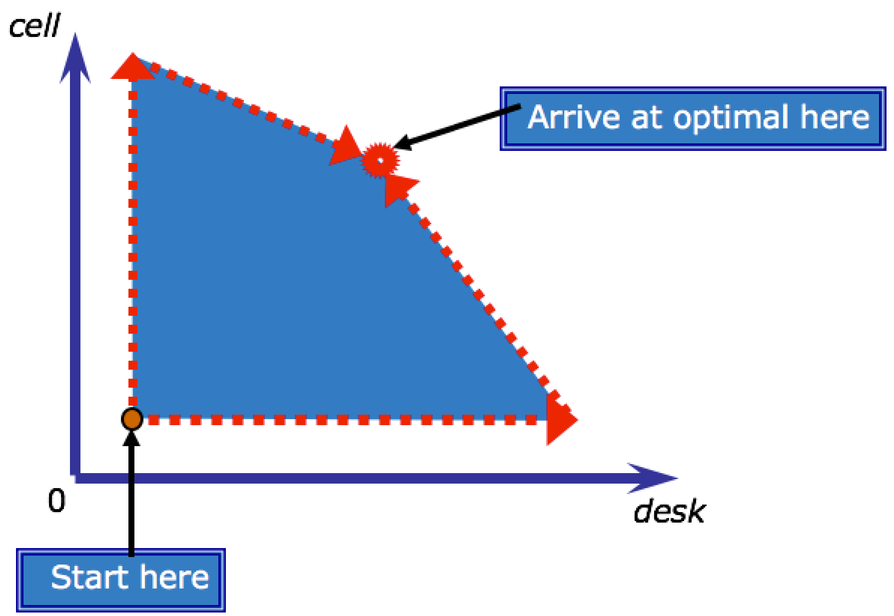


Optimization

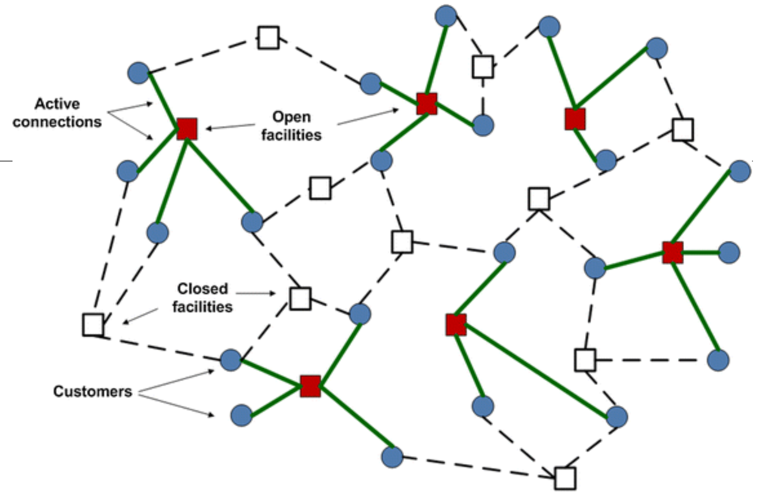
LECTURE 11

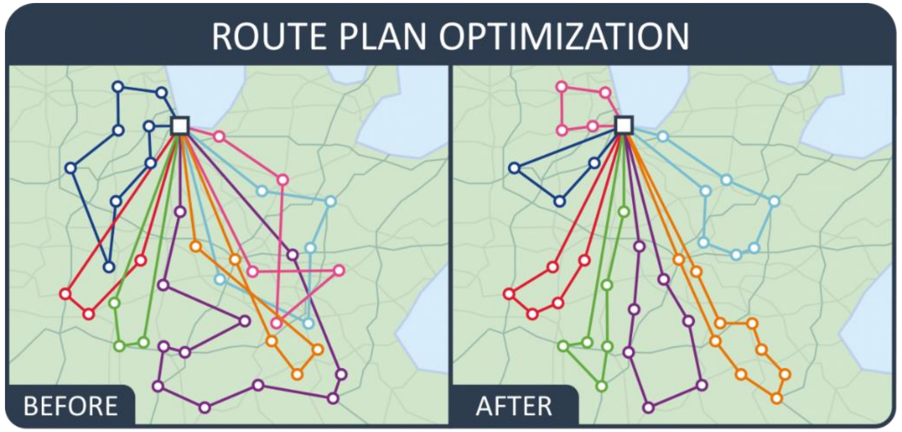
Organization of Lecture 11

- Real World Examples
- Basic Ideas of Optimization
- Types of Optimization Problems
- Solving Optimization Problems
  - Linear Programming and Examples
  - Greedy Algorithms, e.g. Coin Changing Problem
    - Do not guarantee optimal solution, but good enough
- Ant Algorithm

Real World Examples

- Dynamic and Customized Pricing
- Scheduling/Allocation
- Routing/Logistics (UPS, USPS, FedEx, etc.)
- Supply Chain
- Facility Location
- Financial Planning



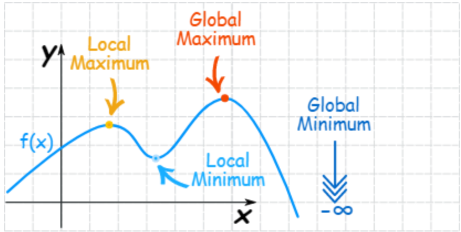


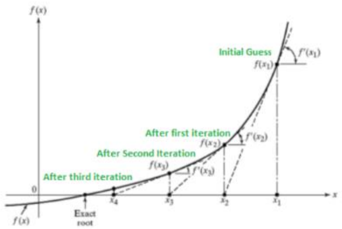
Basic Ideas

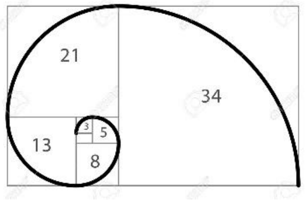
- Major field within Data Analytics, Computational Science, Operations Research and Management Science
- Problem Components include:
  - Decision Variables
  - Objective function (to maximize or minimize)
- Basic Idea
  - Find values of the decision variables the maximize or minimize the objective function value, while staying within constraints.

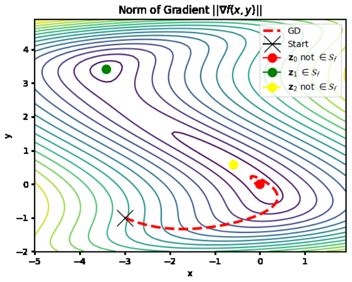
Types of Optimization Problems

- **Unconstrained Optimization Problem** $\min_x F(x)$ or $\max_x F(x)$
  - Analytical method - Differential equations
  - Newton’s Method – step-by-step
  - Golden-section search method
  - Gradient method – steepest ascent (descent method)



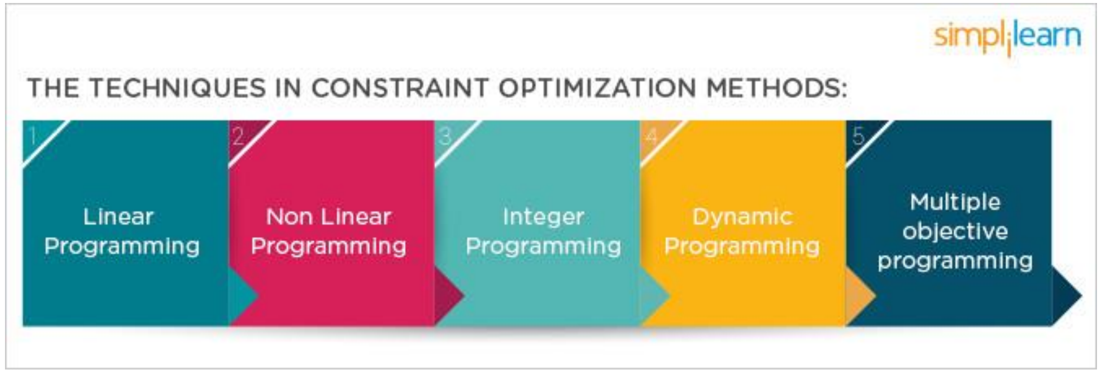


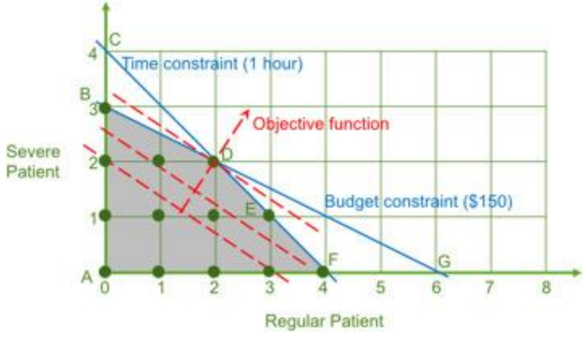


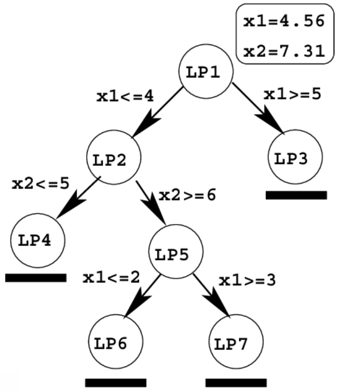


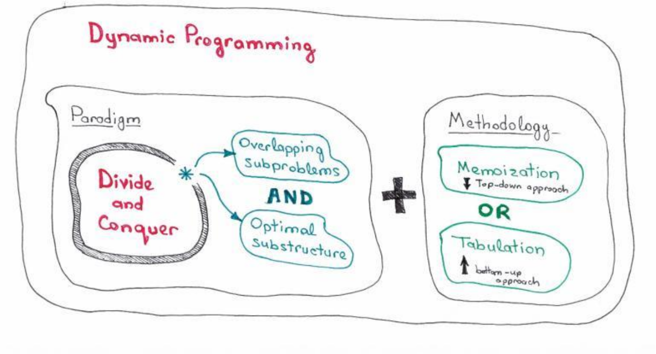
Types of Optimization Problems

- **Constrained Optimization Problem** $\min_x F(x)$ or $\max_x F(x)$ subject to $g(x) = 0$ and/or $h(x) < 0$ or $h(x) > 0$







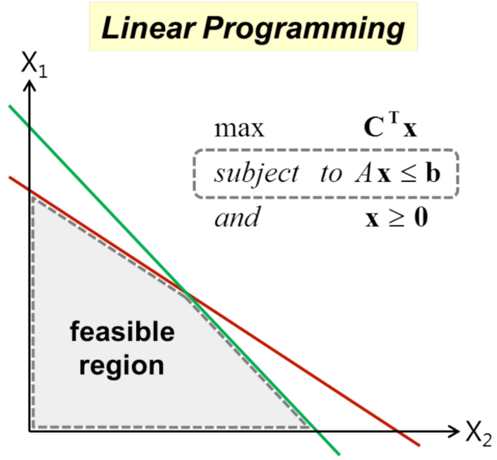


Solving Optimization Problems

- Understand the problem
  - Draw a diagram
  - Write a problem formulation in words
- Write the algebraic formulation of the problem
  - Define the decision variables
  - Write the objective function in terms of those variables
  - Write the constraints in terms of those variables
- Write a computer program or use a tool, eg. Spreadsheet
- Examine results, make corrections to the model
- Interpret the results and draw insights

Linear Programming (LP)

- If objective function and all constraints are linear functions of decision variables – then it is an LP problem.
  - Easier to solve than non-linear functions
  - Real world LPs are solved which contain hundreds to millions of variables (*with specialized hardware).*



Sample Problem

- Karla has a garden store and she makes two kinds of planting mixtures.
  - gardening mixture – takes 1lb of fertilizer and 3 lbs of soil
  - potting mixture – takes 2 lbs of peat moss and 2 lbs of soil
- The garden mixture sells for $3 and the potting mixture sells for $5.
- Karla has at most 4lbs of fertilizer, 12 lbs of peat moss and 18lbs of soil.
- Maximize revenue Z = 3 x_1 + $5x_2$
- Subject to constraints on soil, peat moss, and fertilizer

$1x_1 \leq 4$	*fertilizer*
$2x_2 \leq 12$	*peat moss*
$3x_1 + 2x_2 \leq 18$	*soil*
where $x_1 \geq 0, x_2 \geq 0$	all mixtures sold are non-negative

Python function linprog()

- From sypy.optimize

```
scipy.optimize.linprog(c, A_ub=None, b_ub=None, A_eq=None, b_eq=None,
  bounds=None, method='simplex’,  callback=None,
  options={'maxiter': 5000, 'disp’:  False, 'presolve': True, 'tol': 1e-12,
  'autoscale': False, 'rr': True, 'bland': False}, x0=None)
```

- Parameters
  - C – coefficients for linear objective function to be *minimized*
  - A_ub and b_ub are inequality constraints, A_e1 and b_eq are equality constraints

- Returns
  - x: 1D array containing values of decision variables optimizing objective function
  - fun: optimal value of objective function c @x
  - slack: nominally positive values of slack variables used to turn constraints into equalities
  - status: represents exit status of algorithm see in message
  - nit: the number of iterations performed

Sample Problem

- Original Form:
  - Maximize $Z = 3x_1 + 5x_2$
  - Subject to

$$x_1 \leq 4$$
$$2x_2 \leq 12$$
$$3x_1 + 2x_2 \leq 18$$
$$\text{where } x_1 \geq 0, x_2 \geq 0$$

- Revised form:
  - Maximize $Z = cx$
  - Subject to $Ax \leq b,\ x \geq 0$

- Vector/Matrix Form:
  - C = [-3, -5] (because we are maximizing, we negate the values)
  - A = $\begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{bmatrix}$
  - b = [4, 12, 18]

- Python Form:
  - c = [-3, -5] #coefficients
  - A = [[1,0],[0,2], [3,2]] #Inequality constraint matrix
  - b = [4,12, 18] #inequality constraint vector

Solution

```
from scipy.optimize import linprog

#initialize matrices
c = [-3, -5]
A = [[1,0],[0,2], [3,2]]
b = [4,12,18]

#configure upper and lower bounds
x0_bounds = (0, None)
x1_bounds = (0, None)

#solve with simplex algorithm, display each iteration
result = linprog(c, A_ub=A, b_ub=b, bounds = (x0_bounds, x1_bounds),\
method = 'simplex', options={'disp': True})
print ("Result: ", result)
```

```
Optimization terminated successfully.
         Current function value: -36.000000
         Iterations: 3
Result:       con: array([], dtype=float64)
     fun: -36.0
 message: 'Optimization terminated successfully.'
     nit: 3
   slack: array([2., 0., 0.])
  status: 0
 success: True
       x: array([2., 6.])
>
```

Simplex Method Explained

- Simplex method is **an approach to solving linear programming models by hand using** slack variables, tableaus, and pivot variables as a means to finding the optimal solution of an optimization problem.
- Sample Problem Solved (8 minutes)

Your Turn

- A store sells two types of toys, A and B. The store owner pays $8 and $14 for each unit of toy A and B, respectively. Each to A yields a profit of $2 while each toy B yields a profit of $3. The store owner estimates that no more than 2000 toys will be sold every month and he does not plan to invest more than $20,000 in inventory of these toys. How many units of each type of toys should be stocked in order to maximize his monthly total profit?

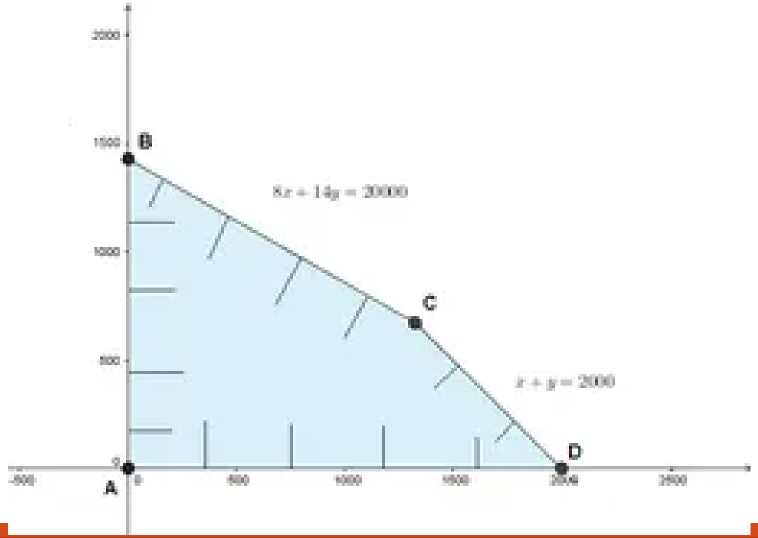
Solution

- Original Form:
  - Maximize $P = 2x_1 + 3x_2$
  - Subject to

    $x_1 + x_2 \leq 2000$

    $8x_1 + 14x_2 \leq 20{,}000$

    where $x_1 \geq 0, x_2 \geq 0$



- Python Form:
  - c = [-3, -5] #coefficients
  - A = [[1,1],[8,14]] #Inequality constraint matrix
  - b = [2000, 20000] #inequality constraint vector

- The maximum profit is at vertex C with x = 1333 and y = 667.
- Hence the store owner has to have 1333 toys of type A and 667 toys of type B in order to maximize his profit.

More Examples

Insect Behavior for Optimization

- Ant Foraging
- Honey bee foraging and Bee hive selection
- Honey bee mate selection – fitness of drone in flight dance, etc.
- Bacterial foraging –waiting for critical mass to emit toxins
- Glow worm search for optimal location to attract mates and avoid predators
- Fish shoaling behavior – for survivability of species without depleting resources

Ant Foraging

- Insects are capable of complex behaviors – sensory inputs modulate their behavior according to stimuli
- Complexity of individual in not sufficient to explain what the societies of insects can achieve
- Only 2% of insects are social, but these comprise 50% of insect biomass
- Suggests social nature of these species contributes to success

Summary of Optimization Methods Seen

- You have been “exposed” to these methods for optimization
  - Greedy algorithms: coin changing problem, Dijkstra’s shortest path and Prim’s and Kruskal’s algorithm for minimum spanning tree
  - Least Squares – minimizes residuals (errors or distance from line)
  - Simulation and stochastic processes – allow you to perform “what-if” analysis on random samples of data
  - Genetic Algorithms – evolutionary algorithms to evolve optimal solution
  - Linear Programming
  - Algorithm inspired by Insect Behavior