DM545/DM871

Linear and Integer Programming

Branch and Bound

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

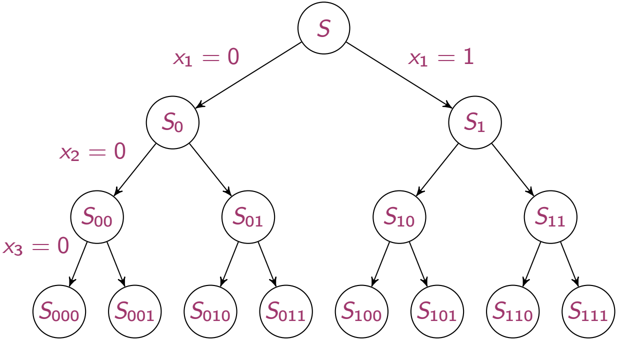
1. Branch and Bound

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:

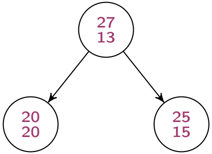


Bounding

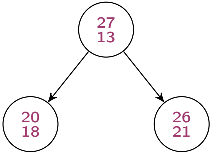
Let's consider a maximization problem

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

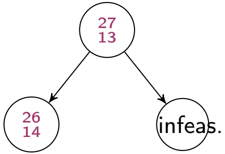
Pruning (Fathoming)



$\overline{z} = 25$
$\underline{z} = 20$
pruned by optimality

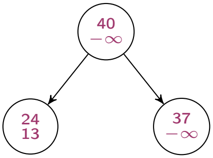


$\overline{z} = 26$
$\underline{z} = 21$
pruned by bounding



$\overline{z} = 26$
$\underline{z} = 14$
pruned by infeasibility

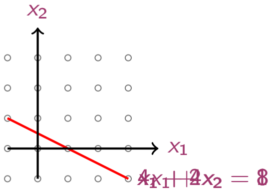
Pruning



$\overline{z} = 37$
$\underline{z} = 13$
nothing to prune

LP Based Branch & Bound: Example

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



- 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 |
```

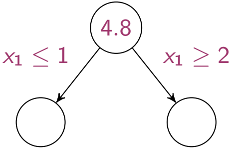
```
|              | x1 | x2   | x3 | x4   | -z |  b |
|--------------+----+------+----+------+----+----|
| I'=I-II'     |  0 | 15/4 |  1 | -1/4 |  0 |  6 |
| II'=1/4II    |  1 | 1/4  |  0 | 1/4  |  0 |  2 |
|--------------+----+------+----+------+----+----|
| III'=III-II' |  0 | 7/4  |  0 | -1/4 | 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

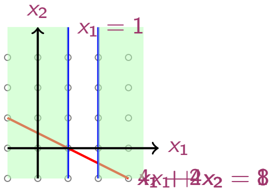
- continuing

	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	-z	b
	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
I'=-15/4I	0	0	-1/4	1	-15/4	0	9/4
II'=II-1/4I	0	1	15/60	0	-1/4	0	7/4
III'=III+I	1	0	0	0	1	0	1
	0	0	-37/60	0	-9/4	1	-90/20

always a b term negative after branching:

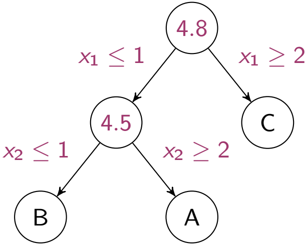
$b_1 = \lfloor \bar{b}_3 \rfloor$

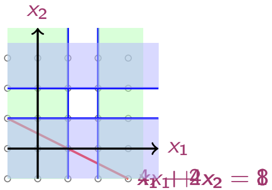
$\bar{b}_1 = \lfloor \bar{b}_3 \rfloor - b_3 < 0$

Dual simplex:

$\min_j \{ |\frac{c_j}{a_{ij}}| \; : \; a_{ij} < 0 \}$

- 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

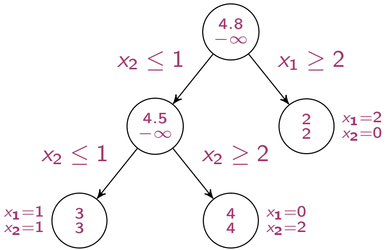
continuing we find:

$x_1 = 0$

$x_2 = 2$

$OPT = 4$

The final tree:



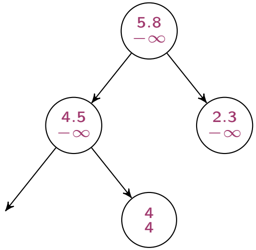
The optimal solution is 4.

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 in S_1 and in S_2.
Which variable to choose?
Eg: Most fractional variable $\arg\max_{j \in C} \min\{f_j, 1 - f_j\}$

Choosing Node for Examination from the list of active (or open):

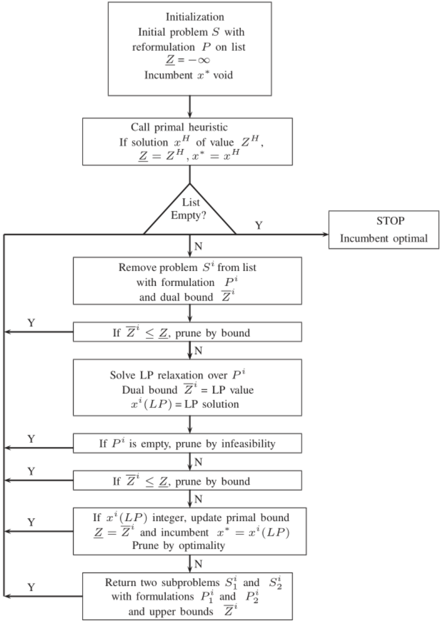
- Depth First Search (a good primal sol. is good for pruning + easier to reoptimize by just adding a new constraint)
- Best Bound First: (eg. largest upper: $\overline{z}^s = \max_k \overline{z}^k$ or largest lower - to die fast)
- Mixed strategies

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!)



[Wolsey, 2021]
Lazy strategy: pruning, calculation, branching, queue insertion; open nodes are stored with the bound of their father

Branch-and-Bound Algorithm ;
begin
1. $m := 1$; $parent[1] := 0$; $Q := \emptyset$;
$z_{OPT} :=$ heuristic solution value (possibly $+\infty$);
2. solve the continuous relaxation $\min\{\mathbf{c}^T\mathbf{x} \,:\, A\mathbf{x} = \mathbf{b}\,,\ \mathbf{x} \geq 0\}$, and let $\mathbf{x}^*$ be the optimal solution found;
3. $LB[1] := \mathbf{c}^T\mathbf{x}^*$;
4. **if** ($\mathbf{x}^*$ integer) **and** ($\mathbf{c}^T\mathbf{x}^* < z_{OPT}$) **then**
begin
$\mathbf{x}_{OPT} = \mathbf{x}^*$; $z_{OPT} := \mathbf{c}^T\mathbf{x}^*$
end ;
5. **if** $LB[1] < z_{OPT}$ **then**
begin
choose the fractional branching variable x_h^* ;
$vbranch[1] := h$; $value[1] := x_h^*$;
$Q := \{1\}$
end ;
6. **while** $Q \neq \emptyset$ **do** /* process the active open nodes*/
begin
7. choose a node $t \in Q$, and set $Q := Q \setminus \{t\}$;
8. $h := vbranch[t]$; $val := value[t]$;
9. **for** $child := 1$ **to** 2 **do** /* generate the children of node t */
begin
10. $m := m + 1$;
if $child = 1$
then $parent[m] := t$
else $parent[m] := -t$;
11. define problem PL_m associated with node m
(constraints of PL_t plus $x_h \leq \lfloor val \rfloor$ if $child = 1$,
or $x_h \geq \lceil val \rceil$ if $child = 2$) ;
12. solve problem PL_m, and let $\mathbf{x}^*$ be the optimal solution found ;
13. $LB[m] := \mathbf{c}^T\mathbf{x}^*$;
14. **if** ($\mathbf{x}^*$ integer) **and** ($\mathbf{c}^T\mathbf{x}^* < z_{OPT}$) **then**
begin /* update the optimal solution */
$\mathbf{x}_{OPT} := \mathbf{x}^*$; $z_{OPT} := \mathbf{c}^T\mathbf{x}^*$;
$Q := Q \setminus \{j \in Q \,:\, LB[j] \geq z_{OPT}\}$
end ;
15. **if** $LB[m] < z_{OPT}$ **then**
begin
choose the fractional branching variable x_k^* ;
$vbranch[m] := k$; $value[m] := x_k^*$;
$Q := Q \cup \{m\}$
end
end
end
end .

Warning: this is for a minimization problem

[Fischetti, 2019]
Eager strategy: branching, calculation, pruning, queue insertion; open nodes are stored with their own bounds

Enhancements

- Preprocessor: constraint/problem/structure specific
  tightening bounds
  redundant constraints
  variable fixing: eg: $\max\{\mathrm{c}^T\mathrm{x} : A\mathrm{x} \leq \mathrm{b}, \mathrm{l} \leq \mathrm{x} \leq \mathrm{u}\}$
  fix $x_j = l_j$ if $c_j < 0$ and $a_{ij} > 0$ for all i
  fix $x_j = u_j$ if $c_j > 0$ and $a_{ij} < 0$ for all i
- User defined branching priorities: establish the next variable to branch (not in gurobi)
- Special ordered sets SOS (or generalized upper bound GUB)

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

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

- 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 of them (in case for limited iterations)
  3. $\overline{z}_j^{\downarrow}, \overline{z}_j^{\uparrow}$ (dual bound of down and up branch)

$$j^* = \arg\min_{j \in C} \max\{\overline{z}_j^{\downarrow}, \overline{z}_j^{\uparrow}\}$$

  ie, choose variable with most change in objective function, ie, largest decrease of dual bound, eg, largest decrease of UB for max problem

There are four common reasons why integer programs can require a significant amount of solution time:

1. There is lack of node throughput due to troublesome linear programming node solves.
2. There is lack of progress in the best integer solution, i.e., the primal bound.
3. There is lack of progress in the best dual bound.
4. There is insufficient node throughput due to numerical instability in the problem data or excessive memory usage.

For 2) or 3) the gap best feasible-dual bound is large:

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

- 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

Relative Optimality Gap

In CPLEX:

$$\text{gap} = \frac{|\text{best dual bound} - \text{best integer}|}{|\text{best integer} + 10^{-11}|}$$

In SCIP and MIPLIB standard:

$$\text{gap} = \frac{pb - db}{\inf\{|z|, z \in [db, pb]\}} \cdot 100 \qquad \text{for a minimization problem}$$

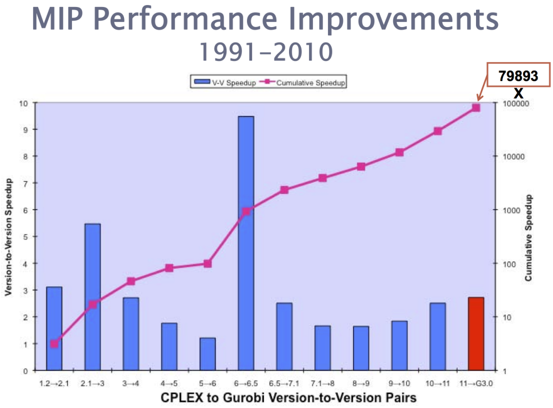
(if $pb \geq 0$ and $db \geq 0$ then $\frac{pb-db}{db}$)
if $db = pb = 0$ then $\text{gap} = 0$
if no feasible sol found or $db \leq 0 \leq pb$ then the gap is not computed.

Last standard avoids problem of non decreasing gap if we go through zero

```
   3186  2520     -666.6217  4096      956.6330     -667.2010  1313338  169.74%
   3226  2560     -666.6205  4097      956.6330     -667.2010  1323797  169.74%
   3266  2600     -666.6201  4095      956.6330     -667.2010  1335602  169.74%
Elapsed real time = 2801.61 sec. (tree size = 77.54 MB, solutions = 2)
*  3324+ 2656                         -125.5775     -667.2010  1363079  431.31%
   3334  2668     -666.5811  4052     -125.5775     -667.2010  1370748  431.31%
   3380  2714     -666.5799  4017     -125.5775     -667.2010  1388391  431.31%
   3422  2756     -666.5791  4011     -125.5775     -667.2010  1403440  431.31%
```

MILP Solvers Breakthroughs

We have seen Fractional Gomory cuts.
The introduction of Mixed Integer Gomory cuts in CPLEX was the major breakthrough of CPLEX 6.5 and produced the version-to-version speed-up given by the blue bars in the chart below



(source: R. Bixby. Mixed-Integer Programming: It works better than you may think. 2010. Slides on the net)

Speedup over the past 25 years:

Hardware 2 000 times
Software 2 000 000 times

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

They are topics of DM872.

In the Exercises

Solve linear programming problems with:

- Simple method tool `https://www.zweigmedia.com/simplex/simplex.php`
- glpk `https://dm871.github.io/notes/glpk.html`
- In Python: scipy.optimize.linprog
- In Python: Python-MIP, pyscipopt, gurobipy

Summary

1. Branch and Bound