DM545

Linear and Integer Programming

Lecture 11
Cutting Plane Algorithms
Branch and Bound

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

1. Cutting Plane Algorithms

2. Branch and Bound

Valid Inequalities

- IP: $z = \max\{c^T x : x \in X\}, X = \{x : Ax \leq b, x \in \mathbb{Z}^n_+\}$
- Proposition: $\text{conv}(X) = \{x : \tilde{A}x \leq \tilde{b}, x \geq 0\}$ is a polyhedron
- LP: $z = \max\{c^T x : \tilde{A}x \leq \tilde{b}, x \geq 0\}$ would be the best formulation
- Key idea: try to approximate the best formulation.

Definition (Valid inequalities)

$ax \leq b$ is a valid inequality for $X \subseteq \mathbb{R}^n$ if $ax \leq b \, \forall x \in X$

Which are useful inequalities? and how can we find them?
How can we use them?

Example: Pre-processing

- $X = \{(x, y) : x \leq 999y; 0 \leq x \leq 5, y \in \mathbb{B}^1\}$

$$x \leq 6y$$

- $X = \{x \in \mathbb{Z}^n_+ : 13x_1 + 20x_2 + 11x_3 + 6x_4 \geq 72\}$

$$2x_1 + 2x_2 + x_3 + x_4 \geq \frac{13}{11}x_1 + \frac{20}{11}x_2 + x_3 + \frac{6}{11}x_4 \geq \frac{72}{11} \geq 6 + \frac{6}{11}$$
$$2x_1 + 2x_2 + x_3 + x_4 \geq 7$$

- UFL:

$$\sum_{i \in M} x_{ij} \leq b_j y_j \quad \forall j \in N \qquad\qquad x_{ij} \leq b_j y_j$$
$$\sum_{j \in N} x_{ij} = a_i \quad \forall i \in M \qquad\qquad x_{ij} \leq a_i$$
$$x_{ij} \geq 0, y_j \in B^n \qquad\qquad x_{ij} \leq \max\{a_i, b_j\} y_j$$

Chvátal-Gomory cuts

- $X \in P \cap \mathbb{Z}_+^n, \quad P = \{x \in \mathbb{R}_+^n : Ax \leq b\}, \quad A \in \mathbb{R}^{n \times m}$
- $u \in \mathbb{R}_+^n$, $\{a_1, a_2, \dots a_n\}$ columns of A

CG procedure to construct valid inequalities

1) $\sum_{j=1}^{n} u a_j x_j \leq ub$ valid: $u \geq 0$

2) $\sum_{j=1}^{n} \lfloor u a_j \rfloor x_j \leq ub$ valid: $x \geq 0$ and $\sum \lfloor u a_j \rfloor x_j \leq \sum u a_j x_j$

3) $\sum_{j=1}^{n} \lfloor u a_j \rfloor x_j \leq \lfloor ub \rfloor$ valid for X since $x \in \mathbb{Z}^n$

Theorem

Every valid inequality for X can be obtained by applying the CG procedure a finite number of times

However often the family of valid inequalities is large and makes the LP hard

Cutting Plane Algorithms

- $X \in P \cap \mathbb{Z}^n_+$
- a family of valid inequalities $\mathcal{F} : a^T x \leq b, (a, b) \in \mathcal{F}$ for X
- we do not find them all a priori, only interested in those close to optimum

Cutting Plane Algorithm

Init.: $t = 0, P^0 = P$

Iter. t: Solve $\bar{z}^t = \max\{c^T x : x \in P^t\}$
let x^t be an optimal solution
if $x^t \in \mathbb{Z}^n$ stop, x^t is opt to the IP
if $x^t \notin \mathbb{Z}^n$ solve separation problem for x^t and $\mathcal{F}$
if (a^t, b^t) is found with $a^t x^t > b^t$ that cuts off x^t

$$P^{t+1} = P \cap \{x : a^i x \leq b^i, i = 1, \ldots, t\}$$

else stop (P^t is in any case an improved formulation)

Gomory's fractional cutting plane algorithm

Cutting plane algorithm + Chvátal-Gomory cuts

- $\max\{c^T x : Ax = b, x \geq 0, x \in \mathbb{Z}^n\}$
- Solve LPR to optimality

$$\left[\begin{array}{c:c:c:c} I & \bar{A}_N = A_B^{-1} A_N & 0 & \bar{b} \\ \hdashline \bar{c}_B & \bar{c}_N (\leq 0) & 1 & -\bar{d} \end{array}\right]$$

$$x_u = \bar{b}_i - \sum_{j \in N} \bar{a}_{uj} x_j, \quad u \in B$$
$$z = \bar{d} + \sum_{j \in N} \bar{c}_j x_j$$

- If basic optimal solution to LPR is not integer then $\exists$ some row u: $b_u \notin \mathbb{Z}^1$.
  The Chvatál-Gomory cut applied to this row is:

$$x_{B_u} + \sum_{j \in N} \lfloor \bar{a}_{uj} \rfloor x_j \leq \lfloor \bar{b}_u \rfloor$$

(B_u is the index in the basis B corresponding to the row u) (cntd)

- Eliminating $x_{B_u} = \bar{b}_i - \sum_{j \in N} \bar{a}_{uj} x_j$ in the CG cut we obtain:

$$\sum_{j \in N} \underbrace{(\bar{a}_{uj} - \lfloor \bar{a}_{uj} \rfloor)}_{0 \leq f_{uj} < 1} x_j \geq \underbrace{\bar{b}_u - \lfloor b_u \rfloor}_{0 < f_u < 1}$$

$$\sum_{j \in N} f_{uj} x_j \geq f_u$$

  $f_u > 0$ or else u would not be row of fractional solution. It implies that x^* in which $x^*_N = 0$ is cut out!

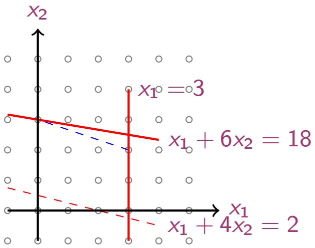
- Moreover: when x is integer, since all coefficient in the CG cut are integer the slack variable of the cut is also integer:

$$s = -f_u + \sum_{j \in N} f_{uj} x_j$$

(theoretically it terminates after a finite number of iterations, but in practice not successful.)

Example

$$\begin{aligned} \max\ & x_1 + 4x_2 \\ & x_1 + 6x_2 \leq 18 \\ & x_1 \qquad\ \ \leq 3 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \text{integer} \end{aligned}$$



	x1	x2	x3	x4	-z	b
	1	6	1	0	0	18
	1	0	0	1	0	3
	1	4	0	0	1	0

	x1	x2	x3	x4	-z	b
	0	1	1/6	-1/6	0	15/6
	1	0	0	1	0	3
	0	0	-2/3	-1/3	1	-13

$x_2 = 5/2, x_1 = 3$
Optimum, not integer

- We take the first row:
  `|     |   0  |   1  | 1/6   | -1/6 |   0  | 15/6  |`

- CG cut $\sum_{j \in N} f_{uj} x_j \geq f_u \rightsquigarrow \frac{1}{6}x_3 + \frac{5}{6}x_4 \geq \frac{1}{2}$

- Let's see that it leaves out x^*: from the CG proof:

$$\begin{array}{l} 1/6\ (x_1 + 6x_2 \leq 18) \\ 5/6\ (x_1 \qquad\quad \leq 3) \\ \hline \quad x_1 + x_2 \leq 3 + 5/2 = 5.5 \end{array}$$

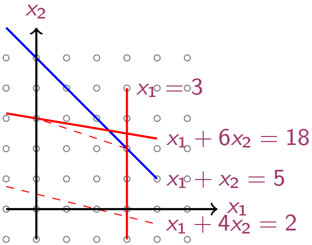
  since x_1, x_2 are integer $x_1 + x_2 \leq 5$

- Let's see how it looks in the space of the original variables: from the first tableau:

$$x_3 = 18 - 6x_2 - x_1$$
$$x_4 = 3 - x_1$$

$$\frac{1}{6}(18 - 6x_2 - x_1) + \frac{5}{6}(3 - x_1) \geq \frac{1}{2} \qquad \rightsquigarrow \qquad x_1 + x_2 \leq 5$$

- Graphically:



- Let's continue:

	x1	x2	x3	x4	x5	-z	b
	0	0	-1/6	-5/6	1	0	-1/2
	0	1	1/6	-1/6	0	0	5/2
	1	0	0	1	0	0	3
	0	0	-2/3	-1/3	0	1	-13

We need to apply dual-simplex (will always be the case, why?)

ratio rule: $\min \left| \frac{c_j}{a_{ij}} \right|$

- After the dual simplex iteration:

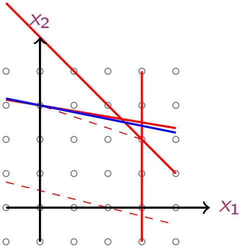
	x1	x2	x3	x4	x5	-z	b
	0	0	1/5	1	-6/5	0	3/5
	0	1	1/5	0	-1/5	0	13/5
	1	0	-1/5	0	6/5	0	12/5
	0	0	-3/5	0	-2/5	1	-64/5

We can choose any of the three rows.

Let's take the third: CG cut:
$\frac{4}{5}x_3 + \frac{1}{5} \geq \frac{2}{5}$

- In the space of the original variables:

$$4(18 - x_1 - 6x_2) + (5 - x_1 - x_2) \geq 2$$
$$x_1 + 5x_2 \leq 15$$



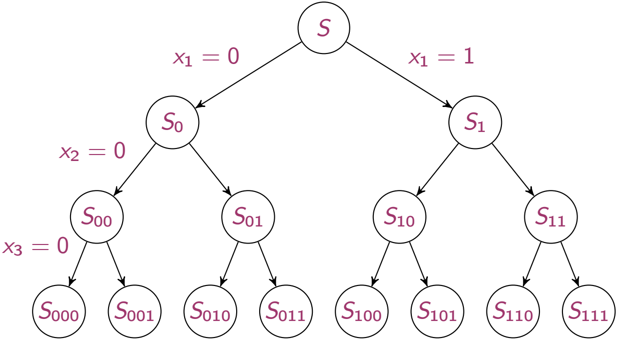
- ...

Outline

Branch and Bound

- Consider the problem $z = \max\{c^T x : x \in S\}$
- Divide and conquer: let $S = S_1 \cup \ldots \cup S_k$ be a decomposition of S into smaller sets, and let $z^k = \max\{c^T x : x \in S_k\}$ for $k = 1, \ldots, K$. Then $z = \max_k z^k$

For instance if $S \subseteq \{0,1\}^3$ the enumeration tree is:

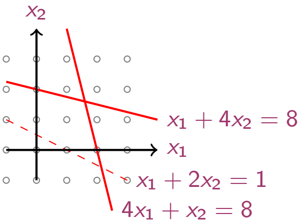


Bounding

- Let $\overline{z}^k$ be an upper bound on z^k
- Let $\underline{z}^k$ be an lower bound on z^k
- $(\underline{z}^k \leq z^k \leq \overline{z}^k)$
- $\overline{z} = \max_k \overline{z}^k$ is an upper bound on z
- $\underline{z} = \max_k \underline{z}^k$ is a lower bound on z

Example

$$\begin{aligned} \max\ & x_1 + 2x_2 \\ & x_1 + 4x_2 \leq 8 \\ & 4x_1 + x_2 \leq 8 \\ & x_1, x_2 \geq 0, \text{integer} \end{aligned}$$



- Solve LP

	x1	x2	x3	x4	-z	b
	1	4	1	0	0	8
	4	1	0	1	0	8
	1	2	0	0	1	0

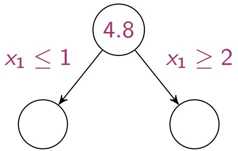
	x1	x2	x3	x4	-z	b
I'=I-II'	0	15/4	1	-1/4	0	6
II'=1/4II	1	1/4	0	1/4	0	2
III'=III-II'	0	7/4	0	-1/4	0	-2

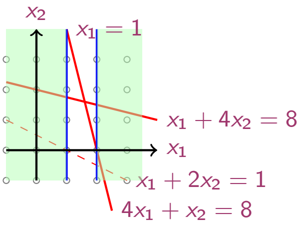
- continuing

	x1	x2	x3	x4	-z	b
I'=4/15I	0	1	4/15	-1/15	0	24/15
II'=II-1/4I'	1	0	-1/15	4/15	0	24/15
III'=III-7/4I'	0	0	-7/15	-3/5	1	-2-14/5

$x_2 = 1 + 3/5 = 1.6$
$x_1 = 8/5$
The optimal solution will not be more than $2 + 14/5 = 4.8$

- Both variables are fractional, we pick one of the two:





- Let's consider first the left branch:

```
|   | x1 | x2 | x3      | x4      | x5 | b | -z     |
|---+----+----+-------+-------+----+---+-------|
|   |  1 |  0 | 0       | 0       |  1 | 0 | 1      |
|   |  0 |  1 | 4/15    | -1/15 |  0 | 0 | 24/15 |
|   |  1 |  0 | -1/15 | 4/15    |  0 | 0 | 24/15 |
|---+----+----+-------+-------+----+---+-------|
|   |  0 |  0 | -7/15 | -3/5    |  0 | 1 | -24/5 |
```

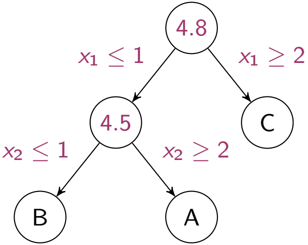
```
|           | x1 | x2 | x3      | x4      | x5 | b | -z     |
|----------+----+----+-------+-------+----+---+-------|
| I'=I-III |  0 |  0 | 1/15    | -4/15 |  1 | 0 | -9/15 |
|           |  0 |  1 | 4/15    | -1/15 |  0 | 0 | 24/15 |
|           |  1 |  0 | -1/15 | 4/15    |  0 | 0 | 24/15 |
|----------+----+----+-------+-------+----+---+-------|
|           |  0 |  0 | -7/15 | -3/5    |  0 | 1 | -24/5 |
```

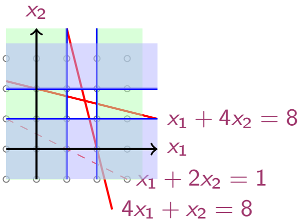
```
|   | x1 | x2 | x3       | x4 | x5      | b | -z      |
|---+----+----+--------+----+-------+---+--------|
|   |  0 |  0 | -1/4     |  1 | -15/4 | 0 | 9/4     |
|   |  0 |  1 | 15/60    |  0 | -1/4    | 0 | 7/4     |
|   |  1 |  0 | 0        |  0 | 1       | 0 | 1       |
|---+----+----+--------+----+-------+---+--------|
|   |  0 |  0 | -37/60 |  0 | -9/4    | 1 | -90/20 |
```

always a b term negative after branching:
$\bar{b}_1 = \lfloor \bar{b}_3 \rfloor$
$\bar{b}_1 = \lfloor \bar{b}_3 \rfloor - b_3 < 0$

Dual simplex:
$\min_j |\frac{c_j}{a_i j}|$

- Let's branch again





We have three open problems. Which one we choose next?
Let's take A.

	x1	x2	x3	x4	x5	x6	b	-z
	0	-1	0	0	0	1	0	-2
	0	0	-1/4	1	-15/4		0	9/4
	0	1	15/60	0	-1/4		0	7/4
	1	0	0	0	1		0	1
	0	0	-37/60	0	-9/4		1	-9/2

	x1	x2	x3	x4	x5	x6	b	-z
III-I	0	0	1/4	0	-1/4	1	0	-1/4
	0	0	-1/4	1	-15/4		0	9/4
	0	1	15/60	0	-1/4		0	7/4
	1	0	0	0	1		0	1
	0	0	-37/60	0	-9/4		1	-9/2

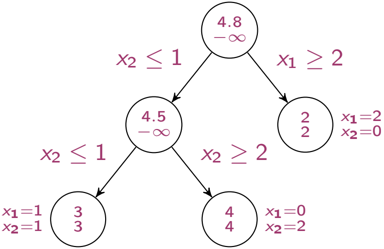
continuing we find:

$x_1 = 0$

$x_2 = 2$

$OPT = 4$

The final tree:

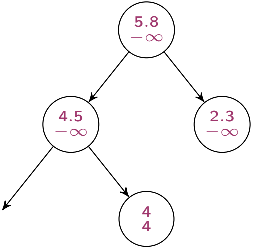


The optimal solution is 4.

Pruning

Pruning:

1. by optimality: $z^k = \max\{c^T x : x \in S^k\}$
2. by bound $\overline{z}^k \leq \underline{z}$
   Example:



3. by infeasibility $S^k = \emptyset$

B&B Components

Bounding:

1. LP relaxation
2. Lagrangian relaxation
3. Combinatorial relaxation
4. Duality

Branching:

$$S_1 = S \cap \{x : x_j \leq \lfloor \bar{x}_j \rfloor\}$$
$$S_2 = S \cap \{x : x_j \geq \lceil \bar{x}_j \rceil\}$$

thus the current optimum is not feasible either in S_1 or in S_2.
Which variable to choose?
Eg: Most fractional variable $\arg\max_{j \in C} \min\{f_j, 1 - f_j\}$

Choosing Node: Examination: nodes to be examined, active (or open):

- Depth First Search (a good primal sol. is good for pruning + easier to reoptimize by just adding a new constraint)
- Best Bound First: (eg. largest upper: $\overline{z}^s = \max_k \overline{z}^k$)
- Mixed strategies

Reoptimizing: dual simplex

Updating the Incumbent: when new best feasible solution is found:

$$\underline{z} = \max\{\underline{z}, 4\}$$

Store the active nodes: bounds + optimal basis (remember the revised simplex!)

Enhancements

- Preprocessor: constraint/problem/structure specific
  tightening bounds
  redundant constraints
  variable fixing: eg: $\max\{c^T x : Ax \leq b, l \leq x \leq u\}$
  fix $\forall a_{ij} > 0, c_j < 0, x_j = l_j; a_{ij} < 0, c_j > 0, x_j = u_j$

- Priorities: establish the next variable to branch

- Special ordered sets SOS (or generalized upper bound GUB)

$$\sum_{j=1}^{k} x_j = 1 \qquad x_j \in \{0, 1\}$$

  instead of: $S_0 = S \cap \{x : x_j = 0\}$ and $S_1 = S \cap \{x : x_j = 1\}$
  $\{x : x_j = 0\}$ leaves $k - 1$ possibilities
  $\{x : x_j = 1\}$ leaves only 1 possibility
  hence tree unbalanced
  here: $S_1 = S \cap \{x : x_{j_i} = 0, i = 1..r\}$ and
  $S_2 = S \cap \{x : x_{j_i} = 0, i = r+1, .., k\}$, $r = \min\{t : \sum_{i=1}^{t} x^*_{j_i} \geq \frac{1}{2}\}$

- Cutoff value: a user-defined primal bound to pass to the system.
- Simplex strategies: simplex is good for reoptimizing but for large models interior points methods may work best.
- Strong branching: extra work to decide more accurately on which variable to branch:
  1. choose a set C of fractional variables
  2. reoptimize for each them (in case for limited iterations)
  3. $\overline{z}_j^D, \overline{z}_j^U$ (UB of down and up branch)

  $$j^* = \arg\min_{j \in C} \max\{z_j^D, z_j^U\}$$

  ie, choose variable with largest decrease of dual bound, UB

- If not finished after a certain time:
  - no feasible solution is found
  - the gap best feasible-dual bound is large

$$GAP = \frac{|\text{Primal bound} - \text{Dual Bound}|}{\text{Primal bound} + \epsilon} \cdot 100$$

  - runs out of memory

- heuristics for finding feasible solutions (generally NP-complete problem)

- find better lower bounds if they are weak: addition of cuts, stronger formulation, branch and cut

- Branch and cut: a B&B algorithm with cut generation at all nodes of the tree. (instead of reoptimizing, do as much work as possible to tighten)

  Cut pool: stores all cuts centrally
  Store for active node: bounds, basis, pointers to constraints in the cut pool that apply at the node

Advanced Techniques

We did not treat:

- LP: Dantzig Wolfe decomposition
- LP: Column generation
- LP: Delayed column generation
- IP: Branch and Price
- LP: Benders decompositions
- LP: Lagrangian relaxation

Summary

1. Cutting Plane Algorithms

2. Branch and Bound