AM 121: Intro to Optimization Models and Methods

Fall 2019

Lecture 15: More cuts, Branch and Cut, other tricks…

Yiling Chen
SEAS

1

Lesson Plan

- Review Chvatal-Gomory cuts
- Cover cuts
- Branch-and-cut
- Other tricks
  - Preprocessing
  - Primal heuristics
  - SOS branching
- CPLEX and cut generation

Jensen & Bard: 8.5

2

Chvátal-Gomory inequalities

- Consider $X = P \cap Z^2$, where P is given by

$$7x_1 - 2x_2 \le 14$$
$$x_2 \le 3$$
$$2x_1 - 2x_2 \le 3$$
$$x \ge 0$$

- Combine with weights u=(2/7, 37/63, 0)

$$2x_1 + 1/63x_2 \le 121/21 \qquad (*)$$

- Round coefficients on LHS down to nearest integer

$$2x_1 + 0x_2 \le 121/21$$

- Use integrality to write:

$$2x_1 \le 5$$

3

Example (Gomory's algorithm)

- Consider the IP

$$z = \max 4x_1 - x_2$$
$$7x_1 - 2x_2 \le 14$$
$$x_2 \le 3$$
$$2x_1 - 2x_2 \le 3$$
$$x_1, \quad x_2 \ge 0, \text{ integer}$$

4

- For row i with fractional RHS, the CG cut is $\sum_{j \in B'} \pi_j x_j \geq \pi_0$; with $\pi_j = a_{ij} - \lfloor a_{ij} \rfloor$ and $\pi_0 = b_i - \lfloor b_i \rfloor$

- $$\begin{array}{lllllll} z & & + 4/7\, x_3 & + 1/7\, x_4 & & = 59/7 \\ & x_1 & + 1/7 x_3 & + 2/7\, x_4 & & = 20/7 \\ & & x_2 & & + x_4 & = 3 \\ & & & -2/7\, x_3 + 10/7 x_4 & + x_5 & = 23/7 \end{array}$$
- B={1, 2, 5}
- Add cut $1/7 x_3 + 2/7 x_4 \geq 6/7$

5

- Add cut, re-optimize. New optimal tableau:

$$\begin{array}{lllllll} z & & & & +\tfrac{1}{2}x_5 & +3x_6 & = 15/2 \\ & x_1 & & & & +x_6 & = 2 \\ & & x_2 & & -\tfrac{1}{2}x_5 & +x_6 & = \tfrac{1}{2} \\ & & & x_3 & -x_5 & -5x_6 & = 1 \\ & & & x_4 & +\tfrac{1}{2}x_5 & +6x_6 & = 5/2 \end{array}$$

- B={1,2,3,4} and $x^* = (2, 1/2, 1, 5/2, 0, 0)$
- Add cut $\tfrac{1}{2}x_5 \geq \tfrac{1}{2}$

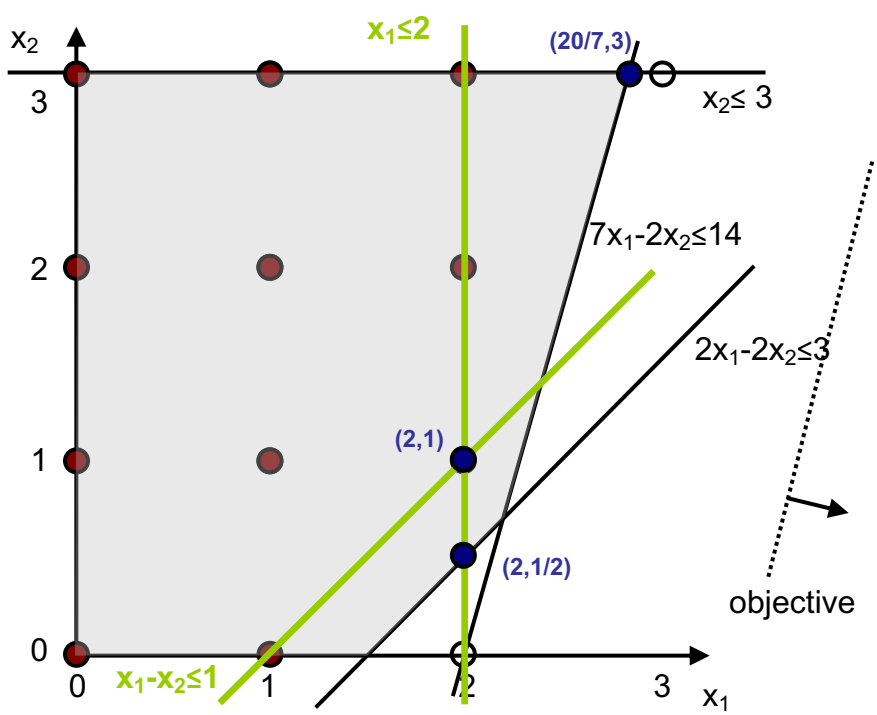
6

- Add cut, re-optimize. New optimal tableau:

$$
\begin{array}{lllllllll}
z & & & & & +3x_6 & + & x_7 & = 7 \\
 & x_1 & & & & +x_6 & & & = 2 \\
 & & x_2 & & & +x_6 & - & x_7 & = 1 \\
 & & & x_3 & & -5x_6 & - & 2x_7 & = 2 \\
 & & & & x_4 & +6x_6 & + & x_7 & = 2 \\
 & & & & x_5 & & - & x_7 & = 1
\end{array}
$$

- $x^*=(2, 1, 2, 2, 1, 0, 0)$

7



8

Cover inequalities

- Defined for problems in which the feasible space is a "0-1 Knapsack set". Example:
- $X=\{x\in\{0,1\}^7 : \underbrace{11x_1+6x_2+6x_3+5x_4+5x_5+4x_6+x_7 \le 19}_{\text{positive coefficients, positive RHS}}\}$

  0/1 variables

- Some (minimal) cover inequalities for X are:

$$
\begin{aligned}
x_1 + x_2 + x_3 \qquad\qquad\qquad &\le 2\\
x_1 + x_2 \qquad\qquad\qquad + x_6 &\le 2\\
x_1 \qquad\qquad + x_5 + x_6 &\le 2\\
x_3 + x_4 + x_5 + x_6 &\le 3
\end{aligned}
$$

9

Cover inequalities

- For $X = \{x\in\{0,1\}^n : \sum_j a_j x_j \le b\}$, with $a_j>0$, $b>0$.
- $N=\{1,\ldots,n\}$
- **Defn**. A set $C\subseteq N$ is a **cover** if $\sum_{j\in C} a_j > b$. A cover is **minimal** if $C \setminus \{j\}$ is not a cover for any $j\in C$
- **Proposition.** If C is a cover, the *cover inequality* $\sum_{j\in C} x_j \le |C|-1$ is valid.
- **Proof.** Consider $x'\in\{0,1\}^n$ with $\sum_{j\in C} x'_j > |C|-1$. Have

$$\sum_j a_j x'_j \ge \sum_{j\in C} a_j x'_j = \sum_{j\in C} a_j > b$$

$a_j>0$ integrality cover

Therefore, this x' is infeasible.

10

Two Variations on Cover Inequal

- Extended Cover Inequalities
- Lifted Cover Inequalities

11

Extended Cover Inequalities

- $X=\{x\in\{0,1\}^7: 11x_1+6x_2+6x_3+5x_4+5x_5+4x_6+x_7\leq 19\}$
- Cover inequality: $x_3 + x_4 + x_5 + x_6 \leq 3$
- Extended cover ineq: $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$
- Still valid. Why? Any four variables have sum coefficients > 19.

- **Rule**: Can **extend** C to E(C), where
  $E(C)= C\cup\{ j : a_j \geq a_i \text{ for all } i\in C\}$.
- Still valid; stronger.

12

Lifted Cover Inequalities

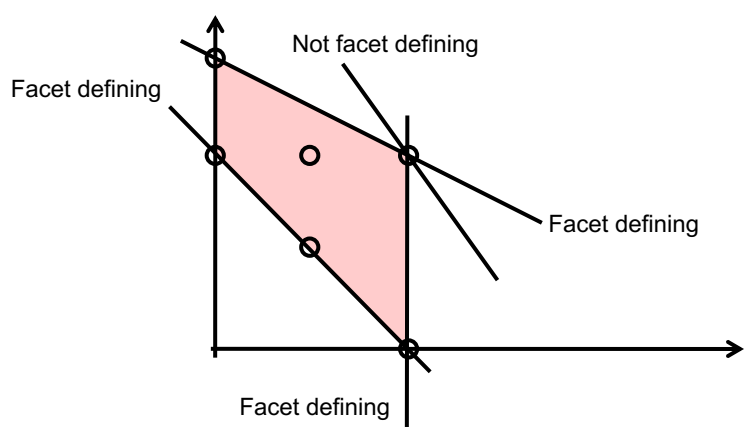
- $X=\{x\in\{0,1\}^7: 11x_1+6x_2+6x_3+5x_4+5x_5+4x_6+x_7\leq 19\}$
- **Lifting**: Given $x_2+x_3+x_4+x_5+x_6\leq 3$, find a valid inequality $\boldsymbol{\alpha_1 x_1}+x_2+x_3+x_4+x_5+x_6\leq 3$, for some $\alpha_1>0$.

- Valid for any soln with $x_1=0$. If $x_1=1$, we know x satisfies: $6x_2+6x_3+5x_4+5x_5+4x_6+x_7\leq 8$ (*)

- **Idea**: pick maximal α_1 s.t. $\alpha_1+x_2+x_3+x_4+x_5+x_6\leq 3$ for all integer solutions satisfying (*)
- Set $\alpha_1= 3 - \max_x \{x_2+x_3+x_4+x_5+x_6:$
  $6x_2+6x_3+5x_4+5x_5+4x_6+x_7\leq 8,$
  $x \in \{0,1\}^5\} = 3 - 1 = 2$
- **Lifted cover inequal**: $2x_1+x_2+x_3+x_4+x_5+x_6 \leq 3$

13

- **Theorem**. Lifted cover inequalities are "facet-defining" for conv(X).
- (Informal) An inequality is **facet defining** if it defines the face of a polyhedron and is as high dimension as possible. Consider:

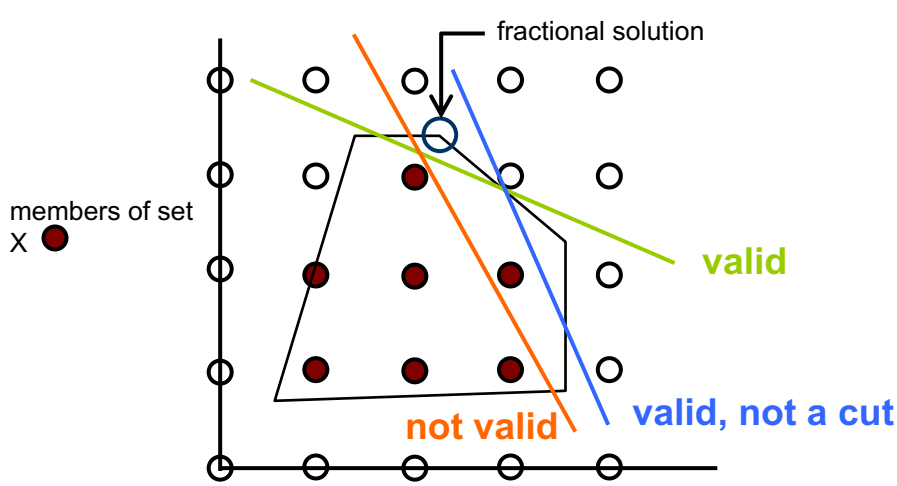


14

How can we use Cover inequalities?

15

- **Defn**. A **cut** is a valid inequality that **separates** the current fractional solution x^*.



16

Separation with Cover inequalities

- Can write cover inequality as
  $\sum_{j\in C} (1 - x_j) \geq 1$ ("at least one not used")
- Given x*, find valid C with $\sum_{j\in C}(1-x^*_j)<1$

- Formulate an IP. Let z_j denote whether $j\in C$.
  $\beta = \min \sum_{j\in N} (1-x^*_j)z_j$
  s.t. $\sum_{j\in N} a_j z_j > b$ (valid cover)
  $z \in \{0,1\}^n$
- If $\beta \geq 1$ then x* satisfies all cover inequalities, if $\beta<1$ then we find a cut.

- Note: this IP is *nonstandard* **(strict** inequality.) But can replace with $\sum_{j\in N} a_j z_j \geq b+1$ when coeffs are integral.

17

Example: Separation

- $X=\{x\in\{0,1\}^6: 45x_1+46x_2+79x_3+54x_4+53x_5+125x_6\leq 178\}$, fractional $x^*=(0, 0, 3/4, 1/2, 1, 0)$
- Solve:
  $\beta = \min z_1 + z_2 + \frac{1}{4} z_3 + \frac{1}{2} z_4 + 0z_5 + z_6$
  s.t. $45z_1+46z_2+79z_3+54z_4+53z_5+125z_6>178$
  $z\in\{0,1\}^6$
- $z^*=(0,0,1,1,1,0)$ with $\beta=3/4$
- Conclude that $x_3 + x_4 + x_5 \leq 2$ is a cut for x*.

- Why is this useful? Because we can try to solve this "auxiliary problem" heuristically and give up if it is too hard.
- Can also combine with **lifting** (complete for 0/1 knapsack sets)

18

- We’ve seen two families of cuts:
  - Chvatal-Gomory cuts
  - Cover cuts
- There are many others:
  - Flow cover
  - Mixed integer rounding
  - Odd hole
  - …
- How can we use these?

19

The Branch-and-Cut Method

20

Reminder

- **Valid inequality**: an inequality satisfied by all feasible solutions
- **Cut**: a valid inequality that is not part of the current formulation

21

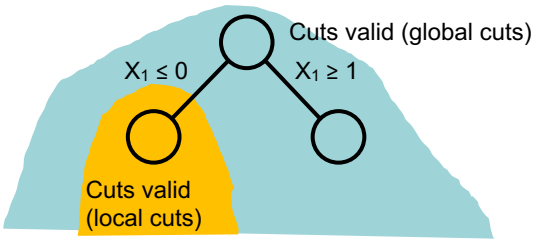
The Branch-and-Cut method

- Generalizes Branch-and-Bound.
- If cannot prune a node, can try to find a **cut.**
- Strengthen the formulation; re-solve the LP, may now be able to prune by bound.
- If no cuts are found, then branch.

22

The Branch-and-Cut method

- Branch-and-Bound + **cuts generated throughout the tree**.
- Cuts generated at root are "global," cuts at subproblems "local." Place into a "cut pool."
- **Goal**: reduce the number of search nodes by improving <u>bounds</u>



23

Example: Branch and Cut

- Generalized assignment problem
- 10 firms, 5 distinct goods (≈100 units each)
- Each firm demands a particular quantity of each good
- Goal: maximize total value

24

Firms		**Goods**				
		A	*B*	*C*	*D*	*E*
1	demand	95	1	21	…	
	value	110	16	25	…	
2	demand	54	53	…		
	value	65	69	…		
3	demand	3	91	…		
	value	19	93	…		
…						
10		69	83	…		45
		88	97	…		51
Supply		91	87	109	88	64

25

Assignment problem

- Firms i, goods j with supply b_j
- c_{ij} value to i for allocation of a_{ij} units of good j
- x_{ij}=1 if a_{ij} units of good j assigned to agent i

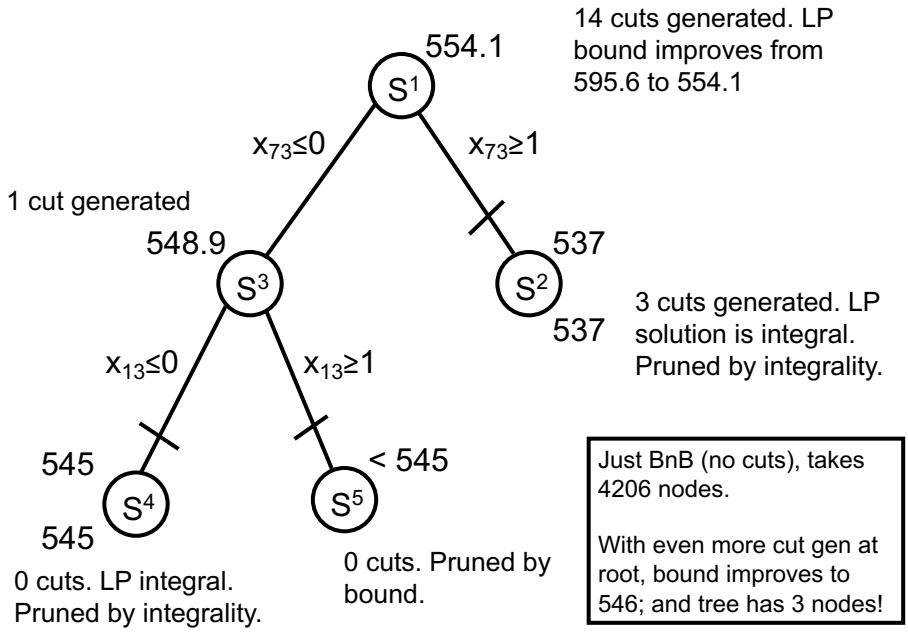
$$\max \sum_i \sum_j c_{ij} x_{ij}$$
$$\text{s.t. } \sum_j x_{ij} \le 1 \quad \text{for all } i = 1,\ldots,m$$
$$\sum_i a_{ij}x_{ij} \le b_j \quad \text{for all } j = 1,\ldots,n$$
$$x_{ij} \in \{0,1\}$$

Has “0/1 knapsack set” structure. Can use Cover cuts.

26

- Generate lifted cover cuts (global and local)

27



28

Historical Context

- Use of cutting planes as a practical tool for solving IPs had been abandoned in ‘60s and ‘70s
- Revisited in early ‘70s following NP-completeness theory (interest in problem structure and facets.)
- Balas, Wolsey, Johnson, Padberg in mid ‘70s developed cover inequalities and “lifting”
- **Branch-and-cut** developed in early ‘80s, and introduced into commercial solvers in 2000s
- Has provided an **order of magnitude** improvement in solve times!

29

Other important components

- Pre-processing
- Primal heuristics
- SOS branching

30

(1 of 3) Preprocessing

- Detect and eliminate redundant constraints and variables, tighten bounds where possible

31

- Consider LP: (Wolsey pp.104)
- max $2x_1 + x_2 - x_3$

  $5x_1 - 2x_2 + 8x_3 \le 15$ (a)

  $8x_1 + 3x_2 - x_3 \ge 9$ (b)

  $x_1 + x_2 + x_3 \le 6$ (c)

  $0 \le x_1 \le 3;\ 0 \le x_2 \le 1;\ 1 \le x_3$
- Isolate x_1 in (a) to obtain $5x_1 \le 15+2x_2 -8x_3 \le 15+2-8=9$, and so $x_1 \le 9/5$.
- Isolate x_3 in (a) to obtain $x_3 \le 17/8$. Not useful to isolate x_2.
- Can also isolate x_1 in (b); obtain $x_1 \ge 7/8$.
- **Repeat**: improve bound on x_3 from (a) by using $x_1 \ge 7/8$; obtain $x_3 \le 101/64$

- Find constraint (c) is redundant, using $x_1 \le 9/5$ and $x_3 \le 101/64$, together with $x_2 \le 1$.

32

"Fixing" values of variables

The modified LP is:

- max $2x_1 + x_2 - x_3$

  s.t. $5x_1 - 2x_2 + 8x_3 \leq 15$

  $8x_1 + 3x_2 - x_3 \geq 9$

  $7/8 \leq x_1 \leq 9/5$; $0 \leq x_2 \leq 1$; $1 \leq x_3 \leq 101/64$

- **Fixing:** In optimal solution, have $x_2 = 1$; and $x_3 = 1$.
- Why? x_2 has positive coeff in obj, and increasing makes constraints less tight. Similarly, best to decrease x_3.

- Obtain problem: $\max\{2x_1 : 7/8 \leq x_1 \leq 9/5\}$. Easy to solve!

33

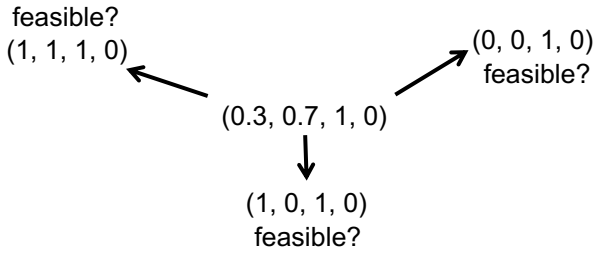
Other important components

- Pre-processing
- Primal heuristics
- SOS branching

34

(2 of 3) Primal heuristics

- At a fractional search node, use a heuristic approach to try to find an integral solution
- For example, round the solution or do a directed "neighborhood search."



- May be able to improve incumbent.

35

Other important components

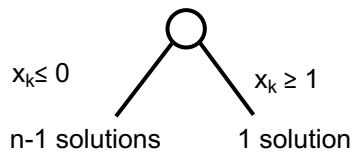
- Pre-processing
- Primal heuristics
- SOS branching

36

(3 of 3) SOS branching

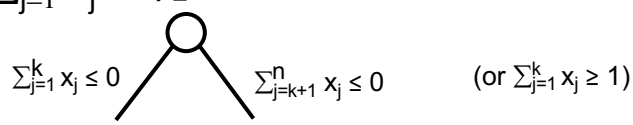
- Consider constraints $\sum_{j=1}^{n} x_j=1$, $x_j \in \{0,1\}$
- Typical to branch on fractional x_k^* :

unbalanced



- "Special-ordered sets" branching. Pick the first k s.t. $\sum_{j=1}^{k} x_j^* \geq ½$

balanced



Example: $x^*=(0,0.2,0.4,0,0.4)$. Branch $x_1 + x_2 + x_3 \leq 0$, $x_4 + x_5 \leq 0$

37

Looking at CPLEX

- IBM's Ilog CPLEX solver is used by AMPL
- Routinely used to solve real world problems of large economic significance

38

Some CPLEX features

- Automated Cut Generation
  - At "root" node (Global cuts)
  - At search nodes (Local cuts)
- Automated bound strengthening
  - Tighten right-hand side
- Primal heuristics
  - Look for integer feasible solutions that are close to current fractional solution

39

Example

```
var X1 integer >= 0;
var X2 integer >= 0;
maximize Obj: 5 * X1 + 8 * X2;
subject to C1: X1 + X2 <= 6;
subject to C2: 5 * X1 + 9 * X2 <= 45;
end;

ampl: model simple.mod;
ampl: option solver cplex;
ampl: solve;
```

40

Enabling feedback to AMPL

```
option cplex_options
   'timing = 1'        # display timing info
   'mipdisplay=2'      # display MIP information
   'mipinterval=1';    # node interval
```

timing=1:	show how much cpu time used to solve the problem
mipdisplay=2:	show # of open nodes
mipinterval=n:	display information every n nodes and whenever it finds an integer solution

41

```
Root relaxation solution time = 0.00 sec. (0.00 ticks)

        Nodes                                          Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                           0.0000       70.0000        2     ---
      0     0       41.2500     2        0.0000       41.2500        2     ---
*     0+    0                          39.0000       41.2500        2    5.77%
*     0+    0                          40.0000       41.2500        2    3.12%
      0     0        cutoff            40.0000                      2    0.00%
Elapsed time = 0.01 sec. (0.03 ticks, tree = 0.00 MB)

Root node processing (before b&c):
  Real time             =    0.01 sec. (0.03 ticks)
Parallel b&c, 16 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.01 sec. (0.03 ticks)

Times (seconds):
Input =  0.000782
Solve =  0.018569
Output = 0.000492
CPLEX 12.6.0.0: optimal integer solution; objective 40
2 MIP simplex iterations
0 branch-and-bound nodes
```

Node: current node
Nodes left: # of open nodes
Objective: value of current LPR
IInf: # integer infeasible variables
Best Integer: best solution found
Best Bound: current upper bound
ItCnt: total #pivots so far
Gap: relative gap between the best integer solution and best bound

42

Disabling Presolve

```
option presolve 0;
option cplex_options
      'timing = 1'
      'mipdisplay=2'
      'mipinterval=1'
      'boundstr=0'
      'dependency=0'
      'coeffreduce=0'
      'presolve=0'
      'cutpass= -1'
      'scale= -1'
      'prerelax= 0'
      'presolvenode = -1';
```

boundstr=0:	no bound strengthening
dependency=0:	no dependency checker in presolve
coeffreduce=0:	do not do coefficient reduction
cutpass =-1:	stop all cut generation
scale = -1:	do not scale the problem
prerelax = 0:	do not presolve at the initial LPR
presolvenode=-1:	do not presolve at each node

43

Disabling Primal Heuristics

```
option cplex_options
      'fpheur = -1'
      'heurfreq = -1'
      'rinsheur = -1';
      …
```

fpheur: Whether to use the feasibility pump heuristic on MIP problems (find initial feasible):
-1 = no
0 = automatic choice (default)

heurfreq: How often to apply "node heuristics" for MIPS
-1= no
20=every twenty nodes

rinsheur Relaxation INduced neighborhood Search HEURistic for MIP problems:
-1 = none
0 = automatic choice of interval (default)
n (for n > 0) = every n nodes.

44

```
Root relaxation solution time = 0.00 sec. (0.00 ticks)

        Nodes                                          Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

      0     0       41.2500     2                    41.2500        2
      0     2       41.2500     2                    41.2500        2
Elapsed time = 0.01 sec. (0.05 ticks, tree = 0.01 MB)
      1     2       41.0000     1                    41.2500        3
*     2     1      integral     0       39.0000       41.2500        4    5.77%
      3     1       40.5556     1       39.0000       41.2500        5    5.77%
*     4     0      integral     0       40.0000       41.2500        6    3.12%

Root node processing (before b&c):
  Real time             =    0.00 sec. (0.01 ticks)
Parallel b&c, 16 threads:
  Real time             =    0.02 sec. (0.05 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.02 sec. (0.05 ticks)

Times (seconds):
Input =  0.000825
Solve =  0.091536
Output = 0.000714
CPLEX 12.6.0.0: optimal integer solution; objective 40
6 MIP simplex iterations
5 branch-and-bound nodes
```

Node: current node
Nodes left: # of open nodes
Objective: value of current LPR
IInf: # integer infeasible variables
Best Integer: best solution found
Best Bound: current upper bound
ItCnt: total #pivots so far
Gap: relative gap between the best integer solution and best bound

45

Summary: Advanced IP solving

- MIPs are a lot harder to solve than LPs
- But IP technology is very sophisticated:
  - Dual simplex pivots
  - Tight formulations
  - Automated cut generation
  - Primal heuristics
  - Node and variable selection heuristics
- Problems with 100,000s variables and 10,000s constraints can be solved in commercially viable times

46