# Linear Programming
# CISC5835, Algorithms for Big Data
# CIS, Fordham Univ.

Instructor: X. Zhang

## Linear Programming

- In a linear programming problem, there is a set of variables, and we want to assign real values to them so as to
  - satisfy a set of linear equations and/or linear inequalities involving these variables, and
  - maximize or minimize a given linear objective function.



## Example: profit maximization

- A boutique chocolatier has two products:
  - its flagship assortment of triangular chocolates, called Pyramide,
  - and the more decadent and deluxe Pyramide Nuit.
- How much of each should it produce to maximize profits?
  - Every box of Pyramide has a a profit of $1.
  - Every box of Nuit has a profit of $6.
  - The daily demand is limited to at most 200 boxes of Pyramide and 300 boxes of Nuit.
  - The current workforce can produce a total of at most 400 boxes of chocolate per day.
- Let $x_1$ be # of boxes of Pyramide, $x_2$ be # of boxes of Nuit



## LP formulation

Objective function: $\max\ x_1 + 6x_2$

Constraints:

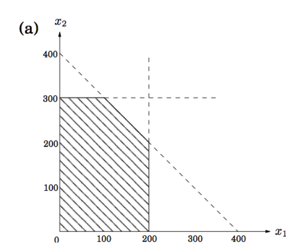
$$x_1 \le 200$$
$$x_2 \le 300$$
$$x_1 + x_2 \le 400$$
$$x_1, x_2 \ge 0$$

A linear equation of $x_1$ and $x_2$ defines a line in the two-dimensional (2D) plane

A linear inequality designates a half-space (the region on one side of the line)

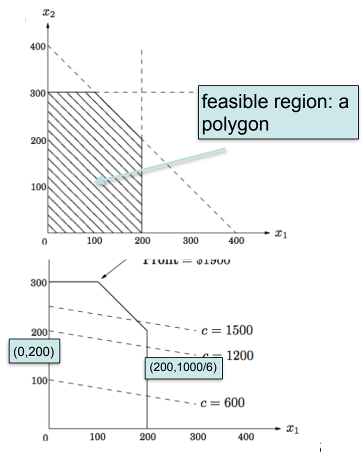
The set of all feasible solutions of this linear program, that is, the points (x1,x2) which satisfy all constraints, is the intersection of five half-spaces.

It is a convex polygon.



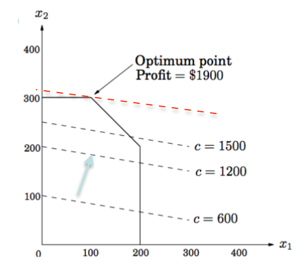
# Maximize Profit

- Find point(s) in **feasible region** (shaded part) at which objective function ($x_1+6x_2$) is maximized.
  - feasible regions decided by linear constraints
- Note: All points on line $x_1 + 6x_2 = c$ (for some constant c) achieve same profit c
  - e.g., points (0, 200), (200, 1000/6) lie on $x_1 + 6x_2 = 1200$, both yield profit \$1200
  - so are all points in the line segment




# Maximize Profit (cont'd)

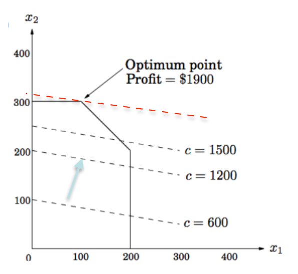
- All points that lie on line $x_1 + 6x_2 = c$ (for some constant c) achieve same profit c
- As c increases, "**profit line**" moves parallel to itself, up and to the right.
  - To maximize c: move line as far up as possible, while still touching feasible region.
- Optimum solution: very last feasible point that profit lines sees and must therefore be a vertex of polygon.






# Maximize Profit (cont'd)

- All points that lie on line $x_1 + 6x_2 = c$ (for some constant c) achieve same profit c
- As c increases, "**profit line**" moves parallel to itself, up and to the right.
  - To maximize c: move line as far up as possible, while still touching feasible region.
- Optimum solution: very last feasible point that profit lines sees and must therefore be a vertex of polygon.






# Simplex Method

**Simplex method:** devised by George Dantzig in 1947.

- Starts at a vertex, and repeatedly looks for an adjacent vertex (connected by an edge of the feasible region) of better objective value.
- In this way it does hill-climbing on vertices of the polygon, walking from neighbor to neighbor so as to steadily increase profit along the way.
- Upon reaching a vertex that has no better neighbor, simplex declares it to be optimal and halts.

## Why does this local test imply global optimality?

considering think of profit line passing through this vertex. Since all the vertex's neighbors lie below the line, the rest of the feasible polygon must also lie below this line.

# A few comments

**Simplex Method is a kind of hill climbing technique:**

- a mathematical optimization technique which belongs to the family of local search.
- It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by incrementally changing a single element of the solution.
- If the change produces a better solution, an incremental change is made to the new solution, repeating until no further improvements can be found.



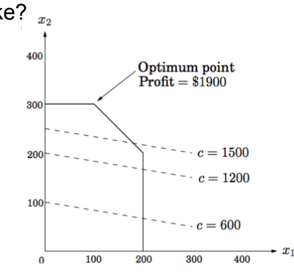
# A few comments

- **Linear programming: a special case of convex optimization.**
  - Convex optimization: minimizing **convex functions** over **convex sets.**
- **Simple ex:** What if objective function is: maximize $x_1^2+x_2^2$ ?
  - What does the "profit" lines look like?

Objective function max $x_1 + 6x_2$

Constraints

$$x_1 \le 200$$
$$x_2 \le 300$$
$$x_1 + x_2 \le 400$$
$$x_1, x_2 \ge 0$$






# Simplex Algorithm: details

- Convert the problem into standard form

In ***standard form***, we are given $n$ real numbers $c_1, c_2, \ldots, c_n$; $m$ real numbers $b_1, b_2, \ldots, b_m$; and $mn$ real numbers $a_{ij}$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. We wish to find $n$ real numbers $x_1, x_2, \ldots, x_n$ that

maximize $\displaystyle\sum_{j=1}^{n} c_j x_j$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, 2, \ldots, m$$
$$x_j \ge 0 \quad \text{for } j = 1, 2, \ldots, n\,.$$



# Simplex Algorithm: detail

- Convert standard form into slack form

maximize $\displaystyle\sum_{j=1}^{n} c_j x_j$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, 2, \ldots, m$$
$$x_j \ge 0 \quad \text{for } j = 1, 2, \ldots, n\,.$$

- slack form: (N, B, A, b, c, v)

N: set of non-basic variables (those on the r
object functions)
B: the set of basic variables
A: matrix ($a_{ij}$)
($b_i$): the vector
($c_i$): the coefficients in object function

$$z = v + \sum_{j \in N} c_j x_j$$
$$x_i = b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for } i \in B\,,$$

**Basic solution**: set all non-basic variables to 0, and calculate basic variables accordingly.

$\text{PIVOT}(N, B, A, b, c, \nu, l, e)$

// Compute the coefficients of the equation for new basic variable $x_e$.
let $\widehat{A}$ be a new $m \times n$ matrix
$\widehat{b}_e = b_l / a_{le}$
**for** each $j \in N - \{e\}$
    $\widehat{a}_{ej} = a_{lj} / a_{le}$
$\widehat{a}_{el} = 1 / a_{le}$
// Compute the coefficients of the remaining constraints.
**for** each $i \in B - \{l\}$
    $\widehat{b}_i = b_i - a_{ie}\widehat{b}_e$
    **for** each $j \in N - \{e\}$
        $\widehat{a}_{ij} = a_{ij} - a_{ie}\widehat{a}_{ej}$
    $\widehat{a}_{il} = -a_{ie}\widehat{a}_{el}$
// Compute the objective function.
$\widehat{\nu} = \nu + c_e\widehat{b}_e$
**for** each $j \in N - \{e\}$
    $\widehat{c}_j = c_j - c_e\widehat{a}_{ej}$
$\widehat{c}_l = -c_e\widehat{a}_{el}$
// Compute new sets of basic and nonbasic variables.
$\widehat{N} = N - \{e\} \cup \{l\}$
$\widehat{B} = B - \{l\} \cup \{e\}$
**return** $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{\nu})$

$\text{SIMPLEX}(A, b, c)$

1 $(N, B, A, b, c, \nu) = \text{INITIALIZE-SIMPLEX}(A, b, c)$
2 let $\Delta$ be a new vector of length $n$
3 **while** some index $j \in N$ has $c_j > 0$
4     choose an index $e \in N$ for which $c_e > 0$
5     **for** each index $i \in B$
6         **if** $a_{ie} > 0$
7             $\Delta_i = b_i / a_{ie}$
8         **else** $\Delta_i = \infty$
9     choose an index $l \in B$ that minimizes $\Delta_i$
10     **if** $\Delta_l == \infty$
11         **return** "unbounded"
12     **else** $(N, B, A, b, c, \nu) = \text{PIVOT}(N, B, A, b, c, \nu, l, e)$
13 **for** $i = 1$ **to** $n$
14     **if** $i \in B$
15         $\bar{x}_i = b_i$
16     **else** $\bar{x}_i = 0$
17 **return** $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$

# What if basic solution not feasible?

- or the problem is not feasible, or is unbounded?

$$\begin{array}{ll} \text{maximize} & 2x_1 - x_2 \\ \text{subject to} & \\ & 2x_1 - x_2 \leq 2 \\ & x_1 - 5x_2 \leq -4 \\ & x_1, x_2 \geq 0 \, . \end{array}$$



$\text{INITIALIZE-SIMPLEX}(A, b, c)$

let $k$ be the index of the minimum $b_i$
**if** $b_k \geq 0$ // is the initial basic solution feasible?
    **return** $(\{1, 2, \ldots, n\}, \{n+1, n+2, \ldots, n+m\}, A, b, c, 0)$
form $L_{\text{aux}}$ by adding $-x_0$ to the left-hand side of each constraint and setting the objective function to $-x_0$
let $(N, B, A, b, c, \nu)$ be the resulting slack form for $L_{\text{aux}}$
$l = n + k$
// $L_{\text{aux}}$ has $n + 1$ nonbasic variables and $m$ basic variables.
$(N, B, A, b, c, \nu) = \text{PIVOT}(N, B, A, b, c, \nu, l, 0)$
// The basic solution is now feasible for $L_{\text{aux}}$.
iterate the **while** loop of lines 3–12 of SIMPLEX until an optimal solution to $L_{\text{aux}}$ is found
**if** the optimal solution to $L_{\text{aux}}$ sets $\bar{x}_0$ to 0
    **if** $\bar{x}_0$ is basic
        perform one (degenerate) pivot to make it nonbasic
    from the final slack form of $L_{\text{aux}}$, remove $x_0$ from the constraints and restore the original objective function of $L$, but replace each basic variable in this objective function by the right-hand side of its associated constraint
    **return** the modified final slack form
**else return** "infeasible"

# Practice

- Consider the following linear program:
  - plot the feasible region and find optimal solution
  - What if objective is to minimize 5x+3y?

$$\textbf{maximize}\ \ 5x + 3y$$
$$5x - 2y \geq 0$$
$$x + y \leq 7$$
$$x \leq 5$$
$$x \geq 0$$
$$y \geq 0$$

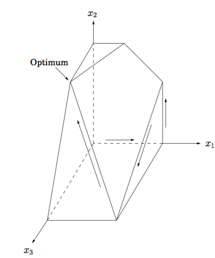


# Higher Dimension

What if there is a third and even more exclusive line of chocolates, called *Pyramide Luxe*. One box of these will bring in a profit of $13.

- Nuit and Luxe require same packaging machinery, except that Luxe uses it three times as much, which imposes another constraint $x_2 + 3x_3 \leq 600$

$$\max\ \ x_1 + 6x_2 + 13x_3$$
$$x_1 \leq 200$$
$$x_2 \leq 300$$
$$x_1 + x_2 + x_3 \leq 400$$
$$x_2 + 3x_3 \leq 600$$
$$x_1, x_2, x_3 \geq 0$$






# Another Problem

**Duckwheat is produced in Kansas and Mexico and consumed in New York and California.**

- Kansas produces 15 shnupells of buckwheat and Mexico 8.
- New York consumes 10 shnupells and California 13.
- Transportation costs per shnupell are $4 from Mexico to New York, $1 from Mexico to California, $2 from Kansas to New York, and $3 and from Kansas to California.

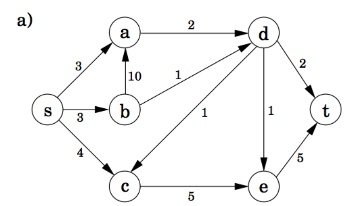
Write a linear program that decides the amounts of duckwheat (in shnupells and fractions of a shnupell) to be transported from each producer to each consumer, so as to minimize the overall transportation cost.



# Transport Networks

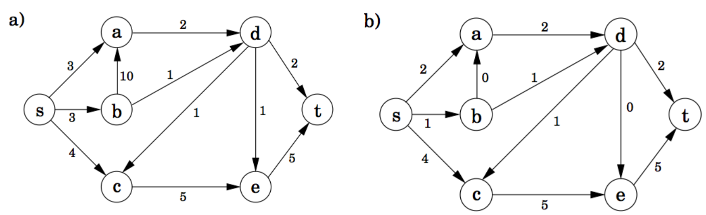
- Given a directed graph G=(V,E), two nodes s, t in V (source and sink), and capacities $c_e$ on edges
  - Model some transport system (a network of oil pipelines, computer networks, …)
  - Question: How to transport as much as goods from s to t using the network using?

**A Network**



# Flow in Networks

- A shipping scheme/plan assign $f_e$ to each edge, and has following properties
  - $0 <= f_e <= c_e$ (capacity)
  - for all nodes u except s and t, amount of flow entering u equals amount leave u (conserved)

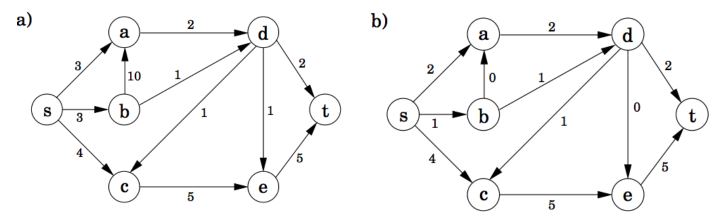



**A Network** **A flow in the network: value is 7**



# Max. Flow in Networks

- Input: G=(V,E), edge capacity $c_e$
- Output: $f_e$ of each edge (# of var = |E|)
- Linear Programming problem
  - constraints are all linear!
  - maximize: $f_{(d,t)}+f_{(e,t)}$



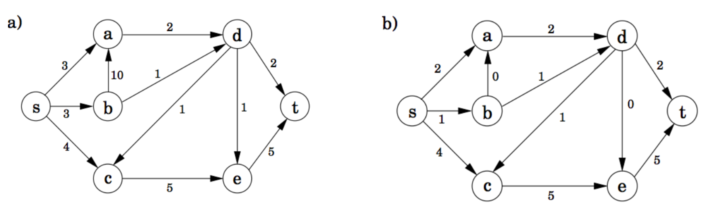

**A Network** **A flow in the network: value is 7**



# Ford-Fulkerson Alg.

- Input: G=(V,E), edge capacity $c_e$
- Output: $f_e$ of each edge (# of var = |E|)

```
Ford-Fulkerson Algorithm
The following is simple idea of Ford-Fulkerson algorithm:
1) Start with initial flow as 0.
2) While there is a augmenting path from source to sink.
           Add this path-flow to flow.
3) Return flow.
```




**A Network** **A flow in the network: value is 7**



# Summary

- Linear Programming: assign values to variables subject to linear constraints, with goal of minimizing (or maximizing) a linear function
- Many problems can be formulated as LP
- if values can only be integer, then it's a harder problem
  - e.g., Knaksack problems
- Ideas of Simplex alg.