Review session: Exam question 2
LP Duality, Matrix Games, and Integer linear programming.

- Lecture 3 (second half)
  - Duality
  - Vanderbei, Chapter 5 (sections 1-4)
- Lecture 4
  - Duality and complementary slackness. Farkas Lemma and Strict complementarity.
  - Vanderbei, Chapter 5 (sections 5-8) and Chapter 10 (sections 4 and 5).
- Lecture 6
  - Game Theory.
  - Vanderbei, Chapter 11.
- Lecture 7
  - Integer linear programming and Branch-and-bound.
  - Vanderbei, Chapter 23
- Lecture 8 (second half)
  - Branch and bound and branch and cut.
  - Vanderbei, Chapter 23

LP Dual

Primal P: Maximize $c^\top x$ subject to $Ax \leq b, x \geq 0.$
Dual D: Minimize $b^\top y$ subject to $A^\top y \geq c, y \geq 0.$

Weak duality theorem:
If x is primal feasible and y is dual feasible, then $c^\top x \leq b^\top y$.
Proof: $c^\top x \leq (y^\top A)x \leq y^\top (Ax) \leq y^\top b$.

Corollary:
If primal is unbounded, then dual is infeasible.
If dual is unbounded, then primal is infeasible.

Strong duality theorem:
If x is primal optimal and y is dual optimal then $c^\top x = b^\top y$.

The Complementary Slackness Property

Primal P: Maximize $c^\top x$ subject to $Ax \leq b, x \geq 0$.

Dual D: Minimize $b^\top y$ subject to $A^\top y \geq c, y \geq 0$.

Solutions x and y are said to satisfy *Complementary Slackness* if and only if:

- for all $i = 1,2,\dots,m$ (at least) one of the following holds:
  a) $\sum_{j=1}^{n} a_{ij}x_j = b_i$ (i-th primal constraint has slack 0)
  b) $y_i = 0$ (i-th dual variable is 0)

- for all $j = 1,2,\dots,n$ (at least) one of the following holds:
  a) $\sum_{i=1}^{m} a_{ij}y_i = c_j$ (j-th dual constraint has slack 0)
  b) $x_j = 0$ (j-th primal variable is 0)

Complementary Slackness Theorem

Primal P: Maximize $c^\top x$ subject to $Ax \leq b, x \geq 0.$

Dual D: Minimize $b^\top y$ subject to $A^\top y \geq c, y \geq 0.$

Theorem

Let x and y be feasible solutions to P and D. Then:

x and y are both optimal

if and only if

x and y satisfy complementary slackness.

Proof of weak duality: $c^\top x \leq (y^\top A)x \leq y^\top(Ax) \leq y^\top b.$

Complementary slackness implies $c^\top x = y^\top b$

Proof of Strong duality theorem

- A dictionary defines a primal solution *and* a dual solution with the same objective value and that satisfies complementary slackness.
- The dictionary of the dual linear program is the negative transpose of the primal dictionary.
- The termination criteria of the simplex algorithm is that the dual solution is feasible.
- Hence if the primal has an optimal solution, the dual has an optimal solution of the same value.

Initial primal and dual dictionary

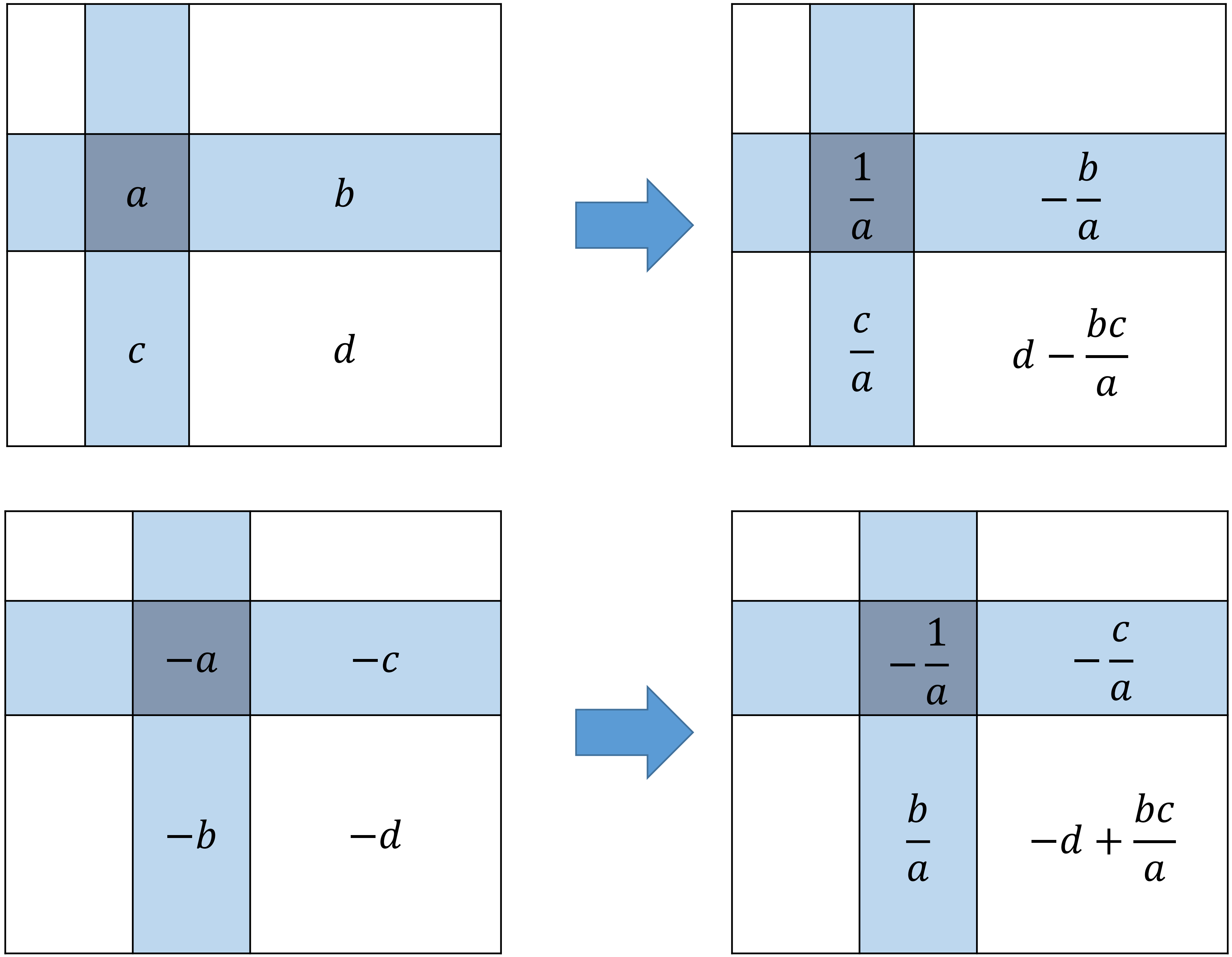
Primal:

$$
\begin{array}{rcl}
z & = & c_1 x_1 + \cdots + c_n x_n \\
\hline
x_{n+1} & = & b_1 - a_{11} x_1 - \cdots - a_{1n} x_n \\
\vdots & = & \vdots \qquad \vdots \qquad\qquad \vdots \\
x_{n+m} & = & b_m - a_{m1} x_1 - \cdots - a_{mn} x_n
\end{array}
$$

Dual:

$$
\begin{array}{rcl}
-w & = & - b_1 y_1 - \cdots - b_m y_m \\
\hline
y_{m+1} & = & -c_1 + a_{11} y_1 + \cdots + a_{m1} y_m \\
\vdots & = & \vdots \qquad \vdots \qquad\qquad \vdots \\
y_{m+n} & = & -c_n + a_{1n} y_1 + \cdots + a_{mn} y_m
\end{array}
$$

Abstract view of pivoting



Dictionaries in Matrix form

$$Ax = [B \quad N] \begin{bmatrix} x_{\mathcal{B}} \\ x_{\mathcal{N}} \end{bmatrix} = Bx_{\mathcal{B}} + Nx_{\mathcal{N}} = b$$
$$\zeta = c^\top x = [c_{\mathcal{B}}^\top \quad c_{\mathcal{N}}^\top] \begin{bmatrix} x_{\mathcal{B}} \\ x_{\mathcal{N}} \end{bmatrix} = c_{\mathcal{B}}^\top x_{\mathcal{B}} + c_{\mathcal{N}}^\top x_{\mathcal{N}}$$

Primal dictionary:

$$\zeta = \zeta^* - {z_{\mathcal{N}}^*}^\top x_{\mathcal{N}}$$
$$x_{\mathcal{B}} = x_{\mathcal{B}}^* - B^{-1} N x_{\mathcal{N}}$$

Dual dictionary:

$$-\xi = -\zeta^* + {x_{\mathcal{B}}^*}^\top z_{\mathcal{B}}$$
$$z_{\mathcal{N}} = z_{\mathcal{N}}^* + (B^{-1}N)^\top z_{\mathcal{B}}$$

Consequence of Strong duality

- Software solving LP programs to optimality can easily be **checked**.
- Give the software the primal program as well as the dual program. The solution to the dual problem is a **certificate** that the solution to the primal program is optimal.
- Solving linear programs to optimality is as easy as solving a system of linear inequalities:

1. Given LP P, check if P is feasible using LI algorithm. Otherwise report **Infeasible**.
2. Construct dual D of P. Use LI algorithm to find (x, y) so that $Ax \leq b, x \geq 0, A^\intercal y \geq c, y \geq 0$, and $c^\intercal x = b^\intercal y$. If no such (x, y) exist, report **Unbounded**. Otherwise return x.

The Dual Simplex Method

- Simple idea: Instead of running the Simplex Algorithm on the primal problem, run it on the dual problem.
- We can even run the Dual Simplex algorithm on the primal dictionary.
- Dual-Based Phase I Algorithm:
  - If we change the objective function to
    $$\max 0$$
    the dual program is origo feasible!
  - The optimal dual solution is feasible in the primal, if a feasible solution exists – this may serve as a replacement for Phase I.

Generalized Duality

Primal:

Maximize	$c_1x_1 + c_2x_2 + c_3x_3$
Subject to	$y_1: a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \le b_1$
	$y_2: a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \ge b_2$
	$y_3: a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$
	$x_1 \ge 0, x_2 \le 0, x_3$ free

Dual:

Minimize	$b_1y_1 + b_2y_2 + b_3y_3$
Subject to	$x_1: a_{11}y_1 + a_{21}y_2 + a_{31}y_3 \ge c_1$
	$x_2: a_{12}y_1 + a_{22}y_2 + a_{32}y_3 \le c_2$
	$x_3: a_{13}y_1 + a_{23}y_2 + a_{33}y_3 = c_3$
	$y_1 \ge 0, y_2 \le 0, y_3$ free

Rules for Taking the Dual

- Constraints in **primal** corresponds to variables in **dual** (and vice versa).
- Coefficients of objective function in **primal** corresponds to constants in **dual** (and vice versa).

Primal (Maximization)	**Dual (Minimization)**
$\leq$ for constraint	≥ 0 for variable
$\geq$ for constraint	≤ 0 for variable
$=$ for constraint	free for variable
≥ 0 for variable	$\geq$ for constraint
≤ 0 for variable	$\leq$ for constraint
free for variable	$=$ for constraint

Certifying infeasibility

Farkas Lemma: Exactly one of the following is true:

1. There exist x s.t. $Ax \leq b$.
2. There exist $y \geq 0$ s.t. $A^\top y = 0$ and $b^\top y < 0$.

Proof: Consider the following program and its dual

P: $\max 0$ s.t. $Ax \leq b$

D: $\min b^\top y$ s.t. $A^\top y = 0, y \geq 0$

Strict Complementary Slackness Property

Primal P: Maximize $c^\top x$ subject to $Ax \leq b, x \geq 0.$
Dual D: Minimize $b^\top y$ subject to $A^\top y \geq c, y \geq 0.$

Solutions x and y are said to satisfy **Strict** *Complementary Slackness* if and only if:

- for all $i = 1,2, \ldots, m$ **exactly** one of the following holds:
  a) $\sum_{j=1}^{n} a_{ij} x_j = b_i$ (i-th primal constraint has slack 0)
  b) $y_i = 0$ (i-th dual variable is 0)

- for all $j = 1,2, \ldots, n$ **exactly** one of the following holds:
  a) $\sum_{i=1}^{m} a_{ij} y_i = c_j$ (j-th dual constraint has slack 0)
  b) $x_j = 0$ (j-th primal variable is 0)

Strict Complementary Slackness Theorem

If P has an optimal solution then there exist optimal solutions x to P and y to D that satisfy strict complementary slackness.

Linear programs used in proof:

P': max x_j s.t. $Ax \leq b, -c^\top x \leq -z^*, x \geq 0$

D': min $b^\top y - z^* t$ s.t. $A^\top y - ct \geq e_j, y \geq 0, t \geq 0$

where z^* is value of optimal solution.

Matrix Games

- A Matrix Game is given by a *payoff matrix* $A = (a_{ij}) \in R^{m\times n}$.
- Row player (Minnie) chooses row $i \in \{1, \dots, m\}$ (without seeing Max's move).
- Column player (Max) chooses column $j \in \{1, \dots n\}$ (without seeing Minnie's move).
- Max *gains* a_{ij} "dollars" and Minnie *loses* a_{ij} "dollars".
- Matrix games are zero-sum games as Max gains exactly what Minnie loses.
- Negative numbers can be used to model money transferred in the other direction.
- **Warning**: In most other texts, the Row player is the maximizer and the Column player is the minimizer.

Column Players Optimal randomized Strategy

Optimal randomized strategy and guaranteed lower bound on exp. gain for Max is $(p_1, p_2, \dots, p_n; g)$ which is a solution to the LP:

$$
\begin{array}{lll}
\max & g & \\
\text{s.t.} & \sum_{j=1}^{n} a_{ij} p_j \geq g & i = 1, \dots, m \\
& \sum_{j=1}^{n} p_j = 1 & \\
& p_j \geq 0 & j = 1, \dots, n
\end{array}
$$

Row Players Optimal randomized Strategy

Optimal randomized strategy and guaranteed upper bound on exp. loss for Minnie is $(q_1, q_2, \ldots, q_m; l)$ which is a solution to the LP:

$$
\begin{array}{lll}
\text{min} & l & \\
\text{s.t.} & \sum_{i=1}^{m} a_{ij} q_i \leq l & \mathrm{j} = 1, \ldots, n \\
& \sum_{i=1}^{m} q_i = 1 & \\
& q_i \geq 0 & \mathrm{i} = 1, \ldots, m
\end{array}
$$

Crucial observation: Max's program and Minnie's program are each others duals!

von Neuman Min-Max Theorem (Theorem 11.1)

For any real matrix A,

$$\max_{p} \min_{q} q^{\top} A p = \min_{q} \max_{p} q^{\top} A p$$

where p (resp. q) are arbitrary probability distributions on columns (resp. rows) of A.

Principle of indifference

Suppose p^* is optimal for Max and q^* is optimal for Minnie.

Let $S \subseteq \{1,2,\ldots,m\}$ be the set of rows to which q^* assigns non-zero probability.

Suppose Max plays p^*. Then his expected payoff is the same, namely the value of the game, *no matter which actual row in* S Minnie chooses.

Proof: Complementary Slackness Theorem!

If $q_i^* > 0$ then i-th constraint is satisfied with equality by p^*.

Exploiting weak opponents

Let $S \subseteq \{1,2,\ldots,m\}$ be the set of rows to which *some* optimal strategy q^* for Minnie assigns non-zero probability.

There exists an optimal strategy p^* for Max so that his expected payoff is *strictly bigger* than the value of the game if Minnie chooses a row that is *not* in S^*.

Proof: Strict Complementary Slackness Theorem!

Best replies and Nash equilibrium

Let y be a randomized strategy for Minnie. A best reply to y is a randomized strategy for Max that maximizes his expected winnings assuming that Minnie plays y.

A pair of randomized strategies x^* and y^* is called a Nash equilibrium if they are best responses to each other.

Theorem: A pair of randomized strategies (x^*, y^*) for a matrix game is a Nash equilibrium if and only if they are both optimal.

Analyzing games with rounds

The theory of matrix games can be used to analyze games where several rounds of actions are made, information is leaked “bit by bit" and external randomness enters the game.

... but the game matrix may become very large : exponential size in the size of the game tree.

An alternative LP formulation gives a linear program of size linear in the game tree.

Yao's Principle

- **Yao's Principle:** The expected complexity of the best randomized algorithm on a worst-case input *equals* the *weighted average* complexity for a worst-case distribution of inputs using the best deterministic algorithm.

Integer Programming

An integer linear program (ILP) is defined exactly as a linear program except that values of variables in a feasible solution have to be integral:

$$\begin{aligned} \max\ & c^\top x \\ \text{s.t. } & Ax \le b \\ & x \ge 0 \\ & x \in \mathbb{Z}^n \end{aligned}$$

- Very useful new possibility: We can now model **Boolean variables**.
- This makes ILPs *much more* powerful than LPs for modelling purposes!
- Unfortunately it also makes them a lot harder to solve computationally!

Traveling Salesman Problem - ILP formulation

$$\min \sum_{i,j=1}^{n} c_{ij} x_{ij}$$

s.t.

$$\sum_{j=0}^{n-1} x_{ij} = 1, \qquad \text{for } i = 0, \dots, n-1$$

$$\sum_{i=0}^{n-1} x_{ij} = 1, \qquad \text{for } j = 0, \dots, n-1$$

$$t_j \geq t_i + 1 - n\left(1 - x_{ij}\right), \text{for } i \geq 0, j \geq 1, i \neq j$$

Bounded Integer Linear Program

P:

$$
\begin{aligned}
\max \ & c^\top x \\
\text{s.t. } & Ax \le b \\
& l_i \le x_i \le u_i \\
& x \in \mathbb{Z}^n
\end{aligned}
$$

Where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$,c $\in \mathbb{R}^n$ and

$l_i < u_i, l_i, u_i \in \mathbb{Z}$.

Branch and Bound

1. Solve P* (using e.g. Simplex algorithm)
2. If P* is infeasible, then P is infeasible (return $(\mathbf{null}, -\infty)$).
3. Otherwise let (x^*, λ^*) be optimal solution to P*.
4. If x^* is integer, return (x^*, λ^*) .
5. If $\lambda^* \leq \mathrm{currentbest}$ **(bound)**, then return from branch with no solution (return $(\mathbf{null}, -\infty)$).
6. Otherwise, pick i such that $x_i^* \notin \mathbb{Z}$.
7. Recursively solve **(branch)** (P*$[u_i \coloneqq \lfloor x_i^* \rfloor]$, currentbest) to get (x^l, λ^l).
8. If $\lambda^l > \mathrm{currentbest}$, update $\mathrm{currentbest} \coloneqq \lambda^r$
9. Recursively solve **(branch)** (P*$[l_i \coloneqq \lceil x_i^* \rceil]$, currentbest) to get (x^r, λ^r).
10. If $\lambda^l > \lambda^r$, return (x^l, λ^l), otherwise return (x^r, λ^r).

Upper and lower bounds can be added to a LP dictionary as an extra constraint. This maintains dual feasibility, so it is convenient to use the dual Simplex algorithm.

Valid inequality for an ILP

- Given integer linear program, a new inequality is called ***valid*** if it does not change the set of ***integer*** solutions to the program.

- It is a ***cutting plane*** if it removes the (non-integer) optimum solution to the relaxed linear program.

Gomory Cutting Plane Algorithm

- Assume that all initial coefficients in (standard form) ILP instance are integer.

- Solve LP-instance using simplex method.

- Suppose the simplex method terminates with a non-integer optimal solution.

Suppose some line in final dictionary reads

$$x_i = b_i + \sum_j a_j x_j$$

where b_i is not integer.

Let $b_i' = b_i - \lfloor b_i \rfloor$ and $a_j' = a_j - \lfloor a_j \rfloor$. Then

$$x_i - \lfloor b_i \rfloor - \sum_j \lfloor a_j \rfloor x_j = b_i' + \sum_j a_j' x_j$$

For every feasible integer solution:

- Left hand side is integer.
- Right hand side is strictly bigger than zero.

Thus

$$x_i - \lfloor b_i \rfloor - \sum_j \lfloor a_j \rfloor x_j \geq 1$$

is a valid cutting plane.

Cutting plane algorithm for TSP

- Find optimal solution x^* of relaxation P^*.
- Let $y_{ij} = x_{ij} + x_{ji}$
- Check if some S, T has $\sum_{i \in S, j \in T} y_{ij} < 2$. (this can be efficiently checked by Max Flow algorithm using the max-flow min-cut theorem).
- Add inequality $\sum_{i \in S, j \in T} y_{ij} \geq 2$.

Branch-and-cut for TSP

- Branch-and-bound with relaxation being LP-relaxation + some set of valid inequalities.
- When to stop adding inequalities and start branching is a matter of heuristics and experiments.
- Yields state-of-the art solver. Many non-trivial implementation issues.