Foundations of Operations Research

Master of Science in Computer Engineering

Roberto Cordone
roberto.cordone@unimi.it

Tuesday 13.15 - 15.15

Thursday 10.15 - 13.15

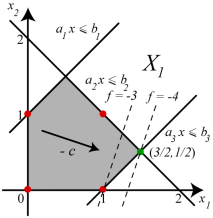
http://homes.di.unimi.it/~cordone/courses/2014-for/2014-for.html

Lesson 18: Cutting planes

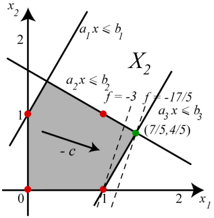
Como, Fall 2013

Actually, different *ILP* formulations can identify the same *ILP* problem

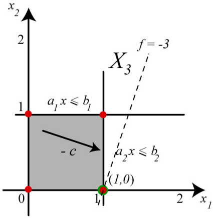
$$\begin{aligned} \min f &= -3x_1 + x_2 \\ -x_1 + x_2 &\leq 1 \\ x_1 + x_2 &\leq 2 \\ x_1 - x_2 &\leq 1 \\ x &\in \mathbb{N}^n \end{aligned}$$



$$\begin{aligned} \min f &= -3x_1 + x_2 \\ -2x_1 + x_2 &\leq 1 \\ x_1 + 2x_2 &\leq 3 \\ 2x_1 - x_2 &\leq 2 \\ x &\in \mathbb{N}^n \end{aligned}$$



$$\begin{aligned} \min f &= -3x_1 + x_2 \\ x_1 &\leq 1 \\ x_2 &\leq 1 \\ x &\in \mathbb{N}^n \end{aligned}$$



The continuous relaxations are different:

- the relaxed optimal solutions are substantially different
- the bounds (relaxed optima) are substantially different

Ideal formulation

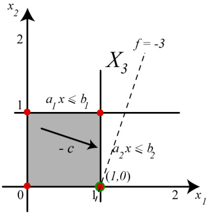
Assume that $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is a bounded (polytope) for the sake of simplicity: the feasible region is $X = P \cap Z^n$

The convex hull of X (conv(X)) is the smallest convex set including X

- the vertices of conv(X) have integer coordinates
- their number is finite and their convex hull is a polytope
- the corresponding formulation is valid for the *ILP* problem
- all basic solutions of its continuous relaxation are integer

The continuous relaxation provided by conv(X) is the tightest one

- its optimal solution is the optimal solution of the *ILP* problem



It is denoted as ideal formulation

An algorithm?

Then, in theory, one could solve any *ILP* problem as follows

1. build the ideal formulation *How?*
2. solve its continuous relaxation *LP* $\Rightarrow$ *polynomial time!*

But *ILP* is $\mathcal{NP}$-complete, whereas *LP* is polynomial!

Where is the contradiction?

There is no contradiction: the problem is hidden in the first step

- the ideal formulation consists of an exponential system of constraints (in general) with respect to the original natural formulation

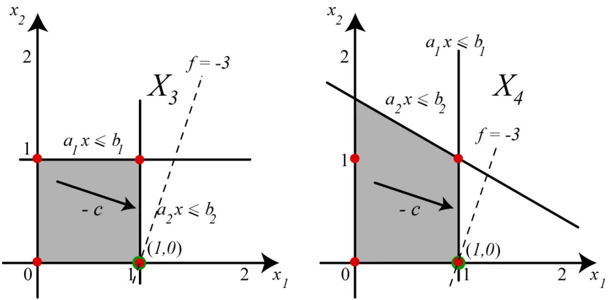
Therefore, this algorithm requires polynomial time with respect to an exponential function of the original size

Moreover, finding the ideal formulation is not trivial

For most problems, the ideal formulation is unknown

A useful remark

It is actually redundant to identify the whole ideal formulation:
only the active constraints in the optimal solution are required



This improves the algorithm

- it is simpler to find the correct formulation
- the number of constraints of the final formulation is smaller

But it is still not trivial to identify the required constraints

Assume that the optimal relaxed solution x^*_{LP} is fractionary (*otherwise, the problem is already solved*)

A cutting plane, in short cut, is an additional constraint $\hat{a}\, x \leq \hat{b}$

- satisfied by all feasible integer solutions

$$\hat{a}\, x \leq \hat{b} \quad \text{for all } x \in X$$

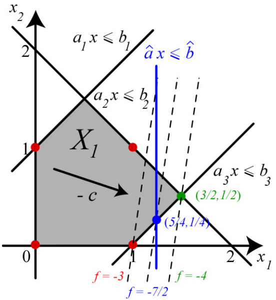
- violated by the optimal relaxed solution x^*_{LP}

$$\hat{a}\, x^*_{LP} > \hat{b}$$

Formulation X_1 admits cutting plane $x_1 \leq 5/4$

Adding a cutting plane to the formulation

- modifies the optimal relaxed solution (the current one becomes unfeasible) $(3/2, 1/2) = x^*_{LP} \neq x^*_{\hat{LP}} = (5/4, 1/4)$
- in general, improves the bound (it is not looser, hopefully tighter) $-4 = f^*_{LP} \leq f^*_{\hat{LP}} = -3.5$



Cutting plane algorithm

This suggests a modified cutting plane algorithm:

1. start from the natural formulation
2. solve the continuous relaxation of the current formulation computing its optimal solution x^*_{LP}
3. if x^*_{LP} is integer, return it and terminate;
   otherwise, find a cutting plane $\hat{a}\, x \leq \hat{b}$,
   introduce it in the current formulation and go to step 2

Now the problems are

1. is it always possible to find a cutting plane at each step?
2. how does one find a cutting plane?
3. is the procedure guaranteed to terminate?

In 1958, Ralph Gomory found a general way to produce cutting planes for all *ILP* problems, and proved that his method always terminates

Given an *ILP* problem P and its continuous relaxation P_{LP}, let

$$x_{b_r} + \sum_{j \in N} \bar{a}_{rj} x_j = \bar{b}_r \qquad r = 1, \ldots, m$$

be the constraints of the optimal basic canonical form of P_{LP}, where

- N is the set of indices of the nonbasic variables
- b_r is the index of the basic variable occurring in row $r = 1, \ldots, m$

If $\bar{b}_r$ is fractionary, a cutting plane for P is given by

$$x_{b_r} + \sum_{j \in N} \lfloor \bar{a}_{rj} \rfloor x_j \leq \lfloor \bar{b}_r \rfloor$$

Proof: *The proof follows a nice two-step process. First, since* $x \geq 0$*, rounding down the left-hand-side coefficients* $\bar{a}_{rj}$ *yields a relaxation of constraint* r

$$x_{b_r} + \sum_{j \in N} \lfloor \bar{a}_{rj} \rfloor x_j \leq \bar{b}_r \qquad r = 1, \ldots, m$$

where $\lfloor \bar{a}_{rj} \rfloor$ *is the largest integer not larger than* $\bar{a}_{rj}$

Gomory cuts (2)

Second, for all solutions x with integer components, the left-hand-side is a combination of integer values, and hence integer.
So, the constraint can be tightened rounding down its right-hand-side

$$x_{b_r} + \sum_{j \in N} \lfloor \bar{a}_{rj} \rfloor x_j \leq \lfloor \bar{b}_r \rfloor \qquad r = 1, \ldots, m$$

All integer nonnegative solutions of P_{LP} satisfy this constraint
Now consider the optimal solution of P_{LP}, denoted as x^*

$$\begin{cases} x^*_{b_r} = \bar{b}_r \\ x^*_j = 0 \text{ for all } j \in N \end{cases} \Rightarrow x^*_{b_r} + \sum_{j \in N} \lfloor \bar{a}_{rj} \rfloor x^*_j = \bar{b}_r > \lfloor \bar{b}_r \rfloor$$

Therefore, x^* violates the cutting plane

Example:

$$\begin{aligned} x_2 + 1.3\, x_4 - 3.2\, x_5 + 4\, x_6 &= 6.7 \Rightarrow \\ \Rightarrow x_2 + 1 \quad x_4 - 4 \quad x_5 + 4\, x_6 &\leq 6.7 \Rightarrow \\ \Rightarrow x_2 + 1 \quad x_4 - 4 \quad x_5 + 4\, x_6 &\leq 6 \end{aligned}$$

Integer and fractional form (1)

It is possible to obtain an alternative equivalent form of the Gomory cut by subtracting the cut from the original constraint

$$\begin{array}{llcll}
\text{Original constraint} & x^*_{b_r} + & \displaystyle\sum_{j\in N} \bar{a}_{rj} x^*_j & = & \bar{b}_r \\
\text{Integer Gomory cut:} & x^*_{b_r} + & \displaystyle\sum_{j\in N} \lfloor \bar{a}_{rj} \rfloor x^*_j & \leq & \lfloor \bar{b}_r \rfloor \\
\text{Fractional Gomory cut:} & & \displaystyle\sum_{j\in N} \left(\bar{a}_{rj} - \lfloor \bar{a}_{rj} \rfloor\right) x^*_j & \geq & \bar{b}_r - \lfloor \bar{b}_r \rfloor
\end{array}$$

Example:

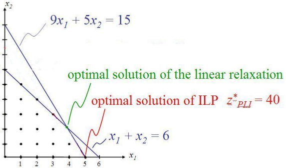
$$\begin{array}{rcl}
x_2 + 1.3\,x_4 - 3.2\,x_5 + 4\,x_6 & = & 6.7 \\
x_2 + 1\,\,\,\, x_4 - 4\,\,\,\, x_5 + 4\,x_6 & \leq & 6 \\
0.3\,x_4 + 0.8\,x_5 & \geq & 0.7
\end{array}$$

The fractional form makes the cutting plane algorithm more efficient

Therefore, it is the one more commonly used

Integer and fractional form (2)

$$\max z = 8\,x_1 + 5\,x_2$$
$$x_1 + x_2 \leq 6$$
$$9\,x_1 + 5\,x_2 \leq 45$$
$$x_1, x_2 \in \mathbb{N}$$



The continuous relaxation has the following optimal *tableau*

41 + 1/4	0	0	5/4	3/4
15/4	1	0	-5/4	1/4
9/4	0	1	9/4	-1/4

The optimal relaxed solution is fractionary: $x^* = (15/4, 9/4) = (3.75, 2.25)$

Both components are fractionary; both rows could provide a Gomory cut
Focus on the first row, and consider two alternative approaches

- add to the *tableau* the integer cut

$$x_1 - 2x_3 \leq 3 \Leftrightarrow x_1 - 2x_3 + x_5 = 3$$

- add to the *tableau* the fractional cut

$$3/4x_3 + 1/4x_4 \geq 3/4 \Leftrightarrow -3/4x_3 - 1/4x_4 + x_5 = -3/4$$

Integer and fractional form (3)

The two approaches are equivalent from a geometric point of view
Let us consider the graphical representation in plane (x_1, x_2); since

$$\begin{cases} x_3 = 6 - x_1 - \ \ x_2 \\ x_4 = 45 - 9\,x_1 - 5\,x_2 \end{cases}$$

- the cut in integer form is

$$x_1 - 2x_3 \leq 3 \Leftrightarrow x_1 - 2\,(6 - x_1 - \ \ x_2) \leq 3 \Leftrightarrow 3x_1 + 2x_2 \leq 15$$

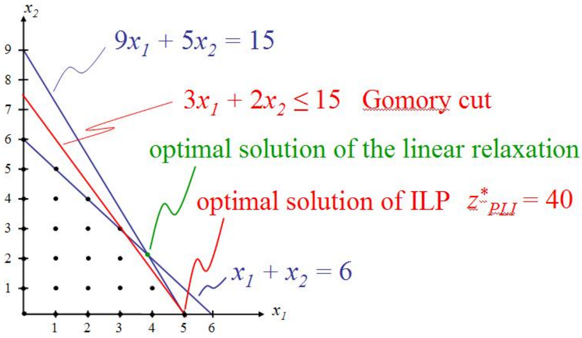
- the cut in fractional form is

$$\frac{3}{4}x_3 + \frac{1}{4}x_4 \geq \frac{3}{4} \Leftrightarrow \frac{3}{4}\,(6 - x_1 - \ \ x_2) + \frac{1}{4}\,(45 - 9\,x_1 - 5\,x_2) \geq \frac{3}{4} \Leftrightarrow$$

$$\Leftrightarrow -3x_1 - 2x_2 \geq -15$$

They are (obviously!) the same constraint

Integer and fractional form (4)



In this case, reoptimization leads to an integer solution $(5, 0)$, which is optimal for the original *ILP* problem, with $f^* = 40$

In general, the resulting solution is still fractional, and other cuts must be introduced; at each step:

- the previous optimal solution is removed
- the lower bound improves

Integer and fractional form (5)

The two forms are equivalent, but their use has a very different impact

1. while the integer cut destroys the canonical form (*column* A_1 *has two* 1*s*)
   the fractional cut keeps the canonical form
2. though the basic solution becomes unfeasible (x^* *violates the added cut!*), it has nonnegative reduced costs

$$b_{m+1} < 0 \text{ but } c_j \geq 0 \text{ for all } j$$

3. it is not necessary to resort to the two-phase method to regain feasibility

Feasibility can be regained with the dual simplex algorithm which

- improves feasibility while keeping superoptimality ($c \geq 0$)
- exactly as the second phase of the simplex algorithm improves the objective while keeping feasibility
- the only difference is the rule to choose the *pivot* element

So, using fractional cuts is much more efficient than using integer ones

Dual simplex algorithm

The dual simplex algorithm can be used only when the current basic solution is superoptimal ($c_j \geq 0$ for all j)

The *pivot* element is selected guaranteeing that

1. the feasibility improves: always choose a *pivot* row with negative right-hand-side
$$b_i < 0$$
2. the new basic variable is feasible: always choose a negative *pivot* element
$$a_{ij} < 0$$
3. the reduced costs keep nonnegative: always choose the *pivot* column with minimum $c_k / |a_{ik}|$ among those with $a_{ik} < 0$
$$j := \arg \min_{k : a_{ik} < 0} \frac{c_k}{|a_{ik}|}$$

There is a relation with duality, not discussed here: the dual *pivot* rule actually

1. builds the dual problem corresponding to the current *tableau*: its basic solution corresponds to the primal one through complementary slackness
2. applies the standard *pivot* rule, obtaining a modified dual basic solution
3. retrieves the corresponding primal basic solution through complementary slackness

Example

Consider the *tableau* introduced above, with the additional Gomory cut

41 + 1/4	0	0	5/4	3/4	0
15/4	1	0	-5/4	1/4	0
9/4	0	1	9/4	-1/4	0
-3/4	0	0	-3/4	-1/4	1

- The *pivot* row is obvious: only the additional row has a negative right-hand-side
- Two elements are negative: $a_{33} = -3/4$ and $a_{34} = -1/4$
- The first one has the minimum ratio c_k/a_{ik}

The *pivot* element is a_{33}

40	0	0	0	1/3	5/3
5	1	0	0	2/3	-5/3
0	0	1	0	-1	3
1	0	0	1	1/3	-4/3

The current basic solution is not only superoptimal, but also feasible

Therefore, it is optimal: $x^* = (5, 0, 1, 0, 0)$ with $z^* = 40$

The algorithm terminates

Termination

If an *ILP* problem admits a finite optimal solution, the Gomory cutting plane method provides such a solution after adding a finite number of Gomory cuts

This is a very good news!

But was not ILP $\mathcal{NP}$*-complete?*

No contradiction: in the worst case, the number of Gomory cuts which must be added to obtain the optimal integer solution is exponential with respect to the size of the original problem

In practice, the phenomenon known as tailing off occurs:

- the first cuts improve the lower bound
- the following ones become weaker and weaker

General cuts and specific cuts

There are many other types of cutting planes, besides Gomory cuts:

- general cuts can be applied to any *ILP* problem
- specific cuts can be applied only to special families of *ILP* problems

Indeed, the specific cuts (for the Travelling Salesman Problem) were discovered some years before Gomory cuts

The strongest cuts are the facets of $\text{conv}(X)$; much research aims to

- characterize classes of facets for different families of problems
- find efficient procedures to generate such facets

The best algorithms for *ILP* combine cutting planes and branch-and-bound

- cutting planes aim to improve the bound for the branch-and-bound subproblems and to find integer solutions
- branching operations interrupt the generation of cuts when tailing off begins