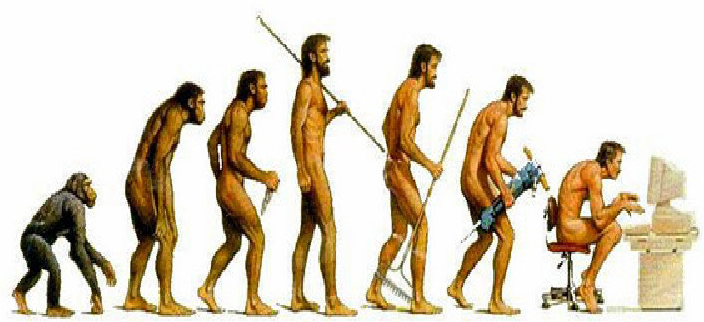
CSE4403 3.0/CSE6602E - Soft Computing
Winter 2011



Lecture 8

Evolutionary Computing: What

Guest lecturer: Xiangdong An
xan@cs.yorku.ca

A bigger picture

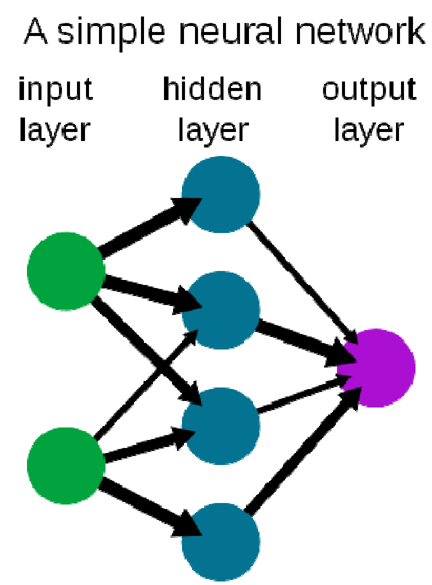
- Evolutionary computing is soft computing
- It is also
  - Natural computing
  - Optimization search
  - Heuristics
    - Local search

Evolutionary computing is natural computing

- Nature's solutions have always been a source of inspiration
  - Natural problem solvers
- DNA (molecular) computing
  - DNA as data structures
    - Molecular level, economical information storage (10^{12} more efficient)
  - Computation by manipulating DNA (extension, cutting, joining)
    - Efficient energy usage (10^{10} times less), massive parallel (comp. power)
    - Solving NP-complete problems in linear time (by exhaustive search)
- Quantum computing
  - Quantum bits as data structures (0, 1 or a superposition of them)
    - Atomic scale information storage
  - Quantum parallelism
    - Solving NP-complete problems in linear time

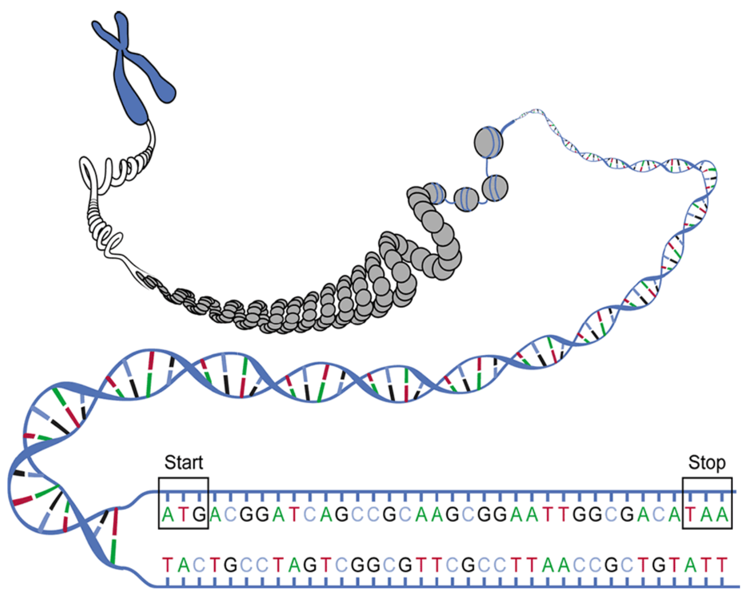
Evolutionary computing is natural computing

- Neurocomputing
  - Human brain (created wheel, Toronto, wars, etc.)
  - Biological neural networks
    - Central nervous systems (brain and spinal cord)
    - Peripheral nervous systems
      - Connect CNS to limbs and organs
    - Neurons connected by axons
  - Artificial neural networks

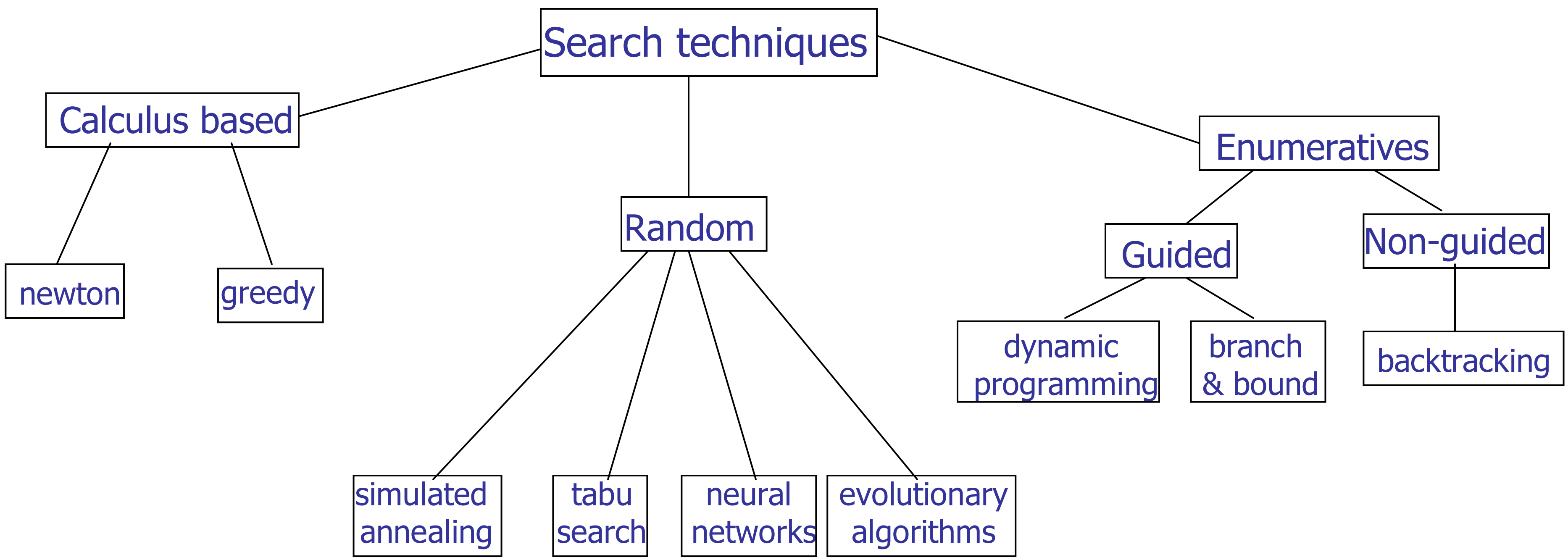


Evolutionary computing is natural computing

- Evolutionary computing
  - Evolutionary process (created the human brain)
    - Genes recombination and mutation, propagation
  - Automated problem solvers

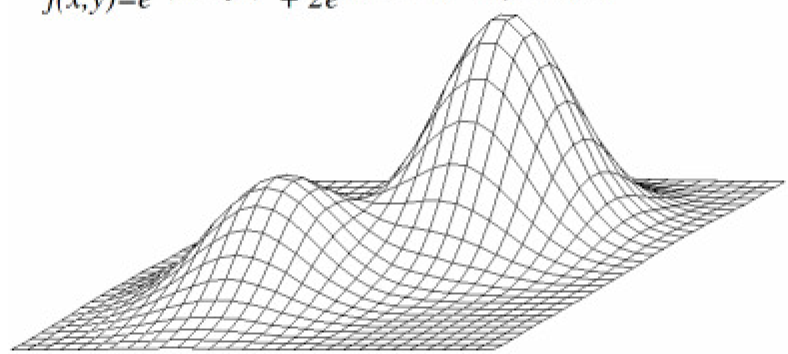


Evolutionary computing is optimal search



Evolutionary computing is optimal search

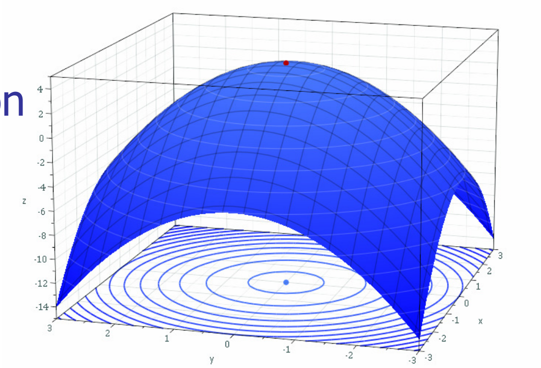
$$f(x,y)=e^{-(x^2+y^2)}+2e^{-((x-1.7)^2+(y-1.7)^2)}$$



- Enumerative search
  - Brute force search
  - Backtracking
  - Branch and bound
  - Dynamic programming
- Calculus-directed search
  - Gradient descent (steepest descent, hill climbing, greedy)
- Random (local) search
  - Simulated annealing
  - Tabu search
  - Neural networks
  - Evolutionary algorithms
    - Starts from a population of solutions

Simulated annealing, Tabu search, Neural networks: Starts from one solution

Problem of local optimal (Calculus-directed search and Random (local) search)

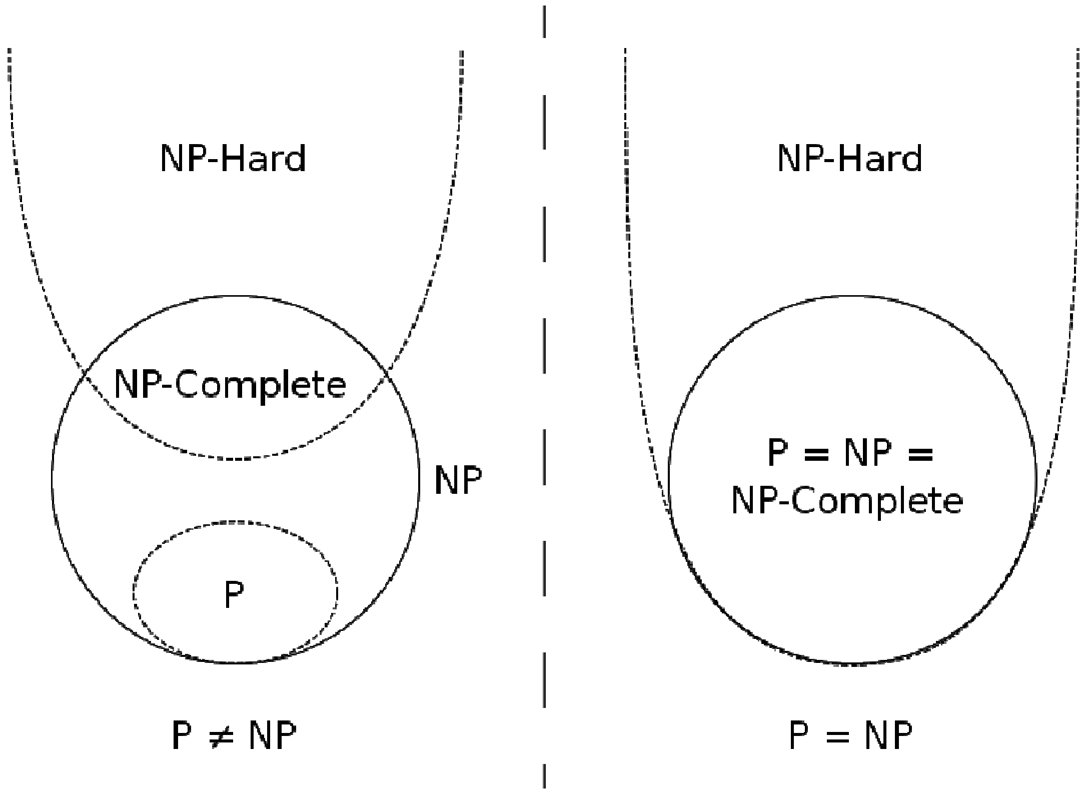


Everybody's life is a local search

What problems need evolutionary computing

- Heruistics are to find a good enough solution where an exhaustive search is impractical
  - P problems
    - Solvable in polynomial time by a deterministic Turing machine
    - Efficiently solvable, tractable, feasible
  - NP problems
    - Non-deterministic polynomial time
  - NP-complete problems
    - Hardest of NP
    - Every problem in NP can be reduced to a problem in NP-complete
    - No polynomial solution by deterministic Turing machine is known
      - Currently intractable, infeasible
    - If any polynomial solution is found, then P = NP
  - NP-hard problems
    - At least as hard as NP-complete problems

What problems need evolutionary computing



A lot of combinatorial problems are NP-hard or NP-complete: a problem can easily take hundreds or millions of years given current computing power

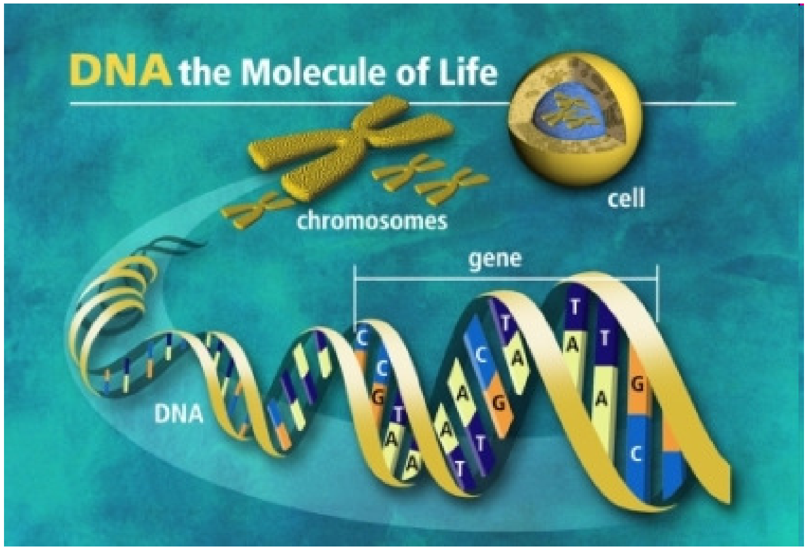
Basic concepts of genetics

- Cell composing the organism
- Cell nucleus contains chromosomes
  - Human cells have 23 pairs of chromosomes
- A chromosome is a macromolecule called DNA
  - Two strands of DNA bond in a double helix structure
    - Complement: one paternal one maternal
  - Each strand is a chain of nucleotides of 4 types (A, G, C, T)
    - A nucleotide is a nucleic acid unit, and A (G) only binds with T (C)
  - A chromosome can have 50-250 million pairs of nucleotides
    - $4^{50} \sim 4^{250} = 2^{100} \sim 2^{500}$ permutations
    - When uncoiled 1.7-8.5cm long, considering the size of cell!
- A functional segment of a chromosome is called a gene
  - 20,000-25,000 human genes

Basic concepts of genetics

- Genotype is genetic markup of a cell or organism
  - Non-evaluable but the evolutionary search takes place
- Phenotype is measurable or observable attributes, traits, characteristics of an organism
  - Evaluable expression
- Genotype encodes phenotype

Genome is all genes of an organism



Darwin’s theory of evolution

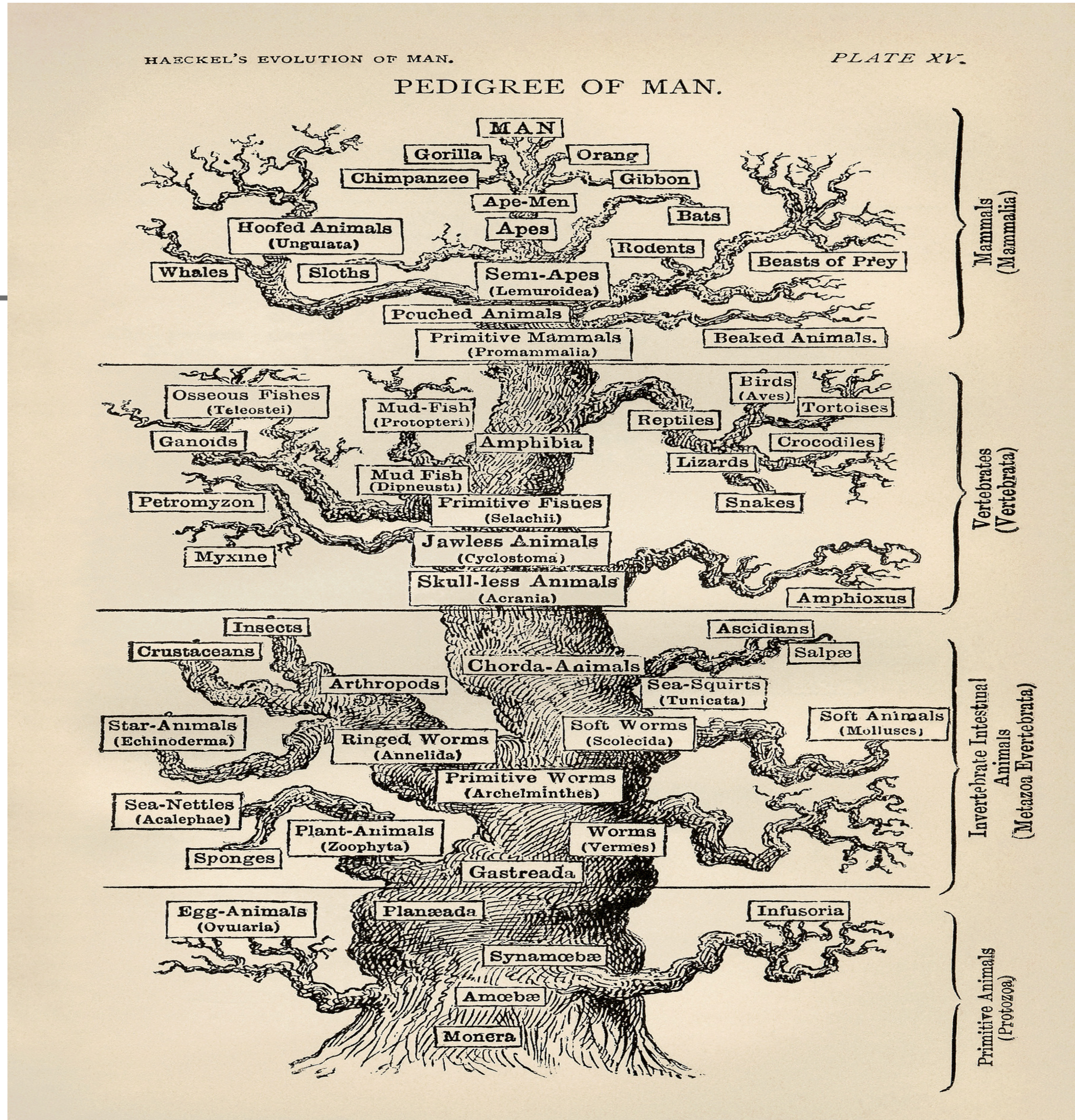
- Survival of the fittest
  - In a non-perfect environment, only a limited number of individuals that adapt or fit to it best survive (selection)
    - Fitness is affected by phenotypic traits
  - Genotypes are propagable to next generation
    - Small and random variations in genotypes occur
    - Hence variations in phenotypes happen
  - A process of trial-and-error (generate-and-test)
    - No guarantee on reaching the best fitness due to randomness
- Two cornerstones of evolution theory
  - Selection acts as a force pushing quality
  - Variation creates necessary diversity to facilitate novelty

Timeline of evolution

Age of the universe: 14 billion years
Age of the Sun: 4.6 billion years
Age of the Earth: 4.54 billion years

- 3.8 billion years of simple cells
- 3 billion years of photosynthesis
- 2 billion years of complex celles
- 1 billion years of multicellular life
- 600 million years of simple animals
- 570 million years of arthropods
- 550 million years of complex animals
- 500 million years fish & proto-amphibians
- 475 million years of land plants
- 400 million years of insects and seeds
- 360 million years of amphibians
- 300 million years of reptiles
- 200 million years of mammals
- 150 million years of birds
- 130 million years of flowers
- 65 million years dinosaurs died out
- 2.5 million years since genus homo
- 200,000 years since homo sapiens
- 25,000 years Neanderthals died out

Cambrian explosion when the rate of evolution suddenly accelerated



- On the tree of evolution, all species are equal. It is there just because it is possible.

Local search by the biological evolution

- All branches of all possible evolution trees form the space the evolution can explore
  - That is a huge space, and our tree of evolution represents only one possible process
- Therefore, human beings may not be the best result (on another evolution tree) if some different variations happened in history
  - Given the huge search space (possibilities)

Search by evolutionary algorithms

- Many local search algorithms (e.g. Tabu, simulated annealing) are naïve evolutionary algorithms
  - Evolutionary algorithms start from a population while local search from one solution
  - Evolutionary algorithms do both recombination (crossover) and mutation while local search only mutation

Evolutionary algorithms

- The fitness (objective, cost, utility) function to optimize
  - Minimize
  - Maximize
- A population of possible solutions
  - Each individual solution is equivalent to a chromosome (DNA)
  - Each element of solution is a gene
  - Individuals are static that do not adapt but the population
  - Size

Evolutionary algorithms

- Evolution by variation and selection
  - Recombination
    - Parent selection
      - Fitter individuals get higher probability to reproduce
      - Cannot be too greedy otherwise get stuck in local optimum
    - What parts to combine and how should be stochastic
  - Mutation
    - A random, unbiased change to possibly reach any solution candidate
    - Therefore, the global optimum is possible to reach
  - Survivor selection (replacement strategy)
    - The fittest survive

Evolutionary algorithms

- Initialization
  - The initial population can be randomly generated
  - Heuristics can be used to generate a fitter population
- Termination condition
  - After a known optimal fitness level is reached
  - After the maximally allowed CPU time elapses
  - After a number of fitness evaluations
  - When the fitness improvement is smaller than a threshold value

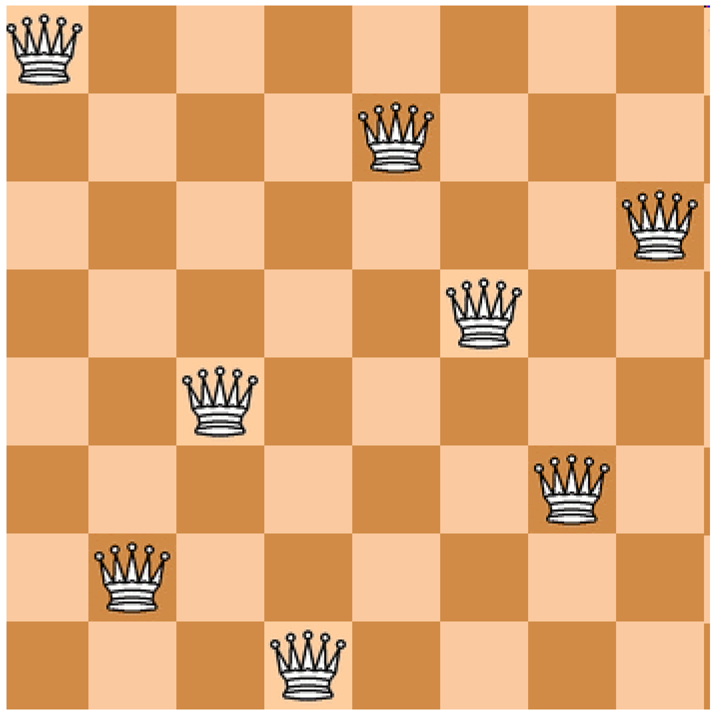
Evolutionary algorithms – pseudo code

Randomly generate the initial population of *m* individuals

Do before *termination condition* is satisfied

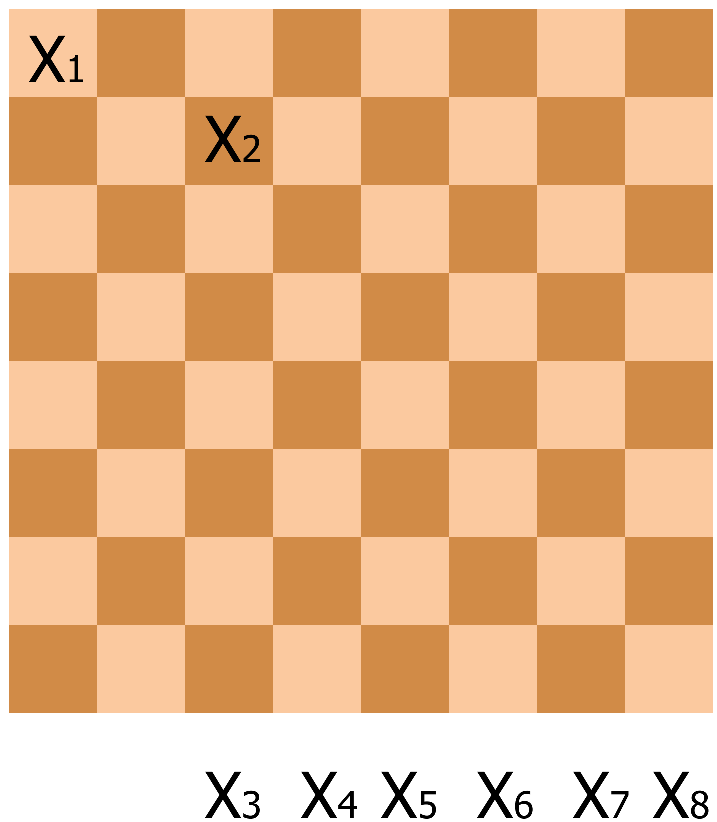
- Randomly select a pair of parents
- Crossover the two parents to generate an offspring
- Mutate the offspring
- Randomly select a candidate and compare its fitness with the offspring's
- If the offspring is fitter, keep the offspring and remove the candidate from the population

Eight-queens problem – an example



Eight-queens problem by backtracking

Incremental construction



Eight-queens problem by evolution algorithm

- Format of a candidate solution in population
  - $[q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8]$
  - Each column has a queen and q_i denotes row number
  - For *n*-queens problem, as *n* grows, the factorial *n*! becomes larger than all polynomials and exponential functions (but slower than double exponential functions) in *n*
- Objective function
  - The number of checking queen pairs
- Initial population
  - A population of 100 randomly generated permutations

Eight-queens problem by evolution algorithm

- Parent selction
  - Choose 5 individuals from population and use two fittest as parents
- Recombination (crossover) operator
  - Cut both parents at the same position (1 ~ 7)
  - The offspring take genes from one segment of a parent and the rest from another

p1: [1, 2, 3, 4, 5, 6, 7, 8]
p2: [2, 4, 1, 8, 3, 5, 7, 6]

Cut at position 3

os: [2, 4, 1, 3, 5, 6, 7, 8]

Eight-queens problem by evolution algorithm

- Mutation operator
  - Select two random positions and swap values on the two positions

[2, 4, 1, 3, 5, 6, 7, 8]

[2, 5, 1, 3, 4, 6, 7, 8]

- Survivor selection
  - Randomly select an individual from population
  - If it is less fit than new offspring, replace it with new offspring
  - Otherwise, discard new offspring
- Easy to be made automated!