Foundations of Operations Research

Master of Science in Computer Engineering

Roberto Cordone
roberto.cordone@unimi.it

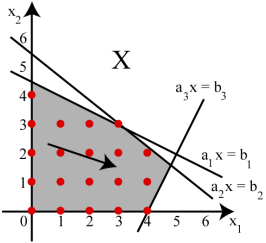
Tuesday 13.15 - 15.15

Thursday 10.15 - 13.15

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

Integer Linear Programming

$$\begin{aligned} \min z &= c\,x + d \quad (\Pi) \\ A\,x &= b \\ x &\geq 0 \\ x &\in \mathbb{Z}^n \end{aligned}$$



It is a family of nonlinear problems: $x \in \mathbb{Z}^n \Leftrightarrow \sum_{j=1}^{n} \sin^2(\pi x_j) = 0$

Except for (relevant) particular cases, there is no polynomial algorithm

- to solve Π
- to decide whether Π admits feasible solutions

...So, what can be done?

Explicit enumeration

We assume that the continuous relaxation has a bounded feasible region (for the sake of simplicity: most conclusions hold anyway)

$$P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\} \text{ bounded} \Leftrightarrow x_j \in \{0, \ldots, K_j\} \subset \mathbb{N}$$

Therefore, the feasible region $X = P \cap \mathbb{Z}^n$ is finite

One can apply the exhaustive algorithm:

- set the best known value $\tilde{z}$ to $+\infty$ (minimization problem)
- subsequently generate each vector x such that $x_j \in \{0, \ldots, K_j\}$
- if the current x is feasible, compare $z(x)$ with $\tilde{z}$:
  save in $\tilde{z}$ the minimum of the two

Unluckily, the number of vectors x to be tested is exponential

The exhaustive method is not practical!

Divide et impera

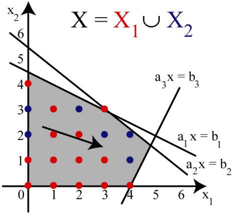
Notice that, if $X = \bigcup_{\ell=1}^{r} X_\ell = X_1 \bigcup \ldots \bigcup X_r$

$$x_i^* = \arg\min_{x \in X} z(x) = \arg\min_{\ell=1,\ldots,r} \left[\min_{x \in X_\ell} z(x) \right]$$

Therefore, one can solve Π by solving r smaller problems $\Pi_1, \ldots, \Pi_r$

$$\min z = c\,x + d \qquad (\Pi_\ell)$$
$$x \in X_\ell$$

$$\begin{aligned} \min z &= -3x_1 + x_2 \quad (\Pi) \\ x_1 + x_2 &\leq 9 \\ 6x_1 + 7x_2 &\leq 39 \\ 2x_1 - x_2 &\leq 8 \\ x &\in \mathbb{N}^n \end{aligned}$$



$$z^* = \min \left[\min_{x \in X_1} z(x), \min_{x \in X_2} z(x) \right] = \min [z(4,0), z(4,1)] = \min(-12, -11) = -12$$

A recursive approach

Why is it better to solve several subproblems Π_ℓ ($\ell = 1, \dots, r$) instead of Π?

- base case: Π_ℓ could be a problem with:
  - no feasible solution: $X_\ell = \emptyset$
  - a single feasible solution: $X_\ell = \{x_\ell\} \Rightarrow z^*_\ell = \min_{x \in X_\ell} z(x) = z(x_\ell)$
  - a polynomial solving algorithm
- recursive case: the decomposition can be repeated on each subproblem Π_ℓ

To apply the scheme recursively, we generate the subproblems by adding affine branching constraints $\hat{A}^{(\ell)} x \leq \hat{b}^{(\ell)}$, so that the subproblems are *ILP* problems

$$\begin{aligned} X_\ell &= X \cap \left\{ x \in \mathbb{R}^n : \hat{A}^{(\ell)} x \leq \hat{b}^{(\ell)} \right\} = \\ &= P \cap \mathbb{Z}^n \cap \left\{ x \in \mathbb{R}^n : \hat{A}^{(\ell)} x \leq \hat{b}^{(\ell)} \right\} = \\ &= P_\ell \cap \mathbb{Z}^n \end{aligned}$$

where $P_\ell = P \cap \left\{ x \in \mathbb{R}^n : \hat{A}^{(\ell)} x \leq \hat{b}^{(\ell)} \right\}$

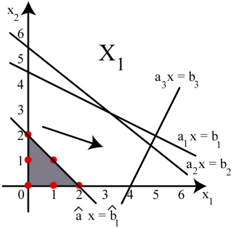
The constraints of the previous page were nonaffine, the subproblems not ILP

Branching rule is the procedure to build the branching constraints $\left(\hat{A}^{(\ell)}, \hat{b}^{(\ell)} \right)$

Parallel constraints: $\hat{a} \in \mathbb{Z}^n$ and $\hat{b}_\ell \in \mathbb{Z}$ for $\ell = 1, \ldots, r-1$

1. $\hat{a}\, x \leq \hat{b}_1$ (e. g., $x_1 + x_2 \leq 2$)
2. $\hat{b}_2 + 1 \leq \hat{a}\, x \leq \hat{b}_3$ (e. g., $3 \leq x_1 + x_2 \leq 4$)
3. $\ldots$
4. $\hat{b}_{2r-2} + 1 \leq \hat{a}\, x$ (e. g., $x_1 + x_2 \geq 5$)

$$
\begin{array}{rcll}
\min z & = & -3x_1 + x_2 & (\Pi_1) \\
x_1 + x_2 & \leq & 9 & \\
6x_1 + 7x_2 & \leq & 39 & \\
2x_1 - x_2 & \leq & 8 & \\
x_1 + x_2 & \leq & 2 & \\
x & \in & \mathbb{N}^n &
\end{array}
$$

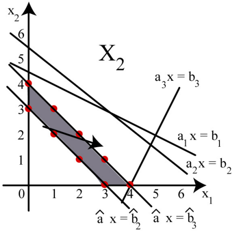


Parallel branching constraints

Parallel constraints: $\hat{a} \in \mathbb{Z}^n$ and $\hat{b}_\ell \in \mathbb{Z}$ for $\ell = 1, \ldots, r-1$

1. $\hat{a}\,x \leq \hat{b}_1$ (e. g., $x_1 + x_2 \leq 2$)
2. $\hat{b}_2 + 1 \leq \hat{a}\,x \leq \hat{b}_3$ (e. g., $3 \leq x_1 + x_2 \leq 4$)
3. $\ldots$
4. $\hat{b}_{2r-2} + 1 \leq \hat{a}\,x$ (e. g., $x_1 + x_2 \geq 5$)

$$
\begin{aligned}
\min z &= -3x_1 + x_2 \quad (\Pi_2)\\
x_1 + x_2 &\leq 9\\
6x_1 + 7x_2 &\leq 39\\
2x_1 - x_2 &\leq 8\\
3 \leq x_1 + x_2 &\leq 4\\
x &\in \mathbb{N}^n
\end{aligned}
$$

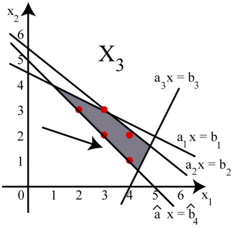


Parallel branching constraints

Parallel constraints: $\hat{a} \in \mathbb{Z}^n$ and $\hat{b}_\ell \in \mathbb{Z}$ for $\ell = 1, \ldots, r-1$

1. $\hat{a}\,x \leq \hat{b}_1$ (e. g., $x_1 + x_2 \leq 2$)
2. $\hat{b}_2 + 1 \leq \hat{a}\,x \leq \hat{b}_3$ (e. g., $3 \leq x_1 + x_2 \leq 4$)
3. $\ldots$
4. $\hat{b}_{2r-2} + 1 \leq \hat{a}\,x$ (e. g., $x_1 + x_2 \geq 5$)

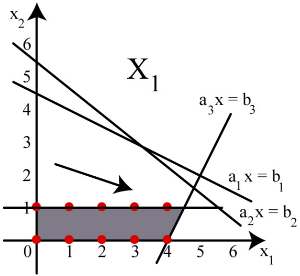
$$
\begin{aligned}
\min z &= -3x_1 + x_2 \quad (\Pi_3)\\
x_1 + x_2 &\leq 9\\
6x_1 + 7x_2 &\leq 39\\
2x_1 - x_2 &\leq 8\\
x_1 + x_2 &\geq 5\\
x &\in \mathbb{N}^n
\end{aligned}
$$

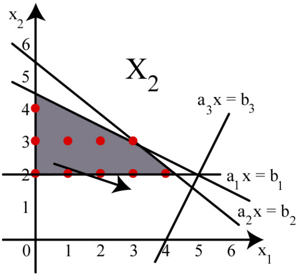


A standard rule

The most common branching rule (by far) consists in
selecting a branching variable x_{j^*} an integer threshold $b \in \mathbb{Z}$, and setting

- $x_{j^*} \leq b$ in Π_1 (e. g., $x_2 \leq 1$)
- $x_{j^*} \geq b + 1$ in Π_2 (e. g., $x_2 \geq 2$)





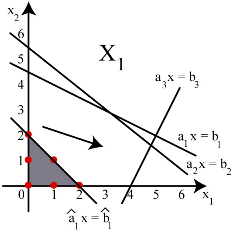
For binary variables, $x_{j^*} = 0$ and $x_{j^*} = 1$

Hierarchical branching constraints

Hierarchical constraints: $\hat{a}^{(i)} \in \mathbb{Z}^n$ and $\hat{b}^{(i)} \in \mathbb{Z}$ for $i = 1, \ldots, r-1$

1. $\hat{a}^{(1)}x \leq \hat{b}^{(1)}$ $\quad (x_1 + x_2 \leq 2)$
2. $\hat{a}^{(1)}x \geq \hat{b}^{(1)} + 1$ and $\hat{a}^{(2)}x \leq \hat{b}^{(2)}$ $\quad (x_1 + x_2 \geq 3,\ x_2 - x_1 \leq 0)$
3. $\ldots$
4. $\hat{a}^{(1)}x \geq \hat{b}^{(1)} + 1, \ldots$ and $\hat{a}^{(r-1)}x \geq \hat{b}^{(r-1)} + 1$ $\quad (x_1 + x_2 \geq 3,\ x_2 - x_1 \geq 1)$

$$
\begin{array}{rcll}
\min z & = & -3x_1 + x_2 & (\Pi) \\
x_1 + x_2 & \leq & 9 & \\
6x_1 + 7x_2 & \leq & 39 & \\
2x_1 - x_2 & \leq & 8 & \\
x_1 + x_2 & \leq & 2 & \\
x & \in & \mathbb{N}^n &
\end{array}
$$

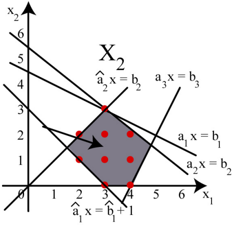


Hierarchical branching constraints

Hierarchical constraints: $\hat{a}^{(i)} \in \mathbb{Z}^n$ and $\hat{b}^{(i)} \in \mathbb{Z}$ for $i = 1, \ldots, r-1$

1. $\hat{a}^{(1)}x \leq \hat{b}^{(1)}$ $(x_1 + x_2 \leq 2)$
2. $\hat{a}^{(1)}x \geq \hat{b}^{(1)} + 1$ and $\hat{a}^{(2)}x \leq \hat{b}^{(2)}$ $(x_1 + x_2 \geq 3,\ x_2 - x_1 \leq 0)$
3. ...
4. $\hat{a}^{(1)}x \geq \hat{b}^{(1)} + 1, \ldots$ and $\hat{a}^{(r-1)}x \geq \hat{b}^{(r-1)} + 1$ $(x_1 + x_2 \geq 3,\ x_2 - x_1 \geq 1)$

$$
\begin{aligned}
\min z &= -3x_1 + x_2 \quad (\Pi)\\
x_1 + x_2 &\leq 9\\
6x_1 + 7x_2 &\leq 39\\
2x_1 - x_2 &\leq 8\\
x_1 + x_2 &\geq 3\\
-x_1 + x_2 &\leq 0\\
x &\in \mathbb{N}^n
\end{aligned}
$$

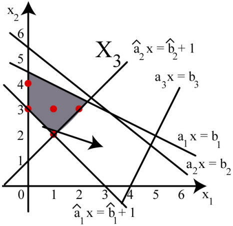


Hierarchical branching constraints

Hierarchical constraints: $\hat{a}^{(i)} \in \mathbb{Z}^n$ and $\hat{b}^{(i)} \in \mathbb{Z}$ for $i = 1, \ldots, r-1$

1. $\hat{a}^{(1)}x \leq \hat{b}^{(1)}$ $\qquad (x_1 + x_2 \leq 2)$
2. $\hat{a}^{(1)}x \geq \hat{b}^{(1)} + 1$ and $\hat{a}^{(2)}x \leq \hat{b}^{(2)}$ $\qquad (x_1 + x_2 \geq 3,\ x_2 - x_1 \leq 0)$
3. ...
4. $\hat{a}^{(1)}x \geq \hat{b}^{(1)} + 1, \ldots$ and $\hat{a}^{(r-1)}x \geq \hat{b}^{(r-1)} + 1$ $\qquad (x_1 + x_2 \geq 3,\ x_2 - x_1 \geq 1)$

$$
\begin{array}{rcll}
\min z & = & -3x_1 + x_2 & (\Pi) \\
x_1 + x_2 & \leq & 9 & \\
6x_1 + 7x_2 & \leq & 39 & \\
2x_1 - x_2 & \leq & 8 & \\
x_1 + x_2 & \geq & 3 & \\
-x_1 + x_2 & \geq & 1 & \\
x & \in & \mathbb{N}^n &
\end{array}
$$



A standard rule

It is possible to focus on a subset of branching variables $x_1, \dots, x_{r-1}$, define integer thresholds $b_1, \dots, b_{r-1}$, and impose

1. $x_1 \leq b_1$
2. $x_1 \geq b_1 + 1$ and $x_2 \leq b_2$
3. $\dots$
4. $x_1 \geq b_1 + 1, \dots, x_{r-2} \geq b_{r-2} + 1$ and $x_{r-1} \leq b_{r-1}$
5. $x_1 \geq b_1 + 1, \dots, x_{r-2} \geq b_{r-2} + 1$ and $x_{r-1} \geq b_{r-1} + 1$

This rule, however, is not common when dealing with general variables

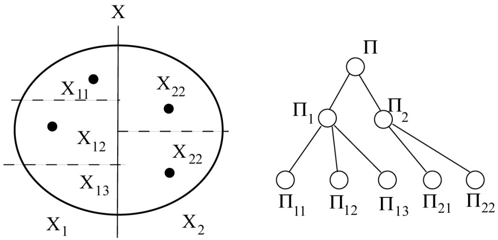
It is common on binary variables x_1, x_2, $\dots$, x_{r-1}:

1. $x_1 = 0$
2. $x_1 = 1$ and $x_2 = 0$
3. $\dots$
4. $x_1 = 1, \dots, x_{r-2} = 1$ and $x_{r-1} = 0$
5. $x_1 = 1, \dots, x_{r-2} = 1$ and $x_{r-1} = 1$

Branching tree

A branch-and-bound algorithm produces a branching tree, that is a tree with nodes associated to subsets of the feasible region X (subproblems)

- the root node is associated to X
- each node is associated to the union of the sets associated to its children
- the leaf nodes are associated to subsets with zero or one solution



Number of nodes (subproblems)	$\geq$	Number of leaves (easy subproblems)	$\geq$	Number of solutions

$\Rightarrow$ The branching tree has exponential size

So, what is the advantage?

Drawbacks associated to the base cases

Given a branch-and-bound algorithm on an *ILP* problem

- the base cases are exponentially many
- the base cases are not trivial
  (it is $\mathcal{NP}$-complete to decide whether $X_\ell = \emptyset$)

The solution is to use additional information to

1. remove subproblems without solving them
2. solve the remaining subproblems efficiently (if possible)

Dominance

Implicit enumeration: examining whole subsets of solutions at a time

Instead of computing the optimal solution of all subproblems, one can compute a bound, or superoptimal estimate, i. e. a value better than the optimum

If for a subset of solutions X_ℓ the bound is worse than a known solution $\tilde{x}$, that subset cannot provide any improvement

Theorem: Let Π be an *ILP* problem and $\tilde{x}$ one of its feasible solutions. Let Π_ℓ be a subproblem of Π with a subset of feasible solutions $X_\ell \subseteq X$, and LB_ℓ be a lower bound on the value of the objective function z in X_ℓ.

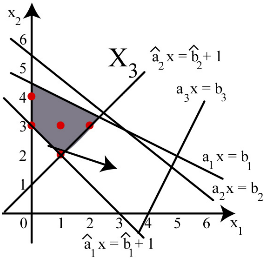
If $LB_\ell \geq z(\tilde{x})$, then at least one optimal solution of Π belongs to $X \setminus X_\ell$

$$\begin{cases} \tilde{x} \in X \\ LB_\ell \leq z(x), \; \forall x \in X_\ell \\ LB_\ell \geq z(\tilde{x}) \end{cases} \Rightarrow \exists x^* \in X \setminus X_\ell \text{ such that } z(x^*) \leq z(x), \; \forall x \in X$$

In such a situation Π_ℓ can be removed without solving it!

An example of dominance

$$\begin{aligned}
\min z &= -3x_1 + x_2 \quad (\Pi) \\
x_1 + x_2 &\leq 9 \\
6x_1 + 7x_2 &\leq 39 \\
2x_1 - x_2 &\leq 8 \\
x_1 + x_2 &\geq 3 \\
-x_1 + x_2 &\geq 1 \\
x &\in \mathbb{N}^n
\end{aligned}$$



Suppose that we know $\tilde{x} = (4, 0) \in X$, with $z(\tilde{x}) = -12$

From the lesson on duality, notice that

$$\begin{cases} -x_1 + x_2 \geq 1 \\ -x_1 - x_2 \geq -9 \end{cases} \quad \text{for all } x \in X_3 \Rightarrow$$

$$\Rightarrow 2(-x_1 + x_2) + (-x_1 - x_2) \geq 2(1) + (-9) \text{ for all } x \in X_3 \Rightarrow$$

$$\Rightarrow z(x) = -3x_1 + x_2 \geq -7 \text{ for all } x \in X_3$$

Since $\tilde{x}$ is (strictly) better than all solutions of X_3 and the optimal solutions will not be worse than $\tilde{x}$, they will be out of X_3

X_3 can be fully removed

Other two ingredients

In order to apply this condition, one needs

1. a heuristic solution $\tilde{x} \in X$, determined with algorithms that do not guarantee optimality, whose value is a suboptimal estimate
2. a bound LB_ℓ or superoptimal estimate of $z(x)$ for $x \in X_i$

Notice that

- the suboptimal estimate is global (worse than the optimum of X)
- the superoptimal estimate is local (better than the optimum of X_ℓ)

A good heuristic solution (small $z(x)$) and a good bound (large LB_ℓ) remove several subproblems Π_i $\Rightarrow$ they accelerate the algorithm

. . . *How to find a good heuristic solution and a good bound?*

Relaxations

A relaxation of a given problem Π is any problem Π'

$$\begin{cases} \min z = f(x) \\ x \in X \end{cases} \qquad \begin{cases} \min z = f'(x) \\ x \in X' \end{cases}$$

such that the following two properties hold

1. $X' \supseteq X$
2. $f'(x) \leq f(x)$ for all $x \in X$

Given the optimal solutions x^* of Π and x'^* of Π'

$$f'(x'^*) \leq f'(x^*) \leq f(x^*) \leq f(x) \text{ for all } x \in X$$

The optimum of any relaxation of Π is a bound on the optimum of Π

Usually, one looks for easy relaxations of hard problems

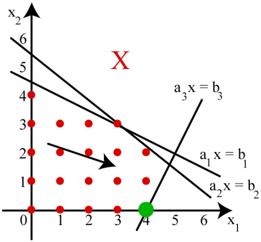
In rare cases, the relaxation yields the optimal solution of the problem:
if $x'^* = x^*$ and $f'(x^*) = f(x^*)$, then x'^* is an optimal solution of Π

Continuous relaxation

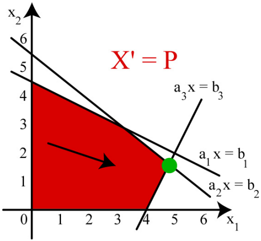
The continuous relaxation adopts $f'(x) = f(x)$ and $X' = P$

$$\begin{aligned} \min z &= cx + d \\ Ax &\leq b \\ x &\geq 0 \\ x &\in X \end{aligned}$$

$$\begin{aligned} \min z &= cx + d \\ Ax &\leq b \\ x &\geq 0 \end{aligned}$$



$x^* = (4,0) \Rightarrow z(x^*) = -12$



$x^* = (4.75,1.5) \Rightarrow z(x^*) = -12.75$

Another (weaker, but faster) way to compute bounds is to find dual feasible solutions

Computation of heuristic solutions

Heuristic solutions provide suboptimal estimation $z(\tilde{x}) \geq z^*$, which are compared to the local bound of each subproblem $LB_\ell \leq z^*_\ell$ in an attempt to remove the subproblem

They can be computed

1. with a heuristic algorithm run before the branch-and-bound method
2. with a heuristic algorithm run at each node, based on auxiliary information produced by the processing of the associated subproblem (e. g., rounding or Lagrangean heuristics)
3. solving exactly one of the subproblems (when possible)

As the method proceeds, new heuristic solutions are discovered, and the best known one is updated, improving the suboptimal estimate

Branching tree visit

As we cannot process simultaneously several nodes of the branching tree, the open ones must be sorted

Visit strategy of the branching tree is the
order in which the open branching nodes are processed

- depth-first: process the last node generated
  - few nodes are open (reduced memory consumption)
  - tends to reduce subproblems
    (quickly providing heuristic solutions)
- best-first: process the most "promising" open node according to some criterium (usually the one with minimum bound)
  - tends to quickly provide better heuristic solutions
- hybrid methods: the strategy changes over time

Obviously, children nodes are visited

- only after the parent node (they do not exist before)
- only if the parent nodes does not represent a base case
  (they are not generated in that case)

Summary

A branch-and-bound algorithm is defined by

1. a branching rule
2. a procedure to compute a heuristic solution
3. a procedure to compute a bound
4. a tree visit strategy

These four elements have a crucial influence on the efficiency, since every choice reflects exponentially on the number of subproblems and consequently on the computational time

Branch-and-bound

Algorithm *Branch-and-Bound*(Π)

$\tilde{x} :=$ *Heuristic*(Π);

$x_\Pi :=$*Relaxation*(Π);

If *Optimal*(x_Π, Π) **then** $\tilde{x} := x_\Pi$;

Insert(Π, x_Π, Q);

While *NotEmpty*(Q) **do**

  (Π, x_Π) := *Extract*(Q);

  If $z(x_\Pi) < z(\tilde{x})$ **then**

    ($\Pi_1, \ldots, \Pi_r$) :=*Branching*(Π);

    For $\ell := 1$ **to** r **do**

      If *Feasible*(Π_ℓ) **then**

        $x_{\Pi_\ell} :=$*Relaxation*(Π_i);

        If *Optimal*(x_{Π_ℓ}, Π_ℓ) **and** $z(x_{\Pi_\ell}) < z(\tilde{x})$

          then $\tilde{x} := x_{\Pi_\ell}$

          else *Insert*($\Pi_\ell, x_{\Pi_\ell}, Q$);

Return $\tilde{x}$;

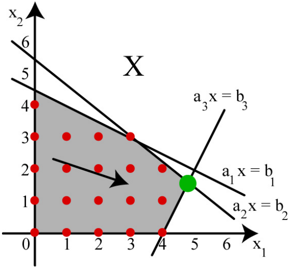
Example

This example applies the branch-and-bound obtained applying

- visit strategy: depth-first, so that Q is a stack
- heuristic: no auxiliary algorithm; start with $\tilde{z} = +\infty$, and replace it as new better solutions are found
- bound: continuous relaxation
- branching rule: standard rule, choosing as branching variable the most fractionary variable x_{j^*} in the relaxed optimal solution (the fractional part of x_{j^*} is closest to 0.5)

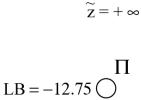
Example

$$\begin{aligned} \min z &= -3x_1 + x_2 \quad (\Pi) \\ x_1 + x_2 &\leq 9 \\ 6x_1 + 7x_2 &\leq 39 \\ 2x_1 - x_2 &\leq 8 \end{aligned}$$



No feasible solution known: set $\tilde{z} = +\infty$

The continuous relaxation provides
$x_\Pi = (4.75, 1.5)$ and $LB = z(x_\Pi) = -12.75$
It is a bound, not a feasible optimal solution



As $-12.75 = LB < \tilde{z} = +\infty$,
Π enters Q, and is immediately extracted

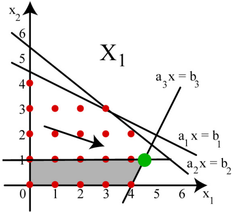
As $-12.75 = LB < \tilde{z} = +\infty$,
branch on the most fractionary variable

- Π_1: $x_2 \leq \lfloor \xi_2^* \rfloor = 1$
- Π_2: $x_2 \geq \lfloor \xi_2^* \rfloor + 1 = 2$

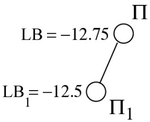
Example

Let us process Π_1

$$
\begin{aligned}
\min z &= -3x_1 + x_2 \quad (\Pi_1) \\
x_1 + x_2 &\leq 9 \\
6x_1 + 7x_2 &\leq 39 \\
2x_1 - x_2 &\leq 8 \\
x_2 &\leq 1
\end{aligned}
$$



$\tilde{z} = +\infty$



The continuous relaxation of Π_1 provides $x_{\Pi_1} = (4.5, 1)$ and $LB_1 = z(x_{\Pi_1}) = -12.5$

It is a bound, not a feasible optimal solution
The suboptimal estimate remains $\tilde{z} = +\infty$

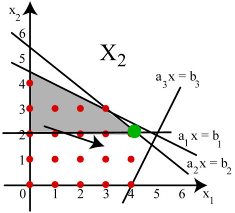
As $-12.5 = z(x_{\Pi_1}) < \tilde{z} = +\infty$, Π_1 enters Q

$Q = (\Pi_1)$

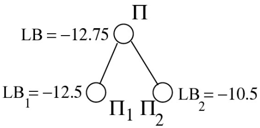
Example

Let us process Π_2

$$\begin{aligned} \min z &= -3x_1 + x_2 \quad (\Pi_2) \\ x_1 + x_2 &\leq 9 \\ 6x_1 + 7x_2 &\leq 39 \\ 2x_1 - x_2 &\leq 8 \\ x_2 &\geq 2 \end{aligned}$$



$\tilde{z} = +\infty$



The continuous relaxation of Π_2 provides $x_{\Pi_1} = (4.1\bar{6}, 2)$ and $LB_2 = z(x_{\Pi_2}) = -10.5$

It is a bound, not a feasible optimal solution
The suboptimal estimate remains $\tilde{z} = +\infty$

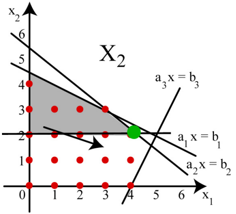
As $-10.5 < LB_2 < \tilde{z} = +\infty$, Π_2 enters Q

$Q = (\Pi_2, \Pi_1)$

Example

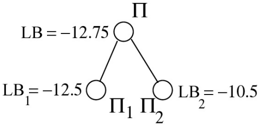
Extract Π_2 from the top of stack Q

$$\begin{aligned} \min z &= -3x_1 + x_2 \quad (\Pi_2) \\ x_1 + x_2 &\leq 9 \\ 6x_1 + 7x_2 &\leq 39 \\ 2x_1 - x_2 &\leq 8 \\ x_2 &\geq 2 \end{aligned}$$



$\tilde{z} = +\infty$

We know that $x_{\Pi_2} = (4.1\bar{6}, 2)$ and $LB_2 = -10.5$



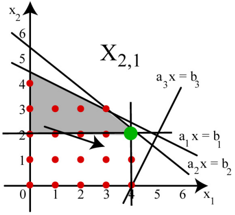
As $LB_2 = -10.5 < \tilde{z} = +\infty$,
branch on the most fractional variable

- $\Pi_{2,1}$: $x_1 \leq \lfloor x_1^* \rfloor = 4$
- $\Pi_{2,2}$: $x_1 \geq \lfloor x_1^* \rfloor + 1 = 5$

Example

Let us process $\Pi_{2,1}$

$$\begin{aligned} \min z &= -3x_1 + x_2 \quad (\Pi_{2,1}) \\ x_1 + x_2 &\leq 9 \\ 6x_1 + 7x_2 &\leq 39 \\ 2x_1 - x_2 &\leq 8 \\ x_2 &\geq 2 \\ x_1 &\leq 4 \end{aligned}$$



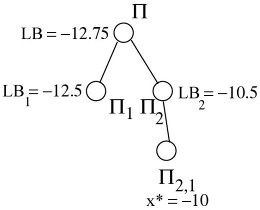
$\tilde{z} = -10$

The continuous relaxation of $\Pi_{2,1}$ provides $x_{\Pi_{2,1}} = (4, 2)$ and $LB_{2,1} = z\left(x_{\Pi_{2,1}}\right) = -10$

It is an optimal solution of $\Pi_{2,1}$:
$\tilde{z}$ is updated to -10

As $LB_{2,1} = -10 \geq \tilde{z} = -10$,
$\Pi_{2,1}$ is closed

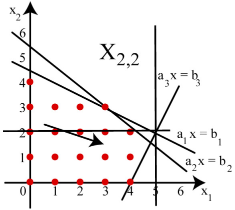
$Q = (\Pi_1)$



Example

Let us process $\Pi_{2,2}$

$$
\begin{aligned}
\min z &= -3x_1 + x_2 \quad (\Pi_{2,2}) \\
x_1 + x_2 &\leq 9 \\
6x_1 + 7x_2 &\leq 39 \\
2x_1 - x_2 &\leq 8 \\
x_2 &\geq 2 \\
x_1 &\geq 5
\end{aligned}
$$



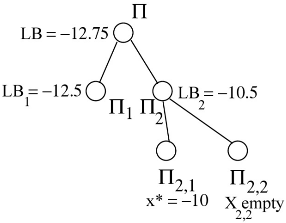
$\tilde{z} = -10$

As the continuous relaxation of $\Pi_{2,2}$ is unfeasible, also $\Pi_{2,2}$ is unfeasible

$\Pi_{2,2}$ is closed

$\tilde{z} = -10$

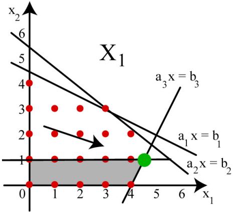
$Q = (\Pi_1)$



Example

Extract Π_1 from the top of stack Q

$$
\begin{aligned}
\min z &= -3x_1 + x_2 \quad (\Pi_1) \\
x_1 + x_2 &\leq 9 \\
6x_1 + 7x_2 &\leq 39 \\
2x_1 - x_2 &\leq 8 \\
x_2 &\leq 1
\end{aligned}
$$

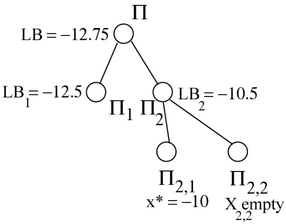


$\tilde{z} = -10$

We know that $x_{\Pi_1} = (4.5, 1)$ and $LB_1 = -12.5$

As $LB_1 = -12.5 < \tilde{z} = -10$,
branch on the most fractionary variable

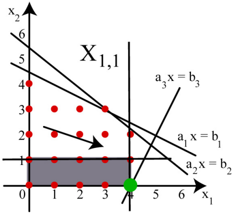
- $\Pi_{1,1}$: $x_1 \leq \lfloor \xi_1^* \rfloor = 4$
- $\Pi_{1,2}$: $x_1 \geq \lfloor \xi_1^* \rfloor + 1 = 5$



Example

Let us process $\Pi_{1,1}$

$$\begin{aligned} \min z &= -3x_1 + x_2 \quad (\Pi_{1,1}) \\ x_1 + x_2 &\leq 9 \\ 6x_1 + 7x_2 &\leq 39 \\ 2x_1 - x_2 &\leq 8 \\ x_2 &\leq 1 \\ x_1 &\leq 4 \end{aligned}$$



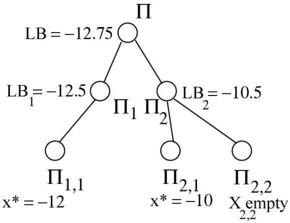
$\tilde{z} = -12$

The continuous relaxation of $\Pi_{1,1}$ provides $x_{\Pi_{1,1}} = (4,0)$ and $LB_{1,1} = z\left(x_{\Pi_{1,1}}\right) = -12$

It is an optimal solution of $\Pi_{1,1}$:
$\tilde{z}$ is updated to -12

As $LB_{1,1} = -12 \geq \tilde{z} = -12$,
$\Pi_{1,1}$ is closed

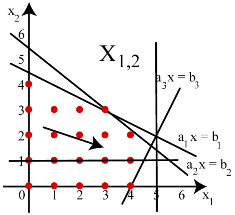
$Q = \emptyset$



Example

Let us process $\Pi_{1,2}$

$$\begin{aligned} \min z &= -3x_1 + x_2 \quad (\Pi_{1,2}) \\ x_1 + x_2 &\leq 9 \\ 6x_1 + 7x_2 &\leq 39 \\ 2x_1 - x_2 &\leq 8 \\ x_2 &\leq 1 \\ x_1 &\geq 5 \end{aligned}$$



$\tilde{z} = -12$

As the continuous relaxation of $\Pi_{1,2}$ is unfeasible, also $\Pi_{1,2}$ is unfeasible

$\Pi_{1,2}$ is closed

$\tilde{z} = -12$

$Q = \emptyset$

As there are no more open subproblems, the optimal solution is $\tilde{x} = (4, 0)$ and the optimum is $\tilde{z} = -12$