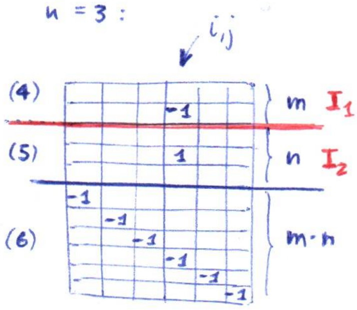
Assumption: $\sum_{i=1}^m p_i \geq \sum_{j=1}^n d_j$ (to guarantee feasibility)

Without loss of generality: $\sum_{i=1}^m p_i = \sum_{j=1}^n d_j$.

$\sum$ capacities $= \sum$ demands

q_{ij} = maximum capacity that can be transported from i to j

$m = 2$
$n = 3$:



We consider just the first part of the matrix. Since it's TU we can add an identity matrix and it still be TU.
We add $-I_{m \cdot n}$.

(Remember that the matrix is the matrix of the coefficients of x_{ij})

Natural ILP formulation:

Variables: x_{ij} = amount of product transported from plant i to client j, with $1 \leq i \leq m$, $1 \leq j \leq n$

$$\begin{array}{llll} \min & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} & & \\ & \boxed{\sum_{j=1}^n x_{ij} \leq p_i \quad \forall i} & & (4) \\ & \sum_{i=1}^m x_{ij} \geq d_j & \forall j & (5) \\ & \boxed{x_{ij} \leq q_{ij} \quad \forall i, \forall j} & & (6) \\ & x_{ij} \geq 0 \ \text{ integer} & \forall i, \forall j & \end{array}$$

We know that if $A\underline{x} \geq \underline{b}$ and $\underline{x} \geq \underline{0}$ and A is TU $\Rightarrow$ all the extreme points are integers
$\Rightarrow$ (4) becomes:
$-\sum_{j=1}^n x_{ij} \geq -p_i \quad \forall i$
(6) becomes:
$-x_{ij} \geq -q_{ij} \quad \forall i \forall j$

Property: The matrix of constraints (4)-(6) is TU.

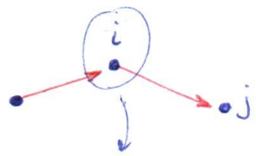
Proof: Problem in consistent inequality form by multiplying constraints (4) and (6) by -1. Since matrix of demand and capacity constraints is TU, just apply the Corollary.

Consequence: If all p_i, d_j and q_{ij} are integer, every extreme point of the feasible region of LP relaxation is integer and, hence, the formulation is ideal.

they must be integers

3) Minimum cost flow problem

Given a directed $G = (V, A)$ with a capacity u_{ij} and a unit cost c_{ij} for each $(i,j) \in A$, and a demand/availability b_i for each $i \in V$ ($b_i < 0$ for sources, $b_i > 0$ for destinations, $\sum_{i \in V} b_i = 0$), determine a feasible flow of minimum total cost satisfying all demands.

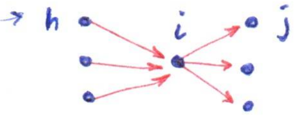


$b_i < 0 \Rightarrow i$ is a source
$b_i > 0 \Rightarrow i$ is a destination
$b_i = 0 \Rightarrow i$ is an intermediate point (all that enters goes out)

Natutal ILP formulation:

$$\begin{array}{lll} \min & \sum_{(i,j)\in A} c_{ij} x_{ij} & \\ & \boxed{\sum_{(h,i)\in\delta^-(i)} x_{hi} - \sum_{(i,j)\in\delta^+(i)} x_{ij} = b_i \quad \forall i \in V} & (7) \\ & x_{ij} \le u_{ij} \quad \forall (i,j) \in A & (8) \\ & x_{ij} \ge 0 \text{ integer} \quad \forall (i,j) \in A & \end{array}$$

where $\delta^+(i) = \{(i,j) \in A : j \in V\}$ and $\delta^-(i) = \{(h,i) \in A : h \in V\}$.



(7) creates: element (i,j)

1
0
0
-1
0
0

exactly 2 non-zero components for every column.
This is called INCIDENCE MATRIX OF THE NETWORK (G).
We can partition it as $A' = I_1$, $I_2 = \emptyset$ and so A' is TU.
(8) creates $-I_{|V|}$, so the matrix $A = \begin{bmatrix} A' \\ -I_{|A|} \end{bmatrix}$ is TU.

Property: The matrix of constraints (7)-(8) is TU.

Proof: The $n \times |A|$ node-arc incidence matrix of any directed graph is TU since it contains exactly one 1 and one -1 per column (take $I_1 = I$ and $I_2 = \emptyset$).

...

We consider the entering flow as positive $\Rightarrow$ when we write about (h,i) we put "+", when we write about (i,j) we put "−".

Consequence: If all demands/availabilities b_i and capacities u_{ij} are integer, every extreme point is integer and, thus, the formulation is ideal.

! Exercise:

Show that the following well-known problems are special cases of the Min cost flow problem.

- Shortest path: Given a directed $G = (V, A)$ with a cost c_{ij} for each arc $(i,j) \in A$, and two prescribed nodes s and t, determine a minimum cost path from s to t.

- Maximum flow: Given a directed $G = (V, A)$ with a capacity u_{ij} for each arc $(i,j) \in A$, a source s and a sink t, determine a feasible flow of maximum value from s to t.

A strategy is reduce these problems as special cases of the minimum cost flow problems

$\Longrightarrow$ the natural formulation is an ideal formulation
$\Longrightarrow$ we have **ideal formulations** for these problems

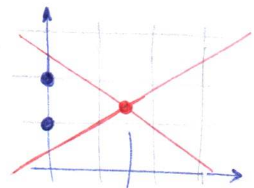
Conclusions:
It's very important to check if A is TU. If it is, we have an IDEAL formulation. If it is not we cannot rely on solving just the single linear programming relaxation.

Rounding optimal solutions of the LP relaxation

In general, when the constraint matrix of an ILP is not TU, the optimal solution $\underline{x}^*_{LP}$ of its LP relaxation is fractional.

Rounding the fractional components of $\underline{x}^*_{LP}$ to the closest integer value does rarely work because

- the rounded solution are often infeasible for ILP,
- even if the rounded solution is feasible for ILP, the error with respect to an optimal ILP solution may be arbitrarily large.



If we round this LP solution we don't obtain a feasible solution for the ILP problem

Rounding $\underline{x}^*_{LP}$ yields to a good approximation of the optimal solution only when the components of $\underline{x}^*_{LP}$ have large values.

Illustration of the different cases:

3.4 Relaxations, heuristics and bounds

Consider a generic Discrete Optimization problem

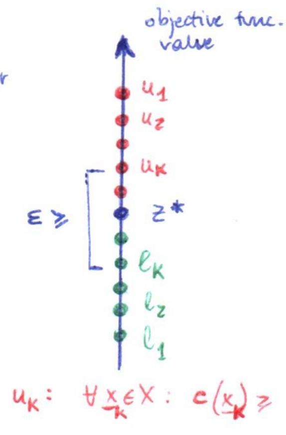
$$\boxed{z^* = \min\{c(\underline{x}) \ : \ \underline{x} \in X\}}$$

$c(\underline{x})$ because it's not necessarily linear (not nec. $\underline{c}^T\underline{x}$)

and an optimal solution $\underline{x}^* \in X$.

Algorithms generate a decreasing sequence of upper bounds $u_1 > \ldots > u_k \ge z^*$ as well as an increasing sequence of lower bounds $l_1 < \ldots < l_k \le z^*$.
Termination criterion: $(u_k - l_k) \le \varepsilon$ for a given $\varepsilon > 0$. accuracy level

We do this to have a solution $\underline{x}_K$ s.t. $\underline{x}_K$ is a feasible solution which has an obj. function of u_K and we know that the optimal objective function will never be smaller than l_K. So we'll know that:
$u_K - l_K \le \varepsilon \Rightarrow \boxed{u_K - z^* \le \varepsilon}$
$\Rightarrow$ we have a guarantee on the quality of the solution



$u_K: \ \forall \underline{x}_K \in X: \ c(\underline{x}_K) \ge z^*$

Primal bounds

For minimization problems, any feasible solution $\overline{x} \in X$ yields an upper bound $\overline{u} = c(\overline{x})$ on z^*, namely $\overline{u} \ge z^*$.

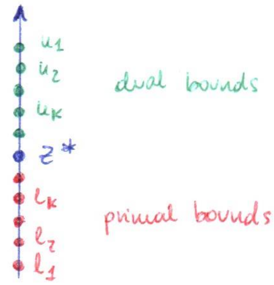
In some cases, even finding a feasible solution may be challenging (NP-hard).

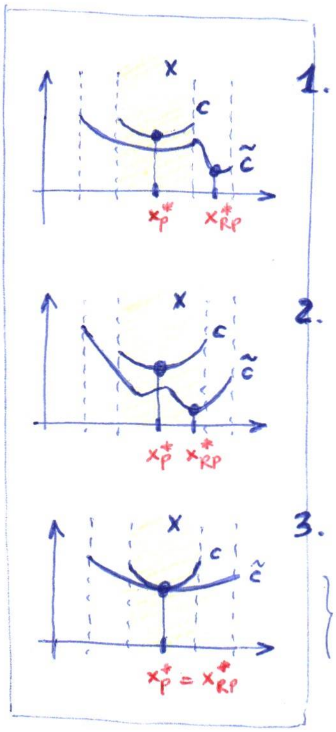
Dual bounds

For minimization problems, lower bounds are obtained by considering a relaxation.

Quality guarantee:
If $\underline{x}_k$ is the best feasible solution found so far and l_k the best dual bound, the termination criterion $(c(\underline{x}_k) - l_k) \leq \varepsilon$, for a given $\varepsilon > 0$, guarantees $(c(\underline{x}_k) - z^*) \leq \varepsilon$.

For maximization problems, the primal (dual) bounds are lower (upper) bounds.





Definition: Given a problem

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

a problem

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

is a **relaxation** of P if

- $X \subseteq \tilde{X}$
- $\tilde{c}(\underline{x}) \leq c(\underline{x})$ for each $\underline{x} \in X$.

We always considered as relaxation the elimination of the integrality constraint (we always kept the same objective funct.). The general concept of relaxation may involve a different objective function.
$(X, c \longrightarrow \tilde{X}, \tilde{c})$

Proposition: If RP is a relaxation of P then $\tilde{z} \leq z^*$.

Proof: Let $\underline{x}^*$ be an optimal solution of P, then $\underline{x}^* \in X \subseteq \tilde{X}$ and $\tilde{c}(\underline{x}^*) \leq c(\underline{x}^*) = z^*$. Since $\underline{x}^* \in \tilde{X}$, we have $\tilde{z} \leq \tilde{c}(\underline{x}^*)$.

! **Proposition**: Let $\underline{x}^*_{RP}$ be an optimal solution of RP. If $\underline{x}^*_{RP}$ is feasible for P ($\underline{x}^*_{RP} \in X$) and $\tilde{c}(\underline{x}^*_{RP}) = c(\underline{x}^*_{RP})$, then $\underline{x}^*_{RP}$ is also optimal for P.

It can generate 3 cases:
1. x^*_{RP} unfeasible for (P)
2. x^*_{RP} feasible for (P) (since $\in X$) but not optimal
3. x^*_{RP} feasible and optimal

Tradeoff between the bound quality $(z^* - \tilde{z})$ and the computational load of RP.

3.4.1 Different types of relaxations

1) Linear programming relaxation

Definition: Given an arbitrary (M)ILP

$$z^* = z_{ILP} = \min \quad \underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y}$$
$$A_1 \underline{x} + A_2 \underline{y} \geq \underline{b}$$
$$\underline{x} \geq \underline{0}, \boxed{\underline{y} \geq \underline{0}, \text{ integer}}$$

its **linear programming relaxation** is

$$\tilde{z} = z_{LP} = \min \quad \underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y}$$
$$A_1 \underline{x} + A_2 \underline{y} \geq \underline{b}$$
$$\underline{x} \geq \underline{0}, \boxed{\underline{y} \geq \underline{0}}$$

where the integrality constraints on y_js are omitted.

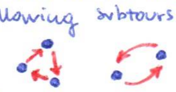
Recall: $z_{LP} \leq z^*$ and the *stronger* the (M)ILP formulation, the *tighter the dual bound* z_{LP}.
$\to z_{ILP}$

2) Relaxation by elimination

by elimination we don't get strong formulations

Simply delete one ore more constraints.

Notice that in this way we're allowing subtours:



in this way $\tilde{z}$ tends to be very bad with respect to z^*.

Examples:

1) Asymmetric TSP

Delete the subtour elimination (cut-set) constraints and just keep the assignment ones.

2) *Multi-dimensional binary knapsack problem*

$$\max \quad \sum_{j=1}^n p_j x_j$$
$$\text{s.t.} \quad \sum_{j=1}^n w_{ij} x_j \leq W_i \qquad \forall i \in \{1, 2, \ldots, m\} \qquad (1)$$
$$x_j \in \{0, 1\} \qquad \forall j \in \{1, 2, \ldots, n\} \qquad (2)$$

items $j \in \{1, \ldots, n\}$
w_{ij} = usage of the i-th dimension for the j-th item

Deleting all but one constraints, we obtain a standard binary knapsack problem.

(Very) weak relaxations.

3) Surrogate relaxation

Idea: Replace a subset of constraints with the surrogate constraint, i.e., their linear combination with multipliers $\lambda_i \geq 0$.

Example: *Multiple binary knapsack problem*

Given m knapsacks of capacities W_i, select m disjoint subsets of items that fit in the knapsacks so as to maximize total profit.

$$
\begin{aligned}
z_{mKP} = \max \quad & \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} \\
\text{s.t.} \quad & \boxed{\textstyle\sum_{j=1}^{n} w_j x_{ij} \leq W_i \quad \forall i \in \{1, 2, \ldots, m\}} && (3) \\
& \textstyle\sum_{i=1}^{m} x_{ij} \leq 1 \quad \forall j \in \{1, 2, \ldots, n\} && (4) \\
& x_{ij} \in \{0, 1\} \quad \forall i, \forall j && (5)
\end{aligned}
$$

$x_{ij} = \mathbb{1}\{j$-th item inserted in the i-th knapsack$\}$

m constraints

w_j = weight of the j-th item

W_i = capacity of the i-th knapsack

every item is being assign to only 1 knapsack

Surrogate relaxation of (3):

$$
\begin{aligned}
z_{S(\underline{\lambda})} = \max \quad & \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} \\
\text{s.t.} \quad & \boxed{\textstyle\sum_{i=1}^{m} \lambda_i \sum_{j=1}^{n} w_j x_{ij} \leq \sum_{i=1}^{m} \lambda_i W_i} && (6) \\
& \textstyle\sum_{i=1}^{m} x_{ij} \leq 1 \quad \forall j \in \{1, 2, \ldots, n\} && (7) \\
& x_{ij} \in \{0, 1\} \quad \forall i, \forall j && (8)
\end{aligned}
$$

S = surrogate

λ = different relaxation for every choice of λ

1 constraint

knapsack problem with m copies of each item j (i-th copy has weight $\lambda_i w_j$ and profit p_j) and at most one copy of each item can be selected.

$$
\begin{aligned}
z_{S(\underline{\lambda})} = \max \quad & \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} \\
\text{s.t.} \quad & \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} (\lambda_i w_j) x_{ij} \leq \sum_{i=1}^{m} \lambda_i W_i && (9) \\
& \textstyle\sum_{i=1}^{m} x_{ij} \leq 1 \quad \forall j \in \{1, 2, \ldots, n\} && (10) \\
& x_{ij} \in \{0, 1\} \quad \forall i, \forall j && (11)
\end{aligned}
$$

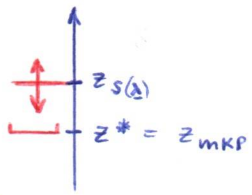
Since for each item j a copy i with smallest λ_i is more convenient, it is a standard binary knapsack problem with capacity $\sum_{i=1}^{m} \lambda_i W_i$.

Clearly $\boxed{z_{mKP} \leq z_{S(\underline{\lambda})}}$. (since it's a max)

To find the tightest upper bound, solve the *surrogate dual*:

$$\min_{\underline{\lambda} \geq \underline{0}} z_{S(\underline{\lambda})}.$$

(we're introducing another problem to solve, but then for a given $\underline{\lambda}$ the initial problem is just a binary knapsack problem)



4) Lagrangian relaxation

Often LP relaxation and relaxation by elimination provide weak bounds (e.g., TSP, UFL).

Idea: Eliminate the "difficult" constraints and add, for each one of them, a term in the objective function with a multiplier u which penalizes its violation (for max: terms ≥ 0 for all feasible solutions).

Example: *Multiple binary knapsack problem*

$$
\begin{aligned}
z_{mKP} = \max \quad & \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} \\
\text{s.t.} \quad & \textstyle\sum_{j=1}^{n} w_j x_{ij} \leq W_i \quad \forall i \in \{1, 2, \ldots, m\} \\
& \boxed{\textstyle\sum_{i=1}^{m} x_{ij} \leq 1 \quad \forall j \in \{1, 2, \ldots, n\}} && (12) \\
& x_{ij} \in \{0, 1\} \quad \forall i, \forall j
\end{aligned}
$$

$1 - \sum_{i=1}^{m} x_{ij} \geq 0 \quad \forall j \in \{1, .., n\}$

we want to delete it but not like in the relaxation by elimination, we want to add a penalty term in the obj. function that penalizes the violation of this constraint

$c(\underline{x})$

since it's max: $\mathrel{\text{I}\!\wedge}$

$\tilde{c}(\underline{x})$

Lagrangian relaxation of (12):

$$
\begin{aligned}
z_{L(\underline{u})} = \max \quad & \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} + \boxed{\textstyle\sum_{j=1}^{n} u_j (1 - \sum_{i=1}^{m} x_{ij})} && (13) \\
\text{s.t.} \quad & \textstyle\sum_{j=1}^{n} w_j x_{ij} \leq W_i \quad \forall i \in \{1, 2, \ldots, m\} && (14) \\
& x_{ij} \in \{0, 1\} \quad \forall i, \forall j && (15)
\end{aligned}
$$

with $u_j \geq 0$ for all j, so that $z_{mKP} \leq z_{L(\underline{u})}$ for every $\underline{u} \geq \underline{0}$.

again, based on $\underline{u}$ we have different relaxations

when the constraints are violated this term is < 0 $\Longrightarrow$ if we want to obtain the maximum is better to respect this constraint

new profit: $\tilde{p}_j$

Since

$$\sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} + \sum_{j=1}^{n} u_j (1 - \sum_{i=1}^{m} x_{ij}) = \sum_{i=1}^{m} \sum_{j=1}^{n} (p_j - u_j) x_{ij} + \sum_{j=1}^{n} u_j,$$

max $\longrightarrow$ +

min $\longrightarrow$ −

in the Lagrangian subproblem (13)-(13) each item j has profit $\tilde{p}_j = p_j - u_j$, weight w_j and can be inserted in several knapsacks.

Equivalent to m independent binary knapsack problems. For $i = 1, 2, \ldots, m$, we have

$$
\begin{aligned}
z_i = \max \quad & \textstyle\sum_{j=1}^{n} \tilde{p}_j x_j \\
\text{s.t.} \quad & \textstyle\sum_{j=1}^{n} w_j x_j \leq W_i && (16) \\
& x_j \in \{0, 1\} \quad \forall j \in \{1, 2, \ldots, n\} && (17)
\end{aligned}
$$

and $\boxed{z_{L(\underline{u})} = \sum_{i=1}^{m} z_i + \sum_{j=1}^{n} u_j.}$

Since we eliminated the linking constraints (the linking between knapsacks) by lagrangianizing the problem, it's like we have m $\parallel$ knapsack problems

To find the tightest Lagrangian bound, solve the *Lagrangian dual*:

$$\min_{\underline{u} \geq \underline{0}} z_{L(\underline{u})}.$$

Lagrangian relaxation will be discussed in detail later.

Simple dominance relations among relaxations

Compare in terms of dual bound the quality of three relaxations when the same constraints are relaxed with optimal multipliers.

Proposition: The surrogate and Lagrangian relaxations dominate the relaxation by elimination.

Obvious, since the latter is equivalent to take $\underline{\lambda} = \underline{0}$ and $\underline{u} = \underline{0}$ in the other relaxations.

Proposition: The surrogate relaxation dominates the Lagrangian relaxation.

The Lagrangian relaxation can be viewed as the Lagrangian relaxation of the surrogate relaxation obtained by relaxing the aggregated (surrogate) constraint with $u = 1$.

In practice Lagrangian relaxation is widely used because

- the Lagrangian subproblem is easier to solve than the surrogate one.
- $\exists$ efficient methods to determine "good" Lagrangian multipliers, unlike for surrogate ones.

In the examples that we saw, in the surrogate relaxation we still keep the constraints (but in a different form). In the Lagrangian relaxation we completely got rid of the linking constraint.
(so of course it's easier)

5) Combinatorial relaxations

i) **Asymmetric TSP**

Given directed $G = (V, A)$ with $V = \{1, 2, \ldots, n\}$ and a cost $c_{ij} \in \mathbb{R}$ for each $(i,j) \in A$, determine a Hamiltonian circuit of minimum total cost.

The set of all the assignments

$$\tilde{X} = \{\underline{x} \in \{0,1\}^{|A|} : \sum_{i:(i,j)\in A} x_{ij} = 1 \ \forall j, \ \sum_{j:(i,j)\in A} x_{ij} = 1 \ \forall i\},$$

is a relaxation of the set of all the Hamiltonian circuits.

ii) **Knapsack problem**

The set

$$\tilde{X} = \{\underline{x} \in \mathbb{Z}^n_+ : \sum_{j=1}^{n} \lfloor w_j \rfloor x_j \leq \lfloor W \rfloor\}$$

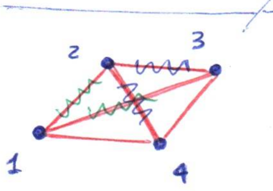
is a relaxation of $X = \{\underline{x} \in \mathbb{Z}^n_+ : \sum_{j=1}^{n} w_j x_j \leq W\}$.

iii) **Symmetric TSP**

Definition: Given undirected $G = (V, E)$ with $V = \{1, \ldots, n\}$, a **1-tree** of G is a subgraph which contains two edges incident to node 1, together with the edges of a spanning tree on nodes $\{2, \ldots, n\}$.

spanning tree on $\{2,..,n\}$: subset of edges which doesn't create a cycle (and touches $\{2,..,n\}$)

Example:



m = spanning tree on $\{2,3,4\}$
+
m = two edges incident in 1

Observation: The set of all 1-trees is a relaxation of the set of all Hamiltonian cycles.

In the example we created a 1-tree which is not Hamiltonian, but $\forall$ Hamiltonian cycle is 1-tree:
$\{$Hamiltonian of $G\} \subset \{$1-trees of $G\}$

Exact algorithm for minimum cost 1-tree:

- determine a minimum cost spanning tree on nodes $\{2, \ldots, n\}$ by using an exact greedy algorithm (Kruskal or Prim),
- select two edges incident in node 1 with smallest cost.

! Kruskal algorithm for minimum spanning tree:

Consider the edges in the order of non-decreasing cost.

At each step, discard the edge if it creates a cycle with previously selected edges.

Stop when selected edges "cover" all the nodes.

3.4.2 Heuristics for primal bounds

1) **Greedy methods**

Construct a feasible solution piece by piece from scratch.

At each step, select an available "piece" that yields the best "local profit", without reconsidering previous choices.

profit w.r.t. the specific choice (choice made at the specific step)

Example 1: Binary Knapsack Problem (BKP)

$$\begin{aligned} z_{ILP} = \max \quad & 16x_1 + 22x_2 + 12x_3 + 8x_4 \\ \text{s.t.} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_1, \ldots, x_4 \in \{0,1\} \end{aligned}$$

Order items by non-increasing profit-volume ratios (p_j/w_j) :

	x_1	x_2	x_3	x_4
p_j	16	22	12	8
w_j	5	7	4	3
p_j/w_j	3.2	3.14	3	2.7

Consider them in that order, select ($x_j = 1$) those that do not violate the residual capacity, skip the others ($x_j = 0$).

$x_1 = 1$ $(w_1 = 5 < 14)$
$x_2 = 1$ $(w_1 + w_2 = 12 < 14)$
$x_3 = 0$ $(w_1 + w_2 + w_3 = 16 > 14)$
$x_4 = 0$ $(w_1 + w_2 + w_4 = 15 > 14)$
$\Rightarrow p = 38$

Feasible solution of greedy procedure: $\overline{\underline{x}} = (1, 1, 0, 0)$ with $\overline{z}_{greedy} = 38$.

Optimal integer solution: $\underline{x}^* = (0, 1, 1, 1)$ with $z_{ILP} = 42$.

Clearly $\overline{z}_{greedy} \leq z_{ILP}$.

How bad can a greedy solution be w.r.t. an optimal one?

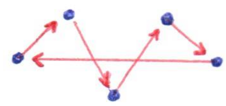
Worst case example:

Example 2: Symmetric TSP with complete graph

Nearest neighbor heuristic: Start from any node, at each step insert the closest node not yet visited, come back to the starting node.

Complexity: $O(n^2)$, where n is the number of nodes.

For visualization of TSP greedy heuristics, see http://bjornson.inhb.de/?p=26



cost = distance in the plane

Empirical performance: on TSPLIB(rary) instances it yields tours whose average cost is about 1.26 times that of optimal tours.

Worst-case performance: there are instances for which it finds tours that are arbitrarily worse than the optimal tours.

It's cheap computationally but it can be bad solution.

Once we use a greedy heuristics to find a reasonable solution, we can improve this solution locally.

2) **Local search methods**

Consider a generic

$$\min_{\underline{x} \in X} c(\underline{x})$$

and try to iteratively improve a current feasible solution.

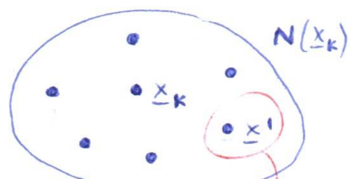
Define, for any feasible solution $\underline{x}$, a neighborhood $N(\underline{x})$, i.e., a subset of "nearby" feasible solutions.

Start from an initial feasible solution $\underline{x}_0$. for instance a solution provided by the greedy algorithm

At iteration k:

- Find a best solution $\underline{x}'$ in $N(\underline{x}_k)$ of the current solution $\underline{x}_k$.
- If $c(\underline{x}') < c(\underline{x}_k)$ then $\underline{x}_{k+1} := \underline{x}'$ and perform iteration $k+1$, otherwise return $\underline{x}_k$ which is a local minimum w.r.t. $N(\underline{x})$.

(the efficiency) It depends all on how the neighborhoods are made: we're always looking for a trade-off between neighborhoods which are rich enough and neighborhoods which are not too expensive to explore from a computational point of view.



we look for the best solution in the neighborhood $\rightarrow \underline{x}_{k+1}$

Example: 2-opt heuristic for Symmetric TSP

Given $G = (V, E)$ and a current tour $H \subseteq E$.

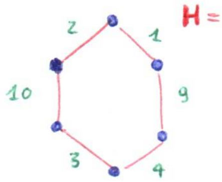
For any pair of nonadjacent e_1 and e_2, try to delete them and replace them with the two unique alternative edges that recombine the two paths into a new tour H'.

The 2-opt neighborhood $N(H)$ contains all tours obtainable with such a "2-interchange".

we try all the possible combinations

If $c(H') < c(H)$ then H' becomes the current solution, otherwise H is a local minimum w.r.t the 2-opt neighborhood.

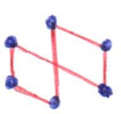
Illustration:



H = given tour

costs

we delete two non-consecutive edges and we re-combine them in the unique possible way:



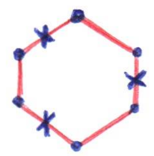
H'

Complexity: $O(n^2)$ with $n = |V|$.

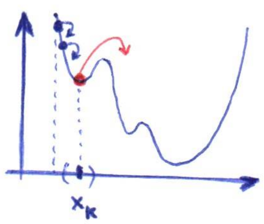
For animation see: https://www.youtube.com/watch?v=UGGPZnAUjPU on average we are about 6% above the optimum

Also k-opt for $k = 3$, with complexity $O(n^3)$.

Empirical performance: on TSPLIB instances 2-opt (3-opt) provides tours about 1.06 (1.04) times the optimum.



Note: for $k > 2$ $\nexists$ a unique way to recombine the path (so computationally it becomes more expensive to explore all the neighborhoods)



we want to escape the **LOCAL MINIMA**

Metaheuristics (for minimization problems)

To try to escape from local optima and improve upon local search heuristics.
E.g., tabu search, simulated annealing or genetic algorithms.

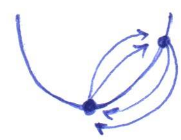
Tabu Search:

we use a list to make "tabu" the moves that have been done recently

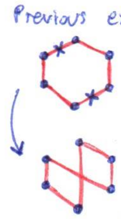
Allow moves to the best neighbor even if it has a worse objective function value. Use a tabu list to avoid cycling (reconsidering previously examined feasible solutions).

- Start from feasible $\underline{x}_0$.
- At iteration k, $\underline{x}_{k+1} := \underline{x}'$ where $\underline{x}'$ is the best solution in $N(\underline{x}_k)$, even if $c(\underline{x}') \geq c(\underline{x}_k)$.
- Prevent to undo recent moves (changes to the feasible solution) for a certain number of iterations. Once a move is peformed it is made tabu for the l successive iterations.

We don't want:



Previous example:



now for a certain number n of iterations we can't remove these two new lines

The best solution found is stored and returned after a prescribed maximum number of iterations. The best solution may not be the last, so we have to store the best solution found along the way and return to it at the end of the process.

Example: Uncapacitated Facility Location (UFL) problem

m clients ($i \in M$) and n depots ($j \in N$)

For any $S \subseteq N$, feasible solution where the depots with indices in S are open and all clients are served by the "cheapest" open depot.

$$\boxed{c(S) = \sum_{i=1}^{m} \min_{j \in S} c_{ij} + \sum_{j \in S} f_j}$$

cost for satisfy all the demand of client i with the depot j (we consider it proportional to the distance between client and depot)

Simple neighborhood:

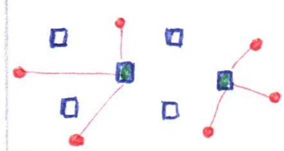
$$N(S) = \{T \subseteq N \; : \; T = S \cup \{j\}, j \notin S \text{ or } T = S \setminus \{j\}, j \in S\}$$

distance 1 neighborhood

We consider as a neighborhood all the subsets that differ from S by just 1 open depot:

$$\underset{\text{n open depots}}{S} \longrightarrow N(S) = \{ \underset{\text{n+1 open depots}}{S'} , \underset{\text{n-1 open depots}}{S''} \}$$

for every $S \subseteq N$:



Among all the depots we open only the ■ ones and each client is served by the closest depot. The total cost $c(S)$ is the cost of opening the depots $(\sum_{j \in S} f_j)$ plus the cost of the service for every client $(\sum_{i=1}^{m} \min_{j \in S} c_{ij})$.

$m = 6$ clients, $n = 4$ depots

to serve all the demand of client 1 with depot 1 it'll cost 6

service cost matrix

$$(c_{ij}) = \underbrace{\begin{pmatrix} 6 & 2 & 3 & 4 \\ 1 & 9 & 4 & 11 \\ 15 & 2 & 6 & 3 \\ 9 & 11 & 4 & 8 \\ 7 & 23 & 2 & 9 \\ 4 & 3 & 1 & 5 \end{pmatrix}}_{\text{depots}} \text{ clients}$$

cost of opening of the depots

$\underline{f} = (21, 16, 11, 24)^t$

Initial solution: $S_0 = \{1, 2\}$ of cost $c(S_0) = 61$. $= 21 + 16 + (2 + 1 + 2 + 9 + 7 + 3)$

Three iterations of Local search/Tabu Search... : $N(S_0) = \{ \{1\}, \{2\}, \{1,2,3\}, \{1,2,4\} \}$

$c(\{1\}) = 63$, $c(\{2\}) = 66$, $c(\{1,2,3\}) = \underline{60}$, $c(\{1,2,4\}) = 84$ $\Longrightarrow$ $S_1 = \{1, 2, 3\}$

$\longrightarrow N(S_1) = \dots$. If we keep on we find $S_3 = \{3\}$ with $c(S_3) = 31$ and it's a local min (that we're not able to improve) w.r.t. this neighborhood. Possible Tabu Search variant: we accept the best neighborhing solution even if it is not a better solution than the current solution. Forbidden: if the last move is to add $\{3\}$ then we cannot remove $\{3\}$ for l iterations.

3.5 Branch and Bound (Summary)

Consider a generic Discrete Optimization problem

$$(P) \qquad z = \max\{c(\underline{x}) \,:\, \underline{x} \in X\},$$

where X is the feasible region, i.e., the set of the feasible solutions.

Branch and Bound is a general semi-enumerative approach (Land and Doig 1960) to explore the feasible region X.

By exploiting bounds on the optimal objective function value

- it avoids explicitly exploring certain parts of the feasible region X,
- it is guaranteed to find an optimal solution.

Two main components:

- "Divide and conquer" strategy (branching)
- Implicit enumeration exploiting bounds on the optimal objective function value (bounding)

1) "Divide and conquer" strategy

Idea: Partition in a recursive way the feasible region X so as to reduce the solution of (P) to the solution of a sequence of smaller and easier subproblems.

Observation: Let $X = X_1 \cup \ldots \cup X_k$ be a *partition* of X in k subsets ($X_i \cap X_j = \emptyset$ for each pair of indices $i \neq j$) and

$$z^i = \max\{c(\underline{x}) \,:\, \underline{x} \in X_i\}$$

for $1 \leq i \leq k$. Then obviously $z = \max_{1 \leq i \leq k} z^i$.

Recursive partition of the feasible region $\equiv$ branching operation

The procedure can be represented by a **enumeration tree** whose root node is associated to X and the other nodes to the subsets X_i.

Examples:

- $X \subseteq \{0,1\}^3$ – binary branching
- X set of all the Hamiltonian circuits of a given digraph $G = (V, A)$ – multiway branching

2) Implicit enumeration

Explicit enumeration is too heavy computationally even for small instances, it is not enough to recursively subdivide the feasible region.

Idea: Exploit **upper** and **lower bounds** (primal and dual bounds) on z^i, with $1 \leq i \leq k$, in order to avoid to explicitly explore some parts of the feasible region X.

Observation: Let $X = X_1 \cup \ldots \cup X_k$ be a partition of X and

$$z^i = \max\{c(\underline{x}) \,:\, \underline{x} \in X_i\}$$

for $1 \leq i \leq k$.

Moreover, let l^i be a lower bound and u^i an upper bound on z^i, namely $l^i \leq z^i \leq u^i$.

Then $l = \max_{1 \leq i \leq k} l^i$ is a lower bound and $u = \max_{1 \leq i \leq k} u^i$ is an upper bound on z, that is $l \leq z \leq u$.

Pruning criteria

Cases in which the primal and dual bounds for the i-th subproblem can be exploited to avoid exploring (discard) the subregion X_i (to prune the corresponding node of the B&B tree):

- **Optimality criterion**: If $u_i = l_i$, it is not necessary to further explore the subregion X_i since we have found an optimal solution in X_i of value $z^i = u_i = l_i$.
- **Bounding criterion**: If the upper bound u_i is lower than
  - the objective function value LB of the best solution $\underline{x}_{LB}$ found so far

  or

  - any lower bound l_j for $j \neq i$,

  it is not necessary to explore the subregion X_i because it cannot contain any better feasible solution.
- **Feasibility criterion**: $X_i = \emptyset$

Four examples of subproblems (node) configurations, including one where the feasible regions of the subproblems must be further explored.

If a subproblem is not "solved", we proceed recursively and generate subproblems (branching step).

Main ingredients of Branch and Bound method (for max problem)

• *Upper bounds*: Efficient method to determine a good quality dual bound u on z.

• *Lower bounds*: Efficient heuristic to look for a feasible solution $\underline{\tilde{x}}$ with a value $c(\underline{\tilde{x}})$, which provides a good lower bound $c(\underline{\tilde{x}})$ on z.

• *Branching rule*: Procedure to (recursively) partition the feasible region X into smaller subregions.

To be stored and updated:

- a list $\mathcal{L}$ of active subproblems with lower and upper bounds on z^i: $l^i \leq z^i \leq u^i$,

- a global upper bound UB on z,

- a global lower bound LB on z provided by the best feasible solution $\underline{x}_{LB}$ found so far.

General method, we "just" need to specify:

1. how to choose the next subproblem (active node) to be "processed"
2. how to generate the subproblems of a given subproblem (the "children" nodes)
3. how to efficiently compute the primal and dual bounds

The performance of a Branch-and-Bound algorithm strongly depends on the efficiency of the branching rule and the quality of primal and dual bounds.

N.B.: A Branch-and-Bound approach is applicable to MILP problems as well as to Nonlinear Optimization problems.

3.5.1 Branch and Bound for ILP problems

Consider an ILP problem:

$$z_{ILP} = \max\{\underline{c}^t\underline{x} \,:\, A\underline{x} = \underline{b}, \underline{x} \geq \underline{0} \text{ integer}\} \tag{1}$$

and suppose we look for an optimal solution $\underline{x}^*_{ILP}$.

Solve the **linear relaxation** of (1) and let $\underline{x}^*_{LP}$ be an optimal solution of value z_{LP}.
Obviously $z_{ILP} = \underline{c}^t\underline{x}^*_{ILP} \leq z_{LP} = \underline{c}^t\underline{x}^*_{LP}$.

If $\underline{x}^*_{LP}$ is integer, then it is also optimal for (1). Otherwise $\underline{x}^*_{LP}$ has at least one fractional component.

Branching

If $\underline{x}^*_{LP}$ is not integer, choose a fractional component x^*_h and generate the two suproblems:

$$z^1_{ILP} = \max\{\underline{c}^t\underline{x} \,:\, A\underline{x} = \underline{b}, x_h \leq \lfloor x^*_h \rfloor, \underline{x} \geq \underline{0} \text{ integer}\}$$
$$z^2_{ILP} = \max\{\underline{c}^t\underline{x} : A\underline{x} = \underline{b}, x_h \geq \lfloor x^*_h \rfloor + 1, \underline{x} \geq \underline{0} \text{ integer}\}$$

with the corresponding subregions X_1 and X_2 of X, which are exhaustive and mutually exclusive.

Clearly $z_{ILP} = \max\{z^1_{ILP}, z^2_{ILP}\}$.

Recursive process: solve the linear relaxation of each subproblem of the ILP and, if necessary, carry out a branching step.

Bounding

Consider the i-th subproblem with feasible subregion X_i.

Solve its **linear relaxation**, let $\underline{x}^*_{LP}$ be the optimal solution and z^i_{LP} its value.

Clearly, if all the coefficients c_i are integer, every feasible solution of the ILP in X_i has value $\leq \lfloor z^i_{LP} \rfloor$.

In Branch and Bound, branching and bounding operations are alternated, while storing and updating the best feasible solution found.

We need to decide:

1. Criterion to select the next subproblem (node) to explore.
2. How to generate the "children" nodes for the node under consideration (choice of the branching variable).
3. Heuristic to determine the lower bounds on the optimal objective function value.

12

1. Choice of the subproblem (node) to be processed

   - *Depth first search strategy* ("deepest" node first): easy to implement but costly if wrong choice.

   - *Best bound first strategy* (most "promising" node first): tend to generate less nodes but the subproblems are less constrained (we rarely update the best solution found so far).

2. Choice of the fractional variable for branching

   - Branching first on a fractional variable whose fractional part is closest to 0.5 (in an attempt to generate two subproblems that are "equally" constrained) is often not the best choice.

   - *Strong branching* ("estimate" the bound improvement if branching on several candidate fractional variables, and branch w.r.t. the best one) is costly but effective for some hard instances.

Exponential example for Branch and Bound:

Let n be an odd positive integer and consider the ILP problem:

$$
\begin{array}{lll}
\max & -x_n & \\
\text{s.t.} & x_0 + 2\sum_{j=1}^{n} x_j = n & \\
 & 0 \leq x_j \leq 1 & \forall j \in \{0,1,2,\ldots,n\} \\
 & x_j \in \mathbb{Z}^+ & \forall j \in \{0,1,2,\ldots,n\}.
\end{array}
$$

It can be verified that, when Branch and Bound is applied to this ILP instance, at least $2^{\frac{n-1}{2}}$ ILP subproblems are inserted in the list $\mathcal{L}$.

Example 1:

Find an optimal solution of the ILP problem

$$
\begin{array}{ll}
\max & 4x_1 - x_2 \\
\text{s.t.} & 4x_1 + 2x_2 \geq 19 \\
 & 10x_1 - 4x_2 \leq 25 \\
 & x_2 \leq \frac{9}{2} \\
 & x_1, x_2 \in \mathbb{Z}^+
\end{array}
$$

with the Branch and Bound method by solving graphically the linear relaxation of the subproblems. Branch first with respect to x_1.

Example 2:

Solve the binary knapsack problem

$$
\begin{array}{ll}
\max & 10x_1 + 12x_2 + 5x_3 + 7x_4 + 9x_5 \\
\text{s.t.} & 5x_1 + 8x_2 + 6x_3 + 2x_4 + 7x_5 \leq 14 \\
 & x_1, \ldots, x_5 \in \{0,1\}
\end{array}
$$

with the Branch and Bound method. Use a simple greedy heuristic to determine the optimal solutions of the linear relaxations.

3.6 Cutting plane methods

Generic ILP

$$\min\{\underline{c}^t\underline{x} : \underline{x} \in X = \{\underline{x} \in \mathbb{Z}^n_+ : A\underline{x} \leq \underline{b}\}\}$$

with rational A and $\underline{b}$.

An ideal formulation exists (Meyer's theorem).

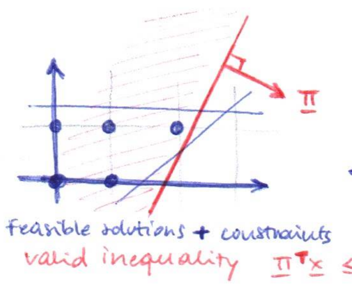
However, for *NP*-hard problems, it is unknown and/or it contains a huge number of constraints.

Idea: Improve initial formulation (better approximation of $conv(X)$) by adding valid inequalities.

Definition: $\underline{\pi}^t\underline{x} \leq \pi_0$ is a **valid inequality** for $X \subseteq \mathbb{R}^n$ if $\underline{\pi}^t\underline{x} \leq \pi_0$ for each $\underline{x} \in X$.

= inequality that is satisfied by all the feasible solutions

Illustration:



Use of valid inequalities:

- add them a priori
- generate them as needed – via a cutting plane method

1) Addition a priori

Advantage: Branch and Bound method with stronger formulation is more efficient (tighter dual bounds).

Example: Given weak UFL formulation with $\sum_{i \in M} x_{ij} \leq m y_j$ for all $j \in N$, add stronger $x_{ij} \leq y_j$, $\forall i \in M, j \in N$, and delete former ones.

Disadvantage: If huge number of valid inequalities, the LP relaxation is extremely heavy and/or standard B&B is impossibile.

2) Cutting plane methods = add the valid inequalities only if they are needed

Generic ILP:

$$\min\{\underline{c}^t \underline{x} \; : \; \underline{x} \in X = P \cap \mathbb{Z}^n\}$$

where $P = \{\underline{x} \in \mathbb{R}^n_+ \; : \; A\underline{x} \leq \underline{b}\}$ is the feasible region of LP relaxation.

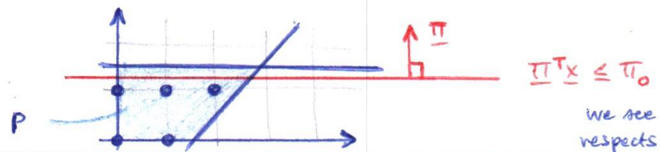
Consider a family $\mathcal{F}$ of inequalities $\underline{\pi}^t \underline{x} \leq \pi_0$ valid for X, $(\underline{\pi}, \pi_0) \in \mathcal{F}$.

Often $|\mathcal{F}|$ is too large to add them a priori (e.g. cut-set/subtour-elimination for ATSP)

Definition: Given $\underline{x}' \in P$ with $\underline{x}' \notin X$, a **cutting plane** is an $\underline{\pi}^t \underline{x} \leq \pi_0$ s.t.

- $\underline{\pi}^t \underline{x} \leq \pi_0$ is valid for $X = P \cap \mathbb{Z}^n$
- $\underline{\pi}^t \underline{x}' > \pi_0$

Illustration:



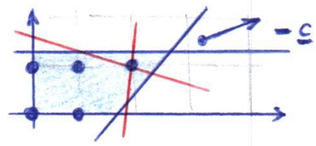
we see that every $\underline{x} \in X$ respects the inequality, moreover $\exists \underline{x} \in P \setminus X$ s.t. doesn't respect the inequality → the inequality is a CUTTING PLANE

Idea of cutting plane methods:

No need for $conv(X)$, it suffices to add a subset of cutting planes providing a good description around $\underline{x}^*_{ILP}$, i.e., bringing out $\underline{x}^*_{ILP}$ as optimal extreme point of LP relaxation polyhedron for given $\underline{c}$.

Illustration:

min $\underline{c}^T \underline{x}$:



It's enough to add: — so that x^*_{ILP} is one vertex of the resulting P.

Subproblem that we have to solve at each iteration of the cutting plane method

Separation problem:

Given any $\underline{x}' \notin X$ and a family of valid inequalities $\mathcal{F}$ for X, find one which separates $\underline{x}'$ from $conv(X)$ or establish that no such cutting plane exists.

Illustration:

Example: Gomory fractional cutting planes for generic ILPs – see below.

Cutting plane algorithm

Initialization $P' := P = \{\underline{x} \in \mathbb{R}^n_+ \; : \; A\underline{x} \leq \underline{b}\}$

1. Solve current LP relaxation $\min\{\underline{c}^t \underline{x} \; : \; \underline{x} \in P'\}$ and let $\underline{x}^*_{LP}$ be an optimal solution.

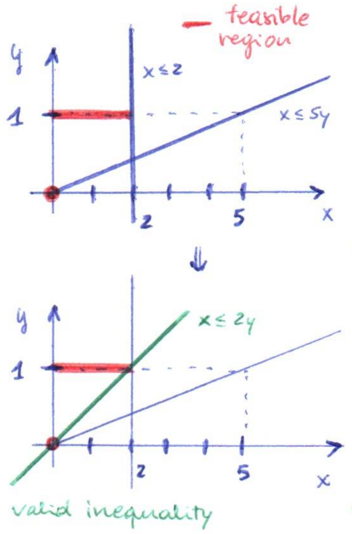
2. IF $\underline{x}^*_{LP} \in \mathbb{Z}^n$ THEN terminate because $\underline{x}^*_{LP}$ is also optimal for ILP

   ELSE Solve the separation problem for $\underline{x}^*_{LP}$, $\mathcal{F}$ and $X = P' \cap \mathbb{Z}^n$

   IF $\underline{\pi}^t \underline{x} \leq \pi_0$ is found THEN $P' := P' \cap \{\underline{x} \in \mathbb{R}^n \; : \; \underline{\pi}^t \underline{x} \leq \pi_0\}$ and go back to (1).

   ELSE stop

N.B.: If $\underline{x}^*_{LP}$ is not integer, P' is anyway stronger than P.



3.6.1 Simple valid inequalities

1) **Binary set**

$$X = \{x \in \{0,1\}^5 : 3x_1 - 4x_2 + 2x_3 - 3x_4 + x_5 \leq -2\}$$

Since $x_2 = x_4 = 0$ infeasible, $x_2 + x_4 \geq 1$ is valid.

Since $x_1 = 1$ and $x_2 = 0$ infeasible, $x_1 \leq x_2$ is valid.

2) **Mixed 0-1 set**

$$X = \{(x, y) : x \leq cy, 0 \leq x \leq b, \ y \in \{0,1\}\} \text{ with } c > b$$

Illustration: $c = 5$ and $b = 2$

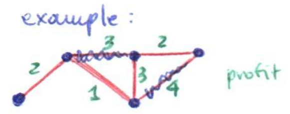
Notice that $x \leq 2y$ (together with the other two inequalities) describes $conv(X)$

Since $X = \{(0,0), (x,1) \text{ with } 0 \leq x \leq b\}$, $x \leq by$ is valid and, with $x \geq 0$ and $y \leq 1$, describe $conv(X)$.

3) **Combinatorial set**

Maximum Matching problem: Given undirected $G = (V, E)$ with a profit $p_e \in \mathbb{R}$ for each $e = \{i,j\} \in E$, determine a **matching**, i.e., a subset of edges without common nodes, of maximum total profit.

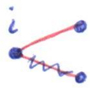
example:



is a matching with total profit equal to 7

Illustration:

$x_e = \begin{cases} 1 & \text{if } e \in E \text{ is selected} \\ 0 & \end{cases}$, for $\forall i$:



we have to pick at most 1 edge ⟹ we make some constraints using $\delta(i)$ = edges that are incidents in i

max $\sum_{e \in E} p_e x_e$

s.t.: $\sum_{e \in \delta(i)} x_e \leq 1 \quad \forall i$

(for every node we take at most one edge)

$\boxed{X = \{\underline{x} \in \{0,1\}^{|E|} : \sum_{e \in \delta(i)} x_e \leq 1, \ i \in V\}}$ all incidence vectors of matchings in G

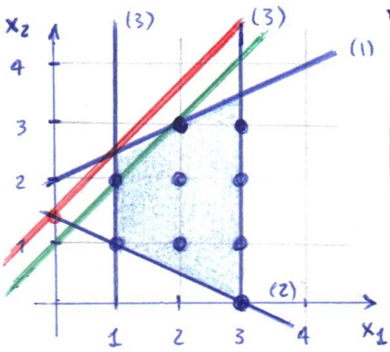
For any $S \subseteq V$ with $|S|$ odd and $|S| \geq 3$,

$$\boxed{\sum_{e \in E(S)} x_e \leq \frac{|S|-1}{2}}$$

is valid for X.

for example, $|S| = 3$: we can pick at most 1 edge:

$\sum_{e \in E(S)} x_e \leq \frac{3-1}{2} = 1$



(1) $-x_1 + 2x_2 \leq 4$

(2) $-x_1 - 2x_2 \leq 3$

(3) $1 \leq x_1 \leq 3$

$-x_1 + x_2 \leq 3/2$

$-x_1 + x_2 \leq \lfloor 3/2 \rfloor = 1$

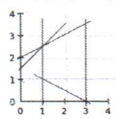
3.6.2 Chvátal cutting planes for ILP

Generate valid inequalities via linear combination and rounding.

Integer rounding principle: Given $X = \{x \in \mathbb{Z} \ : \ x \leq b\}$ where $b \in \mathbb{Q} \setminus \mathbb{Z}$, then $x \leq \lfloor b \rfloor$ is valid for X.

Example 1:

$$X = \{(x_1, x_2)^t \in \mathbb{Z}^2_+ \ : \ -x_1 + 2x_2 \leq 4, \quad -x_1 - 2x_2 \leq -3, \quad 1 \leq x_1 \leq 3\}$$



By adding $-x_1 \leq -1$ and $-x_1 + 2x_2 \leq 4$ multiplied by 1/2, we have: $-x_1 + x_2 \leq 3/2$.
Then

$$-x_1 + x_2 \leq \lfloor 3/2 \rfloor = 1$$

is valid for X and needed to describe $conv(X)$.

Chvátal-Gomory procedure for generating valid inequalities:

Consider $X = P \cap \mathbb{Z}^n$ with $P = \{\underline{x} \in \mathbb{R}^n_+ \ : \ A\underline{x} \leq \underline{b}\}$

$X = \{\underline{x} \in \mathbb{Z}^n_+ \ : \ \sum_{j=1}^n A_j x_j \leq \underline{b}\}$ where A_j is j-th column of A

1) Choose $\underline{u} \in R^m_+$ and consider $\sum_{j=1}^n (\underline{u}^t A_j) x_j \leq \underline{u}^t \underline{b}$

2) Since $\lfloor \underline{u}^t A_j \rfloor \leq \underline{u}^t A_j$ and $x_j \geq 0$,

$$\sum_{j=1}^n \lfloor \underline{u}^t A_j \rfloor x_j \leq \underline{u}^t \underline{b}$$

is valid for P and for $conv(X)$ and X.

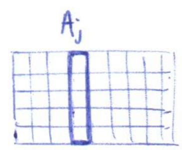
3) Since $x_j \in \mathbb{Z}^n_+$, the stronger

$$\sum_{j=1}^n \lfloor \underline{u}^t A_j \rfloor x_j \leq \lfloor \underline{u}^t \underline{b} \rfloor$$

is valid for $conv(X)$ and X (but not necessarily for P).

$\underline{u} \in \mathbb{R}^m_+$ and not $\underline{u} \in \mathbb{R}^m$ because if $u_j < 0$ then we may reverse some inequalities

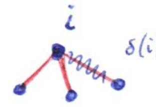
$A =$



we take (possibly) all combinations of the columns and then we round down all the coefficients

Example 2: **Matching polytope**

for every node i we should pick at most 1 edge:



Given an undirected $G = (V, E)$ and $X = \{\underline{x} \in \{0,1\}^{|E|} : \sum_{e \in \delta(i)} x_e \leq 1, \ i \in V\}$.

Proposition 1:
For any $S \subseteq V$ with $|S|$ odd and $|S| \geq 3$,

$$\sum_{e \in E(S)} x_e \leq \frac{|S|-1}{2}$$

is a Chvátal-Gomory inequality w.r.t. the linear description

$$\sum_{e \in \delta(i)} x_e \leq 1 \quad \forall i \in V \tag{1}$$

$$x_e \geq 0 \quad \forall e \in E. \tag{2}$$

proof.

Consider any $S \subseteq V$ with $|S| \geq 3$.

Linear combination of (1) with $u_i = 0.5$ for $i \in S$ and $u_i = 0$ for $i \notin S$, yields

$$\sum_{e \in E(S)} x_e + \frac{1}{2} \sum_{e \in \delta(S)} x_e \leq \frac{|S|}{2}$$

which is valid for X.

Since $x_e \geq 0$ and $x_e \in \mathbb{Z}$ for each $e \in E$, also

$$\sum_{e \in E(S)} x_e \leq \lfloor \frac{|S|}{2} \rfloor \tag{3}$$

is valid for X.

If $|S|$ is even, (3) is implied by the constraints (1) for $i \in S$ and by constraints (2).

If $|S|$ is odd, $\lfloor \frac{|S|}{2} \rfloor = \frac{|S|-1}{2}$ and (3) is not implied.

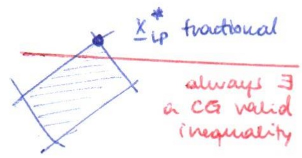
Theorem 1 (Chvátal): Any valid inequality for any X can be obtained by applying Chvátal-Gomory procedure a finite number of times.

Proof for case $X \subseteq \{0,1\}^n$ cf. L. Wolsey, Integer Programming, Wiley, p. 120-121

What does it mean?

Given any fractional extreme point $\underline{x}^*_{LP}$ of P, $\exists \ \underline{u} \geq \underline{0}$ such that the CG inequality $\lfloor \underline{u}^t A \rfloor \underline{x} \leq \lfloor \underline{u}^t \underline{b} \rfloor$ is valid for X and violated by $\underline{x}^*_{LP}$.

If we have a fractional vertex then there is a Chvatal-Gomory cutting plane (and so inequality) s.t. is cutting this fractional vertex:



Definition:
Denote by $A^1 \underline{x} \leq \underline{b}^1$ all inequalities obtainable by varying $\underline{u}$ in $\mathbb{R}^m_+$.
$P_1 = \{\underline{x} \in \mathbb{R}^n_+ \ : \ A\underline{x} \leq \underline{b}, A^1 \underline{x} \leq \underline{b}^1\}$ is the **first Chvátal closure** of P.

polyedron of the initial formulation on which we add all the possible CG valid inequalities that we can derive from $A\underline{x} \leq \underline{b}$. Of course we get a stronger formulation.

Note: $A^1 \underline{x} \leq \underline{b}^1$ is obtained by $\lfloor \underline{u}^T A \rfloor \underline{x} \leq \lfloor \underline{u}^T \underline{b} \rfloor$ for every possible choice of $\underline{u}$.

Obviously $P_1 \subseteq P$, and $P_1 = P$ if and only if P has no fractional vertices, that is $P = conv(X)$.

If $P_1 \neq conv(X)$, we can iterate to obtain Chvátal closures P_k of (higher) rank k, with $k \geq 2$.

Definition: The smallest integer k such that $P_k = conv(X)$ is the **Chvátal rank** of $conv(X)$ with respect to the formulation P.

Clearly $P_k = conv(X) \subset \ldots \subset P_2 \subset P_1 \subset P$.

The **CHVATAL theorem** says that always $\exists k < \infty$ such that $P_k = conv(\underline{X})$.

The problem of this procedure is that in practice we cannot generate P_1 with all the possible $\underline{u} \in \mathbb{R}^m_+$.

3.6.3 Gomory fractional/integer cutting planes – review

Just a special case of the Chvatal-Gomory cutting planes
(just a special choice of the multiplier vector $\underline{u} \in \mathbb{R}^m_+$)

Generic ILP

$$\min\{\, \underline{c}^t \underline{x} \,:\, A\underline{x} = \underline{b},\ \underline{x} \geq \underline{0},\ \underline{x} \in \mathbb{Z}^n \}$$

where $A \in \mathbb{Z}^{m \times n}$, $\underline{b} \in \mathbb{Z}^{m \times 1}$ and $n > m$.

Assumption: A is of full rank m

Idea: At each iteration, generate C-G cuts exploiting the optimal basic feasible solution $\underline{x}^*_{LP}$ of the current LP relaxation.

B is a basis of A associated with $\underline{x}^*_{LP}$.

$A\underline{x} = \underline{b}$, $\underline{x} \geq \underline{0}$ can be expressed in canonical form as

$$\underline{x}_B = B^{-1}\underline{b} - B^{-1}N\underline{x}_N \quad \text{with } \underline{x}_B \geq \underline{0} \text{ and } \underline{x}_N \geq \underline{0},$$

which emphasizes $\underline{x}^*_{LP} = (\underline{x}_B, \underline{x}_N) = (B^{-1}\underline{b}, \underline{0})$.

If $\underline{x}^*_{LP} = B^{-1}\underline{b}$ integer, $\underline{x}^*_{LP}$ is also optimal for ILP.

If $\underline{x}^*_{LP}$ is fractional, generate a C-G cut violated by $\underline{x}^*_{LP}$.

Let x^*_h be a fractional basic variable and row t of the canonical form

$$x_h + \sum_{j \in N} \bar{a}_{tj} x_j = \bar{b}_t \ (= x^*_h) \tag{4}$$

where N corresponds to non basic variables.

Observation: Equation (4) amounts to take $\boxed{\underline{u}^t = \underline{e}^t_t B^{-1}}$ where $\underline{e}_t$ is the t-th m-dimensional unit vector.

$\underline{e}_t = [0, 0, \ldots, 0, \underset{t}{1}, 0, \ldots]^T$

Applying Chvátal procedure to (4):

the **integer form** of the **Gomory cut** generated from row t of LP relaxation

$$x_h + \sum_{j \in N} \lfloor \bar{a}_{tj} \rfloor x_j \leq \lfloor \bar{b}_t \rfloor. \tag{5}$$

Clearly valid for X but violated by $\underline{x}^*_{LP}$ (since $x^*_j = 0$ for $j \in N$ and $x^*_h = \bar{b}_t$ is fractional).

Substracting (5) from (4):

the **fractional form** of the **Gomory cut** generated from row t of LP relaxation

$$\sum_{j \in N} (\bar{a}_{tj} - \lfloor \bar{a}_{tj} \rfloor)\, x_j \geq \bar{b}_t - \lfloor \bar{b}_t \rfloor. \tag{6}$$

If $\{a\} := a - \lfloor a \rfloor \geq 0$ denotes the *fractional part* of $a \in \mathbb{R}$, (6) is equivalent to

$$\sum_{j \in N} \{\bar{a}_{tj}\}\ x_j \geq \{\bar{b}_t\}.$$

Recall: $\{4/3\} = 1/3$ but $\{-4/3\} = -4/3 - (-2) = 2/3$

The fractional and integer forms of a Gomory cut are equivalent.

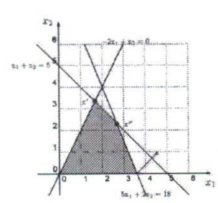
Observation: The difference (slack) between the lhs and rhs of (5) and hence of (6) is always integer when $\underline{x}$ is integer.

Minimal computational requirements.

Example:

$$\begin{array}{ll} \max & x_1 + x_2 \\ \text{s.t.} & x_1 + x_2 \leq 5 \\ & -2x_1 + x_2 \leq 0 \\ & 5x_1 + 2x_2 \leq 18 \\ & x_1, x_2 \in \mathbb{Z}^+ \end{array}$$

1. Graphical solution of LP relaxation:



Two optimal basic solutions: $\underline{x}' = (5/3, 10/3)$ and $\underline{x}'' = (8/3, 7/3)$ of value 5.

2. LP relaxation in standard form:

$$\begin{array}{ll} \max & x_1 + x_2 \\ \text{s.t.} & x_1 + x_2 + x_3 = 5 \\ & -2x_1 + x_2 + x_4 = 0 \\ & 5x_1 + 2x_2 + x_5 = 18 \\ & x_1, \dots, x_5 \geq 0 \end{array}$$

3. Canonical form w.r.t. the optimal basic solution $\underline{x}'' = (8/3, 7/3, 0, 3, 0)$:

$$\begin{aligned} x_1 - \tfrac{2}{3}x_3 + \tfrac{1}{3}x_5 &= \tfrac{8}{3} \\ x_2 + \tfrac{5}{3}x_3 - \tfrac{1}{3}x_5 &= \tfrac{7}{3} \\ -3x_3 + x_4 + x_5 &= 3 \end{aligned}$$

Gomory cut derived from x_1 row:

- integer form: $x_1 - x_3 \leq 2$
- fractional form: $\frac{1}{3}x_3 + \frac{1}{3}x_5 \geq \frac{2}{3}$

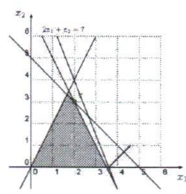
Gomory cut derived from x_2 row:

- integer form: $x_2 + x_3 - x_5 \leq 2$
- fractional form: $\frac{2}{3}x_3 + \frac{2}{3}x_5 \geq \frac{1}{3}$

4. Express Gomory cut associated with x_1 as a function of x_1 and x_2.

Substituting $x_3 = 5 - x_1 - x_2$ in $x_1 - x_3 \leq 2$, we obtain the cut: $2x_1 + x_2 \leq 7$.

5. Add this Gomory cut to LP relaxation and find an optimal solution.



Adding $2x_1 + x_2 \leq 7$ to the original formulation, we obtain an optimal solution of new LP relaxation $\underline{x}_{LP}^* = (2, 3)$ with $z_{LP}^* 5$.

Since $\underline{x}_{LP}^*$ is integer, it is also optimal for ILP.

Theorem 2 (Gomory): A lexicographic cutting plane method based on Gomory fractional/integer cuts terminates after a finite number of iterations.

Provided a careful choice of (i) the basis defining the optimal solution we intend to cut off and (ii) the row of the tableau used to generate the cut.

In practice: Huge number of iterations and such cuts tend to become weaker after a few iterations.

Strategy: Introduce several cuts at each iterations, e.g., all those with $\{\bar{b}_i\} > \varepsilon = 0.01$

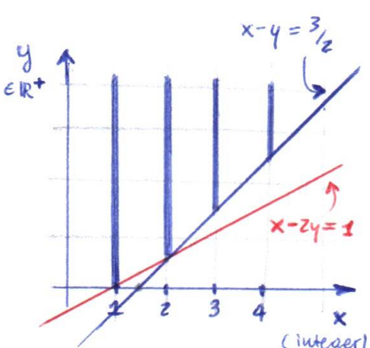
Recall: Gomory fractional/integer cuts are generated via simple integer rounding.

3.6.4 Mixed integer rounding inequalities

: how to generate generic cutting planes (in practice) (since CG fractionals cutting planes are not so efficient)

Consider $X = \{(x, y)^t \in \mathbb{Z} \times \mathbb{R}^+ \ : \ x - y \leq b\}$ where $b \in \mathbb{Q} \setminus \mathbb{Z}$.

Illustration for $b = 3/2$:



Is the blue line $(y = x - 3/2)$ corresponding to $x - y \leq 3/2$ the ideal formulation?

No, to get an ideal formulation we should add the **red** inequality.

Proposition 2: The **mixed-integer rounding (MIR)** inequality

$$\boxed{x - \frac{1}{1 - \{b\}} y \leq \lfloor b \rfloor} \qquad (7)$$

is valid for $conv(X)$.

$\{b\}$ = fractional part of b : $\{\frac{3}{2}\} = \frac{1}{2}$

Proof: Partition X into $X_1 = X \cap \{(x, y)^t \ : \ x \leq \lfloor b \rfloor\}$ and $X_2 = X \cap \{(x, y)^t \ : \ x \geq \lfloor b \rfloor + 1\}$.

By adding $(1 - \{b\})$ times $(x - \lfloor b \rfloor) \leq 0$ and $0 \leq y$, we see that

$$(x - \lfloor b \rfloor)(1 - \{b\}) \leq y$$

is valid for X_1.

By adding $-(x - \lfloor b \rfloor) \leq -1$ and $x - y \leq b$ with multipliers $\{b\}$ and 1, we see that

$$(x - \lfloor b \rfloor)(1 - \{b\}) \leq y$$

is valid for X_2.

Thus (7) is valid for $conv(X_1 \cup X_2) = conv(X)$. □

Observation: $conv(\{(x,y)^t \in \mathbb{Z} \times \mathbb{R}^+ \; : \; x - y \le b\})$ is defined by
$x - y \le b$, $y \ge 0$ and $x - \frac{1}{1-\{b\}} y \le \lfloor b \rfloor$.

Illustration:

Notation: For any $a \in \mathbb{R}$, $a^+ = \max\{a, 0\}$ and $a^- = \max\{-a, 0\}$.

Proposition 3: Let $A \in \mathbb{Q}^{m \times n}$, $\underline{b} \in \mathbb{Q}^m$ and $X = \{\underline{x} \in \mathbb{Z}_+^n \; : \; \sum_{j=1}^n A_j x_j \le \underline{b}\}$.

equivalent to: $A\underline{x} \le \underline{b}$

For every $\underline{u} \in \mathbb{Q}_+^m$, the **MIR aggregated** inequality

$$\boxed{\sum_{j=1}^{n} (\lfloor \underline{u}^t A_j \rfloor + \frac{(\{\underline{u}^t A_j\} - \{\underline{u}^t \underline{b}\})^+}{1 - \{\underline{u}^t \underline{b}\}}) x_j \le \lfloor \underline{u}^t \underline{b} \rfloor} \qquad (8)$$

this is an enforced CG inequality (CG is without the part in the middle)

is valid for X (and $conv(X)$).

Note: stronger inequality than corresponding CG cut but coefficients are not integer.

Proof:

Let $J = \{1, \ldots, n\}$. For $\underline{u} \in \mathbb{Q}_+^m$ denote by $J_1 = \{j \in J : \{\underline{u}^t A_j\} \le \{\underline{u}^t \underline{b}\}\}$ and $J_2 = J \setminus J_1$.

Note that if $\underline{u}^t A_j \in \mathbb{Z}$ then $j \in J_1$.

For all $\underline{x} \in X$ we have

$$\sum_{j \in J_1} \lfloor \underline{u}^t A_j \rfloor x_j + \sum_{j \in J_2} (\underline{u}^t A_j) x_j \le \underline{u}^t \underline{b}. \qquad (9)$$

Since $\underline{u}^t A_j = \lfloor \underline{u}^t A_j \rfloor + \{\underline{u}^t A_j\}$ and $\lceil \underline{u}^t A_j \rceil - \lfloor \underline{u}^t A_j \rfloor = 1$ for all $j \in J_2$, we obtain
$\underline{u}^t A_j = \lceil \underline{u}^t A_j \rceil + \{\underline{u}^t A_j\} - 1$, for all $j \in J_2$.

Substituting to Eq. (9) we obtain $w - z \le \underline{u}^t \underline{b}$ for all $\underline{x} \in X$ with

$$w = (\sum_{j \in J_1} \lfloor \underline{u}^t A_j \rfloor x_j + \sum_{j \in J_2} \lceil \underline{u}^t A_j \rceil x_j) \in \mathbb{Z}, \qquad z = \sum_{j \in J_2} (1 - \{\underline{u}^t A_j\}) x_j \ge 0.$$

Applying Proposition 2 to $w - z \le \underline{u}^t \underline{b}$ we obtain the valid inequality

$$w - \frac{z}{1 - \{\underline{u}^t \underline{b}\}} \le \lfloor \underline{u}^t \underline{b} \rfloor. \qquad (10)$$

Substituting w and z in Eq. (10) gives Eq. (8). □

3.6.5 Gomory mixed integer cutting planes

Generic MILP

$$\begin{aligned} \min \quad & \underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} & \\ s.t. \quad & A_1 \underline{x} + A_2 \underline{y} = \underline{b} & (11) \\ & \underline{x} \ge \underline{0}, \; \underline{y} \ge \underline{0} & (12) \\ & \underline{x} \text{ integer}. & (13) \end{aligned}$$

$(\underline{x}_{LP}^*, \underline{y}_{LP}^*)$ an optimal basic feasible solution of LP relaxation.

Denote by N_1/N_2 the indices in N corresponding to integer/continuous variables.

N_1, N_2 are the indeces of the non-basis variables of (respectively) $\underline{x}$ and $\underline{y}$

If $\underline{x}_{LP}^*$ not integer ($(\underline{x}_{LP}^*, \underline{y}_{LP}^*)$ not optimal), $\exists$ an index $h \in B$ such that $x_h^* \notin \mathbb{Z}$.

Canonical form w.r.t. optimal basis contains a row, say t-th one:

$$x_h + \sum_{j \in N_1} \overline{a}_{tj} x_j + \sum_{j \in N_2} \overline{a}_{tj} y_j = \overline{b}_t \qquad (14)$$

expression of x_h^* in terms of the non-basis variables (with indeces N_1, N_2)

for appropriate $\overline{a}_{tj}$ and $\overline{b}_t$, with $\overline{b}_t \notin \mathbb{Z}$.

Proposition 4: The **Gomory mixed integer (GMI)** inequality

$$\boxed{x_h + \sum_{j \in N_1} (\lfloor \overline{a}_{tj} \rfloor + \frac{(\{\overline{a}_{tj}\} - \{\overline{b}_t\})^+}{1 - \{\overline{b}_t\}}) x_j \le \lfloor \overline{b}_t \rfloor + \sum_{j \in N_2} \frac{(\overline{a}_{tj})^-}{1 - \{\overline{b}_t\}} y_j} \qquad (15)$$

stronger than CG

is valid for the feasible region (11)-(13) and is violated by $(\underline{x}_{LP}^*, \underline{y}_{LP}^*)$.

Proof: Eq. (14) and $\underline{x} \ge \underline{0}$ and $\underline{y} \ge \underline{0}$ imply

$$x_h + \sum_{j \in N_1 : \{\overline{a}_{tj}\} \le \{\overline{b}_t\}} \lfloor \overline{a}_{tj} \rfloor x_j + \sum_{j \in N_1 : \{\overline{a}_{tj}\} > \{\overline{b}_t\}} \overline{a}_{tj} x_j \le \overline{b}_t + \sum_{j \in N_2} (\overline{a}_{tj})^- y_j$$

is valid for the feasible region (11)-(13).

Since $\overline{a}_{tj} = (\lfloor \overline{a}_{tj} \rfloor + 1) - (1 - \{\overline{a}_{tj}\})$ $\forall j$, by splitting the second summation we obtain

$$x_h + \sum_{j \in N_1 : \{\overline{a}_{tj}\} \le \{\overline{b}_t\}} \lfloor \overline{a}_{tj} \rfloor x_j + \sum_{j \in N_1 : \{\overline{a}_{tj}\} > \{\overline{b}_t\}} (\lfloor \overline{a}_{tj} \rfloor + 1) x_j \le \overline{b}_t + \sum_{j \in N_2} (\overline{a}_{tj})^- y_j + \sum_{j \in N_1 : \{\overline{a}_{tj}\} > \{\overline{b}_t\}} (1 - \{\overline{a}_{tj}\}) x_j.$$

Denoting by w the lhs and by z the rhs except $\overline{b}_t$, we have $w \le \overline{b}_t + z$.

Since $w \in \mathbb{Z}$ and $z \ge 0$, Proposition 2 implies that $w \le \overline{b}_t + \frac{z}{1 - \{\overline{b}_t\}}$ is valid. With simple algebraic manipulations we obtain (15).

Clearly $(\underline{x}^*_{LP}, \underline{y}^*_{LP})$ does not satisfy (15) because $x^*_j = 0\ \forall j \in N_1$, $y^*_j = 0\ \forall j \in N_2$, and $x^*_h = \bar{b}_t > \lfloor \bar{b}_t \rfloor$.

□

Recall: For pure ILP

i) GMI cut (15) is potentially stronger than corresponding fractional Gomory cut $(\frac{(\{\bar{a}_{tj}\}-\{\bar{b}_t\})^+}{1-\{\bar{b}_t\}} \geq 0$ and $y_j = 0\ \forall j \in N_2)$,

ii) coefficients are not integer anymore.

Example: integer Gomory cuts versus GMI cuts (from T. Ralphs, ISE 418 Lecture 11)

Unlike for fractional Gomory cuts in pure ILP, no finite termination guarantee for GMI cuts but very effective in practice (see later).

3.7 Strong valid inequalities for structured ILP problems

We are thinking about cutting planes approach and we would like to exploit the structures of specific problems (to get stronger inequalities)

By studying the problem structure, we can derive strong valid inequalities yielding better approximations of $conv(X)$ and hence tighter bounds.

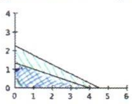
Consider any $P = \{\underline{x} \in \mathbb{R}^n_+ \ : \ A\underline{x} \leq \underline{b}\}$.

Definition: Given $\underline{\pi}^t\underline{x} \leq \pi_0$ and $\underline{\mu}^t\underline{x} \leq \mu_0$ both valid for P, $\underline{\pi}^t\underline{x} \leq \pi_0$ **dominates** $\underline{\mu}^t\underline{x} \leq \mu_0$ if $\exists\ u > 0$ such that $u\underline{\mu} \leq \underline{\pi}$ and $\pi_0 \leq u\mu_0$ with $(\underline{\pi}, \pi_0) \neq (u\underline{\mu}, u\mu_0)$.

if the set of feasible solutions of the first one is included in the set of feasible solutions of the second one

Since $\boxed{u\underline{\mu}^t\underline{x} \leq \underline{\pi}^t\underline{x} \leq \pi_0 \leq u\mu_0}$, clearly $\{\underline{x} \in \mathbb{R}^n_+ : \underline{\pi}^t\underline{x} \leq \pi_0\} \subseteq \{\underline{x} \in \mathbb{R}^n_+ : \underline{\mu}^t\underline{x} \leq \mu_0\}$.

Example: $x_1 + 3x_2 \leq 4$ dominates $2x_1 + 4x_2 \leq 9$ since for $(\underline{\pi}, \pi_0) = (1, 3, 4)$ and $(\underline{\mu}, \mu_0) = (2, 4, 9)$ we have $\frac{1}{2}\underline{\mu} \leq \underline{\pi}$ and $\pi_0 \leq \frac{1}{2}\mu_0$.



$x_1 + 3x_2 \leq 4 \ \rightarrow \ (1,3,4) = (\pi, \pi_0)$

$2x_1 + 4x_2 \leq 9 \ \rightarrow \ (2,4,9) = (\mu, \mu_0)$

we have: $\frac{1}{2}\begin{bmatrix}2\\4\end{bmatrix} \leq \begin{bmatrix}1\\3\end{bmatrix}$, $4 \leq \frac{1}{2}\cdot 9$

$\Longrightarrow u = \frac{1}{2}$

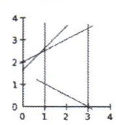
Definition: A valid inequality $\underline{\pi}^t\underline{x} \leq \pi_0$ is **redundant** in the description of P $\ A\underline{x} \leq \underline{b}$ if $\exists\ k \geq 2$ valid inequalities $\underline{\pi}^i\underline{x} \leq \pi^i_0$ for P with $u_i > 0$, $1 \leq i \leq k$, such that

$$(\sum_{i=1}^{k} u_i\underline{\pi}^i)\underline{x} \leq \sum_{i=1}^{k} u_i\pi^i_0 \quad \text{dominates} \quad \underline{\pi}^t\underline{x} \leq \pi_0.$$

a valid inequality (*) is redundant if $\exists$ at least 2 other inequalities (in $A\underline{x} \leq \underline{b}$) that, if linearly combined, generate an inequality which dominates (*).

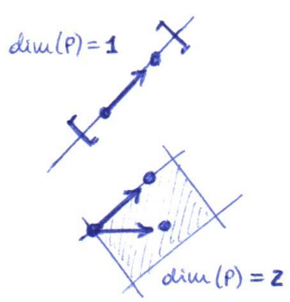
Example:

$P = \{(x_1, x_2) \in \mathbb{R}^2_+ \ : \ -x_1 + 2x_2 \leq 4,\ -x_1 - 2x_2 \leq -3,\ -x_1 + x_2 \leq 5/3,\ 1 \leq x_1 \leq 3\}$



$-x_1 + x_2 \leq 5/3$ is redundant: it is dominated by $-x_1 + x_2 \leq 3/2$, which is implied by $-x_1 + 2x_2 \leq 4$ and $-x_1 \leq -1$ (with $u_1 = u_2 = \frac{1}{2}$).

Observation: It can be very difficult to check redundancy. In practice, try to avoid dominated inequalities.



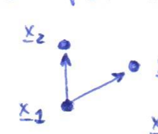
3.7.1 Faces and facets of polyhedra

Consider any $P = \{\underline{x} \in \mathbb{R}^n \ : \ A\underline{x} \leq \underline{b}\}$.

Definitions

- $\underline{x}_1, \ldots, \underline{x}_k \in \mathbb{R}^n$ are **affinely independent** if $k-1$ vectors $\underline{x}_2 - \underline{x}_1, \ldots, \underline{x}_k - \underline{x}_1$ are linearly independent, or equivalently if k vectors $(\underline{x}_1, 1), \ldots, (\underline{x}_k, 1) \in \mathbb{R}^{n+1}$ are linearly independent.
- The **dimension** of P, $\dim(P)$, is equal to the maximum number of affinely independent points of P minus 1.
- P is **full dimensional** if $\dim(P) = n$, i.e., no equation $\underline{a}^t\underline{x} = b$ is satisfied with equality by all the points $\underline{x} \in P$.

example: $\mathbb{R}^2$

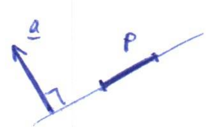


x_1, x_2, x_3 are affinely independent since the two vectors are linearly independent

(equivalently): P is full dimensional if its dimension is equal to the dimension of the space

Illustrations:

P is full dimensional in $\mathbb{R}^2$



P is not full dimensional (Here $\exists\,\underline{a}$ s.t. P is included in $\underline{a}^T\underline{x} = \underline{b}$)

For the sake of simplicity, we assume $dim(P) = n$. (polyedron of full dimension)

Theorem: If $dim(P) = n$, P admits a unique minimal description

$$P = \{\underline{x} \in \mathbb{R}^n : \underline{a}_i^t \underline{x} \leq b_i, i = 1, \ldots, m\}$$

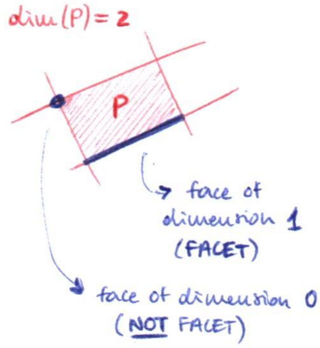
where each inequality is unique within a positive multiple.

Each inequality is necessary: deleting anyone yields a different polyhedron.

Moreover, each valid inequality for P which is not a positive multiple of one $\underline{a}_i^t \underline{x} \leq b_i$ is redundant.

1. Alternative characterization of necessary valid inequalities

dim(P) = 2



Definitions

- Let $F = \{\underline{x} \in P : \underline{\pi}^t \underline{x} = \pi_0\}$ for any valid inequality $\underline{\pi}^t \underline{x} \leq \pi_0$ for P. Then F is a **face** of P and $\underline{\pi}^t \underline{x} \leq \pi_0$ *represents* or *defines* F.
- If F is a face of P and $\dim(F) = \dim(P) - 1$, then F is a **facet** of P.

Illustration:

we need just facets (all facets) to characterize the optimal region

Consequences: The faces of a polyhedron are polyhedra, a polyhedron has a finite number of faces.

Theorem: If P is full dimensional, a valid inequality is necessary to describe P if and only if it defines a facet of P, i.e., if $\exists$ n affinely independent points of P satisfying it at equality.

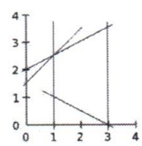
If we want to find valid inequalities → facet inequalities

Example

Consider $P \subset \mathbb{R}^2$ described by:

$$\begin{aligned} x_1 + 2x_2 &\leq 4 && (1)\\ -x_1 - 2x_2 &\leq -3 && (2)\\ -x_1 + x_2 &\leq \frac{3}{2} && (3)\\ x_1 &\leq 3 && (4)\\ x_1 &\geq 1 && (5) \end{aligned}$$

(3) → this does not define a facet, all the others are defining facets (are necessary)



dim(P) = 2

Verify that P is full dimensional (dim(P)=2).

Which inequalities define facets of P or are redundant? All but (3) define facets.

2. Showing that a valid inequality is facet defining

Consider $X \subset \mathbb{Z}_+^n$ and a valid inequality $\underline{\pi}^t \underline{x} \leq \pi_0$ for X.

Assumption: $conv(X)$ is bounded and $dim(conv(X)) = n$.

Simple approaches to show that $\underline{\pi}^t \underline{x} \leq \pi_0$ defines a facet of $conv(X)$:

1) Apply the definition: Find n points $\underline{x}^1, \ldots, \underline{x}^n \in X$ satisfying $\underline{\pi}^t \underline{x} = \pi_0$ and prove that they are affinely independent. } kind of difficult sometimes

2) Indirect approach:

(i) Select t points $\underline{x}^1, \ldots, \underline{x}^t \in X$, with $t \geq n$, satisfying $\underline{\pi}^t \underline{x} = \pi_0$.
Suppose that they all belong to a generic hyperplane $\underline{\mu}^t \underline{x} = \mu_0$.

(ii) Solve linear system

$$\sum_{j=1}^{n} \mu_j x_j^k = \mu_0 \qquad \text{for } k = 1, \ldots, t$$

in $n+1$ unknowns $\mu_0, \mu_1, \ldots, \mu_n$.

(iii) If the only solution is $(\underline{\mu}, \mu_0) = \lambda(\underline{\pi}, \pi_0)$ with $\lambda \neq 0$, then $\underline{\pi}^t \underline{x} \leq \pi_0$ defines a facet of $conv(X)$.

KNAPSACK PROBLEM – 1D

- n objects
- p_i = profit for i $\quad \forall i \in \{1, .., n\} := N$
- a_i = weight for i $\quad \forall i \in \{1, .., n\} := N$
- b = knapsack capacity

goal: decide which objects to select so as to maximize total profit while respecting the capacity constraints

ILP formulation:

Variables: • $x_i = \begin{cases} 1 & \text{if } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$ $\quad i \in N$

$$\max \sum_{i=1}^{n} p_i x_i$$
$$\sum_{i=1}^{n} a_i x_i \le b$$
$$x_i \in \{0,1\} \quad \forall i$$

Cover inequalities:

- A subset $C \subseteq N$ is a cover for $X = \{ \underline{x} \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \le b \}$ if $\sum_{j \in C} a_j > b$ ($\equiv C$ is a cover if it corresponds to a subset of indeces which items will not fit (all together) in the knapsack).
- A cover is minimal if $\forall j \in C$: $C \setminus \{j\}$ is not a cover.
- If $C \subseteq N$ is a cover for X, the cover inequality $\sum_{j \in C} x_j \le |C| - 1$ is valid for X.
- A valid inequality is an equality that is satisfied by all the feasible solutions ($\underline{\pi}^T \underline{x} \le \pi_0$ valid for $X \subseteq \mathbb{R}^n$ if $\underline{\pi}^T \underline{x} \le \pi_0 \quad \forall \underline{x} \in X$)
- Why is it a valid inequality?
  Given a cover C, since it's impossible to put in the knapsack all the elements of C, at most $|C|-1$ ~~elements~~ of them can be selected. This implies that $\sum_{j \in C} x_j \le |C| - 1$ is satisfied by all integers solutions, and so $(\uparrow)$ is a valid inequality.

Separation problem:

Problem:
Given a fractional $\underline{x}^*$ with $0 \le x_j^* \le 1 \quad \forall j \in N$, find a cover inequality that is violated by $\underline{x}^*$ or establish that none exists.

Since $\left[\sum_{j \in C} x_j \le |C| - 1 \right] \equiv \left[\sum_{j \in C} (1 - x_j) \ge 1 \right]$ the problem can be written as:

$$\exists \text{ subset } C \subseteq N \text{ s.t. } \underbrace{\sum_{j \in C} a_j > b}_{C \text{ is a cover}} \text{ and } \underbrace{\sum_{j \in C} (1 - x_j^*) < 1}_{C \text{ violates the cover inequality}} \; ?$$

ILP formulation: $\underline{z} = (z_j)_{j \in N}$: $z_j = 1$ if $j \in C$. $\underline{z} \in \{0,1\}^n$: ($\underline{z}$ defines C)

$$\exists \; \xi := \min \sum_{j \in N} (1 - x_j^*) z_j \quad \text{s.t. } \xi < 1 \, ?$$
$$\text{s.t. } \sum_{j \in N} a_j z_j > b$$
$$\underline{z} \in \{0,1\}^n$$

$$\xi \rightarrow \begin{cases} \geq 1 & \Longrightarrow \quad \underline{x}^* \text{ satisfies all the cover ineq.} \\ < 1 & \Longrightarrow \quad \sum_{j \in C} x_j \leq |C| - 1 \text{ (with } C = \{j : z_j^* = 1,\ j \in N\} \end{cases}$$

where $\underline{z}^*$ is the opt. solution of the problem)
is violated by $\underline{x}^*$ by a quantity $1 - \xi$

Consider $X = \{\underline{x} \in \{0,1\}^6 : 12x_1 + 9x_2 + 7x_3 + 5x_4 + 5x_5 + 3x_6 \leq 14\}$

- list all the minimal cover inequalities that are valid for X
- apply the lifting procedure to ~~cover~~ the cover inequality containing the variables x_3, x_5, x_6

- $x_1 + x_2 \leq 1$
  $x_1 + x_3 \leq 1$
  $x_1 + x_4 \leq 1$
  $x_1 + x_5 \leq 1$
  $x_1 + x_6 \leq 1$
  $x_2 + x_3 \leq 1$
  $x_2 + x_4 + x_5 \leq 2$
  $x_2 + x_4 + x_6 \leq 2$
  $x_2 + x_5 + x_6 \leq 2$
  $x_3 + x_4 + x_5 \leq 2$
  $x_3 + x_4 + x_6 \leq 2$
  $x_3 + x_5 + x_6 \leq 2$

- Lifting procedure to strenghten a cover inequality (to obtain a facet defining one)

$x_3 + x_5 + x_6 \leq 2 \quad : \quad C = \{3, 5, 6\}$

$j \in N \setminus C = \{1, 2, 4\}$

- $\alpha_1 x_1 + x_3 + x_5 + x_6 \leq 2 \quad : \quad \alpha_1 ?$

  If $x_1 = 0 \Rightarrow \alpha_1$ can be $\forall$

  If $x_1 = 1 \Rightarrow \alpha_1 = 2 - \max\{x_3 + x_5 + x_6 : 7x_3 + 5x_5 + 3x_6 \leq 14 - 12 = 2\}$

  $= 2 - 0 = 2$

  $\Rightarrow 2x_1 + x_3 + x_5 + x_6 \leq 2$

- $\alpha_2 x_2 + 2x_1 + x_3 + x_5 + x_6 \leq 2$

  If $x_2 = 0 \Rightarrow \forall \alpha_2$

  If $x_2 = 1 \Rightarrow \alpha_2 = 2 - \max\{2x_1 + x_3 + x_5 + x_6 : 12x_1 + 7x_3 + 5x_5 + 3x_6 \leq 14 - 9 = 5\}$

  $= 2 - 1 = 1$

  $\Longrightarrow 2x_1 + x_2 + x_3 + x_5 + x_6 \leq 2$

- $\alpha_4 x_4 + 2x_1 + x_2 + x_3 + x_5 + x_6 \leq 2$

  If $x_4 = 0 \Rightarrow \forall \alpha_4$

  If $x_4 = 1 \Rightarrow \alpha_4 = 2 - \max\{2x_1 + x_2 + x_3 + x_5 + x_6 : 12x_1 + 9x_2 + 7x_3 + 5x_5 + 3x_6 \leq 14 - 5 = 9\}$

  $= 2 - 2 = 0$

$\longrightarrow \boxed{2x_1 + x_2 + x_3 + x_5 + x_6 \leq 2}$

Since we started from a minimal cover the lifting procedure ended with a facet inequality (facet of conv(X)).

Example:

Consider $X = \{(\underline{x}, y) \in \mathbb{R}^m \times \{0,1\} : \sum_{i=1}^m x_i \le my,\ 0 \le x_i \le 1\ \forall i\}$

they have to satisfy: $x_i \le y$

i) Verify that $dim(conv(X)) = m + 1$. : we have to exhibit $m+2$ points $\in conv(X)$ which are affinely ⊥⊥ :

$(\underline{0}, 0)$, $(\underline{0}, 1)$ and $(\underline{e}_i, 1)$, with $1 \le i \le m$, are $m + 2$ affinely independent points of $conv(X)$.

ii) Show (indirect approach) that, for each i, valid inequality $x_i \le y$ defines a facet of $conv(X)$.

Consider $m + 1$ points $(\underline{0}, 0)$, $(\underline{e}_i, 1)$ and $(\underline{e}_i + \underline{e}_{i'}, 1)$ for $i' \ne i$, which are feasible and satisfy $x_i = y$.

- Since $(\underline{0}, 0)$ lies on hyperplane defined by $\sum_{j=1}^m \mu_j x_j + \mu_{m+1} y = \mu_0$, then $\mu_0 = 0$.
- Since $(\underline{e}_i, 1)$ lies on hyperplane $\sum_{j=1}^m \mu_j x_j + \mu_{m+1} y = 0$, then $\mu_{m+1} = -\mu_i$.
- Since $(\underline{e}_i + \underline{e}_{i'}, 1)$ lies on hyperplane $\sum_{j=1}^m \mu_j x_j - \mu_i y = 0$, then $\mu_{i'} = 0$ for $i' \ne i$.
- Thus the hyperplane is $\mu_i x_i - \mu_i y = 0$ and $x_i \le y$ defines a facet of $conv(X)$.

3.7.2 Cover inequalities for binary knapsack problem

Consider $X = \{\underline{x} \in \{0,1\}^n : \sum_{j=1}^n a_j x_j \le b\}$ with $b > 0$ and $N = \{1, \ldots, n\}$.

indeces of the items

Assumptions: For each j, $a_j \le b$ and $a_j > 0$

capacity of the knapsack

Definition: A subset $C \subseteq N$ is a **cover** for X if $\sum_{j \in C} a_j > b$.
A cover is **minimal** if, for each $j \in C$, $C \setminus \{j\}$ is not a cover.

C is a cover if it corresponds to a subset of indeces which items will not fit (all together) the knapsack

Example: For $X = \{\underline{x} \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \le 19\}$
$\{1, 2, 3\}$ is a minimal cover and $\{3, 4, 5, 6, 7\}$ is a non-minimal cover.

since $\{3,4,5,6\}$ is still a cover

Proposition: If $C \subseteq N$ is a cover for X, the **cover inequality**

$$\sum_{j \in C} x_j \le |C| - 1$$

is valid for X.

we have to delete at least one element from the cover

Example cont.: For above covers $x_1 + x_2 + x_3 \le 2$ and $x_3 + x_4 + x_5 + x_6 + x_7 \le 4$.

Proposition: If $C \subseteq N$ is a cover for X, the cover inequality

$$\sum_{j \in C} x_j \le |C| - 1$$

defines a facet of $P_C := conv(X) \cap \{\underline{x} \in \mathbb{R}^n : x_j = 0, j \in N \setminus C\}$ if and only if C is a minimal cover.

Proof:

Separation of cover inequalities

Separation problem: Given a fractional $\overline{x}$ with $0 \le \overline{x}_j \le 1$, $1 \le j \le n$, find a cover inequality that is violated by $\overline{x}$ or establish that none exists.

If we know how to do this (solve the separation problem) then we'll be able to use the cover inequalities for a cutting plane method

Since $\sum_{j \in C} x_j \le |C| - 1$ can be written as $\sum_{j \in C}(1 - x_j) \ge 1$, it amounts to question:

$\exists$ subset $C \subseteq N$ such that $\sum_{j \in C} a_j > b$ and $\sum_{j \in C}(1 - \overline{x}_j) < 1$?

C is a cover   C violates the inequality

∃ a cover that violates the cover inequality?

$\underline{z} = (z_j)_j$ defines C : $z_j = 1 \Rightarrow$ the j-th item is in C

If $\underline{z} \in \{0,1\}^n$ incidence vector of $C \subseteq N$, it is equivalent to:

$$\zeta = \min\{\sum_{j \in N}(1 - \overline{x}_j) z_j : \sum_{j \in N} a_j z_j > b, \underline{z} \in \{0,1\}^n\} < 1?$$

We have to formulate the problem: (ILP)

Proposition:

(i) If $\zeta \ge 1$, $\overline{x}$ satisfies all cover inequalities.

(ii) If $\zeta < 1$ with optimal solution $\underline{z}^*$, then $\sum_{j \in C} x_j \le |C| - 1$ with $C = \{j : z_j^* = 1, 1 \le j \le n\}$ is violated by $\overline{x}$ by a quantity $1 - \zeta$.

If it is we would have found a $C \subseteq N$ which is a cover which violates the corresponding cover inequality. If not ∄ cover inequality that is violated by the current $\overline{x}$.

Example:

$$\begin{array}{ll} \max & 5x_1 + 2x_2 + x_3 + 8x_4 \\ s.t. & 4x_1 + 2x_2 + 2x_3 + 3x_4 \leq 4 \\ & x_j \in \{0,1\} \quad \forall j \in \{1, \dots, 4\} \end{array}$$

Optimal solution of LP relaxation $\underline{x}^*_{LP} = (1/4, 0, 0, 1)^t$ of value 9.25.

we start from this: we want to solve the separation problem for this fractional solution $\underline{x}^*_{LP}$

Separation problem (binary knapsack-like):

$$\begin{array}{rl} \zeta = \min & \frac{3}{4}z_1 + z_2 + z_3 \\ s.t. & 4z_1 + 2z_2 + 2z_3 + 3z_4 > 4 \\ & z_j \in \{0,1\} \quad \forall j \in \{1, \dots, 4\} \end{array}$$

$\sum_{j \in N} (1 - x_j^*) z_j$: we are trying to minimize the degree of violation of the inequality

$\sum_{j \in N} a_j z_j > b$

where constraint can be replaced with $4z_1 + 2z_2 + 2z_3 + 3z_4 \geq 5$.

Now $\underline{z}^* = (1, 0, 0, 1)^t$ with $\zeta = \frac{3}{4}$.

what is the C cover corresponding to the cover inequality which is violated by $\underline{x}^*_{LP}$?

$C = \{j \in N : z_j = 1\}$

$\Rightarrow C = \{1, 4\}$

$\Rightarrow x_1 + x_4 \leq 1$ cuts away $\underline{x}^*_{LP}$

The cover inequality

$$x_1 + x_4 \leq 1$$

cuts away $\underline{x}^*_{LP}$ by $1 - \zeta = \frac{1}{4}$.

Separation problem is *NP*-hard, in practice fast heuristics.

Strengthening cover inequalities

by extending the cover: (1)

Proposition: If $C \subseteq N$ is a cover for X, the **extended cover inequality**

$$\sum_{j \in E(C)} x_j \leq |C| - 1$$

is valid for X, where $E(C) = C \cup \{j \in N : a_j \geq a_i \text{ for all } i \in C\}$.

Example cont.: $X = \{\underline{x} \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$

extended cover inequality for $C = \{3, 4, 5, 6\}$ is

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$$

which clearly dominates

$$x_3 + x_4 + x_5 + x_6 \leq 3. \qquad (6)$$

$C = \{3,4,5,6\}$ is a cover $\Rightarrow$ valid inequality: $x_3 + x_4 + x_5 + x_6 \leq 3$. We add to C all the j such that $a_j \geq a_i$ $i \in C$: $C = \{3,4,5,6\} \longrightarrow \{1,2,3,4,5,6\}$ since $a_1 = 11 > a_3, a_4, a_5, a_6$, $a_2 = 6 \geq a_3, a_4, a_5, a_6$. $\Rightarrow$ new (stronger) valid inequality: EXTENDED cover inequality: $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$

Systematic way to strengthen a cover inequality to obtain a facet defining one.

minimal cover → lifting procedure → facet

Example lifting procedure:

$$X = \{\underline{x} \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$$

Minimal cover $C = \{3, 4, 5, 6\}$ with $x_3 + x_4 + x_5 + x_6 \leq 3$. ← we want to strengthen this

- Consider x_j with $j \in N \setminus C$ in the order x_1, x_2 and x_7.
- The largest α_1 such that $\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$ is valid for X is
$$\begin{aligned} \alpha_1 &= 3 - \max\{x_3 + x_4 + x_5 + x_6 \ : \ 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19 - 11 = 8, \underline{x} \in \{0,1\}^4\} \\ &= 3 - 1 = 2 \end{aligned}$$
- Largest α_2 such that $\alpha_2 x_2 + 2x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$ is valid for X is
$$\begin{aligned} \alpha_2 &= 3 - \max\{2x_1 + x_3 + x_4 + x_5 + x_6 \ : \ 11x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19 - 6 = 13, \underline{x} \in \{0,1\}^5\} \\ &= 3 - 2 = 1 \end{aligned}$$
- Largest α_7 such that $\alpha_7 x_7 + 2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$ is valid for X is
$$\begin{aligned} \alpha_7 &= 3 - \max\{2x_1 + \textstyle\sum_{j=2}^6 x_j : \ 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19 - 1 = 18, \underline{x} \in \{0,1\}^6\} \\ &= 3 - 3 = 0 \end{aligned}$$
- Thus stronger valid inequality: $2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$.

- $x_1 = 0 \Rightarrow \alpha_1$ can be $\forall$
- $x_1 = 1$ then $\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$ and since $x_1 = 1$:
$\alpha_1 + x_3 + x_4 + x_5 + x_6 \leq 3$
so:
$\alpha_1 + \underbrace{\max\{x_3 + x_4 + x_5 + x_6\}} \leq 3$
$\max\{x_3 + x_4 + x_5 + x_6 : 11 + 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19\}$
$\Rightarrow \alpha_1 + 1 \leq 3$
$\Rightarrow \alpha_1 \leq 2$

Lifting procedure for cover inequalities

(2)

Let $j_1, \dots, j_r$ be an ordering of $N \setminus C$ and set $t = 1$.

$\sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$ valid inequality obtained at iteration $t - 1$.

Iteration t: Determine the maximum α_{j_t} such that

$$\alpha_{j_t} x_{j_t} + \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$$

is valid for X by solving (binary knapsack) problem

$$\boxed{\begin{array}{rl} \sigma_t = \max & \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \\ s.t. & \sum_{i=1}^{t-1} a_{j_i} x_{j_i} + \sum_{j \in C} a_j x_j \leq b - a_{j_t} \\ & \underline{x} \in \{0,1\}^{|C|+t-1} \end{array}}$$

and setting $\alpha_t = |C| - 1 - \sigma_t$.

Terminate when $t = r$.

Note: σ_t = maximum amount of "space" used up by the variables of indices in $C \cup \{j_1, \dots, j_{t-1}\}$ when $x_{j_t} = 1$.

Proposition: If $C \subset N$ is a minimal cover and $a_j < b$ for all $j \in N$, the lifting procedure is guaranteed to yield a facet defining inequality of $conv(X)$.

Example cont.:

$$X = \{\underline{x} \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \le 19\}$$

the valid inequality

$$2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \le 3$$

defines a facet of $conv(X)$.

The resulting facet defining inequality depends on the order of variables $N \setminus C$, that is, on the lifting sequence.

3.7.3 Strong valid inequalities for TSP

STSP: Given undirected $G = (V, E)$ with $n = |V|$ nodes and a cost c_e for every $e = \{i, j\} \in E$, determine a Hamiltonian cycle of minimal total cost.

$$\begin{array}{lll} \min & \sum_{e \in E} c_e x_e & \\ \text{s.t.} & \sum_{e \in \delta(i)} x_e = 2 & i \in V \\ & \sum_{e \in E(S)} x_e \le |S| - 1 & S \subset V, S \neq \emptyset \\ & x_e \in \{0,1\} & e \in E. \end{array}$$

$X = \{$ all incidence vectors $\underline{x} \in \{0,1\}^{|E|}$ of Hamiltonian cycles $\}$

Proposition: For every $S \subseteq V$ with $2 \le |S| \le n/2$ and $n \ge 4$,

$$\sum_{e \in E(S)} x_e \le |S| - 1 \qquad (7)$$

defines a facet of polytope $conv(X)$.

The STSP polytope has a very complicated structure. Many classes of facet defining inequalities are known but its complete description is unknown.

Separation of cut-set inequalities for the ATSP

ILP formulation:

$$\begin{array}{llll} \min & \sum_{(i,j) \in A} c_{ij} x_{ij} & & (8) \\ s.t. & \sum_{(i,j) \in \delta^-(j)} x_{ij} = 1 & \forall j & (9) \\ & \sum_{(i,j) \in \delta^+(i)} x_{ij} = 1 & \forall i & (10) \\ & \sum_{(i,j) \in \delta^+(S)} x_{ij} \ge 1 & \forall S \subset V, 1 \in S & (11) \\ & x_{ij} \in \{0,1\} & \forall (i,j) \in A & (12) \end{array}$$

→ without loss of generality we can consider the subset that contains the node 1

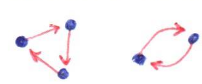
Cutting plane approach:

Start solving LP relaxation of (8)-(12) without (11), namely

$$\begin{array}{llll} \min & \sum_{(i,j) \in A} c_{ij} x_{ij} & & (13) \\ s.t. & \sum_{(i,j) \in \delta^-(j)} x_{ij} = 1 & \forall j & (14) \\ & \sum_{(i,j) \in \delta^+(i)} x_{ij} = 1 & \forall i & (15) \\ & x_{ij} \ge 0 & \forall (i,j) \in A, & (16) \end{array}$$

and iteratively add some which substantially violate the current $\underline{x}^*_{LP}$.

we start from this and iteratively we add some usefull cut-set inequalities. **Remember**: without cut-set inequalities we may have:



Proposition:

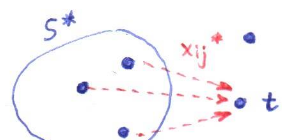
Given $\underline{x}^*_{LP}$ of the current LP relaxation ((13)-(16) with (11) generated so far), a cut-set inequality (11) violated by $\underline{x}^*_{LP}$ can be obtained (if $\exists$) by solving a sequence of instances of the minimum cut problem.

→ after the first iteration we may have added some cut-set inequalities already

Separation algorithm:

We look for violated cut-set inequalities by $\underline{x}^*_{LP}$:

- Given $\underline{x}^*_{LP}$, we look for $S^* \subseteq V$ with $1 \in S^*$ such that $\sum_{(i,j) \in \delta^+(S^*)} x^*_{ij} < 1$.
- Consider directed $G^* = (V, A^*)$ where $A^* := \{(i,j) \in A : x^*_{ij} > 0\}$ with node 1 as source s and capacity x^*_{ij} for each $(i,j) \in A^*$.
- For every choice of sink $t \in V \setminus \{1\}$, we look for a cut $\delta(S^*)$, separating $1 \in S^*$ from $t \in V \setminus S^*$, of minimum capacity.
- If capacity of such minimum cut < 1, S^* induces a cut-set inequality that is violated by $\underline{x}^*_{LP}$, otherwise no such violated inequality (separating 1 from t) exists.
- For each $t \in V \setminus \{1\}$, a minimum cut, separating 1 and t, can be found by looking for a maximum flow from 1 to t in the capacitated network G^* (strong LP duality).



$x_{ij}^* =$ capacity of the arc (i,j)

$(x_{ij}^* = 0 \Rightarrow$ the arc is not included in the graph$)$

Illustration:

Observations:

- The separation problem can be solved in polynomial time.
- The procedure may yield a number of violated cut-set inequalities (one for each t).

With the cutting plane method we have reduced the solution of a optimization problem to the solution of a sequence of separation problems.
Is there an equivalence between the separation problem and the optimization problem?
In fact, there is a nice connection.

3.7.4 Equivalence between separation and optimization

A family of LPs $\min\{\underline{c}^t\underline{x} : \underline{x} \in P_o\}$ with $o \in \mathcal{O}$, where $P_o = \{\underline{x} \in \mathbb{R}^{n_o} : A_o\underline{x} \geq \underline{b}_o\}$ polytope with rational (integer) coefficients and a very large number of constraints.

Examples:
1) Linear relaxation of ATSP with cut-set inequalities ($\mathcal{O}$ set of all graphs)
2) Maximum Matching problem: For each $G = (V, E)$, the matching polytope

$$conv(\{\underline{x} \in \{0,1\}^{|E|} : \sum_{e \in \delta(i)} x_e \leq 1, \forall i \in V\})$$

coincides (Edmonds) with

$$\{\underline{x} \in \mathbb{R}_+^{|E|} : \sum_{e \in \delta(i)} x_e \leq 1, \forall i \in V, \sum_{e \in E(S)} x_e \leq \frac{|S|-1}{2}, \forall S \subseteq V \text{ with } |S| \geq 3 \text{ odd}\}.$$

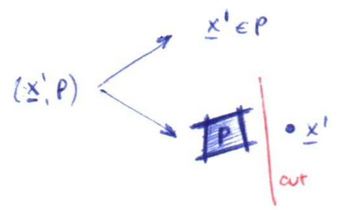
Consider a cutting plane approach.

Assumption: Even though the number of constraints m_o of P_o is exponential in n_o, A_o and $\underline{b}_o$ are specified in a concise way (as a function of a polynomial number of parameters w.r.t. n_o).

- Optimization problem: Given rational polytope $P \subseteq \mathbb{R}^n$ and rational $\underline{c} \in \mathbb{R}^n$, find a $\underline{x}^* \in P$ minimizing $\underline{c}^t\underline{x}$ over $\underline{x} \in P$ or establish that P is empty.

N.B.: P assumed to be bounded just to avoid unbounded problems.

- Separation problem: Given rational polytope $P \subseteq \mathbb{R}^n$ and rational $\underline{x}' \in \mathbb{R}^n$, establish that $\underline{x}' \in P$ or determine a cut that separates $\underline{x}'$ from P.



!

Theorem: (consequence of Grötschel, Lovász, Schriver 1988 theorem)
The separation problem (for a family of polyhedra) can be solved in polynomial time in n and $\log U$ if and only if the optimization problem (for that family) can be solved in polynomial time in n and $\log U$, where U is an upper bound on all a_{ij} and b_i.

Proof based on *Ellipsoid method*, first polynomial algorithm for LP.

Corollary: The LP relaxation of ILP formulation for ATSP with cut-set inequalities can be solved in polynomial time in spite of exponentially many constraints.

(since the corresponding separation problem can be solved in polynomial time)

3.7.5 Remarks on cutting plane methods

Consider a generic Discrete Optimization problem

$$\min\{\underline{c}^t\underline{x} : \underline{x} \in X \subseteq \mathbb{R}_+^n\}$$

with rational coefficients c_i.

When designing a cutting plane method, be aware that:

- It can be difficult to describe one or more families of strong (possibly facet defining) valid inequalities for $conv(X)$.
- The separation problem for a given family $\mathcal{F}$ may require a considerable computational effort (if NP-hard devise heuristics).
- Even when finite convergence is guaranteed (e.g., with Gomory cuts), pure cutting plane methods tend to be very slow.

The subfield of Discrete Optimization studying the polyhedral structure of the ideal formulations ($conv(X)$) is known as Polyhedral Combinatorics.

The state of art for solving ILP is a **combined approach**: it combines the Branch and Bound method and the cutting plane method

3.8 Branch and Cut

Idea: Embed strong valid inequalities into a Branch-and-Bound framework to be able to solve hard/large problems to optimality.

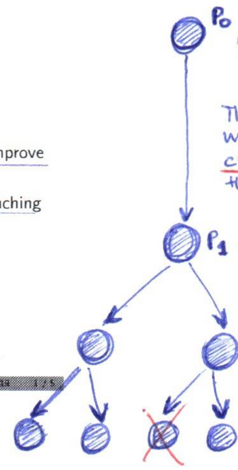
→ *Branch-and-Cut* method

(Strong) valid inequalities are generated throughout the Branch-and-Bound tree.

Advantages:

- stronger LP relaxations of the subproblems yield tighter dual bounds which improve Branch and Bound efficiency,
- slow convergence of pure cutting plane method is contrasted by applying branching steps.

Trade-off between computational load of reoptimization and the quality of the formulations (bounds).

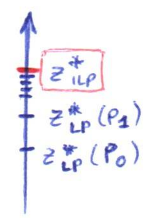


P_0: we have the linear relaxation of the ILP (we obtain a formulation P_0)
This formulation may be weak, so we generate cuts as we can strengthen the formulation.

P_1. At some points the cuts will improve so little that we'll be almost stucked.
So we branch: we subdivide the feasible region in 2 subregions and then we generates cuts in both the subproblems

maybe some of them can be closed

In general we get tighter lower bounds:



Main components of Branch and Cut (min problem)

- Preprocessing
  Delete redundant constraints, strengthen the constraint coefficients and right hand side term, fix variables (whenever possibile).
- Primal heuristics
  Tighter upper bounds lead to a more efficient implicit enumeration.
- Cutting planes pool
  Violated valid inequalities and facets are added by solving corresponding separation problems exactly or heuristically. Many of them are simultaneously added at each node.

  w.r.t. $\underline{x}^*_{LP}$
- Branching strategy
  Different criteria to choose the fractional variable on which to branch based on one/mix of criteria (with largest cost coefficient, "most promising" one based on estimate,...).
- Postprocessing
  When $\underline{x}^*_{LP}$ of value z_{LP} is not integer, primal heuristic yields a feasible solution $\underline{x}_{heur}$ such that $z_{LP} \leq z^* \leq z_{heur}$ ($\underline{x}_{heur}$ often derived by "smart" rounding).
  z^*_{ILP}

For **flow chart** and a basic **example** for the generalized assignment problem

$$
\begin{array}{lll}
\min\ z = & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{s.t.} & \sum_{j \in J} x_{ij} = 1 & \forall i \in I \\
& \sum_{i \in I} w_{ij} x_{ij} \leq b_j & \forall j \in J \\
& x_{ij} \in \{0,1\} & \forall i \in I, \forall j \in J.
\end{array}
$$

see **third computer laboratory** and L. Wolsey, Integer Programming, p. 157-160.

Lab main goals: separate cover inequalities and evaluate the impact of adding them at the root node of the branching tree (Cut and Branch).

Branch and Cut methods are successfully developed to solve to optimality a wide range of discrete optimization problems.

Example: Concorde algorithm for TSP (see http://www.math.uwaterloo.ca/tsp/)

Impact of different features in a MILP solver

From R. Bixby, M. Fenelon, Z. Gu, E. Rothberg and R. Wunderling, Mixed integer programming: A progress report, M. Grötschel ed., The sharpest cut: The impact of Manfred Padberg and his work, MPS/SIAM Series in Optimization (2004) 309-326.

2002 "new generation" Cplex solver for MILPs

Computational experiments on set of 106 benchmark instances

Different features

Feature	Speedup factor
Cuts	**54**
Preprocessing	**11**
Variable fixing	3
Heuristics	1.5

Average speedup for each feature (enabling that feature versus disabling it, while keeping all others active).

Different types of cutting planes

Cut type	Speedup factor
GMI	**2.5**
MIR	**1.8**
Knapsack cover	1.4
Flow cover	1.2
Implied bounds	1.2
Path	1.04
Clique	1.02
GUB cover	1.02

MIR inequalities with heuristic aggregation of constraints

GMI and MIR cuts implementations account for finite precision (avoid invalid cuts or cuts that could slow down LP solution).

3.9 Lagrangian relaxation

Generic ILP
$$\min\{\underline{c}^t\underline{x} : \boxed{A\underline{x} \geq \underline{b}}, D\underline{x} \geq \underline{d}, \underline{x} \in \mathbb{Z}^n\}$$
with integer coefficients.

Suppose $D\underline{x} \geq \underline{d}$ are "complicating" constraints.

For example in the knapsack problem these are the linking constraints

More general setting:
$$\min\{\underline{c}^t\underline{x} : D\underline{x} \geq \underline{d}, \boxed{\underline{x} \in X} \subseteq \mathbb{R}^n\} \qquad (1)$$

Idea: Delete complicating constraints $D\underline{x} \geq \underline{d}$ and, for each one of them, add to objective function a term with a multiplier u_i, which penalizes its violation.

For min problems, ≤ 0 for all feasible solutions of (1).

ILP
↓
Lagrangian sub-problem $\forall \underline{u} \geq \underline{0}$
↓
Lagrangian dual to find the best (tightest) lower bound

Definition: Given
$$z^* = \min\{\underline{c}^t\underline{x} : D\underline{x} \geq \underline{d}, \underline{x} \in X \subseteq \mathbb{R}^n\} \qquad (2)$$
Primal problem

For each *Lagrange multipliers* vector $\underline{u} \geq \underline{0}$, **Lagrangian subproblem** is
$$w(\underline{u}) = \min\{\underline{c}^t\underline{x} + \underline{u}^t(\underline{d} - D\underline{x}) : \underline{x} \in X\} \qquad (3)$$
$= \min L(\underline{x}, \underline{u})$ s.t. $\underline{x} \in X$

where

$L(\underline{x}, \underline{u}) = \underline{c}^t\underline{x} + \underline{u}^t(\underline{d} - D\underline{x})$ *Lagrangian function* of primal (2),

$w(\underline{u}) = \min\{L(\underline{x}, \underline{u}) : \underline{x} \in X\}$ *dual function*.

Proposition: For any $\underline{u} \geq \underline{0}$, the Lagrangian subproblem (3) is a relaxation of (2).

Proof:
Clearly $\{\underline{x} \in X : D\underline{x} \geq \underline{d}\} \subseteq X$.
For any $\underline{u} \geq \underline{0}$ and $\underline{x}$ feasible for (2), we have $w(\underline{u}) \leq \underline{c}^t\underline{x}$.
Indeed $w(\underline{u}) \leq \underline{c}^t\underline{x} + \underline{u}^t(\underline{d} - D\underline{x}) \leq \underline{c}^t\underline{x}$ since $\underline{u}^t(\underline{d} - D\underline{x}) \leq \underline{0}$ for all $\underline{x}$ feasible for (2). □

Corollary: If $z^* = \min\{\underline{c}^t\underline{x} : D\underline{x} \geq \underline{d}, \underline{x} \in X\}$ is finite, then $w(\underline{u}) \leq z^* \quad \forall \underline{u} \geq \underline{0}$.

z^*

↑ $w(\underline{u}) \quad \forall \underline{u} \geq \underline{0}$

we want the tightest lower bound possible

To determine tightest lower bound :

Definition: **Lagrangian dual** of primal problem (2) is
$$\boxed{w^* = \max_{\underline{u} \geq \underline{0}} w(\underline{u})} \qquad (4)$$

$w(\underline{u})$ is **concave** : ∩
maximizing a concave function is like minimizing a convex function

Only nonnegativity constraints.

This depends if we are Lagrangianizing inequalities ($\underline{u} \geq \underline{0}$) or equalities ($\underline{u} \in \mathbb{R}^m$):
$$D\underline{x} = \underline{d} \; = \begin{cases} D\underline{x} \geq \underline{d} & \rightarrow u_i^+ \\ -D\underline{x} \leq -\underline{d} & \rightarrow u_i^- \end{cases}$$
$\Rightarrow u_i^+ \geq 0, \; u_i^- \geq 0$
$\Rightarrow u_i := u_i^+ - u_i^- \in \mathbb{R}$

Note: By relaxing (dualizing) linear constraints, the objective function remains linear.
Any type of other constraints provided subproblem (3) is "sufficiently easy".
For LPs Lagrangian dual coincides with LP dual (exercise 5.2).

WEAK DUALITY RELATION:
so any feasible solution $\tilde{\underline{x}}$ provides an upper bound:
$w(\underline{u}) \leq \underline{c}^T\tilde{\underline{x}}$ (Dual)
and any feasible solution $\tilde{\underline{u}}$ provides a lower bound:
$w(\tilde{\underline{u}}) \leq \underline{c}^T\underline{x}$ (Primal)

$z = \underline{c}^T\underline{x} \quad \forall \underline{x}$

IV

$w(\underline{u}) \quad \forall \underline{u} \geq 0$

Corollary: (Weak Duality)
For any pair of feasible solutions $\underline{x} \in \{\underline{x} \in X : D\underline{x} \geq \underline{d}\}$ of primal (2) and $\underline{u} \geq \underline{0}$ of Lagrangian dual (4), we have
$$w(\underline{u}) \leq \underline{c}^t\underline{x}.$$

Consequences:

!

i) If $\underline{\tilde{x}}$ feasible for primal (2), $\underline{\tilde{u}}$ feasible for Lagrangian dual (4) and $\underline{c}^t\underline{\tilde{x}} = w(\underline{\tilde{u}})$, then $\underline{\tilde{x}}$ and $\underline{\tilde{u}}$ optimal for respectively (2) and (4).

ii) In particular $w^* = \max_{\underline{u}\geq\underline{0}} w(\underline{u}) \leq z^* = \min\{\underline{c}^t\underline{x} : D\underline{x} \geq \underline{d}, \ \underline{x} \in X\}$.
If one problem is unbounded, the other one is infeasible.

Recall: For any primal-dual pair of bounded LPs, we have strong duality ($w^* = z^*$).

Observation: Unlike for LPs, discrete optimization problems can have a *duality gap*, that is $w^* < z^*$.

(gap = difference between z^* and w^*)

$z^* = \underline{c}^T\underline{x}^*$

gap (possible only with discrete optimization)

w^*

ILP with equality constraints:

Lagrangian dual is

$$\max_{\underline{u}\in\mathbb{R}^m} w(\underline{u})$$

with $\underline{u}$ unrestricted in sign.

Example: Uncapacitated Facility Location (UFL)

Variant with profits p_{ij}, fixed costs f_j for opening the depots in the candidate sites, and total profit to be maximized.

MILP formulation:

$$\begin{aligned}
z^* = \max \quad & \textstyle\sum_{i\in M}\sum_{j\in N} p_{ij}x_{ij} - \sum_{j\in N} f_j y_j \\
\text{s.t.} \quad & \boxed{\textstyle\sum_{j\in N} x_{ij} = 1 \qquad \forall i \in M} && (5)\\
& x_{ij} \leq y_j \qquad \forall i \in M, j \in N \\
& y_j \in \{0,1\} \qquad \forall j \in N \\
& 0 \leq x_{ij} \leq 1 \qquad \forall i \in M, j \in N
\end{aligned}$$

M clients i
N candidate sites j
f_j opening costs
p_{ij} profit if all the demand of client i is satisfied in the depot j
x_{ij} fraction of demand of i satisfied by j
$y_j = 1$ if j is open

This is a linking constraint: it links all the candidate sites.
it is considered "complicate" so we want to lagrangianize it:
penalty term:
$u_i\left(1 - \sum_{j\in N} x_{ij}\right) \quad \forall i$
$\longrightarrow \sum_{i\in M}\left(u_i\left(1 - \sum_{j\in N} x_{ij}\right)\right)$

Relaxing constraints (5), Lagrangian subproblem:

$$\begin{aligned}
w(\underline{u}) = \max \quad & \textstyle\sum_{i\in M}\sum_{j\in N}(p_{ij} - u_i)x_{ij} - \sum_{j\in N} f_j y_j + \sum_{i\in M} u_i \\
\text{s.t.} \quad & x_{ij} \leq y_j \qquad \forall i \in M, j \in N && (6)\\
& y_j \in \{0,1\} \qquad \forall j \in N && (7)\\
& 0 \leq x_{ij} \leq 1 \qquad \forall i \in M, j \in N && (8)
\end{aligned}$$

which decomposes into $|N|$ independent subproblems, one for each candidate site j.

Indeed $w(\underline{u}) = \sum_{j\in N} w_j(\underline{u}) + \sum_{i\in M} u_i$ where

$$\begin{aligned}
w_j(\underline{u}) = \max \quad & \textstyle\sum_{i\in M}(p_{ij} - u_i)x_{ij} - f_j y_j && (9)\\
\text{s.v.} \quad & x_{ij} \leq y_j \qquad \forall i \in M \\
& y_j \in \{0,1\} \\
& 0 \leq x_{ij} \leq 1 \qquad \forall i \in M
\end{aligned}$$

we splitted the problem into $|N|$ independent subproblems (easier!)

For each $j \in N$, the subproblem (9) can be solved by inspection:

- If $y_j = 0$, then $x_{ij} = 0$ for each i and objective function value is 0.
- If $y_j = 1$, set $x_{ij} = 1$ for all i such that $p_{ij} - u_i > 0$, with objective function value of

$$\sum_{i\in M} \max\{p_{ij} - u_i, 0\} - f_j.$$

Thus $w_j(\underline{u}) = \max\{0, \sum_{i\in M}\max\{p_{ij} - u_i, 0\} - f_j\}$.

(See Chapter 10 of L. Wolsey, Integer Programming, p. 169-170)

Optimal solutions of Lagrangian subproblem and of the primal

Proposition: If $\underline{u} \geq \underline{0}$ and

i) $\underline{x}(\underline{u})$ is an optimal solution of Lagrangian subproblem (3) $\longrightarrow$ $w(\underline{u}) = \min_{\underline{x}\in X} L(\underline{x},\underline{u})$

ii) $D\underline{x}(\underline{u}) \geq \underline{d}$

iii) $(D\underline{x}(\underline{u}))_i = d_i$ for each $u_i > 0$ (complementary slackness conditions),

then $\underline{x}(\underline{u})$ is also optimal for primal (2).

Proof:

Due to (*i*) we have $w^* \geq w(\underline{u}) = \underline{c}^t\underline{x}(\underline{u}) + \underline{u}^t(\underline{d} - D\underline{x}(\underline{u}))$ and to (*iii*) we have $\underline{c}^t\underline{x}(\underline{u}) + \underline{u}^t(\underline{d} - D\underline{x}(\underline{u})) = \underline{c}^t\underline{x}(\underline{u})$.

According to (*ii*), $\underline{x}(\underline{u})$ is a feasible solution of primal (2) and hence $\underline{c}^t\underline{x}(\underline{u}) \geq z^*$.

Thus $w^* \geq \underline{c}^t\underline{x}(\underline{u}) + \underline{u}^t(\underline{d} - D\underline{x}(\underline{u})) = \underline{c}^t\underline{x}(\underline{u}) \geq z^*$ and, since $w^* \leq z^*$, $\underline{x}(\underline{u})$ is an optimal solution of primal (2). $\square$

Note: If only equalities, conditions (*iii*) are automatically satisfied.

Lagrangian Dual: $\max_{\underline{u} \geq \underline{0}} w(\underline{u})$:

concave function

Proposition: Dual function $w(\underline{u})$ is concave.

Proof:

Consider any $\underline{u}_1 \geq \underline{0}$ and $\underline{u}_2 \geq \underline{0}$.

For any α with $0 \leq \alpha \leq 1$, let $\tilde{\underline{x}}$ be an optimal solution of Lagrangian subproblem (3) for $\tilde{\underline{u}} = \alpha \underline{u}_1 + (1-\alpha)\underline{u}_2$, namely $w(\tilde{\underline{u}}) = \underline{c}^t \tilde{\underline{x}} + \tilde{\underline{u}}^t(\underline{d} - D\tilde{\underline{x}})$.

By definition of $w(\underline{u})$, we have $w(\underline{u}_1) \leq \underline{c}^t \tilde{\underline{x}} + \underline{u}_1^t(\underline{d} - D\tilde{\underline{x}})$ and $\longleftarrow \cdot \alpha$

$$w(\underline{u}_2) \leq \underline{c}^t \tilde{\underline{x}} + \underline{u}_2^t(\underline{d} - D\tilde{\underline{x}}). \quad \longleftarrow \cdot (1-\alpha)$$

Multipying the first inequality by α and the second one by $1-\alpha$, we obtain

$$\alpha w(\underline{u}_1) + (1-\alpha) w(\underline{u}_2) \leq \underline{c}^t \tilde{\underline{x}} + (\alpha \underline{u}_1 + (1-\alpha)\underline{u}_2)^t(\underline{d} - D\tilde{\underline{x}}) = w(\alpha \underline{u}_1 + (1-\alpha)\underline{u}_2).$$

$\forall \underline{u}_1 \geq \underline{0}, \ \forall \underline{u}_2 \geq \underline{0}, \ \forall \alpha \in [0,1]$

□

Illustration: piecewise concave, piecewise linear:

3.9.1 Strength and choice of the Lagrangian dual

How can we chose which one are the complicated constraints and which ones should we lagrangianize? We would like to lagrangianize the constraints that provides the tightest bound if we solve the lagrangian dual.

Let's first focus on the strength of the lagrangian dual.

Characterization in terms of an LP.

Theorem: Generic ILP

$$\min \{\underline{c}^t \underline{x} \ : \ A\underline{x} \geq \underline{b}, \ D\underline{x} \geq \underline{d}, \ \underline{x} \in \mathbb{Z}^n\}$$

with integer coefficients.

Let $w(\underline{u}) = \min \{\underline{c}^t \underline{x} + \underline{u}^t(\underline{d} - D\underline{x}) \ : \ A\underline{x} \geq \underline{b}, \ \underline{x} \in \mathbb{Z}^n\}$ dual function,
$w^* = \max_{\underline{u} \geq \underline{0}} w(\underline{u})$ optimal value of Lagrangian dual and $X = \{\underline{x} \in \mathbb{Z}^n \ : \ A\underline{x} \geq \underline{b}\}$,
then

$$w^* = \min \{\underline{c}^t \underline{x} \ : \ D\underline{x} \geq \underline{d}, \underline{x} \in conv(X)\}.$$

"Convexification" of X.

Corollary 1: Since $conv(X) \subseteq \{\underline{x} \in \mathbb{R}^n \ : \ A\underline{x} \geq \underline{b}\}$,

$$z_{LP} = \min \{\underline{c}^t \underline{x} \ : \ A\underline{x} \geq \underline{b}, D\underline{x} \geq \underline{d}, \underline{x} \in \mathbb{R}^n\} \leq w^* \leq z^*. \quad \Longrightarrow$$

the lagrangian duality solution is better than the LP relaxation solution

These inequalities can be strict, i.e. $z_{LP} < w^* < z^*$.

Illustration: from D. Bertsimas, R. Weismantel, Optimization over integers, Dynamic Ideas, 2005, p. 144-146 (latest pages)

$$\begin{array}{llll} \min & 3x_1 - x_2 & & \\ s.t. & x_1 - x_2 & \geq -1 & (10) \\ & -x_1 + 2x_2 & \leq 5 & (11) \\ & 3x_1 + 2x_2 & \geq 3 & (12) \\ & 6x_1 + x_2 & \leq 15 & (13) \\ & x_1, x_2 & \geq 0 \ \text{integer} & \end{array}$$

$\longleftarrow$ we want to lagrangianize this one

- Represent feasible region, and optimal solutions of ILP and LP relaxation:
  $\underline{x}_{ILP} = (1,2)$ with $z_{ILP} = 1$ and $\underline{x}_{LP} = (1/5, 6/5)$ with $z_{LP} = -3/5$.

- Dualize (10):
  For every $u \geq 0$, Lagrangian subproblem:

$$w(u) = \min_{(x_1,x_2) \in X} 3x_1 - x_2 + u(-1 - x_1 + x_2)$$

  where X is set of all integer solutions of (11)-(13).

- Use theorem to find optimal u^* of Lagrangian dual:

$$w^* = \max_{u \geq 0} w(u)$$

and optimal solution $\underline{x}_D = \underline{x}(u^*)$.

Represent $conv(X)$ and $conv(X) \cap \{(x_1, x_2) \in \mathbb{R}^2 \ : \ x_1 - x_2 \geq -1\}$.

We obtain $\underline{x}_D = (1/3, 4/3)$ with $w^* = -1/3$.

Thus, we have: $z_{LP} = -3/5 < w^* = -1/3 < z_{ILP} = 1$

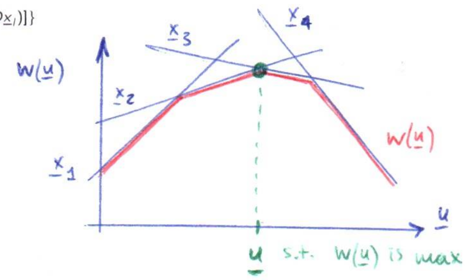
Drawing $w(u)$ it is possibile to verify that $u^* = 5/3$ with $w^* = -1/3$.

Proof: Consider case where $X = \{\underline{x}_1, \ldots, \underline{x}_k\}$ with k finite even though huge.

Lagrangian dual amounts to maximize a nondifferentiable piecewise linear concave function:

$$w^* = \max_{\underline{u}\geq\underline{0}} w(\underline{u}) = \max_{\underline{u}\geq\underline{0}}\{\min_{\underline{x}\in X}[\underline{c}^t\underline{x} + \underline{u}^t(\underline{d} - D\underline{x})]\} = \max_{\underline{u}\geq\underline{0}}\{\min_{1\leq l\leq k}[\underline{c}^t\underline{x}_l + \underline{u}^t(\underline{d} - D\underline{x}_l)]\}$$

and it can be expressed as the LP: →



$$\begin{aligned} w^* &= \max \quad y \\ &\text{s.t.} \quad \underline{c}^t\underline{x}_l + \underline{u}^t(\underline{d} - D\underline{x}_l) \geq y \quad \forall l \in \{1, \ldots, k\} \\ &\qquad \underline{u} \geq \underline{0},\ y \in \mathbb{R}. \end{aligned}$$

Taking its dual and applying strong duality, we obtain:

$$\begin{aligned} w^* = \min \quad & \sum_{l=1}^{k}(\underline{c}^t\underline{x}_l)\mu_l \\ \text{s.t.} \quad & \sum_{l=1}^{k}(D\underline{x}_l - \underline{d})\mu_l \geq \underline{0} \\ & \sum_{l=1}^{k}\mu_l = 1 \\ & \mu_l \geq 0 \quad \forall l \in \{1, \ldots, k\}. \end{aligned}$$

$$\begin{aligned} w^* = \min \quad & \sum_{l=1}^{k}(\underline{c}^t\underline{x}_l)\mu_l \\ \text{s.t.} \quad & \sum_{l=1}^{k}(D\underline{x}_l - \underline{d})\mu_l \geq \underline{0} \\ & \sum_{l=1}^{k}\mu_l = 1 \\ & \mu_l \geq \underline{0} \quad \forall l \in \{1, \ldots, k\}. \end{aligned}$$

Setting $\underline{x} = \sum_{l=1}^{k}\mu_l\underline{x}_l$ with $\sum_{l=1}^{k}\mu_l = 1$ and $\mu_l \geq \underline{0}$ for each l, we have

$$\begin{aligned} w^* = \min \quad & \underline{c}^t\underline{x} \\ \text{s.t.} \quad & D\underline{x} \geq \underline{d} \\ & \underline{x} \in conv(X). \end{aligned}$$

Proof can be extended to the feasible region X of any ILP. □

Illustration $w(\underline{u})$:

concave, piecewise linear
(we cannot use methods for smooth problems)

In some cases Lagrangian relaxation is as weak as LP relaxation.

LP relaxation of X

If conv(X) is equivalent to the LP relaxation of X
$\Rightarrow w^* = z_{LP}$

Corollary 2: If $X = \{\underline{x} \in \mathbb{Z}^n : A\underline{x} \geq \underline{b}\}$ and $conv(X) = \{\underline{x} \in \mathbb{R}^n : A\underline{x} \geq \underline{b}\}$, then

$$w^* = \max_{\underline{u}\geq\underline{0}} w(\underline{u}) = z_{LP} = \min\{\underline{c}^t\underline{x} : A\underline{x} \geq \underline{b}, D\underline{x} \geq \underline{d}, \underline{x} \in \mathbb{R}^n\}.$$

Example: Binary knapsack problem

$$\begin{aligned} \max \quad & z = \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_j x_j \leq b \\ & x_j \in \{0, 1\} \quad \forall j \end{aligned}$$

and its LP relaxation

$$z_{LP-KP} = \max_{\underline{x}\in[0,1]^n}\{\sum_{j=1}^{n} p_j x_j : \sum_{j=1}^{n} a_j x_j \leq b\}.$$

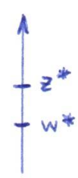
$X = \{\underline{x} \in \{0,1\}^n\}$ and obviously $conv(X) = \{\underline{x} \in [0,1]^n\}$, and $0 \leq x_j \leq 1$ are already contained in LP relaxation.

Corollary 2 implies: $w^* = z_{LP-KP}$. $\Rightarrow$ for the binary knapsack problem the Lagrangian relaxation is as weak as the LP relaxation

Choice of the Lagrangian dual

Primal:
$z^* = \min\{\underline{c}^T\underline{x} : D\underline{x} \geq \underline{d},\ \underline{x} \in X \subseteq \mathbb{R}\}$

Which constraints to relax to get tighter bounds?



Choice **criteria**:

i) strength of the bound w^* obtained by solving Lagrangian dual, = how close will be w^* and z^*

ii) difficulty of solving Lagrangian subproblems

$$w(\underline{u}) = \min\{\underline{c}^t\underline{x} + \underline{u}^t(\underline{d} - D\underline{x}) : \underline{x} \in X \subseteq \mathbb{R}^n\},$$ ← Lagrangian subproblem

iii) difficulty of solving Lagrangian dual: $w^* = \max_{\underline{u}\geq\underline{0}} w(\underline{u})$. ← Dual

For (i) we have the LP characterization (Theorem),

(ii) depends on the specific problem,

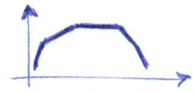
(iii) depends, among others, on the number of dual variables.

We look for a reasonable trade-off.

See **exercise 5.4** for an example.

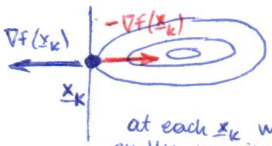
3.9.2 Solution of the Lagrangian duals

$\max_{\underline{u} \geq \underline{0}} w(\underline{u})$, $w(\underline{u})$:



piecewise linear concave function
$\Rightarrow$ not differentiable everywhere
(if we have $\min w(\underline{u})$ then $w(\underline{u})$ is convex)

Gradient method:



at each $\underline{x}_k$ we go on the opposite direction of the gradient
(gradient method for C^1 funct.)

Generalization of the gradient method for C^1 functions to convex piecewise C^1 ones (not everywhere differentiable).

Definition: Let $C \subseteq \mathbb{R}^n$ and $f : C \to \mathbb{R}$ be convex.

- $\underline{\gamma} \in \mathbb{R}^n$ is a *subgradient* of f at $\overline{\underline{x}} \in C$ if

$$f(\underline{x}) \geq f(\overline{\underline{x}}) + \underline{\gamma}^t(\underline{x} - \overline{\underline{x}}) \qquad \forall \underline{x} \in C$$

- the *subdifferential*, denoted by $\partial f(\underline{x})$, is the set of all subgradients of f at $\underline{x}$.

Example: For $f(x) = |x|$, $\gamma = 1$ if $\overline{x} > 0$, $\gamma = -1$ if $\overline{x} < 0$, and $\partial f(\overline{x}) = [-1, 1]$ if $\overline{x} = 0$

Properties:
A convex $f : C \to \mathbb{R}$ has at least one subgradient at each interior point $\overline{\underline{x}}$ of C.
$\underline{x}^*$ is a global minimum of f if and only if $\underline{0} \in \partial f(\underline{x}^*)$.

Generalization of the gradient method:

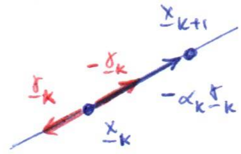
Subgradient method

Given $\min_{\underline{x} \in \mathbb{R}^n} f(\underline{x})$ with $f(\underline{x})$ convex.

- Start from an abitrary $\underline{x}_0$.
- At k-th iteration: consider $\underline{\gamma}_k \in \partial f(\underline{x}_k)$ and set

$$\underline{x}_{k+1} := \underline{x}_k - \alpha_k \underline{\gamma}_k$$

with $\alpha_k > 0$



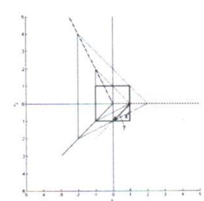
Observation: No 1-D search (optimization) because for nondifferentiable functions a subgradient $\underline{\gamma} \in \partial f(\underline{x})$ is not necessarily a descent direction!

With the gradient method we know that we're optimizing the objecting function if we move in the $-$gradient direction. In the case of not differentiable functions, the subgradient γ is not necessarily a descent direction: in the direction $-\gamma_k$ the obj. funct. may increase.

for example:

Example: $\min_{-1 \leq x_1, x_2 \leq 1} f(x_1, x_2)$ with $f(x_1, x_2) = \max\{-x_1, x_1 + x_2, x_1 - 2x_2\}$

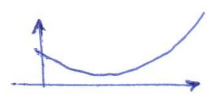
Level curves in brown, points of nondifferentiability in green (of type: $(t, 0)$, $(-t, 2t)$ and $(-t, -t)$ for $t \geq 0$), global minimum $\underline{x}^* = (0, 0)$.



If $\underline{x}_k = (1, 0)$ and we consider $\underline{\gamma}_k = (1, 1) \in \partial f(\underline{x}_k)$, $f(\underline{x})$ increases along $\{\underline{x} \in \mathbb{R}^2 : \underline{x} = \underline{x}_k - \alpha_k \underline{\gamma}_k, \alpha_k \geq 0\}$ but if α_k is sufficiently small then $\underline{x}_{k+1} = \underline{x}_k - \alpha_k \underline{\gamma}_k$ is closer to $\underline{x}^*$.

(From Chapter 8, Bazaraa et al., Nonlinear Programming, Wiley, 2006, p. 436-437)

Theorem:
If f is convex, $\lim_{\|\underline{x}\| \to \infty} f(\underline{x}) = +\infty$, $\lim_{k \to \infty} \alpha_k = 0$ and $\sum_{k=0}^{\infty} \alpha_k = \infty$, the subgradient method terminates after a finite number of iterations with an optimal solution $\underline{x}^*$ or infinite sequence $\{\underline{x}_k\}$ admits a subsequence converging to $\underline{x}^*$.



Condition on the steps length: $\lim_{k \to \infty} \alpha_k = 0$ (we want the steps to be smaller and smaller) and $\sum_{k=0}^{\infty} \alpha_k = \infty$, so the length of α_k's must be a divergent series

Stepsize:
In practice $\{\alpha_k\}$ as above (e.g., $\alpha_k = 1/k$) are too slow.

$\rho :=$ contraction factor $\rightarrow$ An option: $\alpha_k = \alpha_0 \rho^k$, for a given $\rho < 1$. A more popular one (min problems):

$$\alpha_k = \varepsilon_k \frac{f(\underline{x}_k) - \hat{f}}{\|\underline{\gamma}_k\|^2},$$

where $0 < \varepsilon_k < 2$ and $\hat{f}$ is either the optimal value $f(\underline{x}^*)$ or an estimate.

either the optimal value (if we know it) or it is an estimate of it (if we don't know it)

Stopping criterion: prescribed maximum number of iterations (even if $\underline{0} \in \partial f(\underline{x}_k)$ it may non be considered at $\underline{x}_k$).

Need to store the best solution $\underline{x}_k$ found.

Simple extension for bounds (projections).

Subgradient method for Lagrangian dual

$$\max_{\underline{u} \geq \underline{0}} w(\underline{u})$$

where $w(\underline{u}) = \min\{\underline{c}^t\underline{x} + \underline{u}^t(\underline{d} - D\underline{x}) : \underline{x} \in X \subseteq \mathbb{R}^n\}$ is concave and piecewise linear.

Simple characterization of the subgradients of $w(\underline{u})$:

We're looking for $\underline{\gamma} \in \partial w(\tilde{\underline{u}})$

$\underline{x} = \underline{x}(\tilde{\underline{u}})$

Proposition:
Consider $\tilde{\underline{u}} \geq \underline{0}$ and $X(\tilde{\underline{u}}) = \{\underline{x} \in X : w(\tilde{\underline{u}}) = \underline{c}^t\underline{x} + \tilde{\underline{u}}^t(\underline{d} - D\underline{x})\}$ set of optimal solutions of Lagrangian subproblem (3).
Then

- For each $\underline{x}(\tilde{\underline{u}}) \in X(\tilde{\underline{u}})$, the vector $(\underline{d} - D\underline{x}(\tilde{\underline{u}})) \in \partial w(\tilde{\underline{u}})$.
- Each subgradient of $w(\underline{u})$ at $\tilde{\underline{u}}$ can be expressed as a convex combination of subgradients $(\underline{d} - D\underline{x}(\tilde{\underline{u}}))$ with $\underline{x}(\tilde{\underline{u}}) \in X(\tilde{\underline{u}})$.

in this way we obtain the generating vectors of the convex set of all the subgradients at $\tilde{\underline{u}}$.

Procedure:

1) Select initial $\underline{u}_0$ and set $k := 0$.

2) Solve Lagrangian subproblem

$$w(\underline{u}_k) = \min\{\underline{c}^t\underline{x} + \underline{u}_k^t(\underline{d} - D\underline{x}) : \underline{x} \in X\}.$$

If $\underline{x}(\underline{u}_k)$ optimal solution found, $(\underline{d} - D\underline{x}(\underline{u}_k))$ is a subgradient of $w(\underline{u})$ at $\underline{u}_k$.

3) Update Lagrange multipliers:

$$\underline{u}_{k+1} = \max\{\underline{0}, \underline{u}_k + \alpha_k(\underline{d} - D\underline{x}(\underline{u}_k))\}$$

with, for instance, $\alpha_k = \varepsilon_k \frac{\hat{w} - w(\underline{u}_k)}{\|\underline{d} - D\underline{x}(\underline{u}_k)\|^2}$, where $\hat{w}$ is an estimate of optimal value w^*.

4) Set $k := k + 1$

(General) issues:
- how to pick the step length
- how to estimate the optimal value ($\hat{w}$)

here we have $\max\{\underline{0}, ..\}$ because we have $\underline{u} \geq \underline{0}$ constr.
Moreover we have:
$\underline{u}_{k+1} = \underline{u}_k + \alpha_k \underline{\gamma}_k$
because we're maximizing: $\max_{\underline{u} \geq \underline{0}} w(\underline{u})$

here we have $\hat{w} - w(\underline{u}_k)$ and not the opposite since we're maximizing

3.9.3 Lagrangian relaxation for the STSP (Held & Karp)

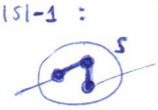
Symmetric TSP: Given undirected $G = (V, E)$ with a cost $c_e \in \mathbb{Z}^+$ for each $e \in E$, determine a Hamiltonian cycle of minimum total cost.

$$\begin{array}{lllr}
\min & \sum_{e \in E} c_e x_e & & \\
s.t. & \boxed{\sum_{e \in \delta(i)} x_e = 2 \quad \forall i \in V} & & (14) \\
& \sum_{e \in E(S)} x_e \leq |S| - 1 & \forall S \subseteq V, 2 \leq |S| \leq n - 1 & (15) \\
& x_e \in \{0, 1\} & \forall e \in E &
\end{array}$$

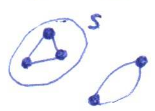
where $E(S) = \{\{i, j\} \in E : i \in S, j \in S\}$

for each node i we select two incident edges:

SEC: subtours elimination constraints
= for each subsets of nodes we should not select more edges than $|S| - 1$:



If we have subcycles then they must be at least 2:



inside and outside of S

$\Rightarrow$ if we specify the non-subcycles condition for S then it's automatically respected for V/S
(Remember
SEC for S satisfied $\Leftrightarrow$ SEC for $V \setminus S$ satisfied)

Observations:

i) Due to (14), half of the (15) are redundant:
$\sum_{e \in E(S)} x_e \leq |S| - 1$ iff $\sum_{e \in E(\bar{S})} x_e \leq |\bar{S}| - 1$, where $\bar{S} = V \setminus S$.
Thus all (15) with $1 \in S$ can be deleted. (we can take $\forall$node, we chose node 1 w.l.o.g.)

ii) Summing over all (14) and dividing by 2, we obtain $\sum_{e \in E} x_e = n$ that can be added.

which means that we should select exactly n edges

Recall: a Hamiltonian cycle is a 1-tree (i.e., a spanning tree on nodes $\{2, \ldots, n\}$ plus two edges incident to node 1) in which all nodes have exactly two incident edges.

Since $\sum_{e \in E} c_e x_e + \boxed{\sum_{i \in V} u_i(2 - \sum_{e \in \delta(i)} x_e)} = \sum_{e = \{i, j\} \in E}(c_e - u_i - u_j)x_e + 2\sum_{i \in V} u_i$,
relaxing the degree constraints (14) for all nodes except node 1,
Lagrangian subproblem:

$$\begin{array}{llll}
w(\underline{u}) = \min & \sum_{e \in E}(c_e - u_i - u_j)x_e + 2\sum_{i \in V} u_i & \\
s.t. & \sum_{e \in \delta(1)} x_e = 2 & \\
& \sum_{e \in E(S)} x_e \leq |S| - 1 & \forall S \subseteq V, 2 \leq |S| \leq n - 1, 1 \notin S \\
& \sum_{e \in E} x_e = n & \\
& x_e \in \{0, 1\} & \forall e \in E
\end{array}$$

can be solved to optimality with greedy algorithms

(we said it's enough to consider only half of the SEC)

where $u_1 = 0$ and $E(S) = \{\{i, j\} \in E : i \in S, j \in S\}$.

Note: Set of feasible solutions $\equiv$ set of all 1-trees.

Lagrangian dual:

$$\max_{\underline{u} \in \mathbb{R}^{|V|} : u_1 = 0} w(\underline{u})$$

can be solved with the subgradient method

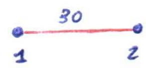
Procedure :

- $\underline{u}^k$
- $c_{ij}^k = c_{ij} - u_i^k - u_j^k$
- $\underline{x}(\underline{u}^k)$ (through C)
- $\underline{\gamma}^k = \left(2 - \sum_{e \in \delta(i)} x_e^k\right)_i$
- $\underline{u}^{k+1}$

Example: taken from L. Wolsey, Integer Programming, p. 175-177

Undirected $G = (V, E)$ with 5 nodes and cost matrix:

(for example) $c_{12} = 30$



$$C = \begin{pmatrix} - & 30 & 26 & 50 & 40 \\ - & - & 24 & 40 & 50 \\ - & - & - & 24 & 26 \\ - & - & - & - & 30 \\ - & - & - & - & - \end{pmatrix}$$

$\underline{u}_0 = \underline{0}$

Dual function:

$$w(\underline{u}^k) = \min \left\{ \sum_{e=\{i,j\} \in E} (c_e - u_i^k - u_j^k) x_e^k + 2 \sum_{i \in V} u_i^k : \underline{x}^k \text{ incidence vector of a 1-tree} \right\}$$

Notation: $c_{ij}^k = c_e - u_i^k - u_j^k$ for $e = \{i, j\} \in E$

Subgradient $\underline{\gamma}^k$ with $\gamma_i^k = (2 - \sum_{e \in \delta(i)} x_e^k)$, where $\underline{x}^k = \underline{x}(\underline{u}^k)$ is an optimal solution of Lagrangian subproblem at k-th iteration.

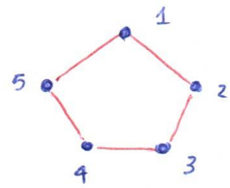
Since $\sum_{e \in \delta(1)} x_e = 2$ is not relaxed, $\gamma_1^k = 0$ for all k.
Starting from $u_1^0 = 0$ we then have $u_1^k = 0$ for all $k \geq 1$.

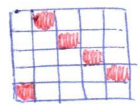
that's why we can write $\sum_{i \in V} u_i^k$ without removing node 1 (because it'll always be zero)

First thing: find an itial feasible solution

Feasible solution of cost 148 found with primal heuristic:

$$x_{12} = x_{23} = x_{34} = x_{45} = x_{51} = 1 \text{ and } x_{ij} = 0 \text{ for all other } \{i, j\} \in E$$



$C =$ 

Solution of Lagrangian dual starting from $\underline{u}^0 = \underline{0}$ with $\varepsilon = 1$:

Solving Lagrangian subproblem with costs:

$c_{ij}^k = c_{ij} - u_i - u_j$

$$c^0 = c = \begin{pmatrix} - & 30 & 26 & 50 & 40 \\ - & - & 24 & 40 & 50 \\ - & - & - & 24 & 26 \\ - & - & - & - & 30 \\ - & - & - & - & - \end{pmatrix}$$

($c_e^0 = c_e$ for each $e \in E$ since $\underline{u}^0 = \underline{0}$),

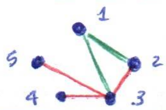
we find $\underline{x}(\underline{u}^0)$ corresponding to 1-tree of cost 130:

$$x_{12} = x_{13} = x_{23} = x_{34} = x_{35} = 1 \text{ and } x_{ij} = 0 \text{ for all other } \{i, j\} \in E$$

Find the min cost 1-tree on $\{2, .., 5\}$ + 2 min cost edges incidents to 1

$$C = \begin{bmatrix} \cancel{-} & \cancel{30} & \cancel{26} & \cancel{50} & \cancel{40} \\ - & - & 24 & 40 & 50 \\ - & - & - & 24 & 26 \\ - & - & - & - & 30 \\ - & - & - & - & - \end{bmatrix}$$

Not considering the first row (since we're looking for the 1-tree on $\{2, .., 5\}$) we pick the min costs till we not create a cycle



+ 2 minimum cost edges incidents in 1 : $[-\ 30\ 26\ 50\ 90]$

OPTIMAL SOLUTION $\underline{x}(\underline{u}^0)$

Knowing $\underline{x}(\underline{u}^0)$, we can compute $w(\underline{u}^0) = 130 + 0$ (cost of 1-tree + $2 \sum_{i \in V} u_i^0$).

Now, from here we should derive the subgradient (by plugging $\underline{x}(\underline{u}^0)$ in the violation part)

Subgradient

$$\underline{\gamma}^0 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 1 \\ 1 \end{pmatrix}$$

Update Lagrange multipliers:

$$\underline{u}^1 = \underline{u}^0 + \frac{(\hat{w} - w(\underline{u}^0))}{\|\underline{\gamma}_0\|^2} \underline{\gamma}^0 = \underline{0} + \frac{(148 - 130)}{6} \begin{pmatrix} 0 \\ 0 \\ -2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -6 \\ 3 \\ 3 \end{pmatrix}$$

we use the objective function value of the best feasible solution that we found so far (148)

Since

$$c^0 = \begin{pmatrix} - & 30 & 26 & 50 & 40 \\ - & - & 24 & 40 & 50 \\ - & - & - & 24 & 26 \\ - & - & - & - & 30 \\ - & - & - & - & - \end{pmatrix}$$

we have

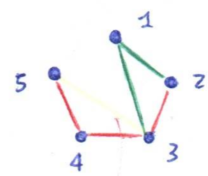
$$c^1 = \begin{pmatrix} - & 30 & 32 & 47 & 37 \\ - & - & 30 & 37 & 47 \\ - & - & - & 27 & 29 \\ - & - & - & - & 24 \\ - & - & - & - & - \end{pmatrix}$$

$$\rightarrow C = \begin{bmatrix} \cancel{-} & \cancel{30} & \cancel{32} & \cancel{47} & \cancel{37} \\ - & - & 30 & 37 & 47 \\ - & - & - & 27 & 29 \\ - & - & - & - & 24 \\ - & - & - & - & - \end{bmatrix}$$

As optimal solution $\underline{x}(\underline{u}^1)$ of Lagrangian subproblem with matrix C^1 we find 1-tree of cost 143:

$$x_{12} = x_{13} = x_{23} = x_{34} = x_{45} = 1 \text{ and } x_{ij} = 0 \text{ for all other } \{i, j\} \in E$$

and $w(\underline{u}^1) = 143 + 2 \sum_{i \in V} u_i^1 = 143$.



we cannot take this (even if it is 29 < 30 because otherwise we would create a cycle)

Since

$$\underline{\gamma}^1 = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 1 \end{pmatrix},$$

we have

$$\underline{u}^2 = \underline{u}^1 + \frac{(\hat{w} - w(u^1))}{\|\underline{\gamma}_1\|^2} \underline{\gamma}^1 = \begin{pmatrix} 0 \\ 0 \\ -6 \\ 3 \\ 3 \end{pmatrix} + \frac{(148 - 143)}{2} \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{-17}{2} \\ 3 \\ \frac{11}{2} \end{pmatrix}$$

Therefore

$$c^2 = \begin{pmatrix} - & 30 & 34.5 & 47 & 34.5 \\ - & - & 32.5 & 37 & 44.5 \\ - & - & - & 29.5 & 29 \\ - & - & - & - & 21.5 \\ - & - & - & - & - \end{pmatrix}$$

and we obtain $\underline{x}(\underline{u}^2)$ that corresponds to 1-tree of cost 147.5:

$$x_{12} = x_{15} = x_{23} = x_{35} = x_{45} = 1 \text{ and } x_{ij} = 0 \text{ for all other } \{i,j\} \in E$$

and $w(\underline{u}^2) = 147.5 + 0$.

Since all costs c_e are integer, the feasible solution of cost 148 found by the heuristic is optimal.

148 — z^*
147.5 — $w(\underline{u}^2)$
143 — $w(\underline{u}^1)$
130 — $w(\underline{u}^0)$

$z^* \geq w(\underline{u}) \ \forall \underline{u}$
$z^* \in \mathbb{Z}_+$

this means that any feasible solution of the original symmetric TSP have a objective function value ≥ 147.5
and since all the costs are integer the total cost will be integer
$\Rightarrow$ the closest integer lower bound is 148 $\rightarrow$ upper = lower = 148

Heuristic upper bound
Lagrangian lower bounds

148 (upper bound)
147.5 (2^{nd} iteration)
143 (1^{st} iteration)
130 (0^{th} iteration)

3.10 Column generation method

Many relevant decision-making problems can be formulated as ILP problems with a very large (exponential) number of variables.

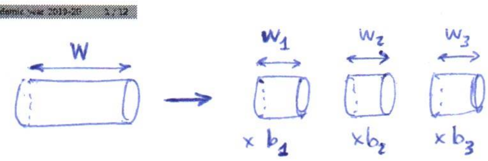
(Previously we discussed a way to deal with large (exponential) number of constraints: cutting plane methods)

Examples: cutting stock, crew scheduling, vehicle routing, combinatorial auctions, multicommodity flows,...

General idea:

- enumerate all partially feasible solutions and represent any additional constraints in a set covering/packing/partitionning type of formulation. (which will include a very large number of variables since we've enumerating partially feasible solutions and we have a variable for each partially feasible solution)
- do not consider all variables explicitly, new variables are generated when needed.

Example: 1-D cutting stock problem



A paper company produces large rolls of width W.

Demand: b_i small rolls of width w_i ($w_i \leq W$), $i \in I = \{1, \ldots, m\}$.

Smaller rolls are obtained by cutting large rolls according to certain patterns.

E.g., if $W = 15$, $w_1 = 6$ and $w_2 = 2$, feasible patterns $\binom{2}{1}$ with a waste of 1.

another feasible patterns : $\binom{0}{7}, \binom{1}{4}$

Given

- large rolls of width W,
- demands for b_i small rolls of width w_i, with $i \in I$

decide how to cut large rolls into small rolls so as to minimize the number of large rolls used, while satisfying demand.

Illustration:

NP-hard problem = difficult problem

Classical ILP formulation (Kantorovich)

K is the index set of the large rolls (we have K rolls available (large rolls))

x_i^k = number of times i-th small roll is cut in k-th large roll, $i \in I$, $k \in K$ = for example : x_4^2 = how many rolls of width w_4 are cut from the 2nd large roll

$y_k = 1$ if k-th large roll is cut, with $k \in K$

$$\begin{array}{lll} z_{K-ILP} = \min & \sum_{k \in K} y_k & \\ \text{s.t.} & \sum_{k \in K} x_i^k \geq b_i & \forall i \in I = \{1, \ldots, m\} \quad \text{(demand)} \\ (*) & \sum_{i \in I} w_i x_i^k \leq W y_k & \forall k \in K \quad \text{(width of large roll)} \\ & \boxed{x_i^k \in \mathbb{Z}_+, \ y_k \in \{0,1\} \quad \forall i \in I, k \in K} & \end{array}$$

$\rightarrow$ relaxation : $x_i^k \geq 0$, $0 \leq y_i \leq 1$

Very weak formulation

Trivial LP relaxation bound:

$$z_{K-LP} = \sum_{k \in K} y_k = \sum_{k \in K} \sum_{i \in I} \frac{w_i x_i^k}{W} = \sum_{i \in I} w_i \sum_{k \in K} \frac{x_i^k}{W} = \frac{\sum_{i=1}^m w_i b_i}{W}$$

(*)

$x_3 = 4 \Rightarrow$ 4 large rolls must be cut with pattern 3

Set covering ILP formulation (Gilmore and Gomory) : it's like we'll enumerate all the possible patterns

Note that the constraints related to the compatibility of the width are included in the possible patterns : if a pattern j exists then it respect the constraints about compatibility

x_j = number of large rolls cut according to j-th pattern, $1 \leq j \leq n$ (in the previous example some patterns were: $\binom{2}{1}, \binom{0}{7}, \binom{1}{4}$)

$$
\begin{array}{lll}
z_{ILP} = \min & \sum_{j=1}^{n} x_j & \\
s.t. & \sum_{j=1}^{n} a_{ij} x_j \geq b_i & \forall i \in I = \{1, \ldots, m\} \quad \text{(demand)} \\
& x_j \in \mathbb{Z}_+ & \forall j \in J = \{1, \ldots, n\}
\end{array}
$$

where a_{ij} is number of small rolls of width w_i in j-th cutting pattern.

Number n of variables (patterns) grows exponentially with number m of rows (types of small rolls).

! Observations: $\rightarrow x_j \geq 0$

- at LP optimality at most m of the n variables have nonzero value; since $m \ll n$ only a very small subset of them (columns) is needed.
- for large integer b_is, rounding an optimal solution of LP relaxation leads to satisfactory integer solutions,

$\Rightarrow$ it seems a good idea to solve the LP problem

Column generation scheme

Idea: no need to include all variables a priori, new variables are generated when needed.

Main steps:

1) consider LP relaxation of original ILP, choose initial subset of variables $J_0 \subseteq J$, and set $k = 0$,

should be rich enough to guarantee at least one feasible solution

2) solve *LP Restricted Master problem* (LPRM) with subset J_k,

"restricted" because we only consider J_k variables "master" because we'll define auxiliary problems to optimize the master problem

3) solve *pricing subproblem* for LPRM with J_k to search for an improving non basic variable x_l (with negative reduced cost if min problem) and the associated column,

auxiliary problem

4) if $\exists$ such x_l, update $J_{k+1} := J_k \cup \{l\}$, set $k := k + 1$ and goto (2); otherwise LPRM optimal solution is also optimal for LP relaxation of original ILP.

If $\nexists$ then the optimal solution of the LP restricted master problem is an optimal solution also of the LP relaxation of the original problem and so we're done.

Observation: Column generation (CG) yields an optimal solution of LP relaxation of ILP and hence a **bound** on optimal ILP solution value. !

Example cont.: 1-D cutting stock problem

LP relaxation of Master problem (LPM):

$$
\begin{array}{lll}
z_{LPM} = \min & \sum_{j=1}^{n} x_j & \\
s.t. & \sum_{j=1}^{n} a_{ij} x_j \geq b_i & \forall i \in I = \{1, \ldots, m\} \\
& x_j \geq 0 & \forall j \in J = \{1, \ldots, n\}.
\end{array}
$$

When solving with Simplex method:

- Since $\underline{\bar{c}}_N = \underline{c}_N - \underline{c}_B^t B^{-1} N$, reduced cost of non basic variable x_j is $\boxed{\bar{c}_j = 1 - \sum_{i=1}^{m} a_{ij} y_i}$ where $\underline{y} = \underline{c}^t B^{-1}$ is (complementary) dual solution.
- Current basic feasible solution is optimal if $\bar{c}_j \geq 0$ for all non basic variables.

$\bar{c}_j = 1 - \sum_{i=1}^{m} a_{ij} y_i$

Dual of LPM:

$$
\begin{array}{lll}
\max & \sum_{i=1}^{m} b_i y_i & \\
s.t. & \boxed{\sum_{i=1}^{m} a_{ij} y_i \leq 1} & \forall j \in J = \{1, \ldots, n\} \\
& y_i \geq 0 & \forall i \in I = \{1, \ldots, m\}.
\end{array}
$$

$\rightarrow \underline{\bar{c}} \geq \underline{0}$

Start with **LP Restricted Master problem** (LPRM) with subset $J_0 \subset J = \{1, \ldots, n\}$ of patterns, guaranteeing a feasible solution.

LPRM with J_0:

$$
\begin{array}{lll}
z_{LPRM} = \min & \sum_{j=1}^{n} x_j & \\
s.t. & \sum_{j \in J_0} a_{ij} x_j \geq b_i & \forall i \in I = \{1, \ldots, m\} \\
& x_j \geq 0 & \forall j \in J_0.
\end{array}
$$

Reduced cost of non basic variable x_j is still $\bar{c}_j = 1 - \sum_{i=1}^{m} a_{ij} y_i$.

Dual of LPRM with J_0:

$$
\begin{array}{lll}
\max & \sum_{i=1}^{m} b_i y_i & \\
s.t. & \sum_{i=1}^{m} a_{ij} y_i \leq 1 & \forall j \in J_0 \\
& y_i \geq 0 & \forall i \in I = \{1, \ldots, m\}.
\end{array}
$$

Let $\underline{x}^*$ and $\underline{y}^*$ be optimal solutions of LPRM and its dual, respectively.

Systematic search for new "improving" non basic variables (columns/patterns):

Look for a non basic variable with smallest reduced cost and corresponding pattern $\underline{\alpha} \in \mathbb{Z}_+^m$ by solving the **pricing subproblem**:

$$\begin{array}{lll} \min & \overline{c} = 1 - \sum_{i=1}^{m} y_i^* \alpha_i & \\ s.t. & \sum_{i=1}^{m} w_i \alpha_i \leq W & \\ & \alpha_i \in \mathbb{Z}_+ & \forall i \in I = \{1, \dots, m\} \end{array} \quad (1)$$

Integer Knapsack problem that can be solved in $O(mW)$ using Dynamic Programming.

Two cases:

- if $\overline{c}^* \geq 0$ then the optimal solution of current LPRM is also optimal for LP relaxation of original ILP,
- adding to current LPRM any non basic variable associated to a cutting pattern $\underline{\alpha} \in \mathbb{Z}_+^m$ with $\overline{c} < 0$, improves (decreases) the objective function value.

$\left(\begin{array}{l} LPRM \rightarrow J_k \\ LP \rightarrow J \end{array} \right)$

Example:

1-D Cutting stock instance with $W = 3.9$ m, $\underline{w} = \begin{pmatrix} 1.25 \\ 1 \\ 0.8 \end{pmatrix}$ and $\underline{b} = \begin{pmatrix} 35 \\ 171 \\ 133 \end{pmatrix}$.

Initial patterns: $A_1 = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$ waste of 0.05, $A_2 = \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix}$ waste of 0.5,

$A_3 = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$ waste of 0.6, $A_4 = \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix}$ waste of 0.9

From J. Lundgren, M. Rönnqvist, P. Värbrand, Optimization, Studentlitteratur AB, Lund, Sweden, 2010.

- LP Restriced Master problem:

$$\begin{array}{ll} \min & z = \sum_{j=1}^{4} x_j \\ s.t. & \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} x_1 + \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix} x_2 + \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} x_3 + \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix} x_4 \geq \begin{pmatrix} 35 \\ 171 \\ 133 \end{pmatrix} \\ & x_j \geq 0 \qquad \forall j \in J_0 = \{1, 2, 3, 4\} \end{array}$$

Optimal solution of LPRM: $\underline{x}^* = (35, 21, 0, 38.33)^t$ with value $z^* = 94.33$ $\Longrightarrow$ for sure we don't need more than 95 rolls to satisfy the demand

Optimal dual solution: $\underline{y}^* = (\frac{2}{9}, \frac{1}{3}, \frac{2}{9})^t$

- Pricing subproblem:

$$\begin{array}{lll} \min & \overline{c} = 1 - (\frac{2}{9}\alpha_1 + \frac{1}{3}\alpha_2 + \frac{2}{9}\alpha_3) & \\ s.t. & 1.25\alpha_1 + 1.0\alpha_2 + 0.8\alpha_3 \leq 3.9 & \\ & \alpha_1, \alpha_2, \alpha_3 \geq 0 & \text{integer} \end{array}$$

Optimal solution (integer knapsack): $\underline{\alpha}^* = (0, 3, 1)^t$ with value $\overline{c} = -\frac{2}{9}$.

- Since $\overline{c} < 0$, adding new pattern $A_5 = (0, 3, 1)^t$ will improve (decrease) the objective function value.

Optimal solution of LPRM with $J_1 = \{1, 2, 3, 4, 5\}$: $\underline{x}^* = (35, 6.625, 0, 0, 43.125)^t$ with value $z^* = 84.75$.

Optimal dual solution: $\underline{y}^* = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4})^t$

- Pricing subproblem:

$$\begin{array}{lll} \min & \overline{c} = 1 - (\frac{1}{4}\alpha_1 + \frac{1}{4}\alpha_2 + \frac{1}{4}\alpha_3) & \\ s.t. & 1.25\alpha_1 + 1.0\alpha_2 + 0.8\alpha_3 \leq 3.9 & \\ & \alpha_1, \alpha_2, \alpha_3 \geq 0 & \text{integer} \end{array}$$

with optimal solution $\underline{\alpha}^* = (0, 3, 1)^t$ (as before!) and $\overline{c} = 0$.

Thus $\underline{x}^* = (35, 6.625, 0, 0, 43.125)^t$ is an optimal sol. of LP relaxation of original ILP.

Rounding up: integer solution $\underline{x} = (35, 7, 0, 0, 44)^t$ with $z = 86$.

Since $z_{LPM} = 84.75$, lower bound is 85.

Optimal ILP solution: $\underline{x}_{ILP} = (36, 6, 0, 0, 43)^t$ with $z_{ILP} = 85$.

General remarks

- Initial set of columns (indexed by J_0) has a strong impact: rich enough to guarantee initial feasible solution but not too large to reduce computational load.
- Use heuristics for pricing subproblem as long as an improving variable (column) is found. Exact method only to certify that LPRM solution is also optimal for LPM.
- CG methods can be viewed as cutting plane methods to solve dual of LPM.
- Strong practical impact of CG due to great flexibility to model complicated restrictions.
- To find an optimal solution of original ILP, CG can be embedded in a Branch-and-Bound framework $\Rightarrow$ **Branch-and-Price method**.

Computer lab 3 devoted to a Column Generation approach to the airline crew pairing problem.

Therefore, the Lagrangean dual is equivalent to and has the same optimal value as the problem

$$\begin{array}{ll} \text{maximize} & \min\limits_{k\in K}\left(\mathbf{c}'\mathbf{x}^k + \boldsymbol{\lambda}'(\mathbf{b}-\mathbf{A}\mathbf{x}^k)\right) \\ \text{subject to} & (\mathbf{c}'-\boldsymbol{\lambda}'\mathbf{A})\mathbf{w}^j \geq 0, \qquad j\in J, \\ & \boldsymbol{\lambda}\geq \mathbf{0}, \end{array}$$

or equivalently,

$$\begin{array}{llll} \text{maximize} & y & & \\ \text{subject to} & y + \boldsymbol{\lambda}'(\mathbf{A}\mathbf{x}^k - \mathbf{b}) & \leq \mathbf{c}'\mathbf{x}^k, & k\in K, \\ & \boldsymbol{\lambda}'\mathbf{A}\mathbf{w}^j & \leq \mathbf{c}'\mathbf{w}^j, & j\in J, \\ & \boldsymbol{\lambda}\geq \mathbf{0}. & & \end{array}$$

Taking the dual of the above problem, and using strong duality, we obtain that Z_{D} is equal to the optimal cost of the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{c}'\left(\sum\limits_{k\in K}\alpha_k\mathbf{x}^k + \sum\limits_{j\in J}\beta_j\mathbf{w}^j\right) \\ \text{subject to} & \sum\limits_{k\in K}\alpha_k = 1, \\ & \mathbf{A}\left(\sum\limits_{k\in K}\alpha_k\mathbf{x}^k + \sum\limits_{j\in J}\beta_j\mathbf{w}^j\right)\geq \mathbf{b}, \\ & \alpha_k,\beta_j\geq 0, \qquad k\in K,\ j\in J. \end{array}$$

Since

$$\mathrm{conv}(\mathcal{F}) = \left\{\sum_{k\in K}\alpha_k\mathbf{x}^k + \sum_{j\in J}\beta_j\mathbf{w}^j \;\middle|\; \sum_{k\in K}\alpha_k = 1,\ \alpha_k,\beta_j\geq 0,\ k\in K,\ j\in J\right\},$$

the result follows. □

Example 4.3 (Illustration of Lagrangean relaxation)
Consider the problem

$$\begin{array}{lrcrcr} \text{minimize} & 3x_1 & - & x_2 & & \\ \text{subject to} & \boxed{x_1 & - & x_2 & \geq & -1,} \\ & -x_1 & + & 2x_2 & \leq & 5, \\ & 3x_1 & + & 2x_2 & \geq & 3, \\ & 6x_1 & + & x_2 & \leq & 15, \\ & x_1, x_2\geq 0, & & & & \\ & x_1, x_2\in\mathbb{Z}. & & & & \end{array}$$

this is the one
that we want to
Lagrangianize

We relax the first constraint $x_1 - x_2$
vectors that satisfy the remaining con
then

$$\mathcal{F} = \{(1,0)', (2,0)', (1,1)', (2,1)$$

For $\lambda\geq 0$, we have

$$Z(\lambda) = \min_{(x_1,x_2)'\in\mathcal{F}}\Big(3x$$

which is plotted in Figure 4.3.

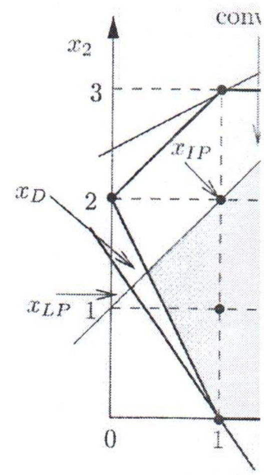


Figure 4.2: The points show
of $\mathcal{F}$ is the set outlined by the t
represents the intersection of c
satisfy $x_1 - x_2 \geq -1$. The op
$\mathbf{x}_{\mathrm{D}} = (1/3, 4/3)'$, and its cost
the optimal solution to the lin
$(1/5, 6/5)'$, resulting in a lower
solution to the integer optimiz
$Z_{\mathrm{IP}} = 1$. Note that $Z_{\mathrm{LP}} < Z_{\mathrm{D}}$

Since there are nine points in $\mathcal{F}$,
linear functions:

$$3-2\lambda,\ 6-3\lambda,\ 2-\lambda,\ 5$$

In Figure 4.3, we plot the function $Z($

$$Z(\lambda) = \begin{cases} -2 + \\ 3 - 2 \\ 6 - 3 \end{cases}$$

uivalent to and has the same optimal

$^{k} + \lambda'(\mathbf{b} - \mathbf{A}\mathbf{x}^k))$

$\lambda)\mathbf{w}^j \geq 0, \qquad j \in J,$

$\mathbf{b}) \leq \mathbf{c}'\mathbf{x}^k, \qquad k \in K,$

$\leq \mathbf{c}'\mathbf{w}^j, \qquad j \in J,$

and using strong duality, we obtain
f the problem

$$\mathbf{x}^k + \sum_{j\in J} \beta_j \mathbf{w}^j \Big)$$

$$\mathbf{x}^k + \sum_{j\in J} \beta_j \mathbf{w}^j \Big) \geq \mathbf{b},$$

$$k \in K,\ j \in J.$$

optimal solution that we find by solving the lagrangian dual (closer than the optimal obtained with the LP relaxation)

$$\alpha_k = 1,\ \alpha_k, \beta_j \geq 0,\ k \in K,\ j \in J \Big\},$$

□

gean relaxation)

x_2
$x_2 \geq -1,$
$2x_2 \leq 5,$
$2x_2 \geq 3,$
$x_2 \leq 15,$
$\geq 0,$
$\in \mathbb{Z}.$

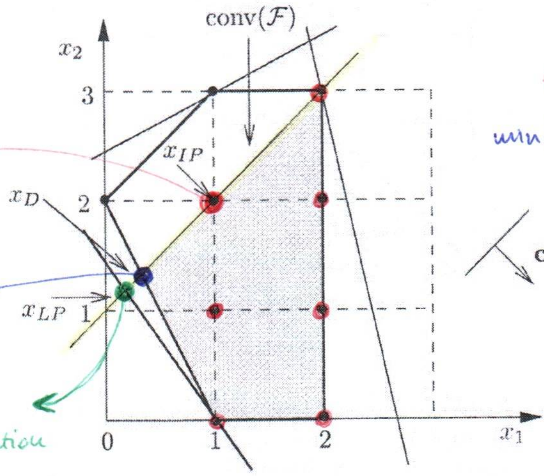
We relax the first constraint $x_1 - x_2 \geq -1$, and we let $\mathcal{F}$ be the set of integer vectors that satisfy the remaining constraints. The set $\mathcal{F}$, shown in Figure 4.2, is then

$$\mathcal{F} = \{(1,0)', (2,0)', (1,1)', (2,1)', (0,2)', (1,2)', (2,2)', (1,3)', (2,3)'\}.$$

For $\lambda \geq 0$, we have

$$Z(\lambda) = \min_{(x_1,x_2)' \in \mathcal{F}} \Big(3x_1 - x_2 + \lambda(-1 - x_1 + x_2)\Big),$$

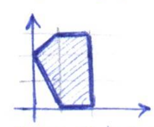
which is plotted in Figure 4.3.



• = X
feasible region

min $\{\underline{c}^T\underline{x} : D\underline{x} \geq \underline{d},$ conv(**X**)$\}$

where
$X = \{\underline{x} \in \mathbb{Z}^m : A\underline{x} \geq \underline{b}\}$
So, conv(X) :



(since X has not $D\underline{x} \geq \underline{d}$)

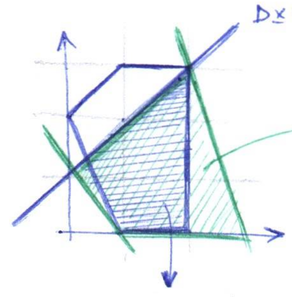
Figure 4.2: The points shown are elements of $\mathcal{F}$. The convex hull of $\mathcal{F}$ is the set outlined by the thicker lines. The shaded polyhedron represents the intersection of conv($\mathcal{F}$) with the set of vectors that satisfy $x_1 - x_2 \geq -1$. The optimal solution to problem (4.27) is $\mathbf{x}_D = (1/3, 4/3)'$, and its cost Z_D is equal to $-1/3$. Note that the optimal solution to the linear relaxation is the vector $\mathbf{x}_{LP} = (1/5, 6/5)'$, resulting in a lower bound $Z_{LP} = -3/5$. The optimal solution to the integer optimization problem is $\mathbf{x}_{IP} = (1,2)'$, and $Z_{IP} = 1$. Note that $Z_{LP} < Z_D < Z_{IP}$.

Since there are nine points in $\mathcal{F}$, $Z(\lambda)$ is the minimum of the following nine linear functions:

$$3 - 2\lambda,\ 6 - 3\lambda,\ 2 - \lambda,\ 5 - 2\lambda,\ -2 + \lambda,\ 1,\ 4 - \lambda,\ \lambda,\ 3.$$

In Figure 4.3, we plot the function $Z(\lambda)$. It follows from Figure 4.3 that

$$Z(\lambda) = \begin{cases} -2 + \lambda, & 0 \leq \lambda \leq 5/3, \\ 3 - 2\lambda, & 5/3 \leq \lambda \leq 3, \\ 6 - 3\lambda, & \lambda \geq 3. \end{cases}$$



!

we can conclude that
$[(D\underline{x} \geq \underline{d}) \cap (\text{conv}(\underline{x}))] \subseteq$ LP polyhedron

So the boundaries that we get with lagrangian duality are always at least as good (usually much better) than the boundaries that we obtain by linear programming relaxation.

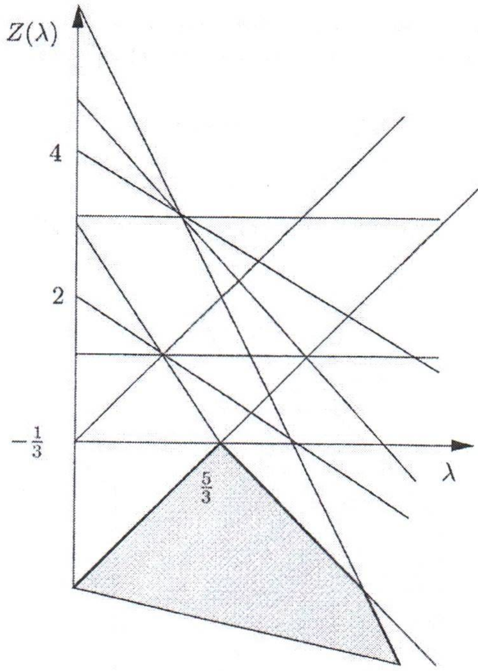


Figure 4.3: The function $Z(\lambda)$. Each line is the plot of the function $3x_1 - x_2 + \lambda(-1 - x_1 + x_2)$, where (x_1, x_2) is set to some particular element of $\mathcal{F}$. The lower envelope of these lines is the function $Z(\lambda)$. The maximum of $Z(\lambda)$ is $-1/3$ and is attained for $\lambda = 5/3$.

The Lagrangean dual is maximized for $\lambda = 5/3$, and the optimal value is $Z_D = Z(5/3) = -1/3$.

In order to illustrate Theorem 4.9, we find first $\text{conv}(\mathcal{F})$, and intersect it with the constraint $x_1 - x_2 \geq -1$, forming the shaded polyhedron in Figure 4.2. Optimizing the original objective function $3x_1 - x_2$ over this polyhedron, we obtain that the optimal solution is $\mathbf{x}_D = (1/3, 4/3)'$ with value $-1/3$, which is the same as Z_D. To illustrate the key insight of the proof of Theorem 4.9, we have for $\lambda = 5/3$:

$$Z(5/3) = \min_{(x_1,x_2)' \in \mathcal{F}} 3x_1 - x_2 + \frac{5}{3}(-1 - x_1 + x_2) = \min_{(x_1,x_2)' \in \mathcal{F}} \frac{1}{3}(4x_1 + 2x_2 - 5).$$

Notice that the vectors $(1, 0)'$ and $(0, 2)'$ are optimal solutions for this problem. As the proof of Theorem 4.9 demonstrates, $\mathbf{x}_D$ can be written as a convex combination of $(1, 0)'$ and $(0, 2)'$: $\mathbf{x}_D = (1/3, 4/3)' = 2/3(0, 2)' + 1/3(1, 0)'$.

Although we presented the method for the case where the relaxed constraints were inequalities, the method is exactly the same even if we

have equality constraints. The or
Lagrange multipliers are unrestrict

Having characterized the opti
solution to a linear optimization p
the optimal cost Z_{IP} and the optin

minimiz
subject t

In general, the following ordering h

$Z_{\text{LP}} \leq$

The first inequality follows from Th

$\text{conv}(\mathcal{F}) \subseteq \{\mathbf{x}$

and the second inequality follows fr
we show that, depending on the ol
be strict.

Example 4.4 We refer again to Ex
the following possibilities:

(a) For the original objective functi
(b) If we change the objective funct
(c) For the objective function $-x_1$ -

One can also construct an example (F
$Z_D < Z_{\text{IP}}$ holds. In this example, how

Using Theorem 4.9, we can make tl

Corollary 4.1
(a) *We have $Z_{\text{IP}} = Z_D$, for all c*

$\text{conv}\Big(\mathcal{F} \cap \{\mathbf{x} \mid \mathbf{Ax} \geq \mathbf{b}$

(b) *We have $Z_{\text{LP}} = Z_D$, for all c*

$\text{conv}(\mathcal{F})$

It is interesting to observe tl
represents the convex hull of $\mathcal{F}$, tha
is equal to the optimal cost of the l

How to strength formulations of the relaxations of ILP's?

Cutting planes method:

separation problem :
- Chvatal-Gomory procedure
- MIR aggregated inequalities
- GMI

Theorical (↑). In practice?

- Cover inequalities for binary knapsack problem
  (separation of cover inequalities + strengthening of cover inequalities)
- Traveling salesman problem (TSP)
  (separation of cut set inequalities + strengthening of subtour elimination ineq.)
  cutting plane approach — facet defining inequalities

State of art: Branch and Cut
(combination of Branch and Bound and cutting plane method)

Lagrangian relaxation

ILP (primal problem) ⟶ Lagrangian sub-problem ⟶ Lagrangian dual

$z^* = \min\{\underline{c}^T\underline{x} : D\underline{x} \geq \underline{d}, \underline{x} \in X\}$

$\underbrace{w(\underline{u})}_{\text{dual function}} = \min_{\underline{x}} \underbrace{L(\underline{x},\underline{u})}_{\text{lagrangian function}}$

$w^* = \max_{\underline{u}} \underbrace{w(\underline{u})}_{\text{concave piecewise linear}}$

z^*

$w(\underline{u}) \quad \underline{u} \geq \underline{0}$

- The boundaries that we get with lagrangian duality are always at least as good as the boundaries that we obtain with the LP relaxation $\left((D\underline{x} \geq \underline{d}) \cap (\mathrm{conv}(X)) \subset LP \right)$
  $z_{LP} \leq w^* \leq z^*$
- Solution of the lagrangian dual: subgradient method

$\underline{u}_k \longrightarrow$ solve $w(\underline{u}_k) = \min L(\underline{x},\underline{u}_k)$ $\longrightarrow$ $\underline{u}_{k+1} = \underline{u}_k + \alpha_k(\underline{d} - D\underline{x}(\underline{u}_k))$

$\underline{x}(\underline{u}_k)$ optimal $\Rightarrow (\underline{d} - D\underline{x}(\underline{u}_k)) \in \partial w(\underline{u}_k)$

$\alpha_k = \varepsilon_k \dfrac{\hat{w} - w(\underline{u}_k)}{\|\underline{d} - D\underline{x}(\underline{u}_k)\|^2}$

Column generation

ILMP (integer linear master problem) ⟶ LPM (LP relaxation of the master problem) ⟶ LPRM (restricted (we consider only the variables in J_K, not all vars in J)) ⟶ Pricing subproblem (search for a new "improving" non basic variable)

If ∃ a non basic variable which improve: $J_{k+1} := J_K \cup \{new\}$ ⟶ back to LPRM

If ∄ ⟶ the optimal solution of LPRM is also optimal for LPM

- Modelling techniques
  1. Binary choice
     knapsack problem, set covering/packing/partitioning
  2. Association between entities
     assignment problem
  3. Forcing constraints
     uncapacitated facility location (UFL))
  4. Piecewise linear cost function
     minimization of piecewise linear cost funcion
  5. Modeling with exponentially many constraints
     asymmetric traveling salesman problem (ATPS)
  6. Disjunctive constraints
     introduction (+ shceduling problem (lab))
  7. Linearization of products of variables

- Alternative, strong and ideal formulations
  - LP relaxation
  - (Alternative) formulations: *TSP, UFL*
  - Stronger formulations: *UFL, STSP*
  - Ideal formulations: *assignment problem, perfect matching problem*
  - Extended formulations
  - Comparison between formulations (Fourier-Motzkin elimination method): *ULS*
  - Stronger extended formulations: *FCNF*

- "Easy" ILP problems and TU matrices
  - TU matrices: sufficient conditions for being TU and characterization of TU
  - Natural formulations that are ideal: *assignment problem, transportation problem, minimum cost flow problem*

- Relaxations, heuristics and bounds
  - Primal and dual bounds
  - Relaxation
  - Different relaxations: LP, elimination, surrogate, lagrangian, combinatorial
  - Heuristic for primal bounds: greedy methods, local search methods, metaheuristics

- Branch and Bound

- Cutting plane methods
  - Valid inequalities
  - Methods: addition a priori, cutting plane methods
  - Simple valid inequalities: binary set, mixed 0-1 set, combinatorial set
  - Chvatal cutting planes for ILP: integer rounding, Chvatal-Gomory procedure
  - Gomory fractional/integer cutting planes
  - Mixed integer rounding (MIR) inequalities
  - Gomory mixed integer cutting planes (GMI)

- Strong valid inequalities for structured ILP problems
  - Dominant inequalities, redundant inequalities
  - Faces and facets
  - Cover inequalities for binary *knapsack problem*
  - Strong valid inequalities for *TSP*
  - Equivalence between separation and optimization

- Lagrangian relaxation
  - Lagrangian subproblem
  - Lagrangian dual
  - Optimal solutions of Lagrangian subproblem and of the primal
  - Strength and choice of the Lagrangian dual
  - Solution of the Lagrangian duals
  - Lagrangian relaxation for the *STSP*

- Column generation method