Monday 7/20

Applied Data Science

Optimization 1

Constraint Satisfaction Problems (CSP)
The Problem of Optimization
Introduction to genetic algorithms

Motivating Problem (chapter 5 PCI)

Demo

- A group of people plan to make a round-trip to NYC from their respective home towns. Determine a group flight schedule that minimizes a cost function. A constraint is that all members of the group need to rendezvous at the LGA airport. This means that once a person arrives at LGA he/ she needs to wait for the last person to arrive. Similarly, on their return journey all people need to get the airport together (which implies that they all need to get there in time for the person with the earliest flight).

Cost

- Various costs that are incurred are: The cost of the tickets
- The wait time both on arrival at LGA and on departure from LGA
- The time for travel

Constraint Satisfaction Problems (CSP)

Constraint Satisfaction Problems (CSP)

- What we are given:
  - a set of variables
  - each variable with a domain of values
  - a set of constraints that limit the values the variables can take on
- What we seek:
  - a value assignment to each variable such that all constraints are satisfied
- Variations
  - does a solution exist?
  - find some solution
  - find all solutions
  - find the "best" solution (by some metric)

A well known puzzle

Two men meet on the street (they have not seen each other for many years):

A: All three of my sons celebrate their birthday today. Can you tell me how old each one is?
B: Yes, but you have to tell me something about them…

A: The product of their ages is 36.
B: I need more info…

A: The sum of their ages is equal to the number of windows in the building next to us…
B: I need more info…

A: My oldest son has blue eyes.
B: That is sufficient!

Puzzle 5.1 (contd)

- The product of their ages is 36:

x	y	z
36	1	1
18	2	1
12	3	1
9	4	1
9	2	2
6	6	1
6	3	2
4	3	3

A: The product of their ages is 36.
B: I need more info…

A: The sum of their ages is equal to the number of windows in the building next to us…
B: I need more info…

A: My oldest son has blue eyes.
B: That is sufficient!

Puzzle 5.1 (contd)

- The sum of their ages is equal to the number of windows in the building next to us…

x		y		z		
36	+	1	+	1	=	38
18	+	2	+	1	=	21
12	+	3	+	1	=	16
9	+	4	+	1	=	14
9	+	2	+	2	=	13
6	+	6	+	1	=	13
6	+	3	+	2	=	11
4	+	3	+	3	=	10

A: The product of their ages is 36.
B: I need more info…

A: The sum of their ages is equal to the number of windows in the building next to us…
B: I need more info…

A: My oldest son has blue eyes.
B: That is sufficient!

Puzzle 5.1 (contd)

- The sum of their ages is equal to the number of windows in the building next to us…

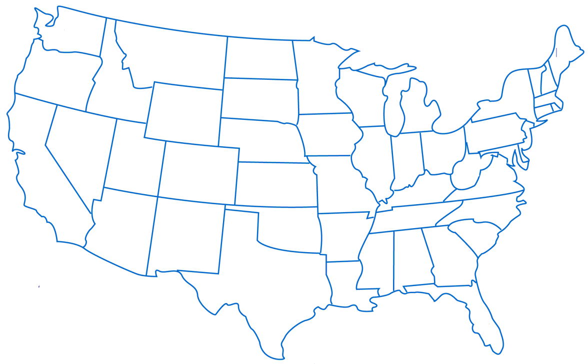
x		y		z		
36	+	1	+	1	=	38
18	+	2	+	1	=	21
12	+	3	+	1	=	16
9	+	4	+	1	=	14
9	+	2	+	2	=	13
6	+	6	+	1	=	13
6	+	3	+	2	=	11
4	+	3	+	3	=	10

A: The product of their ages is 36.
B: I need more info…

A: The sum of their ages is equal to the number of windows in the building next to us…
B: I need more info…

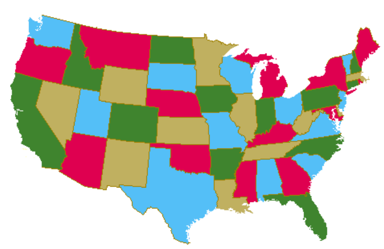
A: My oldest son has blue eyes.
B: That is sufficient!

How many colors are needed to color this map?



A Famous CSP: 4 color problem

Any plane separated into regions (such as a political map of the states of a country), may be colored using no more than four colors in such a way that no two adjacent regions receive the same color.



Real-world CSPs

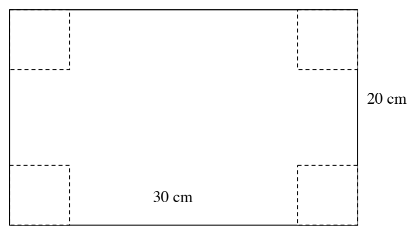
- Scheduling
- Temporal reasoning
- Building design
- Planning
- Optimization/satisfaction
- Vision
- Graph layout
- Network management
- Natural language processing
- Molecular biology / genomics
- VLSI design

Constraint Satisfaction Problems vs. Optimization Problems

- Constrain Satisfaction
  - number of variables
  - each variable defined on a domain of possible values
  - number of constraints
  - find a solution satisfying all the constraints
- Optimization
  - number of variables
  - each variable defined on a domain of possible values
  - number of constraints
  - find the best solution satisfying all the constraints
  - quality measure
    or
    evaluation function

Traditional Optimization

From a piece of rectangular paper 20×30 centimeters, cut corners in such a way that after bending the sides of the paper we get a box of the largest possible volume:



Calculus to the aid

$V = (20 - 2x)(30 - 2x)x =$

$= 4x^3 - 100x^2 + 600x$

This is example of a non-linear problem – *nonlinear programming* techniques must be used.

Solution: Take the derivative

$12x^2 - 200x + 600$
$x = 200 \pm \text{sqrt}(200^2 - 4 \cdot 12 \cdot 600) / (2 \cdot 12)$

Solution: $x \approx 3.92$ cm, and $V \approx 1056.31$ cm^3

Bridge Crossing

Four people (say A, B, C, D) have to cross a narrow bridge at night. Due to darkness they need to use a flashlight to cross. The bridge can only hold two people at a time. Each person travels at different speeds:

Person	A	B	C	D
Time to cross the bridge in mins	1	2	5	8

When two people are simultaneously crossing the bridge they move at the slower person's pace (e.g., B and D traveling together would take 8 mins to cross). Show that all of the people can cross from one side to the other side of the bridge in ? minutes. Keep in mind that anytime a person or a pair crosses the bridge they need to use the flashlight and that the flashlight can not be thrown from one side to the other (i.e., somebody has to take the flashlight back).

Bridge Crossing

Four travellers approached a bridge…

A(1)
B(2)
C(5)
D(8)

An "obvious" solution:
A is the fastest so have A take every one over

ABCD			0
CD		AB	2 (+2)
A CD		B	3 (+1)
D		AC B	8 (+5)
AD		BC	9 (+1)
		AD BC	17 (+8)

Bridge Crossing

Four travellers approached a bridge…

A(1)
B(2)
C(5)
D(8)

Insight: C and D take the most time; hence minimize their travel cost

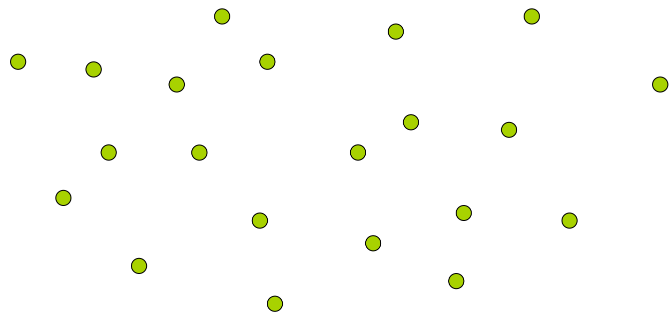
ABCD			0
CD		AB	2 (+2)
A CD		B	3 (+1)
A		CD B	11 (+8)
AB		CD	13 (+2)
		AB CD	15 (+2)

General Structure of a Solution

X & **X** ↑
X ↓
X & **X** ↑
X ↓
X & **X** ↑
X ↓
X & **X** ↑
X ↓
X & **X** ↑

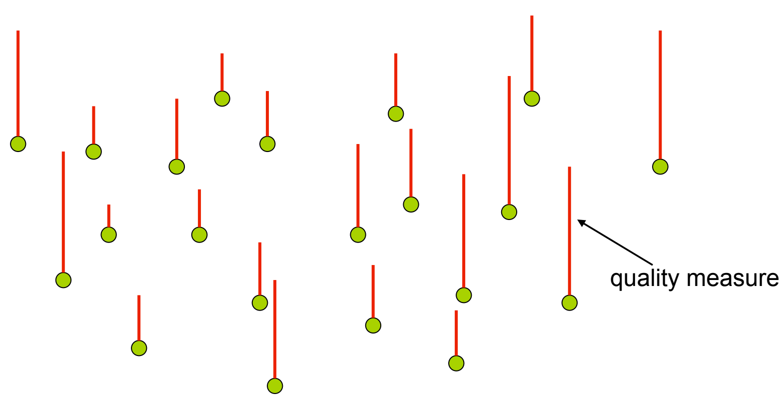
Optimization

x & x ↑
x ↓
x & x ↑
x ↓
x & x ↑
x ↓
x & x ↑
x ↓
x & x ↑



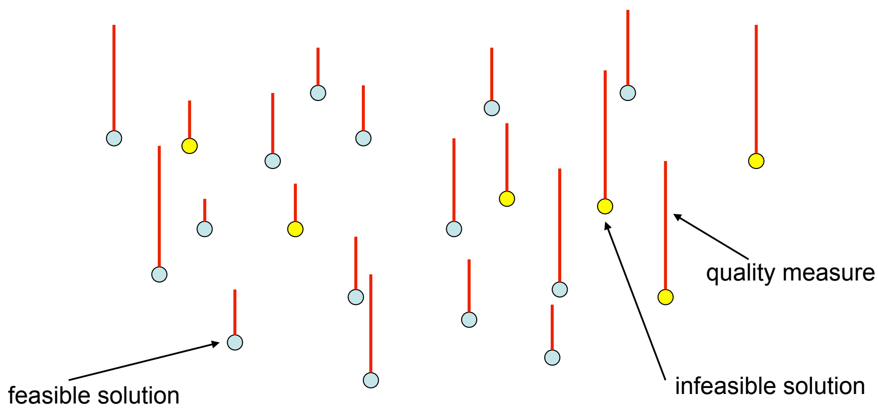
all solutions
(search space)

Optimization



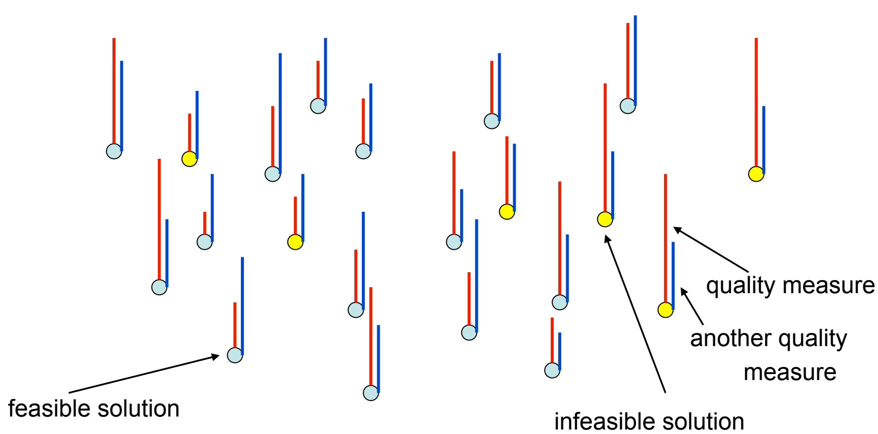
all solutions
(search space)

Optimization



Optimization task: Find the highest quality feasible solution…

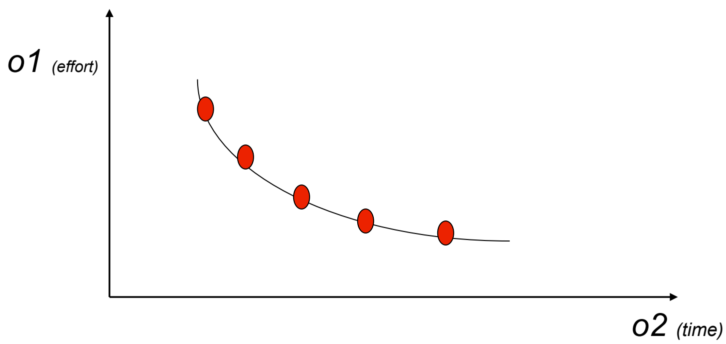
Optimization



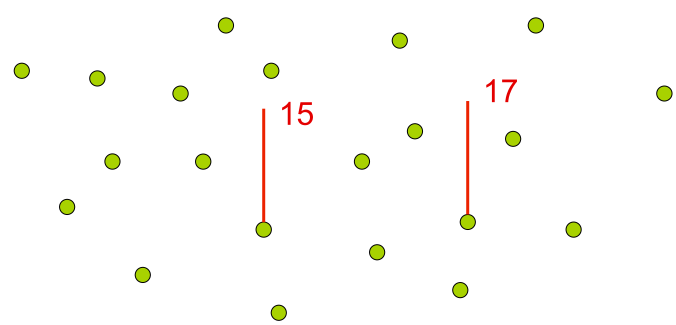
Optimization task: Find the highest quality feasible solution…

Multi-Objective Optimization

For more than one objective, often the best we can do is to explore the tradeoffs between various solutions…



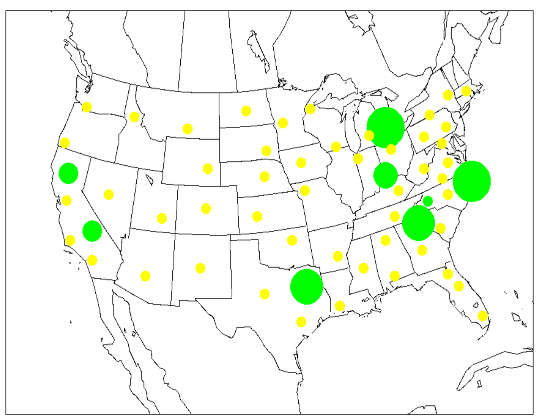
What is the optimal solution?



Case Study:
Optimal distribution of auction cars

A Sample Problem: Car Distribution

- A car company receives thousands (2000-7000) of leased cars back each day.
- Green (darker) circles show where and how many cars were returned from their leases at so-called "ramps".
- Yellow (brighter) circles represent stationary auction sites where the cars need to be sold.
- Each day this leasing company must make decisions where to send thousands of cars.



Sample decision

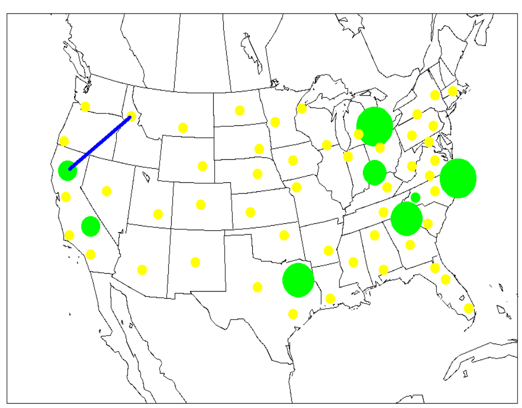
Car	1	2	3	4	…	2999	3000
Site	23	41	5	23		19	41

Nature of Complex Business problems

- Many possible solutions.
- Multi-criteria objectives
- Time changing environment
- Heavily constrained
- Incomplete / noisy data

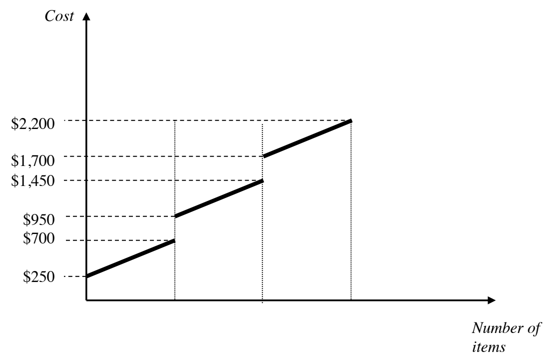
A Greedy Approach?

- We could attempt to assign one car at a time to an auction site where we expect the highest sales value.
- This would end in a complete disaster and the leasing company would loose Millions of USD.
- Let us look at the inherent issues.



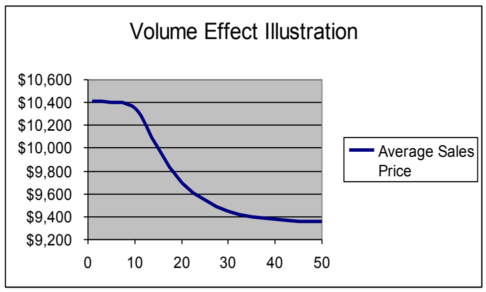
Transportation

- Transportation cost is not linear (and may look like illustrated below).
  - This is due to the fact that cars are transported on trucks that can carry 10-15 cars. Must be taken into account such that we maximize whole truckloads of cars.
  - Further, some routes from ramps to auction sites are not feasible.



Volume Effect

- If you sent one white Ford Mustang to an auction the buyers would bid and this would result in a higher price.
- If you sent hundreds of white Ford Mustangs to the same auction the buyers would not bid since there are so many of them. This would result in a lower price.
- The volume effect for a particular auction could look like illustrated to the right.
- Hence, we can not just consider one car at a time or we would loose money.



Price Depreciation

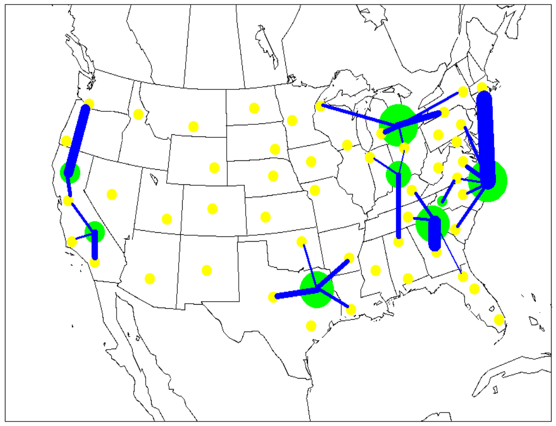
- A car looses about $10 USD in value *each day* while it remains not sold.
- The price depreciation is based on the fact that if the car was sold quickly the money could be invested and earn interests.
- Hence, we would like to sell the cars as quickly as possible, and the transportation time (and distance) should be minimized.
- Example:
  - We might have 1,000 cars at a particular auction site, but only 250 of our cars are sold (on average) at each sale.
  - It will take approximately four auction sales to sell our current inventory.
  - Hence, if we ship an additional 100 cars to this particular auction, they would first be sold some two months later!
  - This would result in a huge price depreciation.

Dynamic Market Changes

- Car prices change all the time and in different ways across the country.
- There are many issues:
  - *Regions* change in different ways and this must be tracked properly.
  - *User preferences* change, e.g., large SUV' s fall in price.
  - *Seasonality* effects were prices of certain types of cars goes up and down, e.g., prices of convertibles in Detroit in the winter versus the summer.
    - Seasonality can of course be exploited by sending convertibles to Detroit in February *if* it takes 2 months for the cars to get to Detroit and by April the prices of convertibles go up in Detroit.
  - *Sudden changes* like peaks in oil prices, stock market fluctuations or error prone Bridgestone/Firestone tires that had to be recalled (3.85 M tires!).
  - *Trend detection* must be done properly, e.g., one car sold at higher than expected price does not make a trend, but ten cars with higher price might represent a new trend if usually ~500 cars are sold at a particular auction.
  - New models are introduced to the market each year.

The Desired Solution

- The problem is far from trivial, but a decision has to be made each day where to send thousands of cars. The following illustrates a solution to the problem.



Car	1	2	3	4	…	2999	3000
Site	23	41	5	23		19	41

Impact of ABI

- Since the problem is so complex, every leasing company has a dedicated team for assigning off-lease cars to auctions sites.
- These teams face the formidable task of recommending the best possible solution for each daily load of cars.
- A small mistake with an inferior recommendation that results in a net loss of "only" \$150 per car, may cost the company hundreds of thousands of dollars in a single day!
- If an ABI system was used to improve the daily distribution of cars by \$200 per car then the increased annual profits would translate into *hundreds of millions of dollars*!
  - Note: \$200 corresponds to an increase of only 1.33% in the price of an average car.
- Hence, to maximize the overall net profit, an ABI system must decide where to ship each car.

Requirements of an Optimization Technique

- a representation for the solution
- an objective to be maximized
- an evaluation function measuring the goodness of a solution

Alternative Representations for the Car Distribution

Car	1	2	3	4	…	2999	3000
Site	23	41	5	23		19	41

sites	linked list of cars assigned to site				
1	2340	902	1198	87	2949
2	781				
…					
49					
50	1007	1459	2541		

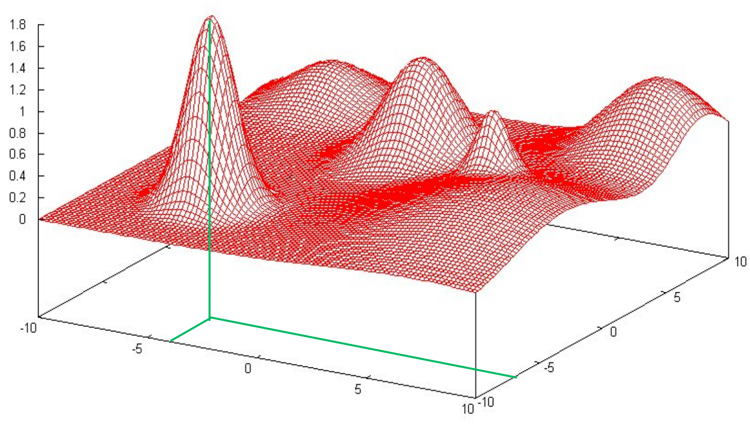
Car	1	2	3	4	…	2999	3000
Closest Site	1	2	1	3		1	5

Evaluation Function

- Usually:
  - Evaluation_function(S) = Objective(S) – Penalties(S)
    where S is a solution.
  - The evaluation function describes the *quality* of the solution.

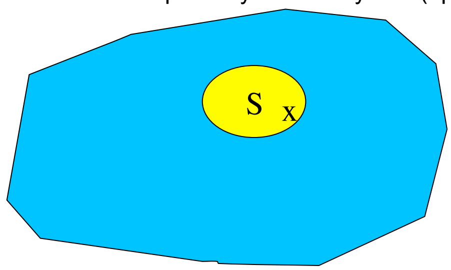
- Example:
  - The car example has many requirements: Maximize the sale price per car, minimize transportation distance, sell each car as soon as possible, etc.
  - We could formulate the evaluation function as:
    Evaluation_function(S) = $money
    *– a*total_distance*
    *– b*total_no_days_to_sell …*
  - This common approach uses penalty functions to reduce undesired effects while maximizing the objective. However, you need to tune the constants *a*, *b*, …

Fitness Landscape defined by eval. fn.



Concept of "neighborhood" or "moveset"

- *Neighborhood* of a solution S.
- Examples: Change auction site of *one* car at a time, *or*
  Change auction sites of possibly *all* cars by one (up or down)



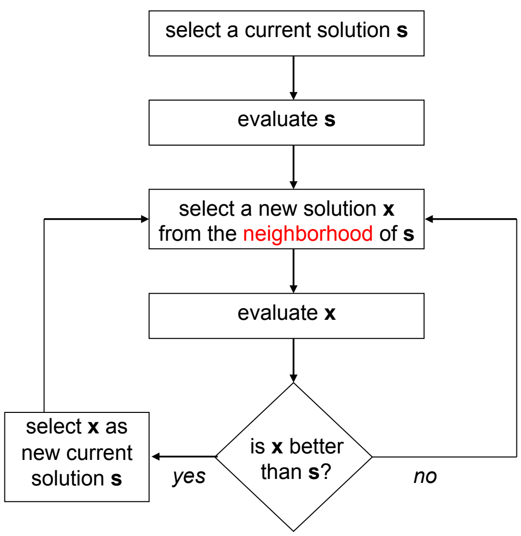
1 2 3 3000

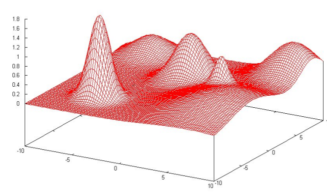
s = (23, 7, 45, 7, … , 19)

↓

x = (22, 7, 45, 8, … , 19)

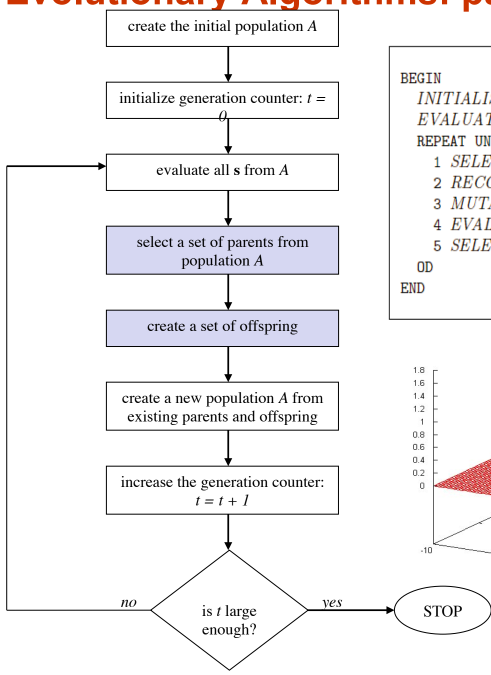
Hill Climber



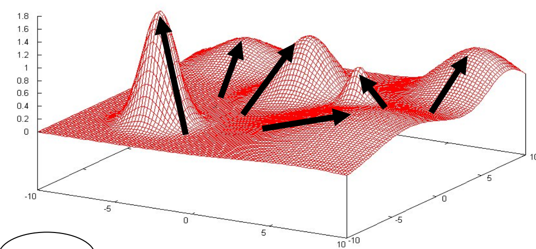


- *What are the weaknesses of this algorithm? What are the strengths?*
- *Does it help to repeat the above process many times?*

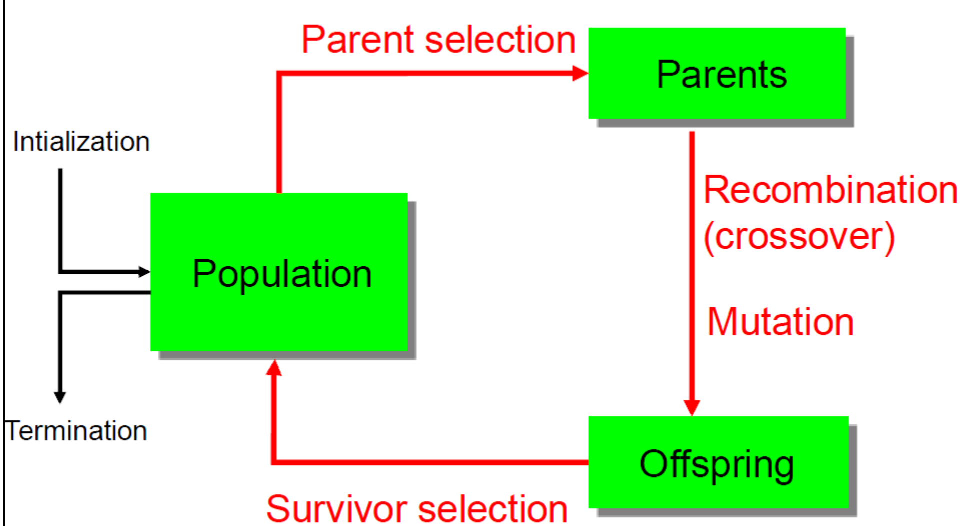
Evolutionary Algorithms: parallel hillclimbing



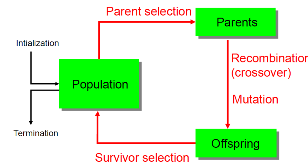

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```



The Evolutionary Cycle

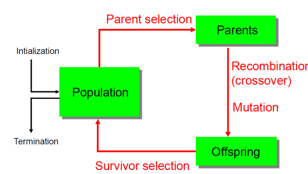


Survivor Selection



- a.k.a. ***replacement***
- Most EAs use fixed population size so need a way of going from (parents + offspring) to next generation
- Often deterministic
  – Fitness based : e.g., rank parents+offspring and take best
  – Age based: make as many offspring as parents and delete all parents
- Sometimes do combination (elitism)

Initialization / Termination



- Initialization usually done at random,
  – Need to ensure even spread and mixture of possible allele values
  – Can include existing solutions, or use problem-specific heuristics, to “seed” the population
- Termination condition checked every generation
  – Reaching some (known/hoped for) fitness
  – Reaching some maximum allowed number of generations
  – Reaching some minimum level of diversity
  – Reaching some specified number of generations without fitness improvement

Motivating Problem (chapter 5 PCI)

Demo

- A group of people plan to make a round-trip to NYC from their respective home towns. Determine a group flight schedule that minimizes a cost function. A constraint is that all members of the group need to rendezvous at the LGA airport. This means that once a person arrives at LGA he/ she needs to wait for the last person to arrive. Similarly, on their return journey all people need to get the airport together (which implies that they all need to get their in time for the person with the earliest flight).

Cost

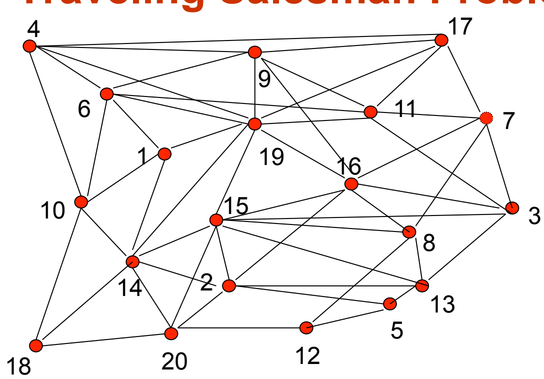
- Various costs that are incurred are: The cost of the tickets
- The wait time both on arrival at LGA and on departure from LGA
- The time for travel

Self-study Questions

- When does the 2nd flight from Dallas for LaGuardia leave?
- When does it arrive at LaGuardia?
- How much does it cost?

Problems vs. Solutions

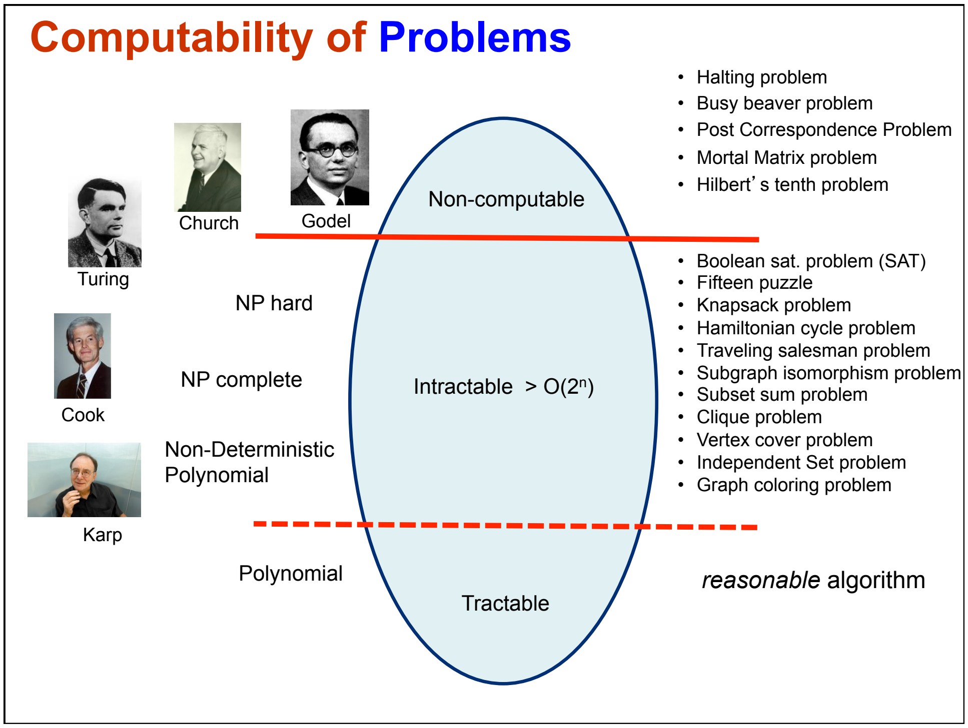
Traveling Salesman Problem

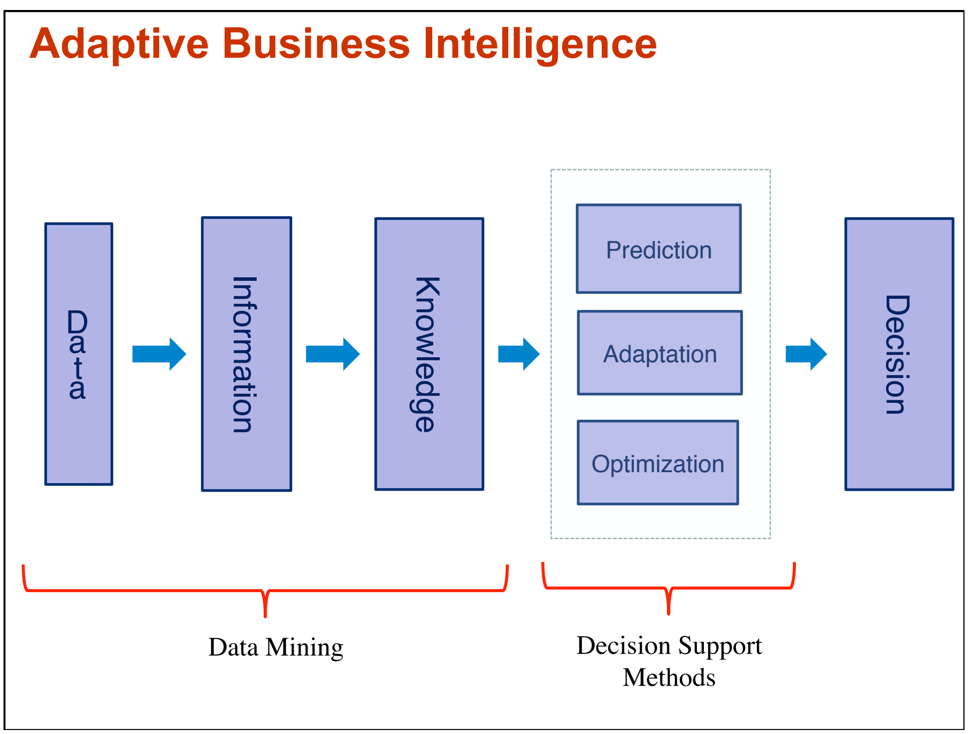


In general, the number of possible solutions is:

$$\frac{(n-1)!}{2}$$

n	# of possible solutions
5	12
10	181,440
20	60822550204416000 $\approx 10^{16}$
50	3041409320171337804361260816606476884437764156896051200000000000 $\approx 10^{62}$

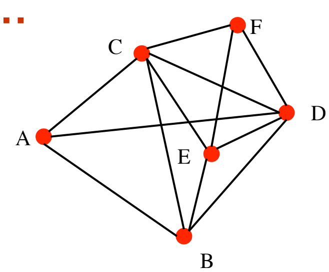

Computability of Problems
Church
Godel
Turing
Cook
Karp
Non-computable
NP hard
NP complete
Non-Deterministic Polynomial
Intractable > O(2ⁿ)
Polynomial
Tractable
Halting problem
Busy beaver problem
Post Correspondence Problem
Mortal Matrix problem
Hilbert's tenth problem
Boolean sat. problem (SAT)
Fifteen puzzle
Knapsack problem
Hamiltonian cycle problem
Traveling salesman problem
Subgraph isomorphism problem
Subset sum problem
Clique problem
Vertex cover problem
Independent Set problem
Graph coloring problem
reasonable algorithm


Adaptive Business Intelligence
Data
Information
Knowledge
Prediction
Adaptation
Optimization
Decision
Data Mining
Decision Support Methods

Requirements of an Optimization Technique

- a representation for the **solution**
- an objective to be maximized
- an evaluation function measuring the goodness of a **solution**

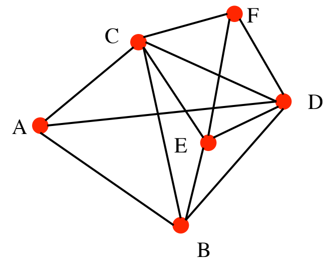
Proper encoding can help …



- Consider the following two tours:
  - BDAFCE and EACBFD
  - if we perform a cross-over operation at position 5 we get
  - BDAFFD and EACBCE
  - these are not valid tours!

Representation: Tour in the TSP

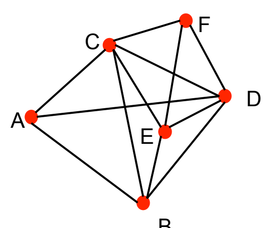
- List of cities
  - B D A F C E
- Offset from a standard list
  - Setup a "standard ordering"
    - ABCDEF (0 based indexing)
    - 012345
  - A sequence like BDAFCE has the code: 120200
    - note the position of a letter in the standard ordering and then remove it
      - B is 1 in ABCDEF
      - D is 2 in ACDEF
      - A is 0 in ACEF
      - F is 2 in CEF
      - C is 0 in CE
      - E is 0 in E



Exercise: Encode EACBFD

Proper encoding can help create valid off-spring

What interesting advantage does this encoding offer?



Consider the two tours: BDAF|CE EACB|FD if we perform a cross-over operation at position 5 we get BDAFFD EACBCE these are not valid tours!

If we use the encoded sequences, we get ABCDEF 012345

BDAFCE 1202|00
EACBFD 4010|10

if we perform a cross-over operation at position 5 we get 120210 401000 which decodes to BDAFEC EACBDF

Demo PCI Chapter 5 code