CMSC5706 Topics in Theoretical Computer Science

Week 2: Linear Programming

Instructor: Shengyu Zhang
Motivating examples

Introduction to algorithms

Simplex algorithm
  - On a particular example
  - General algorithm

Duality

An application to game theory
Example 1: profit maximization

- A company has two types of products: P, Q.
- Profit:  P --- $1 each;  Q --- $6 each.
- Constraints:
  - Daily productivity (including both P and Q) is 400
  - Daily demand for P is 200
  - Daily demand for Q is 300
- **Question:** How many P and Q should we produce to maximize the profit?
  - $x_1$ units of P, $x_2$ units of Q
How to solve?

- $x_1$ units of P
- $x_2$ units of Q

Constraints:
- Daily productivity (including both P and Q) is 400
- Daily demand for P is 200
- Daily demand for Q is 300

Question: how much P and Q to produce to maximize the profit?

Variables:
- $x_1$ and $x_2$.

Constraints:
- $x_1 + x_2 \leq 400$
- $x_1 \leq 200$
- $x_2 \leq 300$
- $x_1, x_2 \geq 0$

Objective:
\[
\max x_1 + 6x_2
\]
Illustrative figures

(a) 

(b) Optimum point
Profit = $1900

\[ \begin{align*}
  x_1 & \quad x_2 \\
  0 & \quad 100 & \quad 200 & \quad 300 & \quad 400 \\
  100 & \quad 200 & \quad 300 & \quad 400 \\
  \end{align*} \]
Example 2

- We are managing a network with **bandwidth** as shown by numbers on edges.
  - Bandwidth: max units of flows
- 3 **connections**: AB, BC, CA
  - We get $3, $2, $4 for providing them respectively.
  - Two routes for each connection: short and long.
- **Question**: How to route the connections to maximize our revenue?
Example 2

- Variables:
  - $x_{AB}, x'_{AB}, x_{BC}, x'_{BC}, x_{AC}, x'_{AC}$.

- Constraints:
  - $x_{AB} + x'_{AB} + x_{AC} + x'_{AC} \leq 12$ (edge $(A, a)$)
  - $x_{AB} + x'_{AB} + x_{BC} + x'_{BC} \leq 10$ (edge $(B, b)$)
  - $x_{BC} + x'_{BC} + x_{AC} + x'_{AC} \leq 8$ (edge $(C, c)$)
  - $x_{AB} + x'_{BC} + x'_{AC} \leq 6$ (edge $(a, b)$)
  - $x'_{AC} + x_{AB} + x_{BC} \leq 13$ (edge $(b, c)$)
  - $x_{AB} + x'_{BC} + x'_{AC} \leq 11$ (edge $(a, c)$)
  - $x_{AB}, x'_{AB}, x_{BC}, x'_{BC}, x_{AC}, x'_{AC} \geq 0$

- Objective:
  \[
  \max 3(x_{AB} + x'_{AB}) + 2(x_{BC} + x'_{BC}) + 4(x_{AC} + x'_{AC})
  \]
LP in general

- Max/min a **linear** function of variables
  - Called the *objective function*
- All constraints are **linear** (in)equalities
- Equational form:

  \[
  \begin{align*}
  & \text{max } c^T x \\
  \text{s.t. } & Ax = b \\
  \end{align*}
  \]

- **Equational form:**
  \[
  \begin{align*}
  & \text{max } c_1 x_1 + \cdots + c_n x_n \\
  \text{s.t. } & a_{i1} x_1 + \cdots + a_{in} x_n = b_i, \\
  & \quad \forall i = 1, \ldots, m \\
  & x \geq 0 \\
  & x_i \geq 0, \forall i = 1, \ldots, n \\
  \end{align*}
  \]

- **Variables:** \(x\)
  - Coefficients in constraints: \((A, b)\)
Transformations between forms

- **Min vs. max:**
  - $\min c^T x \Leftrightarrow \max -c^T x$

- **Inequality directions:**
  - $a_i^T x \geq b_i \Leftrightarrow -a_i^T x \leq -b_i$

- **Equalities to inequalities:** ($a_i$: row $i$ in matrix $A$)
  - $a_i^T x = b_i \Leftrightarrow a_i^T x \geq b_i$, and $a_i^T x \leq b_i$. 
Transformations between forms

- **Inequalities to equalities:**
  - \( \mathbf{a}_i^T \mathbf{x} \geq b_i \iff \mathbf{a}_i^T \mathbf{x} = b_i + s_i, s_i \geq 0 \)
  - The newly introduced variable \( s_i \) is called *slack variable*

- **“Unrestricted” to “nonnegative constraint”:**
  - \( x_i \) unrestricted \( \iff x_i = s - t, s \geq 0, t \geq 0 \)
feasibility

- The constraints of the form $ax_1 + bx_2 = c$ is a line on the plane of $(x_1, x_2)$.
- $ax_1 + bx_2 \leq c$? half space.
  - $x_1 \leq 200$
  - $x_2 \leq 300$
  - $x_1 + x_2 \leq 400$
  - $x_1, x_2 \geq 0$
- All constraints are satisfied: the intersection of these half spaces. --- feasible region.
  - Feasible region nonempty: LP is feasible
  - Feasible region empty: LP is infeasible
Adding the objective function into the picture

- The objective function is also linear
  - also a line for a fixed value.
- Thus the optimization is:
  try to move the line towards the desirable direction s.t. the line still intersects with the feasible region.

![Graph showing the objective function and feasible region with an optimum point at (200, 300).](image)
Possibilities of solution

- **Infeasible**: no solution satisfying $Ax = b$ and $x \geq 0$.
  - Example? Picture?

- **Feasible but unbounded**: $c^T x$ can be arbitrarily large.
  - Example? Picture?

- **Feasible and bounded**: there is an optimal solution.
  - Example? Picture?
Three Algorithms for LP

■ **Simplex** algorithm (Dantzig, 1947)
  - Exponential in worst case
  - Widely used due to the practical efficiency

■ **Ellipsoid** algorithm (Khachiyan, 1979)
  - First polynomial-time algorithm: $O(n^4L)$
    - $L$: number of input bits
  - Little practical impact.

■ **Interior point** algorithm (Karmarkar, 1984)
  - More efficient in theory: $O(n^{3.5}L)$
  - More efficient in practice (compared to Ellipsoid).
Simplex method: geometric view

- Start from any vertex of the feasible region.
- Repeatedly look for a better neighbor and move to it.
  - Better: for the objective function
- Finally we reach a point with no better neighbor
  - In other words, it’s locally optimal.
- For LP: locally optimal $\Leftrightarrow$ globally optimal.
  - Reason: the feasible region is a convex set.
Simplex algorithm: Framework

- A sequence of (simplex) tableaus
  1. Pick an initial tableau
  2. Update the tableau
  3. Terminate

- What’s a tableau?
  1. How?
  2. What’s the rule?
  3. When to terminate? Why optimal?

Complexity?
An introductory example

Consider the following LP

\[
\begin{align*}
\text{max} & \quad x_1 + x_2 \\
\text{s.t.} & \quad -x_1 + x_2 + x_3 = 1 \\
& \quad x_1 + x_4 = 3 \\
& \quad x_2 + x_5 = 2 \\
& \quad x_1, \ldots, x_5 \geq 0
\end{align*}
\]

The equalities are \( Ax = b \),

\[
A = \begin{pmatrix}
-1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1
\end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}
\]

Let \( z = \text{obj} = x_1 + x_2 \).

Rewrite equalities as follows. (A tableau.)

\[
\begin{align*}
x_3 &= 1 + x_1 - x_2 \\
x_4 &= 3 - x_1 \\
x_5 &= 2 - x_2 \\
z &= x_1 + x_2
\end{align*}
\]
An introductory example

- The equalities are \( Ax = b \),
  \[
  A = \begin{pmatrix}
  -1 & 1 & 1 & 0 & 0 \\
  1 & 0 & 0 & 1 & 0 \\
  0 & 1 & 0 & 0 & 1
  \end{pmatrix},
  b = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}
  \]
- Let \( z = \text{obj} = x_1 + x_2 \).
- \( B = \{3, 4, 5\} \) is a basis: \( A_B = I_3 \) is non-singular.
  - \( A_B \): columns \( \{j: j \in B\} \) of \( A \).
- The basis is feasible:
  \[
  A_B^{-1} b = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.
  \]
  - Rewrite equalities as follows.
    \[
    x_3 = 1 + x_1 - x_2 \\
    x_4 = 3 - x_1 \\
    x_5 = 2 - x_2 \\
    z = x_1 + x_2
    \]
  - Set \( x_1 = x_2 = 0 \), and get \( x_3 = 1, x_4 = 3, x_5 = 2 \).
  - And \( z = 0 \).
  - \[
    \begin{pmatrix}
    x_1 & x_2 & x_3 & x_4 & x_5 & z
    \end{pmatrix}
  \]

An introductory example

- Now we want to improve $z = \text{obj} = x_1 + x_2$.
- Clearly one needs to increase $x_1$ or $x_2$.
- Let's say $x_2$.
  - we keep $x_1 = 0$.
- How much can we increase $x_2$?
  - We need to maintain the first three equalities.
- Rewrite equalities as follows.
  
  \[
  \begin{align*}
  x_3 &= 1 + x_1 - x_2 \\
  x_4 &= 3 - x_1 \\
  x_5 &= 2 - x_2 \\
  z &= x_1 + x_2
  \end{align*}
  \]
- Set $x_1 = x_2 = 0$, and get $x_3 = 1, x_4 = 3, x_5 = 2$.
- And $z = 0$.
- \[
  \begin{pmatrix}
  x_1 & x_2 & x_3 & x_4 & x_5 & z \\
  0 & 0 & 1 & 3 & 2 & 0
  \end{pmatrix}
  \]
- Setting $x_1 = 0$, the first three equalities become:
  \[ x_3 = 1 - x_2 \]
  \[ x_4 = 3 \]
  \[ x_5 = 2 - x_2 \]

- To maintain all $x_i \geq 0$, we need $x_2 \leq 1$ and $x_2 \leq 2$.
  - Obtained from the first and third equalities above.

- So $x_2$ can increase to 1.

- And $x_3$ becomes 0.

- Rewrite equalities as follows:
  \[ x_3 = 1 + x_1 - x_2 \]
  \[ x_4 = 3 - x_1 \]
  \[ x_5 = 2 - x_2 \]
  \[ z = x_1 + x_2 \]

- Set $x_1 = 0$, $x_2 = 1$, and update other variables:
  \[ x_3 = 0, x_4 = 3, x_5 = 1. \]

- And $z = 1$.

\[
\begin{pmatrix}
  x_1 & x_2 & x_3 & x_4 & x_5 & z \\
  0 & 1 & 0 & 3 & 1 & 1
\end{pmatrix}
\]
An introductory example

- Now basis becomes \( \{2, 4, 5\} \)
  - the basis is feasible.
- Compare to previous basis \( \{3, 4, 5\} \), one index (3) leaves and another (2) enters.
- This process is called a pivot step.
- Rewrite the tableau by putting variables in basis to the left hand side.

- Rewrite equalities as follows.
  \[
  x_3 = 1 + x_1 - x_2 \\
  x_4 = 3 - x_1 \\
  x_5 = 2 - x_2 \\
  z = x_1 + x_2
  \]
An introductory example

- Now basis becomes \{2,4,5\}
  - the basis is feasible.

- Compare to previous basis \{3,4,5\}, one index (3) leaves and another (2) enters.

- This process is called a pivot step.

- Rewrite the tableau by putting variables in basis to the left hand side.

- Rewrite equalities as follows.
  - \( x_2 = 1 + x_1 - x_3 \)
  - \( x_4 = 3 - x_1 \)
  - \( x_5 = 1 - x_1 + x_3 \)
  - \( z = 1 + 2x_1 - x_3 \)
An introductory example

- Repeat the process.
- To increase $z$, we can increase $x_1$.
  - Increasing $x_3$ decreases $z$ since the coefficient is negative.
- We keep $x_3 = 0$, and see how much we can increase $x_1$.
- We can increase $x_1$ to 1, at which point $x_5$ becomes 0.

- Rewrite equalities as follows.
  - $x_2 = 1 + x_1 - x_3$
  - $x_4 = 3 - x_1$
  - $x_5 = 1 - x_1 + x_3$
  - $z = 1 + 2x_1 - x_3$

- Set $x_3 = 0$, $x_1 = 1$, and update other variables
  - $x_2 = 2$, $x_4 = 2$, $x_5 = 0$.
- And $z = 3$.

\[
\begin{pmatrix}
  x_1 & x_2 & x_3 & x_4 & x_5 & z \\
  1 & 2 & 0 & 2 & 0 & 3
\end{pmatrix}
\]
An introductory example

- The new basis is \{1,2,4\}.
- Rewrite the tableau.

- Rewrite equalities as follows.
  \[
  \begin{align*}
  x_2 &= 1 + x_1 - x_3 \\
  x_4 &= 3 - x_1 \\
  x_5 &= 1 - x_1 + x_3 \\
  z &= 1 + 2x_1 - x_3 
  \end{align*}
  \]
- Set \(x_3 = 0\), \(x_1 = 1\), and update other variables
  \(x_2 = 2, x_4 = 2, x_5 = 0\).
- And \(z = 3\).

\[
\begin{pmatrix}
  x_1 & x_2 & x_3 & x_4 & x_5 & z \\
  1 & 2 & 0 & 2 & 0 & 3
\end{pmatrix}
\]
An introductory example

- The new basis is \{1,2,4\}.
- Rewrite the tableau.
- See which variable should increase to make \(z\) larger.
  - \(x_3\) in this case.
- See how much we can increase \(x_3\).
  - \(x_3 = 2\).
- Update \(x_i\)'s and \(z\).

- Rewrite equalities as follows.
  \[
  \begin{align*}
  x_1 &= 1 + x_3 - x_5 \\
  x_2 &= 2 - x_5 \\
  x_4 &= 2 - x_3 + x_5 \\
  z &= 3 + x_3 - 2x_5
  \end{align*}
  \]
- Set \(x_5 = 0, x_3 = 2\), and update other variables \(x_1 = 3, x_2 = 2, x_4 = 0\).
- And \(z = 5\).
An introductory example

- The new basis is \{1,2,3\}.
- Rewrite the tableau.
- See which variable should increase to make \(z\) larger.
  - None!
    - Both coefficients for \(x_4\) and \(x_5\) are negative now.
- Claim: We’ve found the optimal solution and optimal value! 😊

- Rewrite equalities as follows.
  \[
  x_1 = 3 - x_4 \\
  x_2 = 2 - x_5 \\
  x_3 = 2 - x_4 + x_5 \\
  z = 5 - x_4 - x_5
  \]
- Set \(x_5 = 0, x_3 = 2\), and update other variables \(x_1 = 3, x_2 = 2, x_4 = 0\).
- And \(z = 5\).

\[
\begin{pmatrix}
  x_1 & x_2 & x_3 & x_4 & x_5 & z \\
  3 & 2 & 2 & 0 & 0 & 5
\end{pmatrix}
\]
Now we make the intuitions formal.

We will rigorously define things like basis, feasible basis, tableau, ...

discuss the pivot steps,

and formalize the above procedure for general LP.
Basis

- In the matrix $A_{m \times n}$, a subset $B \subseteq [n]$ is a *basis* if those columns of $A$ in $B$ are linearly independent.
  - In other words, $A_B$ is nonsingular.
- Denote $N = [n] - B$.
  - $[n] = \{1,2,\ldots,n\}$.
- A basis $B$ is **feasible** if $A_B^{-1}b \geq 0$.
  - The inequality is entry-wise.
A (simplex) tableau $T(B)$ w.r.t. feasible basis $B$ is the following system of equations

$$T(B): \begin{cases} x_B = A_B^{-1}b - A_B^{-1}A_N x_N \\ z = c_B^T A_B^{-1} b + (c_N^T - c_B^T A_B^{-1} A_N) x_N \end{cases} \tag{1}$$

It looks complicated, but it just

- writes basis variables $x_B$ in terms of non-basis variables $x_N$
- add a new variable $z$ for the objective function value $c^T x$. (Details next.)
Tableau \( T(B) \):

\[
\begin{align*}
    x_B &= A_B^{-1}b - A_B^{-1}A_Nx_N \\
    z &= c_B^TA_B^{-1}b + (c_N^T - c_B^TA_B^{-1}A_N)x_N
\end{align*}
\] 

(1)

(2)

[Prop 1] If \( A_B \) is nonsingular, then 

\((x, z)\) satisfies \( T(B) \iff Ax = b, z = c^T x \)

Proof.

\( \Rightarrow: \) 

\[
    Ax = (A_B, A_N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = A_Bx_B + A_Nx_N = b
\]

\[
    c^T x = (c_B^T, c_N^T) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = c_B^Tx_B + c_N^Tx_N
\]

\[
    = c_B^TA_B^{-1}b - c_B^TA_B^{-1}A_Nx_N + c_N^Tx_N
\]

\( \Leftarrow: \) 

\[
    b = Ax = A_Bx_B + A_Nx_N.
\]

\[
    \therefore A_B^{-1}b = x_B + A_B^{-1}A_Nx_N.
\]

\[
    z = c^T x = c_B^Tx_B + c_N^Tx_N
\]

\[
    = c_B^TA_B^{-1}b - c_B^TA_B^{-1}A_Nx_N + c_N^Tx_N
\]
Tableau $T(B)$: \[
\begin{align*}
x_B &= A_B^{-1}b - A_B^{-1}A_Nx_N \\
z &= c_B^T A_B^{-1}b + (c_N^T - c_B^T A_B^{-1}A_N)x_N
\end{align*}
\]  

- Recall: A basis $B$ is feasible basis if $A_B^{-1}b \geq 0$.
- A feasible basis induces a feasible solution $x$, defined by $x_B = A_B^{-1}b$, $x_N = 0$.
- [Prop 2] If all the coefficients of $x_N$ in (2) are $\leq 0$, then the induced $x$ is optimal.
- Proof: $\forall$ feasible solution $x'$: $Ax' = b$ and $x' \geq 0$. Let $z' = c^T x'$, then by Prop 1, $(x', z')$ satisfies $T(B)$. So $c^T x' = z' = c_B^T A_B^{-1}b + (c_N^T - c_B^T A_B^{-1}A_N)x_N'$ \[
\begin{align*}
\leq c_B^T A_B^{-1}b + (c_N^T - c_B^T A_B^{-1}A_N)0 \\
= c_B^T A_B^{-1}b = c_B^T x_B = c^T x
\end{align*}
\]  // $x_B = A_B^{-1}b$, $x_N = 0$
Updating... \( T(B) \):

\[
\begin{align*}
x_B &= A_B^{-1}b - A_B^{-1}A_Nx_N \\
z &= c_B^T A_B^{-1}b + (c_N^T - c_B^T A_B^{-1}A_N)x_N
\end{align*}
\]

\( (1) \)

\( (2) \)

- When updating a tableau, we move a variable from \( N \) to \( B \), then move a variable from \( B \) to \( N \).
- The set of variables in \( N \) allowed to join \( B \) is:
  \[
  E = \{ j : \text{coefficient of } x_j \text{ in (2) is positive} \}
  \]
  - If \( E = \emptyset \), the induced \( x \) is optimal (by Prop 2). Output it.
- The set of variables in \( B \) allowed to leave is:
  \[
  L = \{ i : \text{as } x_j \uparrow, x_i \text{ in (1) drops below 0 the earliest} \}
  \]
  - If \( L = \emptyset \), then the LP is unbounded, because
    \[
    c^T x = z = c_B^T A_B^{-1}b + (c_N^T - c_B^T A_B^{-1}A_N)x_N
    \]
    gets increased with \( x_j \) to \(+\infty\).
The updating rule maintains the tableaus:

Theorem. \( \forall j \in E, i \in L, \)

\[ B \text{ is a feasible basis } \Rightarrow \text{So is } B \cup \{j\}\backslash\{i\}. \]

Proof omitted.

Geometric meaning: walk from one vertex to another.
Pivoting rule: which $j$ in $E$ (and which $i$ in $L$) to pick?

- **Largest coefficient** in (2).
  - Dantzig’s original.
- **Largest increase** of $z$.
- **Steepest edge**: i.e. closest to the vector $c$.
  - Champion in practice.
- **Bland’s rule**: smallest index.
  - Prevents cycling.
- **Random**:
  - Best provable bounds.
Picking the initial feasible solution

- Assume \( b \geq 0 \). \( \times (-1) \) on some rows if needed.
- [Fact] \( \exists x \in \mathbb{R}^n \) s.t. \( Ax = b \) and \( x \geq 0 \)
  \( \iff \) the following LP has optimal value 0
  \[
  \max \quad -(y_{n+1} + y_{n+2} + \cdots + y_{n+m})
  \]
  \[s.t. \quad (A, I_m) \begin{pmatrix} y_1 \\ \vdots \\ y_{n+m} \end{pmatrix} = b\]
  \( y_1, \ldots, y_n, y_{n+1}, \ldots, y_{n+m} \geq 0 \)
  - The new LP has variables \( y_1, \ldots, y_n, y_{n+1}, \ldots, y_{n+m} \).

- Proof. \( \Rightarrow \): ① \( \text{opt} \leq 0 \). ② \( y = (x, 0^m) \) achieves 0.
  \( \iff \) Take \( x = (y_1, \ldots, y_n)^T \). \( \therefore \) \( \text{opt} = 0, y_{n+1}, \ldots, y_{n+m} \geq 0, \therefore y_{n+1} = \cdots = y_{n+m} = 0 \). So \( Ax = b \) and \( x \geq 0 \).
Solve the new LP first

- Note that the new LP has a feasible basis easily found: \( B^0 = \{n + 1, \ldots, n + m\} \).
  - \( A_{B^0} = I_m \), and thus \( A_{B^0}^{-1}b = b \geq 0 \).
- Solve this new LP, obtaining an opt. solution \( y \)
  - If optimal value \( \neq 0 \): the original LP is not feasible.
  - If optimal value \( = 0 \): \( y_{n+1} = \cdots = y_{n+m} = 0 \)
    - \( B_+ \) \( \equiv \{i: y_i > 0\} \subseteq [n] \).
- Columns in \( B_+ \subseteq [n] \) are linearly independent. Expand it to \( m \) linearly independent columns \( B \subseteq [n] \). Then \( B \) is a feasible basis for the original LP.
  - \( A_B^{-1}b = A_B^{-1}(A, I)y = A_B^{-1}(A_By_B + A_Ny_N) = y_B \geq 0 \).
**Simplex Alg: putting everything together**

- If no feasible basis is available,
  - solve
    
    \[
    \max \quad - (y_{n+1} + y_{n+2} + \cdots + y_{n+m})
    \]
    
    \[
    \begin{align*}
    \text{s.t.} \quad (A, I_m) \begin{pmatrix}
    y_1 \\
    \vdots \\
    y_{n+m}
    \end{pmatrix} &= b \\
    y_1, \ldots, y_n, y_{n+1}, \ldots, y_{n+m} &\geq 0
    \end{align*}
    \]
    
  - If optimal value \(\neq 0\): original LP is infeasible.
  - If optimal value = 0: get a feasible basis \(B\) for the original LP.
Simplex Algorithm: continued

- For the feasible basis $B \subseteq [n]$, compute tableau

  \[
  T(B): \begin{cases}
  x_B = A_B^{-1}b - A_B^{-1}A_Nx_N \\
  z = c_B^T A_B^{-1}b + (c_N^T - c_B^T A_B^{-1} A_N)x_N
  \end{cases}
  \]

  \hspace{1cm} (1)

- if all coefficients of $x_N$ in (2) are $\leq 0$
  - output optimal solution $x = (x_B, x_N)$, with $x_B$ in (1), and $x_N = 0$. (opt value: $c^T x = z$.)

- else
  - pick $j \in E$ by some pivoting rule.
  - if the column of $j$ in tableau $\geq 0$, output “LP is unbounded”.
  - else
    - $E = \{j: \text{coefficient of } x_j \text{ in (2) is positive}\}$
    - $L = \{i: \text{as } x_j \uparrow, x_i \text{ in (1) drops below 0 the earliest}\}$
    - Pick $i \in L$ by some pivoting rule
    - $B \leftarrow B \cup \{j\} \setminus \{i\}$ and go to the first step in this slide.
In practice: Very efficient.

- Typical: $2m \sim 3m$ pivoting steps.
  - $m$: number of constraints

In theory:

- Finite: Some pivoting rules prevent cycling.
- Worst case complexity is exponential for most known deterministic pivoting rules.
- No “pivoting rule”, deterministic or randomized, with polynomial worst-case complexity known.
- Best bound: $e^{\Theta(\sqrt{n \log n})}$ with $n$ variables and $n$ constraints
Theory of simplex method

- Actually we don’t even know the complexity of best possible pivoting rule.
- Hirsch Conj: It’s $O(n)$.
- Best upper bound (Kalai-Kleitman): $n^{1+\ln(n)}$.
- **Smoothed complexity:** For any LP, perturbing its coefficients by small random amounts makes the simplex method (w/ a certain pivoting rule) polynomial time complexity.
  - See [here](#) for surveys/papers.
Duality

- Recall our problem:
  - max $x_1 + 6x_2$
  - s.t. $x_1 \leq 200$ (1)
    $x_2 \leq 300$ (2)
    $x_1 + x_2 \leq 400$ (3)
    $x_1, x_2 \geq 0$ (4)

- Let’s see how good the solution could be.
- $(1) + 6 \times (2)$:
  - $x_1 + 6x_2 \leq 200 + 6 \times 300 = 2000$
- It’s an upper bound.
- $5 \times (2) + (3)$:
  - $5x_2 + (x_1 + x_2) \leq 5 \times 300 + 400 = 1900$
- It’s a better upper bound.
- What’s the best upper bound obtained this way?
Duality

Recall our problem:
- \( \max x_1 + 6x_2 \)
- s.t. \( x_1 \leq 200 \) \( \quad \) (1)
  \( x_2 \leq 300 \) \( \quad \) (2)
  \( x_1 + x_2 \leq 400 \) \( \quad \) (3)
  \( x_1, x_2 \geq 0 \) \( \quad \) (4)

In general:
- \( y_1 \times (1) + y_2 \times (2) + y_3 \times (3): \)
  \( (y_1 + y_3)x_1 + (y_2 + y_3)x_2 \leq 200y_1 + 300y_2 + 400y_3. \)
- If \( y_1 + y_3 \geq 1 \) and \( y_2 + y_3 \geq 6 \), we get an upper bound:
  \( x_1 + 6x_2 \leq 200y_1 + 300y_2 + 400y_3. \)

The best upper bound?
\( \min 200y_1 + 300y_2 + 400y_3 \)
- s.t. \( y_1 + y_3 \geq 1 \)
  \( y_2 + y_3 \geq 6 \)
  \( y_1, y_2, y_3 \geq 0 \)

This is another linear programming problem. --- dual of the original LP.
Making it formal

- **Primal**
  \[
  \begin{align*}
  \text{max} & \quad c^T x \\
  \text{s.t.} & \quad Ax \leq b \\
  & \quad x \geq 0
  \end{align*}
  \]

- **Dual**
  \[
  \begin{align*}
  \text{min} & \quad b^T y \\
  \text{s.t.} & \quad A^T y \geq c \\
  & \quad y \geq 0
  \end{align*}
  \]
### Dualization Recipe

<table>
<thead>
<tr>
<th></th>
<th>Primal linear program</th>
<th>Dual linear program</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variables</strong></td>
<td>$x_1, x_2, \ldots, x_n$</td>
<td>$y_1, y_2, \ldots, y_m$</td>
</tr>
<tr>
<td><strong>Matrix</strong></td>
<td>$A$</td>
<td>$A^T$</td>
</tr>
<tr>
<td><strong>Right-hand side</strong></td>
<td>$b$</td>
<td>$c$</td>
</tr>
<tr>
<td><strong>Objective function</strong></td>
<td>$\max c^T x$</td>
<td>$\min b^T y$</td>
</tr>
<tr>
<td><strong>Constraints</strong></td>
<td>$i$th constraint has $\leq$</td>
<td>$y_i \geq 0$</td>
</tr>
<tr>
<td></td>
<td>$\geq$</td>
<td>$y_i \leq 0$</td>
</tr>
<tr>
<td></td>
<td>$=$</td>
<td>$y_i \in \mathbb{R}$</td>
</tr>
<tr>
<td></td>
<td>$x_j \geq 0$</td>
<td>$j$th constraint has $\geq$</td>
</tr>
<tr>
<td></td>
<td>$x_j \leq 0$</td>
<td>$\leq$</td>
</tr>
<tr>
<td></td>
<td>$x_j \in \mathbb{R}$</td>
<td>$=$</td>
</tr>
</tbody>
</table>

- **Primal**
  - $\max c^T x$
  - s.t. $Ax \leq b$
  - $x \geq 0$
- **Dual**
  - $\min b^T y$
  - s.t. $A^T y \geq c$
  - $y \geq 0$

- **Primal**
  - $\max c^T x$
  - s.t. $Ax = b$
  - $x \geq 0$
- **Dual**
  - $\min b^T y$
  - s.t. $A^T y \geq c$
## Dualization Recipe

<table>
<thead>
<tr>
<th></th>
<th>Primal linear program</th>
<th>Dual linear program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>$x_1, x_2, \ldots, x_n$</td>
<td>$y_1, y_2, \ldots, y_m$</td>
</tr>
<tr>
<td>Matrix</td>
<td>$A$</td>
<td>$A^T$</td>
</tr>
<tr>
<td>Right-hand side</td>
<td>$b$</td>
<td>$c$</td>
</tr>
<tr>
<td>Objective function</td>
<td>$\max c^T x$</td>
<td>$\min b^T y$</td>
</tr>
<tr>
<td>Constraints</td>
<td>$i$th constraint has $\le$ $y_i \ge 0$</td>
<td>[ y_i \ge 0 ]</td>
</tr>
<tr>
<td></td>
<td>$\ge$</td>
<td>$y_i \le 0$</td>
</tr>
<tr>
<td></td>
<td>$=$</td>
<td>$y_i \in \mathbb{R}$</td>
</tr>
<tr>
<td></td>
<td>$x_j \ge 0$</td>
<td>$j$th constraint has $\ge$</td>
</tr>
<tr>
<td></td>
<td>$x_j \le 0$</td>
<td>$\le$</td>
</tr>
<tr>
<td></td>
<td>$x_j \in \mathbb{R}$</td>
<td>$= \quad$</td>
</tr>
</tbody>
</table>

### Primal
- $\max x_1 + 6x_2$
- s.t. $x_1 \leq 200$  \[ (1) \]
  $x_2 \leq 300$  \[ (2) \]
  $x_1 + x_2 \leq 400$  \[ (3) \]
  $x_1, x_2 \geq 0$

### Dual
- $\min 200y_1 + 300y_2 + 400y_3$
- s.t. $y_1 + y_3 \geq 1$  \[ (1) \]
  $y_2 + y_3 \geq 6$  \[ (2) \]
  $y_1, y_2, y_3 \geq 0$
Strong duality

- The primal gives **lower bounds** for the dual
- The dual gives **upper bounds** for the primal
- [Strong duality] For linear programming, **optimal primal value = optimal dual value**
  - If both exist, then they are equal
  - If one is infinity, then the other is infeasible
A physical interpretation of duality

- Consider

**Primal**
\[
\begin{align*}
\text{max } & \quad c^T x \\
\text{s.t. } & \quad Ax \leq b
\end{align*}
\]

**Dual**
\[
\begin{align*}
\text{min } & \quad b^T y \\
\text{s.t. } & \quad A^T y \geq c \\
& \quad y \geq 0
\end{align*}
\]

- Rotate s.t. \( c \) points downward.
- Each inequality \( a_i^T x \leq b_i \) gives a half-space, with outer normal \( a_i \).
  - Denote the face by \( S_i \).
A physical interpretation of duality

- **Primal**
  \[
  \begin{align*}
  \text{max} & \quad c^T x \\
  \text{s.t.} & \quad Ax \leq b 
  \end{align*}
  \]

- **Dual**
  \[
  \begin{align*}
  \text{min} & \quad b^T y \\
  \text{s.t.} & \quad A^T y \geq c, \quad y \geq 0 
  \end{align*}
  \]

- Drop a steel ball and let it rolls down to the lowest point \( x^* \).
  - \( x^* \) is an optimal solution.
  - \( x^* \) touches some faces \( S_i \).
    - Let \( D = \{i: x^* \text{ touches } S_i\} \).
    - Note: \( x^* \text{ touches } S_i \iff a_i^T x^* = b_i \).
A physical interpretation of duality

- **Primal**
  \[
  \text{max } c^T x \quad \text{min } b^T y \\
  \text{s.t. } Ax \leq b \quad \text{s.t. } A^T y \geq c
  \]
  \(y \geq 0\)

- **Dual**

Consider the gravity force \(F\).
- It’s decomposed into forces of pressure on the faces \(S_i (i \in D)\): \(F = \sum_{i \in D} F_i\).
- \(F_i\) is directed outward, along the direction \(a_i\).
- So \(\sum_{i \in D} y_i^* a_i = c\) and \(y_i^* \geq 0, \forall i \in D\).
A physical interpretation of duality

- **Primal**
  \[
  \begin{align*}
  \text{max } & \quad c^T x \\
  \text{s.t. } & \quad Ax \leq b
  \end{align*}
  \]

- **Dual**
  \[
  \begin{align*}
  \text{min } & \quad b^T y \\
  \text{s.t. } & \quad A^T y \geq c \\
  & \quad y \geq 0
  \end{align*}
  \]

- Now set \( y_i^* = 0 \), \( \forall i \notin D \).

- \[
  \sum_{i=1}^{m} y_i^* a_i = \sum_{i \in D} y_i^* a_i = c.
  \]
  - That is, \( A^T y^* = c \).

- So this \( y^* \) is feasible for Dual.
A physical interpretation of duality

- **Primal**
  \[
  \begin{align*}
  \text{max} & \quad c^T x \\
  \text{s.t.} & \quad Ax \leq b
  \end{align*}
  \]

- **Dual**
  \[
  \begin{align*}
  \text{min} & \quad b^T y \\
  \text{s.t.} & \quad A^T y \geq c \\
  & \quad y \geq 0
  \end{align*}
  \]

Consider \((y^*)^T(Ax^* - b)\).

- For \(i \in D\): \(a_i^T x^* = b_i\), so \(a_i^T x^* - b_i = 0\).
- For \(i \notin D\): \(y_i^* = 0\)

Thus \((y^*)^T(Ax^* - b) = 0\).

Therefore,
\[
(y^*)^T b = (y^*)^T A x^* = (A^T y^*)^T x^* = c^T x^*
\]

We just “proved” strong duality by physics!
Application: Zero-sum game

- Two players: **Row** and **Column**

- **Payoff matrix**
  - \((i, j)\): Row pays to Column when Row takes strategy \(i\) and Column takes strategy \(j\)

- **Row wants to minimize; Column wants to maximize.**
- **Game:** You don’t know others’ strategy.
Who moves first?

- They both want to minimize their loss in the worst case (of the other’s strategy).
  - **Row:** \(\min_i \max_j a_{ij}\)
  - **Column:** \(\max_j \min_i a_{ij}\)
- **Fact:** \(\min_i \max_j a_{ij} \geq \max_j \min_i a_{ij}\)
- **Game theoretical interpretation:**
The player making the first move has disadvantage.
  - Consider the Rock-Paper-Scissors game: If you move first, then you’ll lose for sure.
Mixed strategy

- Mixed strategy: a randomized choice.
  - Row: strategy \( i \) with prob. \( p_i \).
  - Column: strategy \( j \) with prob. \( q_j \).

- Now the tasks are:
  - Row: \( \min_{p_i} \max_{q_j} \sum_i p_i q_j a_{ij} \)
  - Column: \( \max_{q_j} \min_{p_i} \sum_j p_i q_j a_{ij} \)

- Fact: the inner opt can be achieved by a deterministic strategy.

- So the tasks become:
  - Row: \( \min_{p_i} \max_j \sum_i p_i a_{ij} \)
  - Column: \( \max_{q_j} \min_i \sum_j q_j a_{ij} \)
Minimax

- Minimax theorem:
  \[
  \min_{\{p_i\}} \max_j \sum_i p_i a_{ij} = \max_{\{q_j\}} \min_i \sum_j q_j a_{ij}
  \]

- The player who moves first doesn’t have disadvantage any more!
  - Consider the Rock-Paper-Scissors game again: Each player wants to use \(\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)\) distribution on her choices.
Proof by LP duality

Row:

\[
\min_{\{p_i\}} \max_j \sum_i p_i a_{ij}
\]

- min \(z\)
- s.t. \(\sum_i p_i a_{ij} \leq z, \forall j\)
  \(0 \leq p_i \leq 1\)
  \(\sum_i p_i = 1\)

Column:

\[
\max_{\{q_j\}} \min_i \sum_j q_j a_{ij}
\]

- max \(w\)
- s.t. \(\sum_j q_j a_{ij} \geq w, \forall i\)
  \(0 \leq q_j \leq 1\)
  \(\sum_j q_j = 1\)

Observation: These two LP’s are dual to each other.

Thus they have the same optimal value.
Summary

- **Linear program**: a very useful framework

- **Algorithms**:
  - Simplex: exponential in worst-case, efficient in practice.
  - Ellipsoid: polynomial in worst-case but usually not efficient enough for practical data.
  - Interior point: polynomial in worst-case and efficient in practice.

- **Duality**: Each LP has a **dual LP**, which has the same optimal value as the primal LP if both are feasible.

Many references on LP or other optimization theories.