Polyhedral combinatorics

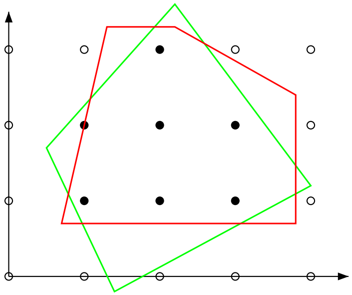
Operational Research Complements

Giovanni Righini

Università degli Studi di Milano

Formulations

Discrete linear optimization problems do *not* have a unique formulation.



Formulations

Since they are not unique, it makes sense

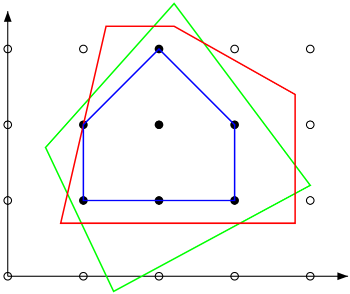
- *to compare* formulations
- *to improve* formulations.

A better formulation translates into a less time-consuming algorithm.

The ideal formulation of a MILP problem is the one which allows solving it as a LP problem.

Ideal formulation

A formulation of a linear programming problem corresponds to a *polyhedron*.



The constraints of the ideal formulation correspond to the *convex hull* of the integer solutions.

Convex hull

Given a discrete set

$$X = \{x_1, \ldots, x_t\} \text{ with } x_i \in \Re^n \ \forall i = i, \ldots, t,$$

its *convex hull* is the polyhedron

$$conv(X) = \{x \in \Re^n : x = \sum_{i=1}^{t} \lambda_i x_i, \sum_{i=1}^{t} \lambda_i = 1, \lambda_i \geq 0 \ \forall i = 1, \ldots, t\}.$$

It is a *polyhedron* whose extreme points are the elements of the discrete set X.

Given a valid formulation P and the discrete set X of its feasible solutions, it holds:

$$X \subseteq conv(X) \subseteq P$$

Polyhedral combinatorics

In general

- we do not know the ideal formulations of discrete linear optimization problems
- the number of their constraints grows exponentially with the size of the instance.

We know the ideal formulations of some particular combinatorial optimization problems (the shortest path problem, the minimum cost bipartite matching problem, the minimum spanning tree problem,...).

The research stream aiming at the selection and improvement of linear formulations for discrete linear optimization problems is called polyhedral combinatorics.

Selection of formulations: example

In many MILP problems involving capacity constraints (Bin Packing Problem, Facility Location Problem,...), there are constraints of this form:

$$\sum_{i \in \mathcal{N}} x_{ij} \leq |\mathcal{N}| y_j \quad \forall j \in \mathcal{M} \tag{1}$$

that is used to express a logical condition on binary variables x and y:

$$\begin{cases} \exists (i,j) \in \mathcal{N} \times \mathcal{M} : x_{ij} > 0 & \Rightarrow y_j = 1 \\ \exists j \in \mathcal{M} : y_j = 0 & \Rightarrow x_{ij} = 0 \;\; \forall i \in \mathcal{N} \end{cases}$$

The same condition can be expressed as

$$x_{ij} \leq y_j \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M}. \tag{2}$$

Formulation (1) requires $|\mathcal{M}|$ constraints.
Formulation (2) requires $|\mathcal{M}||\mathcal{N}|$ constraints.

Selection of formulations: example

Summing up constraints (2) for each $i \in \mathcal{N}$ we get

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

i.e. constraints (1): $\sum_{i \in \mathcal{N}} x_{ij} \leq |\mathcal{N}| y_j \quad \forall j \in \mathcal{M}$.

Therefore each constraint (1) is a *surrogate* of constraints (2).
Constraints (2) imply constraints (1) but not vice versa.
There are solutions that satisfy (1) but violate (2):

$$\begin{cases} x_{ij} = 1 & \forall j \in \mathcal{M}, \forall i \in \mathcal{N} : i \in [k(j-1)+1, \ldots, kj] \\ y_j = 1/|\mathcal{M}| & \forall j \in \mathcal{M} \end{cases}$$

where $k = |\mathcal{N}|/|\mathcal{M}|$.

Constraints (2) yield a better formulation than constraints (1).
The polyhedron with (1) contains the polyhedron with (2).

Comparing formulations: example

Consider these two formulations of the lot-sizing problem.

$$
\begin{aligned}
P1) s_{t-1} + x_t &= d_t + s_t & \forall t \in \mathcal{T} \\
x_t &\leq K y_t & \forall t \in \mathcal{T} \\
s_0 &= 0 & \\
s_t &\geq 0 & \forall t \in \mathcal{T} \\
x_t &\geq 0 & \forall t \in \mathcal{T} \\
0 \leq y_t &\leq 1 & \forall t \in \mathcal{T}
\end{aligned}
$$

$$
\begin{aligned}
P2) \sum_{i=1}^{t} w_{it} &= d_t & \forall t \in \mathcal{T} \\
w_{it} &\leq d_t y_i & \forall i, t \in \mathcal{T} : i \leq t \\
w_{it} &= 0 & \forall i, t \in \mathcal{T} : i > t \\
w_{it} &\geq 0 & \forall i, t \in \mathcal{T} : i \leq t \\
0 \leq y_t &\leq 1 & \forall t \in \mathcal{T}
\end{aligned}
$$

where $K = \sum_{t \in \mathcal{T}} d_t$.

They use different variables, so that they cannot be directly compared. However, they can be compared by *projection*, i.e. variables replacement.

Comparing formulations: example

Projecting $P2$ onto the subspace of variables x, s and y, we obtain a polyhedron contained in $P1$.

To project a polyhedron onto the space of the other one, we need a relationship between the variables of $P1$ and $P2$.

$$x_i = \sum_{t=i}^{|\mathcal{T}|} w_{it} \;\; \forall i \in \mathcal{T}$$

Let $\mathcal{T}$ be $[1, \ldots, T]$.

Comparing formulations: example

Part 1: demand satisfaction constraints.
To prove that every point of $P2$ is projected into a point of $P1$, we rewrite the lhs and the rhs of the constraints of $P1$ in an equivalent way. We use the following equalities and inequalities:

$$\sum_{i=1}^{t} x_i \geq \sum_{j=1}^{t} d_j \quad \forall t \in \mathcal{T} \tag{3}$$

$$\sum_{i=1}^{j} w_{ij} = d_j \quad \forall j \in \mathcal{T} \tag{4}$$

$$w_{ij} = 0 \quad \forall i > j \in \mathcal{T} \tag{5}$$

$$w_{ij} \geq 0 \quad \forall i \leq j \in \mathcal{T} \tag{6}$$

$$x_i = \sum_{j=i}^{T} w_{ij} \quad \forall i \in \mathcal{T} \tag{7}$$

Comparing formulations: example

By means of (4), the right-hand-side of (3) can be rewritten:

$$\sum_{j=1}^{t} d_j = \sum_{j=1}^{t} \sum_{i=1}^{j} w_{ij} \quad \forall t \in \mathcal{T} \tag{8}$$

Now we observe that for any given index $j \leq t$

$$\sum_{i=1}^{j} w_{ij} = \sum_{i=1}^{t} w_{ij}$$

because $w_{ij} = 0$ when $i > j$, owing to (5). The right-hand-side can now be rewritten swapping the two sums:

$$\sum_{j=1}^{t} \sum_{i=1}^{t} w_{ij} = \sum_{i=1}^{t} \sum_{j=1}^{t} w_{ij}$$

Comparing formulations: example

Now we observe that for any given index $i \in \mathcal{T}$

$$\sum_{j=1}^{t} w_{ij} = \sum_{j=i}^{t} w_{ij}$$

because $w_{ij} = 0$ when $j < i$, owing to (5).
Therefore the right-hand-side is

$$\sum_{i=1}^{t} \sum_{j=i}^{t} w_{ij}$$

By means of (7), the left-hand-side of (3) can be rewritten:

$$\sum_{i=1}^{t} x_i = \sum_{i=1}^{t} \sum_{j=i}^{T} w_{ij} \qquad (9)$$

Comparing formulations: example

With the new (but equivalent) expressions (8) for the rhs and (9) for the lhs, inequality (3) can be reformulated as follows:

$$\sum_{i=1}^{t}\sum_{j=i}^{T} w_{ij} \geq \sum_{i=1}^{t}\sum_{j=i}^{t} w_{ij} \quad \forall t \in \mathcal{T}$$

This inequality is always satisfied because of (6).

So, we have proven that every point satisfying (4) also satisfies (3).

Comparing formulations: example

Part 2: constraints linking production and periods.
In polyhedron $P1$ we have

$$x_i \leq K y_i \quad \forall i \in \mathcal{T} \tag{10}$$

with $K = \sum_{t=1}^{T} d_t$.
In polyhedron $P2$ we have

$$w_{ij} \leq d_j y_i \;\; \forall i \leq j \in \mathcal{T}. \tag{11}$$

Using the projection $x_i = \sum_{j=i}^{T} w_{ij}$ as before, inequality (10) can be rewritten as

$$\sum_{j=i}^{T} w_{ij} \leq K y_i \quad \forall i \in \mathcal{T}. \tag{12}$$

Comparing formulations: example

From (11) we know that

$$\sum_{j=i}^{T} w_{ij} \leq \sum_{j=i}^{T} d_j y_i$$

Since $\sum_{j=i}^{T} d_j \leq K$, (12) is satisfied: every point satisfying (11) also satisfies (10).

Therefore formulation $P2$ is tighter than $P1$ (by the way, $P2$ it is also an ideal formulation).

Comparing formulations: example

The converse is not true: for instance the solution

$$\begin{cases} x_t = d_t & \forall t \in \mathcal{T} \\ y_t = d_t/K & \forall t \in \mathcal{T} \end{cases}$$

is feasible for $P1$ but not for $P2$.

Cutting planes algorithms

Cutting planes algorithms iteratively solve the linear relaxation L of a discrete optimization problem P and reinforce its formulation, generating additional constraints (cutting planes), so that the optimal solution of the linear relaxation at iteration k is infeasible in iteration $k+1$.

- Pros:
  - if cutting planes are generated in a clever way, the algorithm can guarantee to produce an optimal solution without having recourse to any other technique (e.g. branching);
  - a stronger formulation, although not ideal, can provide tighter *dual bounds* to be used in a branch-and-bound algorithm.
- Cons:
  - a procedure is needed to generate valid and useful inequalities at each iteration: it is called *separation algorithm*. If the original problem is difficult, the separation problem also is.

Cutting planes algorithms

Given a discrete linear optimization problem

$$P^{(k)} = \max\{cx : Ax \leq b, x \in \mathcal{Z}^n_+\}$$

we consider its continuous relaxation

$$L^{(k)} = \max\{cx : Ax \leq b, x \in \Re^n_+\}$$

and its optimal solution $x^{*(k)}$. A cutting planes algorithm generates one or more *valid inequalities* $Qx \leq q$ s.t.

- $Qx \leq q \;\; \forall x \in \mathcal{Z}^n_+ : Ax \leq b$
- $Qx^{*(k)} > q$

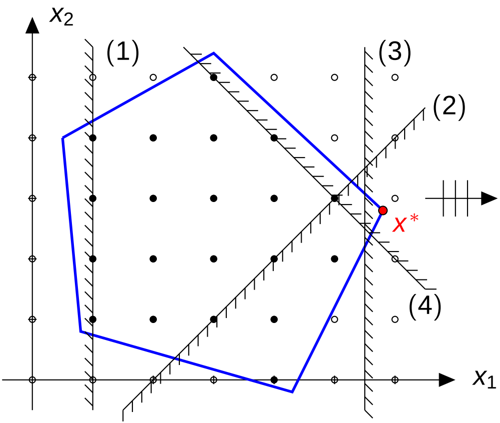
yielding a new (stronger) formulation

$$P^{(k+1)} = \max\{cx : Ax \leq b, Qx \leq q, x \in \mathcal{Z}^n_+\}.$$

Cutting planes algorithms: pseudo-code

Begin Cutting plane algorithm
$t \leftarrow 0$; $P^{(0)} \leftarrow P$; [P is the linear relaxation]
repeat
  $z^{*(t)} \leftarrow \max\{cx : x \in P^{(t)}\}$
  $x^{*(t)} \leftarrow \text{argmax}\{cx : x \in P^{(t)}\}$
  if $x^{*(t)} \notin \mathcal{Z}^n$ **then**
    Find a valid inequality $\pi x \leq \pi_0 : \pi x^{*(t)} > \pi_0$
    $P^{(t+1)} \leftarrow P^{(t)} \cap \{x : \pi x \leq \pi_0\}$
    $t \leftarrow t + 1$
  end if
until ($x^{*(t)} \in \mathcal{Z}^n$) or (no inequalities found)
End Cutting plane algorithm

Valid inequalities: example



Inequality (1) is valid but useless: it does not cut off x^*.
Inequality (2) is not valid: it cuts off feasible integer points.
Inequality (3) is valid and useful.
Inequality (4) is also *facet defining*.

Cutting planes algorithms

After every iteration $z^{*(t)}$ is a valid dual bound.

It may happen that no valid inequality is found if we restrict the separation algorithm to some specific subsets of inequalities, with a special structure, that do not define the convex hull of the original problem.

Example 1: logical implications

Consider the formulation

$$X = \{x \in \mathcal{B}^5 : 3x_1 - 4x_2 + 2x_3 - 3x_4 + x_5 \leq -2\}.$$

Fixing $x_2 = x_4 = 0$ would imply $3x_1 + 2x_3 + x_5 \leq -2$ which is impossible. Therefore

$$x_2 + x_4 \geq 1$$

is a valid inequality.
It cuts off some fractional solutions. For instance $\bar{x} = [0\ \frac{1}{3}\ 0\ \frac{1}{3}\ 0]$.

Fixing $x_1 = 1$ and $x_2 = 0$ would imply $3 + 2x_3 - 3x_4 + x_5 \leq -2$ which is impossible. Therefore

$$x_1 \leq x_2$$

is a valid inequality.
It cuts off some fractional solutions. For instance $\bar{x} = [1\ \frac{1}{2}\ 0\ 1\ 0]$.

Example 2a: combination of fixed and variable bounds

Consider the mixed-integer formulation (with a binary variable)

$$X = \{(x, y) \in \Re \times \mathcal{B} : x \leq 100y, 0 \leq x \leq 5\}.$$

The constraint

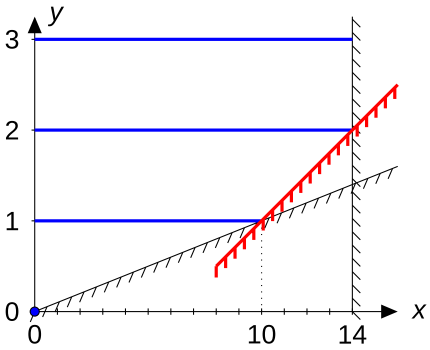
$$x \leq 5y$$

is a valid inequality (stronger than $x \leq 100y$).

Example 2b: combination of fixed and variable bounds

Consider the mixed-integer formulation (with an integer variable)

$$X = \{(x, y) \in \Re \times \mathcal{Z}_+ : x \leq 10y, 0 \leq x \leq 14\}.$$



The constraint $x \leq 6 + 4y$ is a valid inequality.

Example 3: combinatorial inequalities

Consider this formulation of the matching problem on a graph $G = (V, E)$:

$$X = \{x \in \mathcal{B}^{|E|} : \sum_{e \in \delta(i)} x_e \leq 1 \ \forall i \in V\}.$$

where $\delta(i) = \{e \in E : \exists j \in V : e = [i, j]\}$.

The constraint

$$\sum_{e \in E(S)} x_e \leq \frac{|S| - 1}{2} \quad \forall S \subseteq V : |S| \geq 3 \text{ and odd}$$

is a valid inequality.

Example 4: integer rounding

Consider this formulation with integer variables:

$$X = \{x \in \mathcal{Z}_+^4 : 13x_1 + 20x_2 + 11x_3 + 6x_4 \geq 72\}.$$

We divide the constraint by 11:

$$\frac{13}{11}x_1 + \frac{20}{11}x_2 + x_3 + \frac{6}{11}x_4 \geq \frac{72}{11}.$$

We round up the coefficients of the lhs (relaxing the constraint)

$$2x_1 + 2x_2 + x_3 + x_4 \geq \frac{72}{11}.$$

We can round up the coefficients of the rhs (tightening the constraint)

$$2x_1 + 2x_2 + x_3 + x_4 \geq 7.$$

This inequality is valid (because x is integer).

Example 5: mixed integer rounding

Consider this formulation with integer variables:

$$X = \{(x, y) \in \mathcal{Z}_+^4 \times \Re_+ : 13x_1 + 20x_2 + 11x_3 + 6x_4 + y \geq 72\}.$$

We divide the constraint by 11:

$$\frac{13}{11}x_1 + \frac{20}{11}x_2 + x_3 + \frac{6}{11}x_4 \geq \frac{72 - y}{11}.$$

Since

$$\left\lceil \frac{72 - y}{11} \right\rceil \begin{cases} = 7 & \text{if } y < 6 \\ \leq 6 & \text{if } y \geq 6 \end{cases}$$

we obtain

$$\left\lceil \frac{72 - y}{11} \right\rceil \geq 7 - \frac{y}{6}$$

and therefore

$$2x_1 + 2x_2 + x_3 + x_4 + \frac{1}{6}y \geq 7.$$

is a valid inequality.