CSC3160: Design and Analysis of Algorithms

Week 6: Linear Programming

Instructor: Shengyu Zhang

LP

- 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

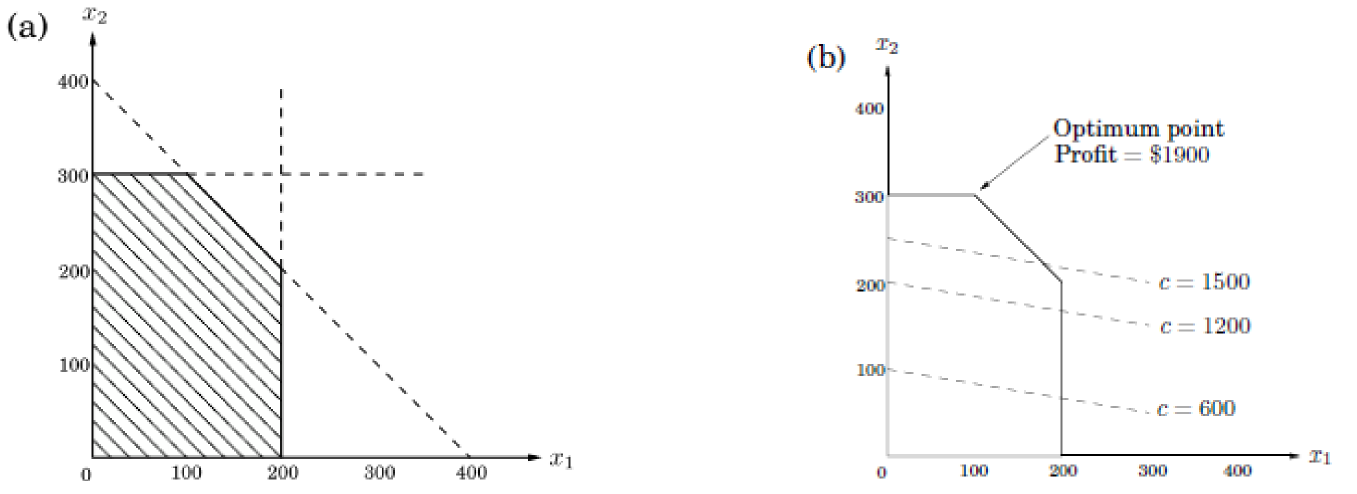
How to solve?

- 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:

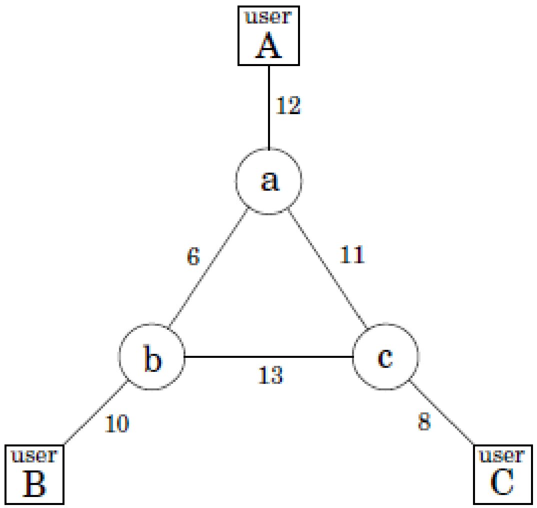
$$\max x_1 + 6x_2$$

Illustrative figures



Example 2

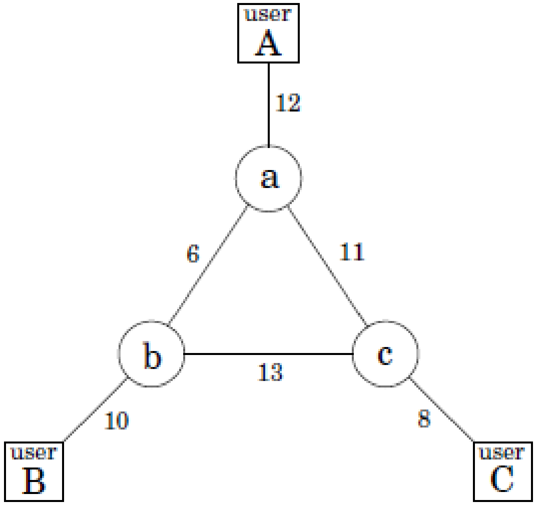
- We are managing a network with bandwidth as shown by numbers on edges.
  - Bandwidth: max units of flows
- 3 connections: AB, BC, CA
  - We get \$3, \$2, \$4 for providing them respectively.
  - Two routes for each connection: short and long.
- *Question: How to route the connections to maximize our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



Example 2

x_{AB}: amount of flow of the short route
x'_{AB}: amount of flow of the long route

- 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:

Superscript ^{T}: transpose of vectors.

$$\begin{aligned} \max \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \max \quad & c_1 x_1 + \cdots + c_n x_n \\ \text{s.t.} \quad & a_{i1} x_1 + \cdots + a_{in} x_n = b_i, \\ & \qquad \forall i = 1, \ldots, m \\ & x_i \geq 0, \ \forall i = 1, \ldots, n \end{aligned}$$

Inequality: entry-wise

  - $\boldsymbol{x}$: variables.
  - $(A, \boldsymbol{b})$: coefficients in constraints

Transformations between forms

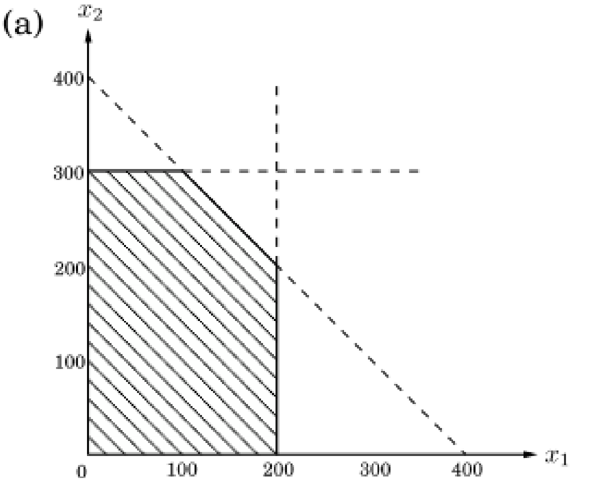
- Min vs. max:
  - $\min \boldsymbol{c}^T\boldsymbol{x} \Leftrightarrow \max -\boldsymbol{c}^T\boldsymbol{x}$
- Inequality directions:
  - $\boldsymbol{a}_{\boldsymbol{i}}^T\boldsymbol{x} \geq b_i \Leftrightarrow -\boldsymbol{a}_{\boldsymbol{i}}^T\boldsymbol{x} \leq -b_i$
- Equalities to inequalities: ($\boldsymbol{a}_{\boldsymbol{i}}$: row i in matrix A)
  - $\boldsymbol{a}_{\boldsymbol{i}}^T\boldsymbol{x} = b_i \Leftrightarrow \boldsymbol{a}_{\boldsymbol{i}}^T\boldsymbol{x} \geq b_i$, and $\boldsymbol{a}_{\boldsymbol{i}}^T\boldsymbol{x} \leq b_i$.

Transformations between forms

- Inequalities to equalities:
  - $\boldsymbol{a_i^T x} \geq b_i \Leftrightarrow \boldsymbol{a_i^T x} = b_i + s_i, s_i \geq 0$
    - The newly introduced variable s_i is called *slack variable*
- "Unrestricted" to "nonnegative constraint":
  - x_i unrestricted $\Leftrightarrow x_i = s - t, s \geq 0, t \geq 0$

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

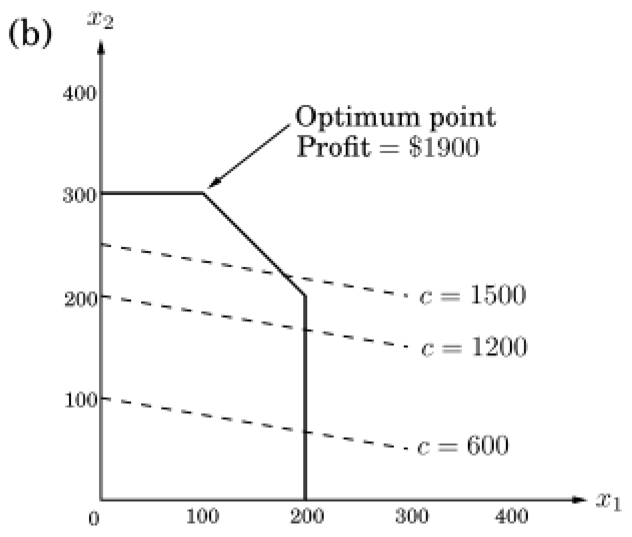


- All constraints are satisfied: the intersection of these half spaces. --- feasible region.
  - Feasible region nonempty: LP is feasible
  - Feasible region empty: LP is infeasible

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.



Possibilities of solution

- Infeasible: no solution satisfying

$$A\boldsymbol{x} = \boldsymbol{b} \text{ and } \boldsymbol{x} \geq 0.$$

  - Example? Picture?
- Feasible but unbounded: $\boldsymbol{c}^T\boldsymbol{x}$ can be arbitrarily large.
  - Example? Picture?
- Feasible and bounded: there is an optimal solution.
  - Example? Picture?

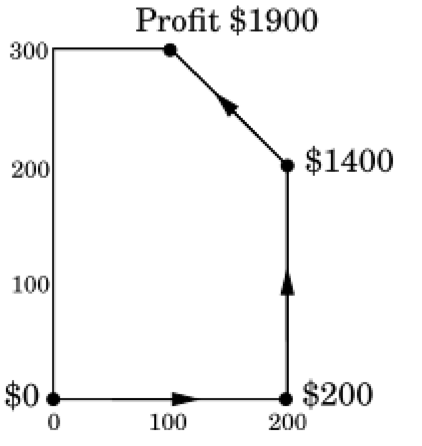
Three Algorithms for LP

- 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).

Weakly polynomial time

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 ⇔ 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

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & -x_1 + x_2 + x_3 = 1 \\ & x_1 + x_4 = 3 \\ & x_2 + x_5 = 2 \\ & x_1, \dots, x_5 \geq 0 \end{aligned}$$

- 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 $z = obj = x_1 + x_2$.

- Rewrite equalities as follows. (A tableau.)

$$\begin{aligned} x_3 &= 1 + x_1 - x_2 \\ x_4 &= 3 - x_1 \\ x_5 &= 2 - x_2 \\ z &= x_1 + x_2 \end{aligned}$$

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 $z = obj = x_1 + x_2$.
- $B = \{3,4,5\}$ 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 $z = 0$.
- $\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & z \\ 0 & 0 & 1 & 3 & 2 & 0 \end{pmatrix}$

An introductory example

- Now we want to improve $z = obj = x_1 + x_2$.
- 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.

$$x_3 = 1 + x_1 - x_2$$
$$x_4 = 3 - x_1$$
$$x_5 = 2 - x_2$$
$$z = x_1 + x_2$$

- Set $x_1 = x_2 = 0$, and get $x_3 = 1, x_4 = 3, x_5 = 2$.
- And $z = 0$.
- $\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & z \\ 0 & 0 & 1 & 3 & 2 & 0 \end{pmatrix}$

An introductory example

- 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.

$$x_2 = 1 + x_1 - x_3$$
$$x_4 = 3 - x_1$$
$$x_5 = 1 - x_1 + x_3$$
$$z = 1 + 2x_1 - x_3$$

- 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.
- 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.
  $$x_1 = 1 + x_3 - x_5$$
  $$x_2 = 2 - x_5$$
  $$x_4 = 2 - x_3 + x_5$$
  $$z = 3 + x_3 - 2x_5$$
- Set $x_5 = 0, x_3 = 2$, and update other variables $x_1 = 3, x_2 = 2, x_4 = 0.$
- And $z = 5$.
- $\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & z \\ 3 & 2 & 2 & 0 & 0 & 5 \end{pmatrix}$

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 $z = 5$.
- $\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & z \\ 3 & 2 & 2 & 0 & 0 & 5 \end{pmatrix}$

Formal treatment

- Now we make the intuitions formal.

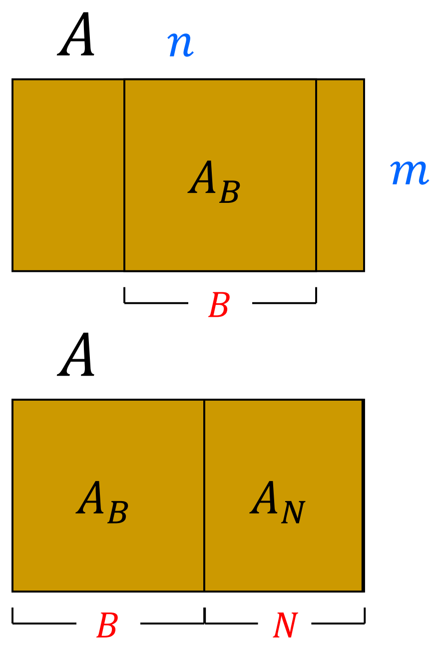
- We will rigorously define things like basis, feasible basis, tableau, …
- discuss the pivot steps,
- and formalize the above procedure for general LP.

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

$$A_B^{-1} \boldsymbol{b} \geq \mathbf{0}.$$

  - The inequality is entry-wise.



(Simplex) tableau

- A (simplex) tableau $T(B)$ w.r.t. feasible basis B is the following system of equations

$$T(B): \begin{cases} \boldsymbol{x}_B = A_B^{-1}\boldsymbol{b} - A_B^{-1}A_N\boldsymbol{x}_N & (1) \\ z = \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} + (\boldsymbol{c}_N^T - \boldsymbol{c}_B^T A_B^{-1}A_N)\boldsymbol{x}_N & (2) \end{cases}$$

- It looks complicated, but it just
  - writes basis variables $\boldsymbol{x}_B$ in terms of non-basis variables $\boldsymbol{x}_N$
  - add a new variable z for the objective function value $\boldsymbol{c}^T\boldsymbol{x}$. (Details next.)

Tableau $T(B)$: $\begin{cases} \boldsymbol{x}_B = A_B^{-1}b - A_B^{-1}A_N\boldsymbol{x}_N & (1) \\ z = \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} + (\boldsymbol{c}_N^T - \boldsymbol{c}_B^T A_B^{-1}A_N)\boldsymbol{x}_N & (2) \end{cases}$

- [Prop 1] If A_B is nonsingular, then
  $(\boldsymbol{x}, z)$ satisfies $T(B)$ $\Leftrightarrow$ $A\boldsymbol{x} = \boldsymbol{b}, z = \boldsymbol{c}^T\boldsymbol{x}$
- Proof.
  - $\Rightarrow$: $A\boldsymbol{x} = (A_B, A_N)\begin{pmatrix}\boldsymbol{x}_B \\ \boldsymbol{x}_N\end{pmatrix} = A_B\boldsymbol{x}_B + A_N\boldsymbol{x}_N$
    $= \boldsymbol{b} - A_N\boldsymbol{x}_N + A_N\boldsymbol{x}_N = \boldsymbol{b}$
    $\boldsymbol{c}^T\boldsymbol{x} = (\boldsymbol{c}_B^T, \boldsymbol{c}_N^T)\begin{pmatrix}\boldsymbol{x}_B \\ \boldsymbol{x}_N\end{pmatrix} = \boldsymbol{c}_B^T\boldsymbol{x}_B + \boldsymbol{c}_N^T\boldsymbol{x}_N$
    $= \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} - \boldsymbol{c}_B^T A_B^{-1}A_N\boldsymbol{x}_N + \boldsymbol{c}_N^T\boldsymbol{x}_N$
  - $\Leftarrow$: $\boldsymbol{b} = A\boldsymbol{x} = A_B\boldsymbol{x}_B + A_N\boldsymbol{x}_N$. $\therefore A_B^{-1}\boldsymbol{b} = \boldsymbol{x}_B + A_B^{-1}A_N\boldsymbol{x}_N$.
    $z = \boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{c}_B^T\boldsymbol{x}_B + \boldsymbol{c}_N^T\boldsymbol{x}_N$
    $= \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} - \boldsymbol{c}_B^T A_B^{-1}A_N\boldsymbol{x}_N + \boldsymbol{c}_N^T\boldsymbol{x}_N$

Tableau $T(B)$: $\begin{cases} \boldsymbol{x}_B = A_B^{-1}b - A_B^{-1}A_N\boldsymbol{x}_N & (1) \\ z = \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} + (\boldsymbol{c}_N^T - \boldsymbol{c}_B^T A_B^{-1}A_N)\boldsymbol{x}_N & (2) \end{cases}$

- Recall: A basis B is feasible basis if $A_B^{-1}\boldsymbol{b} \geq \boldsymbol{0}$.
- A feasible basis induces a feasible solution $\boldsymbol{x}$, defined by $\boldsymbol{x}_B = A_B^{-1}\boldsymbol{b}$, $\boldsymbol{x}_N = \boldsymbol{0}$.
- [Prop 2] If all the coefficients of $\boldsymbol{x}_N$ in (2) are ≤ 0, then the induced $\boldsymbol{x}$ is optimal.
- Proof: ∀ feasible solution $\boldsymbol{x}'$: $A\boldsymbol{x}' = \boldsymbol{b}$ and $\boldsymbol{x}' \geq 0$. Let $z' = \boldsymbol{c}^T\boldsymbol{x}'$, then by Prop 1, $(\boldsymbol{x}', \boldsymbol{z}')$ satisfies $T(B)$. So

$$\begin{aligned} \boldsymbol{c}^T\boldsymbol{x}' = z' &= \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} + (\boldsymbol{c}_N^T - \boldsymbol{c}_B^T A_B^{-1}A_N)\boldsymbol{x}'_N \\ &\leq \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} + (\boldsymbol{c}_N^T - \boldsymbol{c}_B^T A_B^{-1}A_N)\boldsymbol{0} \qquad // \ \boldsymbol{x}' \geq \boldsymbol{0} \\ &= \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} = \boldsymbol{c}_B^T\boldsymbol{x}_B = \boldsymbol{c}^T\boldsymbol{x} \qquad // \ \boldsymbol{x}_B = A_B^{-1}\boldsymbol{b}, \boldsymbol{x}_N = \boldsymbol{0} \end{aligned}$$

Updating… $T(B)$: $\begin{cases} \boldsymbol{x}_B = A_B^{-1}b - A_B^{-1}A_N\boldsymbol{x}_N & (1) \\ z = \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} + (\boldsymbol{c}_N^T - \boldsymbol{c}_B^T A_B^{-1}A_N)\boldsymbol{x}_N & (2) \end{cases}$

- 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

    $$\boldsymbol{c}^T\boldsymbol{x} = z = \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} + (\boldsymbol{c}_N^T - \boldsymbol{c}_B^T A_B^{-1}A_N)\boldsymbol{x}_N$$

    gets increased with x_j to $+\infty$.

Updating

- The updating rule maintains the tableaus:

- Theorem. $\forall j \in E, i \in L,$

  B is a feasible basis $\Rightarrow$ So is $B \cup \{j\} \setminus \{i\}$.

- 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 $\boldsymbol{b} \geq 0$. $\times(-1)$ on some rows if needed.
- [Fact] $\exists \boldsymbol{x} \in \mathbb{R}^n$ s.t. $A\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{x} \geq \boldsymbol{0}$

  $\Leftrightarrow$ the following LP has optimal value 0

$$\max \quad -(y_{n+1} + y_{n+2} + \cdots + y_{n+m})$$
$$s.t. \quad (A, I_m)\begin{pmatrix} y_1 \\ \vdots \\ y_{n+m} \end{pmatrix} = \boldsymbol{b}$$
$$y_1, \dots, y_n, y_{n+1}, \dots, y_{n+m} \geq 0$$

  - The new LP has variables $y_1, \dots, y_n, y_{n+1}, \dots, y_{n+m}$.
- Proof. $\Rightarrow$: ① opt ≤ 0. ② $y = (\boldsymbol{x}, 0^m)$ achieves 0.
  $\Leftarrow$: Take $\boldsymbol{x} = (y_1, \dots, y_n)^T$. $\because$ opt $= 0$, $y_{n+1}, \dots, y_{n+m} \geq 0$, $\therefore y_{n+1} = \cdots = y_{n+m} = 0$. So $A\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{x} \geq \boldsymbol{0}$.

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 $A_{B^0}^{-1}\boldsymbol{b} = \boldsymbol{b} \geq 0$.
- Solve this new LP, obtaining an opt. solution y
  - If optimal value $\neq 0$: the original LP is not feasible.
  - If optimal value $= 0$: $y_{n+1} = \cdots = y_{n+m} = 0$
    - $B_+ \stackrel{\text{def}}{=} \{i: y_i > 0\} \subseteq [n]$.
    - 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.
      - $A_B^{-1}\boldsymbol{b} = A_B^{-1}(A, I)\boldsymbol{y} = A_B^{-1}(A_B\boldsymbol{y}_B + A_N\boldsymbol{y}_N) = \boldsymbol{y}_B \geq 0$.

Simplex Alg: putting everything together

- If no feasible basis is available,
  - solve

$$\begin{aligned} \max \quad & -(y_{n+1} + y_{n+2} + \cdots + y_{n+m}) \\ s.t. \quad & (A, I_m)\begin{pmatrix} y_1 \\ \vdots \\ y_{n+m} \end{pmatrix} = \boldsymbol{b} \\ & y_1, \dots, y_n, y_{n+1}, \dots, y_{n+m} \geq 0 \end{aligned}$$

  - If optimal value ≠ 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

$$T(B): \begin{cases} \boldsymbol{x}_B = A_B^{-1}\boldsymbol{b} - A_B^{-1}A_N\boldsymbol{x}_N & (1) \\ z = \boldsymbol{c}_B^T A_B^{-1}\boldsymbol{b} + (\boldsymbol{c}_N^T - \boldsymbol{c}_B^T A_B^{-1}A_N)\boldsymbol{x}_N & (2) \end{cases}$$

- **if** all coefficients of x_N in (2) are ≤ 0
  - output optimal solution $\boldsymbol{x} = (\boldsymbol{x}_B, \boldsymbol{x}_N)$, with $\boldsymbol{x}_B$ in (1), and $\boldsymbol{x}_N = 0$. (opt value: $\boldsymbol{c}^T\boldsymbol{x} = z$.)
- **else**

  $E = \{j\text{: coefficient of } x_j \text{ in } (2) \text{ is positive}\}$

  - pick $j \in E$ by some pivoting rule.
  - **if** the column of j in tableau ≥ 0, output "LP is unbounded".
  - **else**

    $L = \{i\text{: as } x_j \uparrow, x_i \text{ in } (1) \text{ drops below 0 the earliest}\}$

    - Pick $i \in L$ by some pivoting rule
    - $B \leftarrow B \cup \{j\} \backslash \{i\}$ and go to the first step in this slide.

Efficiency

- 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: $e^{\Theta\left(\sqrt{n \log n}\right)}$ 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. $x_1 \le 200$ (1)

    $x_2 \le 300$ (2)

    $x_1 + x_2 \le 400$ (3)

    $x_1, x_2 \ge 0$ (4)

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

- In general:
  - $y_1 \times (1) + y_2 \times (2) + y_3 \times (3)$: $(y_1 + y_3)x_1 + (y_2 + y_3)x_2 \le 200y_1 + 300y_2 + 400y_3$.
  - If $y_1 + y_3 \ge 1$ and $y_2 + y_3 \ge 6$, we get an upper bound: $x_1 + 6x_2 \le 200y_1 + 300y_2 + 400y_3$.
- The best upper bound?

  min $200y_1 + 300y_2 + 400y_3$

  s.t. $y_1 + y_3 \ge 1$

  $y_2 + y_3 \ge 6$

  $y_1, y_2, y_3 \ge 0$

Making it formal

- Primal

$$\begin{aligned} \max \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

→

- Dual

$$\begin{aligned} \min \quad & \boldsymbol{b}^T \boldsymbol{y} \\ \text{s.t.} \quad & A^T\boldsymbol{y} \geq \boldsymbol{c} \\ & \boldsymbol{y} \geq 0 \end{aligned}$$

General form of the LP-duality

- Primal

  max $\boldsymbol{c}^T\boldsymbol{x}$

  s.t. $A\boldsymbol{x} \leq \boldsymbol{b}$

  $\boldsymbol{x} \geq 0$

  $\Longleftrightarrow$

  Dual

  min $\boldsymbol{b}^T\boldsymbol{y}$

  s.t. $A^T\boldsymbol{y} \geq \boldsymbol{c}$

  $\boldsymbol{y} \geq 0$

- max $\boldsymbol{c}^T\boldsymbol{x}$

  s.t. $A\boldsymbol{x} = \boldsymbol{b}$

  $\boldsymbol{x} \geq 0$

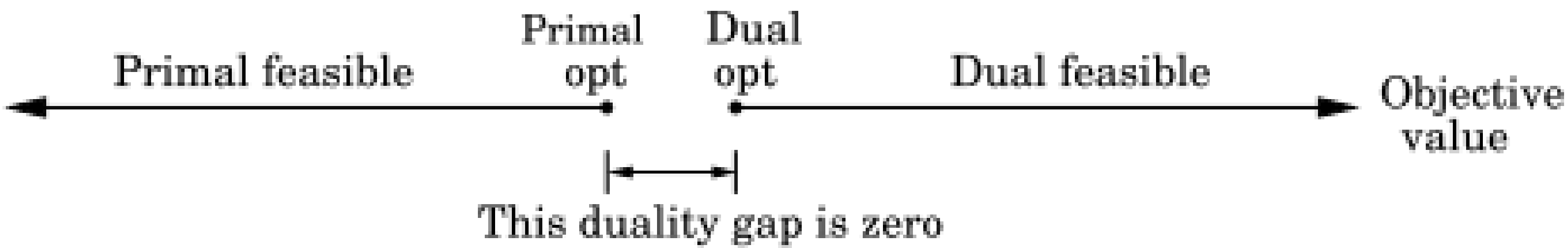
  $\Longleftrightarrow$

  min $\boldsymbol{b}^T\boldsymbol{y}$

  s.t. $A^T\boldsymbol{y} \geq \boldsymbol{c}$

- variable $\leftrightarrow$ constraint
- max $\leftrightarrow$ min
- $\boldsymbol{b} \leftrightarrow \boldsymbol{c}$
- contraints $\geq$ / $\leq$ / $=$ $\leftrightarrow$ variables ≤ 0 / ≥ 0 / unrestricted
- variable ≥ 0/ ≤ 0 $\leftrightarrow$ constraint $\geq$ / $\leq$

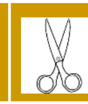
Strong duality



- 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

Application: Zero-sum game

- Two players: Row and Column

			
	0	1	-1
	-1	0	1
	1	-1	0

- 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:

  $$\min_{\{p_i\}} \max_j \Sigma_i p_i a_{ij} = \max_{\{q_j\}} \min_i \Sigma_j q_j a_{ij}$$

- 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:
  $\min_{\{p_i\}} \max_j \Sigma_i p_i a_{ij}$
  - min $\quad z$
  - s.t. $\quad \sum_i p_i a_{ij} \le z, \forall j$
    $0 \le p_i \le 1$
    $\sum_i p_i = 1$
- Column:
  $\max_{\{q_j\}} \min_i \Sigma_j q_j a_{ij}$
  - max $\quad w$
  - s.t. $\quad \sum_j q_j a_{ij} \ge w, \forall\, i$
    $0 \le q_j \le 1$
    $\sum_j q_j = 1$
- Observation: These two LP's are dual to each other.
- Thus they have the same optimal value.

Extra

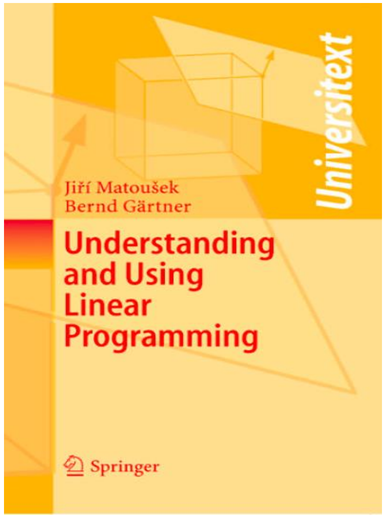
- Application in CS: Yao's principle.
  - Row: deterministic algorithms/protocols/…
  - Column: inputs
- Row/Us: design the best randomized algorithm s.t. the worst-case error is small.
- Column/Adversary: give the worst input distribution s.t. any deterministic algorithm has a big error.
- Thus to prove a lower bound for randomized algorithm complexity (on worst input), it is enough to prove a lower bound for any deterministic algorithm on a random input.

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.

References

- Our introduction to LP largely follows the book



Understanding and Using Linear Programming, Jiři Matoušek and Bernd Gärtner, *Springer*, 2006.

- Many references on LP or other optimization theories.

Convex Optimization, Boyd and Vandenberghe, *Cambridge University Press*, 2004.