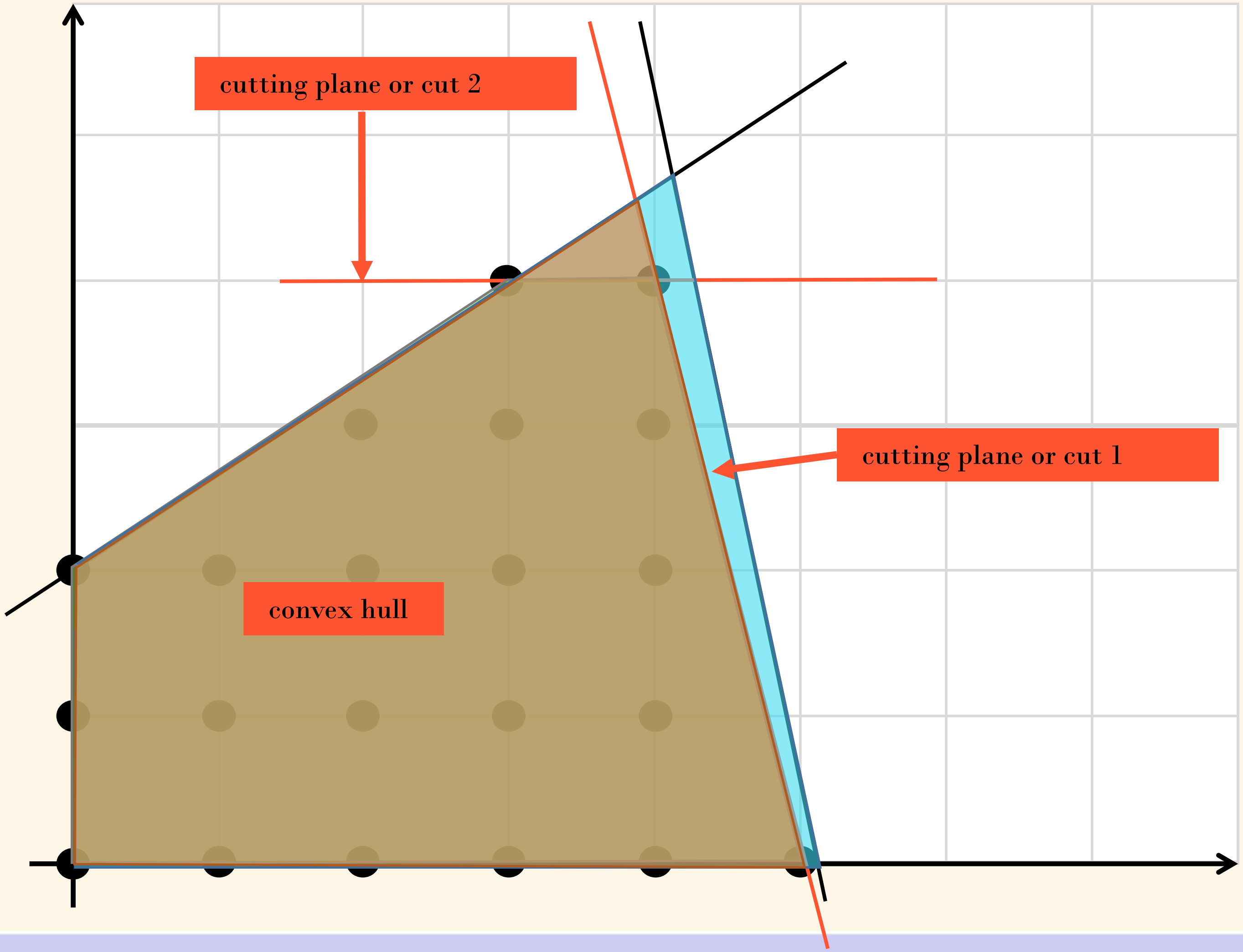
Solution Methods for Mixed-Integer Optimization Problems :

The Cutting Plane Algorithm

UNIVERSITY OF WATERLOO

The Concept of Cutting Planes

Integer program → ignore integer requirements → Linear Programming (LP) relaxation

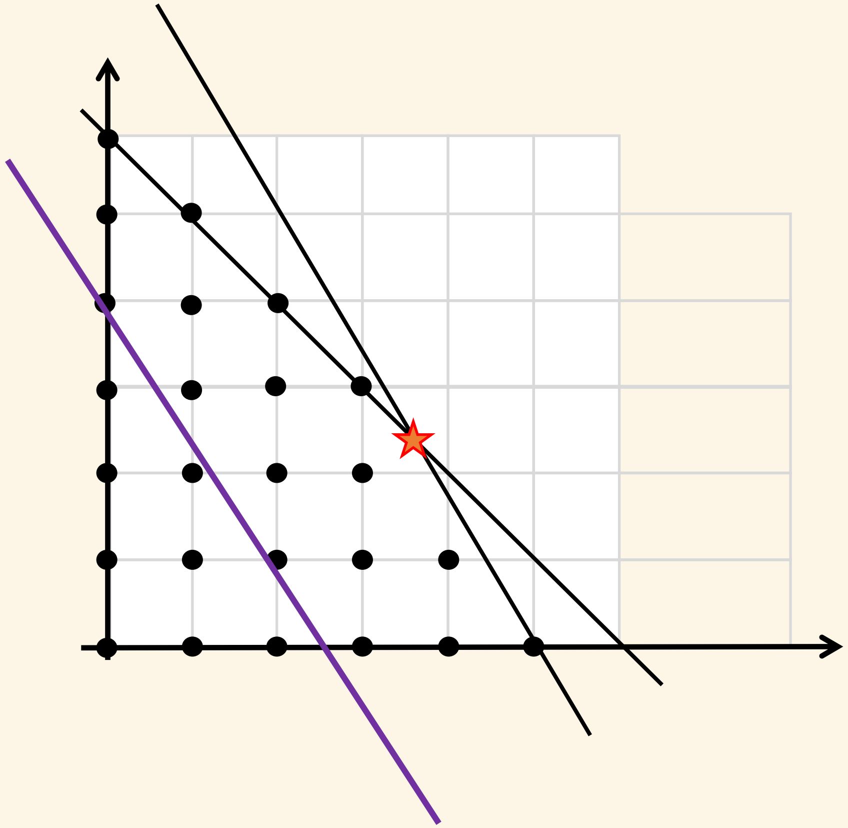


max $\quad 8x_1+5x_2$

s.t. $\quad x_1 + x_2 \leq 6$

$9x_1+5x_2 \leq 45$

$x_1, x_2 \geq 0$, ~~integer~~



$z = 41.25$; $x_1 = 2.25$; $x_2 = 3.75$

x_1 and x_2 are non-integer.

- Normally branch if branch-and-bound is used
- Cutting plane method- add cuts

z	x_1	x_2	s_1	s_2	rhs
1	0	0	1.25	0.75	41.25
0	0	1	2.25	- 0.25	2.25
0	1	0	-1.25	0.25	3.75

Gomory's Cutting Plane Algorithm

- Pick a row corresponding to a basic variable that has to be integer with a fractional value

  row 2: $x_1 + 0x_2 - 1.25s_1 + .25s_2 = 3.75$

z	x_1	x_2	s_1	s_2	rhs
1	0	0	1.25	0.75	41.25
0	0	1	2.25	- 0.25	2.25
0	1	0	-1.25	0.25	3.75

- Rewrite it as:

  $x_1 - 2s_1 + 0.75s_1 + 0s_2 + 0.25s_2 = 3 + .75$

- And *as*

  $x_1 - 2s_1 - 3 = -0.75s_1 - 0.25s_2 + 0.75$

- Gomory cut: $-0.75s_1 - 0.25s_2 + 0.75 \leq 0$

  $0.75s_1 + 0.25s_2 \geq 0.75$

  $3s_1 + s_2 \geq 3$

- As a function of x_1, x_2:

  Use: $s_1 = 6 - x_1 - x_2$ & $s_2 = 45 - 9x_1 - 5x_2$, so

  $3s_1 + s_2 \geq 3 \Rightarrow 3(6 - x_1 - x_2) + (45 - 9x_1 - 5x_2) \geq 3$

  $\Rightarrow 3x_1 + 2x_2 \leq 15$

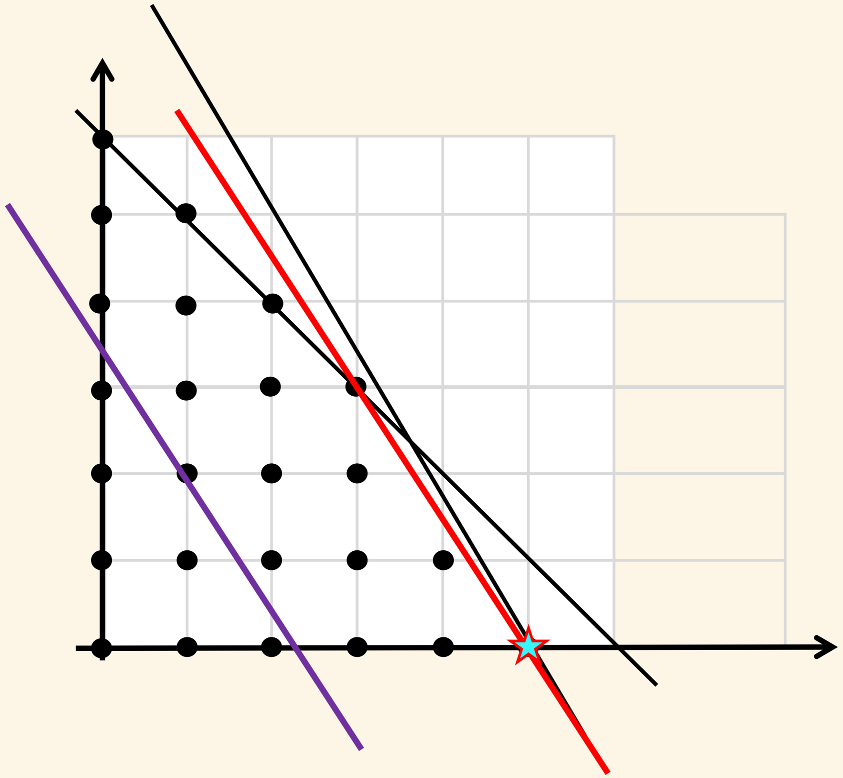
Note 1:(usually the one closest to .5 is picked).

Note 2: $-1.25 = 2 + .75$
$0.25 = 0 + .25$

UNIVERSITY OF WATERLOO

Gomory's Cutting Plane Algorithm

$$\begin{aligned} \max \quad & 8x_1 + 5x_2 \\ \text{s.t.} \quad & x_1 + x_2 \le 6 \\ & 9x_1 + 5x_2 \le 45 \\ & \color{red}{3x_1 + 2x_2 \le 15 \leftarrow cut} \\ & x_1, x_2 \ge 0 \end{aligned}$$



Gomory's Cutting Plane Algorithm

$$\begin{array}{ll} \max & 8x_1 + 5x_2 \\ \text{s.t.} & x_1 + x_2 + s_1 = 6 \\ & 9x_1 + 5x_2 + s_2 = 45 \\ & 0.75s_1 + 0.25s_2 \geq 0.75 \leftarrow \text{cut} \\ & x_1, x_2, s_1, s_2 \geq 0 \end{array}$$

z	x_1	x_2	s_1	s_2	s_3	rhs
1	0	0	1.25	0.75	0	41.25
0	0	1	2.25	-0.25	0	2.25
0	1	0	-1.25	0.25	0	3.75
0	0	0	-0.75	-0.25	1	-0.75

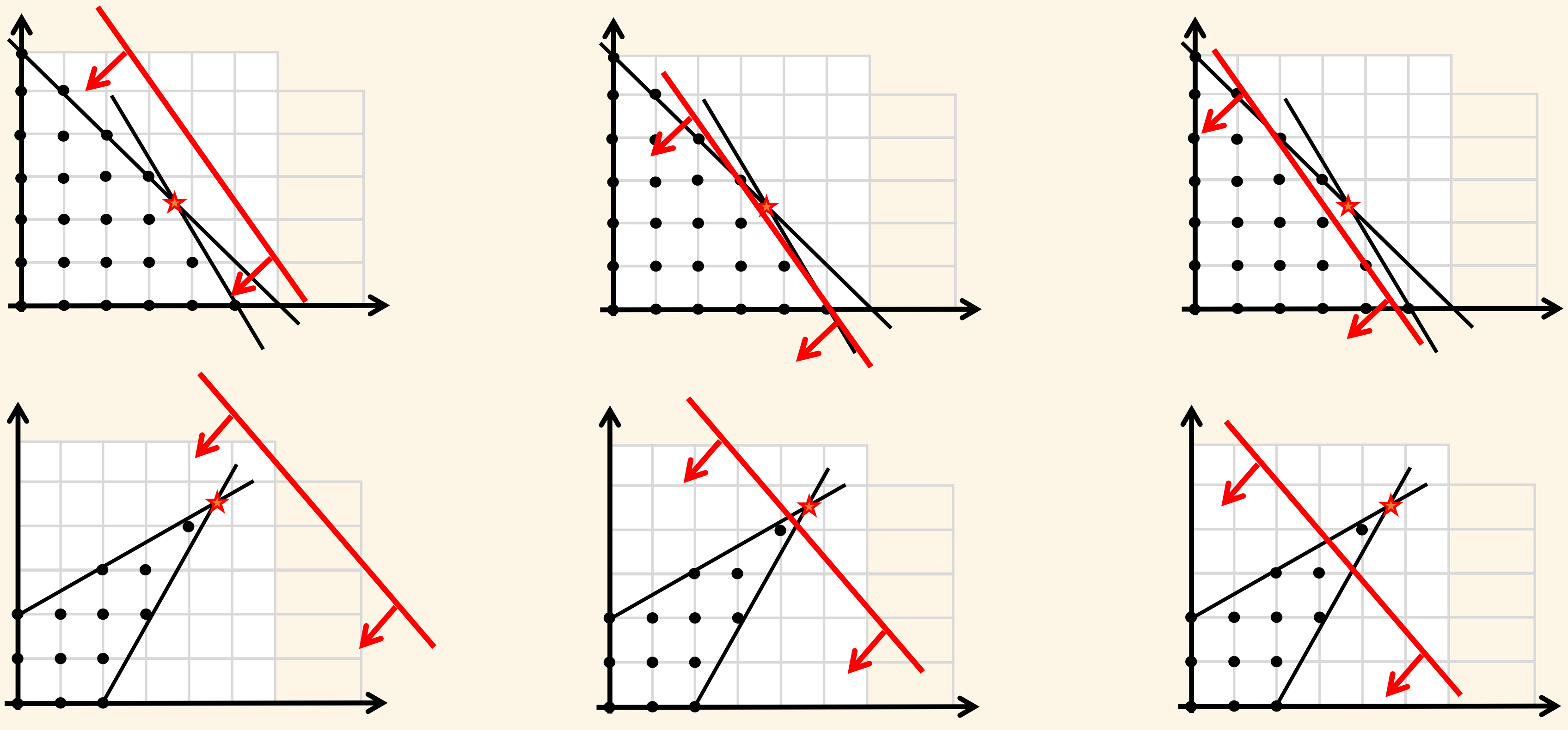
$$\begin{array}{ll} \max & 8x_1 + 5x_2 \\ \text{s.t.} & x_1 + x_2 + s_1 = 6 \\ & 9x_1 + 5x_2 + s_2 = 45 \\ & -0.75s_1 - 0.25s_2 + s_3 = -0.75 \\ & x_1, x_2, s_1, s_2, s_3 \geq 0 \end{array}$$

z	x_1	x_2	s_1	s_2	s_3	rhs
1	0	0	0	0.33	1.67	40
0	0	1	0	-1	3	0
0	1	0	0	0.67	-1.67	5
0	0	0	1	0.33	-1.33	1

UNIVERSITY OF WATERLOO

Cutting Plane Methods: Main Concepts:

- Valid Inequality: A constraint that is satisfied by all feasible integer points
- A cut or a cutting plane: A valid inequality that "cuts off" a non-integer point, usually the optimal LP solution.
- Cutting Plane Methods: solve a series of improved approximations of the convex hull till a feasible integer solution is reached. At each iteration, an improved linear program is solved. If the solution is feasible to the original problem, then the algorithm stops, otherwise, a cut is added. i.e. a valid inequality that cuts off the current solution.



Valid Inequality:

Valid cut:

Neither

Solution Methods for Mixed-Integer Optimization Problems :

The Branch-and-Cut Algorithm

Branch-and-cut

$$\begin{array}{ll} \min & -5x_1 - 6x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 7 \\ & 2x_1 - x_2 \leq 3 \\ & x_1, x_2 \geq 0, \text{integer}. \end{array}$$

$$\begin{array}{llll} \min & -5x_1 - 6x_2 & & \\ \text{s.t.} & x_1 + 2x_2 + s_1 & & = 7 \\ & 2x_1 - x_2 & +s_2 & = 3 \\ & x_1, x_2 \geq 0, & \text{integer}. & \end{array}$$

z	x_1	x_2	s_1	s_2	rhs
1	0	0	3.4	0.8	26.2
0	0	1	.4	-.2	2.2
0	1	0	.2	.4	2.6

$$x_1^* = 2.2; x_2^* = 2.6; z^* = 26.2$$

UNIVERSITY OF WATERLOO

Branch-and-cut

As the solution is not all integer, we should branch if we want to proceed with branch and bound. We could also add Gomory cuts if we want to proceed with a cutting plane algorithm. Let us add a Gomory cut based on row 1:

z	x_1	x_2	s_1	s_2	rhs
1	0	0	3.4	0.8	26.2
0	0	1	.4	-.2	2.2
0	1	0	.2	.4	2.6

$$x_2 + \frac{2}{5}s_1 - \frac{1}{5}s_2 = 2.2 \Rightarrow x_2 - s_2 - 2 = .2 - \frac{2}{5}s_1 - \frac{4}{5}s_2$$

$2s_1 + 4s_2 \geq 1.$

$x_1 \leq 2.5$

$\boxed{x_1 \leq 2}$

Branch-and-cut

Add $x_1 + s_3 = 2$ and reoptimize:

z	x_1	x_2	s_1	s_2	s_3	rhs
1	0	0	3	0	2	25
0	1	0	0	0	1	2
0	0	1	0.5	0	-0.5	2.5
0	0	0	0.5	1	-2.5	1.5

- $x_1 = 2$, $x_2 = 2.5$ $z = 25$. As x_2 is non-integer, we should proceed.
- add an other cut or branch?
- Branch
- Two nodes are created: $x_2 \leq 2$ versus $x_2 \geq 3$

Branch-and-cut

Node 0

$$\begin{array}{ll} \min & -5x_1 - 6x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 7 \\ & 2x_1 - x_2 \leq 3 \\ & x_1, x_2 \geq 0, \\ & \mathbf{x_1} \leq \mathbf{2} \leftarrow \text{Valid cut} \end{array}$$

$x_1 = 2$, $x_2 = 2.5$ $z = -25$.

Incumbent = ?

$x_2 \leq 2$

$x_2 \geq 3$

Node 1

$$\left\{ \begin{array}{ll} \min & -5x_1 - 6x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 7 \\ & 2x_1 - x_2 \leq 3 \\ & x_1, x_2 \geq 0, \\ & \mathbf{x_1} \leq \mathbf{2} \leftarrow \text{Valid cut} \\ & \mathbf{x_2} \leq \mathbf{2} \leftarrow \text{Branching constraint} \end{array} \right\}$$

$x_1 = 2$, $x_2 = 2$; $z=-22$.

Incumbent = -22

Node 2

$$\left\{ \begin{array}{ll} \min & -5x_1 - 6x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 7 \\ & 2x_1 - x_2 \leq 3 \\ & x_1, x_2 \geq 0, \\ & \mathbf{x_1} \leq \mathbf{2} \leftarrow \text{Valid cut} \\ & \mathbf{x_2} \geq \mathbf{3} \leftarrow \text{Branching constraint} \end{array} \right\}$$

$x_1 = 1$, $x_2 = 3$ $z=-23$.

Incumbent = ~~-22~~, -23

The Branch-and-Cut Algorithm

1. In branch-and-cut, at every node, we either add cuts or decide to branch.
2. If the addition of cuts is only done at the root node, then it is branch-and-bound with a preprocessing step where cuts are added to enhance the formulation.
3. At child nodes, there are usually two types of cuts, local and global. Local cuts are only valid for that node and its children. Global cuts are valid for all nodes in the tree, including the root node.
4. The dual Simplex method is very efficient is finding the optimal solution after a cut or a branching constraint is added.

Branch-and-Cut for Mixed Integer Programs

Summary:

- Branch-and-cut method, combines
  - The cutting plane method
  - The branch-and-bound method
  - is the leading algorithm for MIPs