COMPX216-24A
Artificial Intelligence

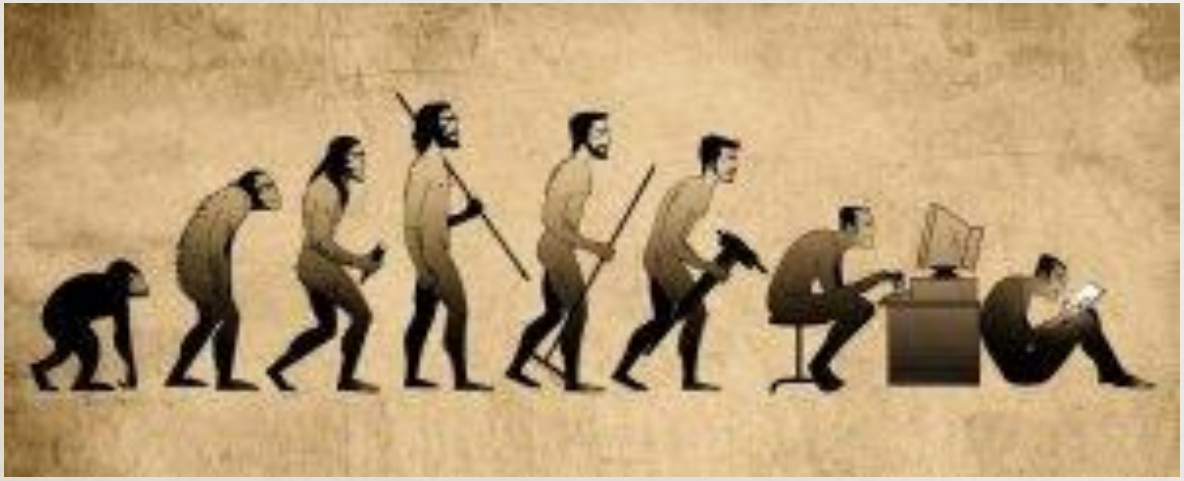
Evolutionary algorithms

Today: Evolutionary algorithms

- Mimicking evolution and natural selection
- Design options for evolutionary algorithms
- A genetic algorithm for the 8-queens problem
- Pseudo code of a genetic algorithm
- Theoretical considerations
- Genetic algorithms vs. evolution in biology
- An application: evolved antennas

Mimicking evolution and natural selection

- **Population** of **individuals** exists in an **environment** with limited resources
- Competition for those resources causes selection of fitter individuals that are **better adapted**
- Those individuals reproduce to form new generation of individuals through **recombination** and **mutation**
- New individuals have **fitness** evaluated, high fitness individuals chosen to reproduce, pass on good traits
- Over time, natural selection causes **fitness to rise**



https://mymodernmet.com/new-evolutionary-theories-man-from-europe/

Mimicking evolution and natural selection

- Each set of states encountered in the search can be viewed as a **generation**, with members obtained from the **parent** generation
- Each parent produces **offspring** and *k* individuals from the parents' offspring will be selected based on their **fitness**
- Loose analogue in the natural world: obtaining a new generation with asexual reproduction and natural selection
- Evolutionary algorithms take this further and produce offspring by **recombining** parent states
- If there are two parents, this is an analogue of the most common form of reproduction in nature

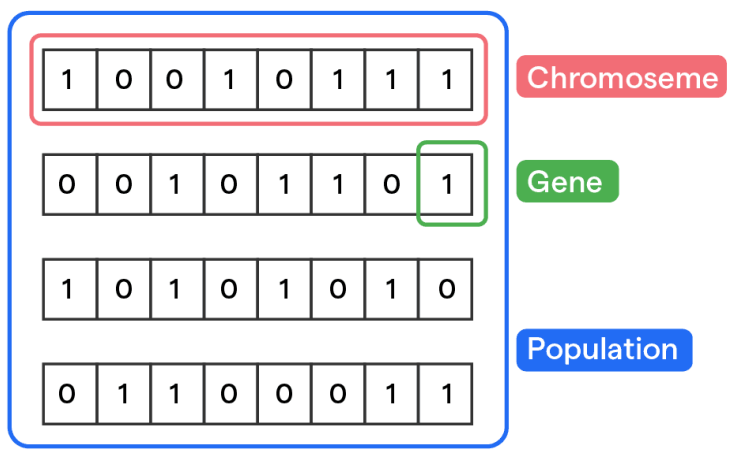
Design options for evolutionary algorithms

- Initialization usually done randomly
  - need to ensure even mixture of candidates
- Termination condition
  - Reach some known/desired fitness
  - Reach a generation limit
  - Reach a minimum diversity
  - Reach a generation limit of no improvement
- The fitness function (i.e., the objective function)
  - How 'fit' an individual is to reproduce

Design options for evolutionary algorithms

- The population:
  - holds (representation of) possible solutions
  - size of the population (i.e., the parameter k), usually fixed
  - selection operators usually take whole population into account
  - diversity of population refers to the different number of fitnesses
- The representation (genetic material) of an individual:
  - **genetic algorithms** use a string (binary, integer)
  - **evolution strategies** use sequence of real numbers (real-valued vectors)
  - **genetic programming** use computer programs (tree structure)
  - **evolutionary programming** use finite state machines

Design options for evolutionary algorithms



https://botpenguin.com/glossary/genetic-algorithms

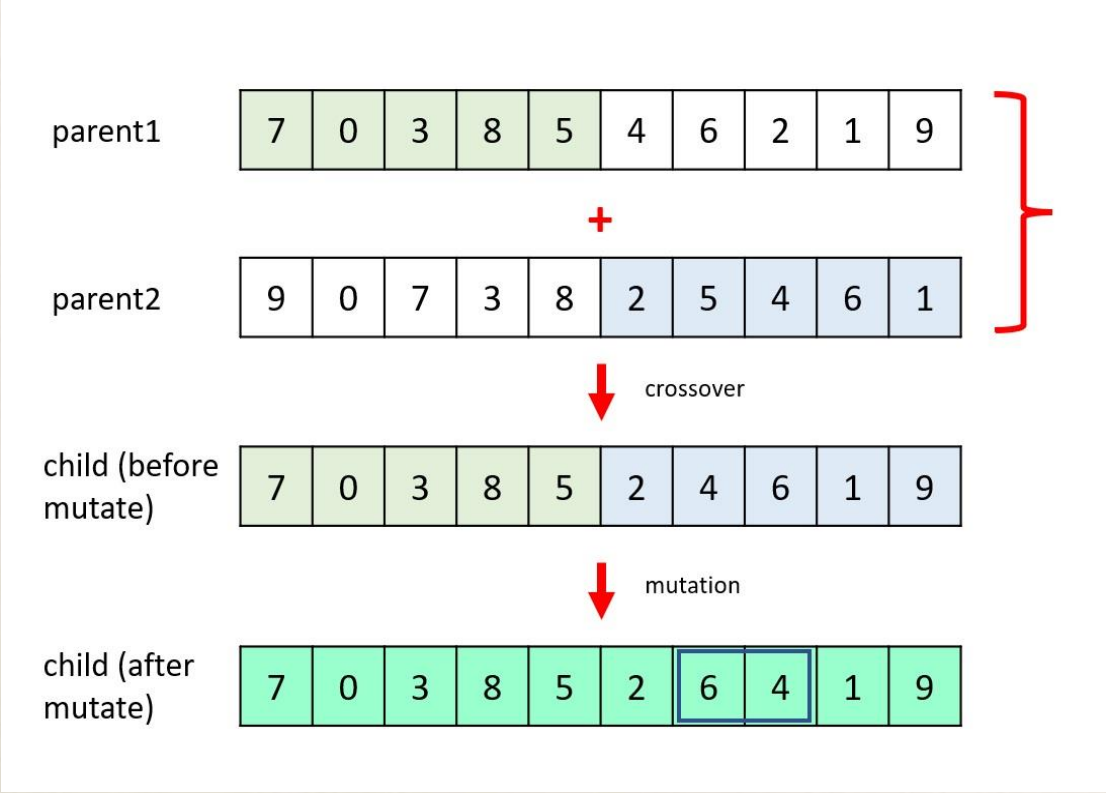
Biological concept	Genetic algorithm equivalent	Description
Chromosome	Candidate solution (Individual)	A complete solution represented as a string (binary, integer, real-valued vectors, etc.)
Gene	Single element in a solution	A part of the solution (one bit in a binary string, one number in a vector)
Population	Set of candidate solutions	A collection of chromosomes (solutions) that evolve over generations

Design options for evolutionary algorithms

- The number of parents (more than two are possible!)
- The selection of the next parent generation:
  - Roulette wheel selection (fitness proportionate selection): selects individuals with probability proportional to their fitness
  - Tournament selection: randomly select a few individuals and pick the best one.
  - Rank-based selection: ranks individuals based on fitness and selects them with probabilities proportional to their ranks, **not actual fitness values**
  - Elitism: keeps the fittest parents from the previous generation
  - Boltzmann selection (Simulated Annealing-Inspired): Uses a probability function that allows weaker individuals to be selected early on, but as evolution progresses, stronger individuals dominate.
  - Culling: discard individuals with poor fitness (**culling**)

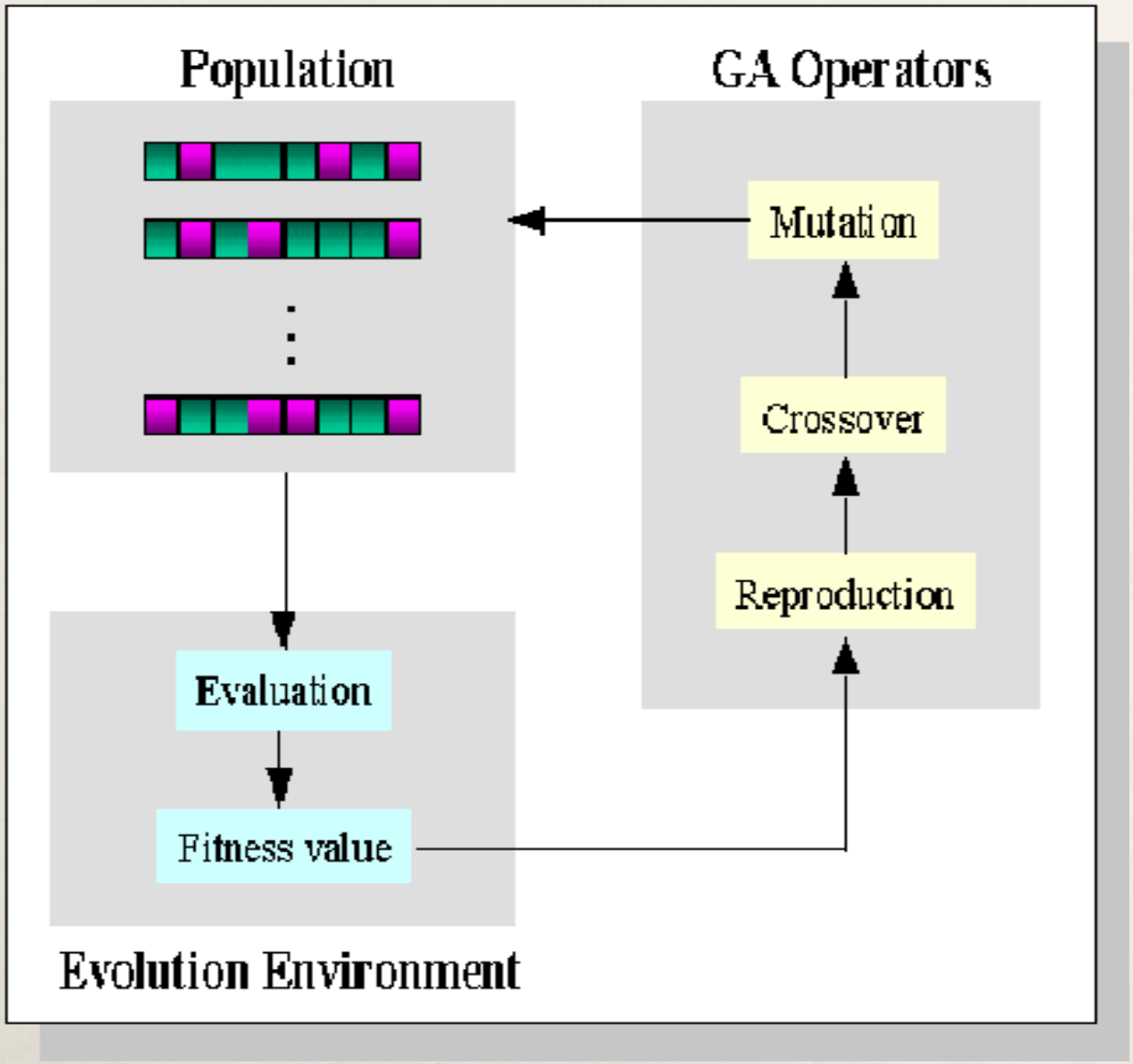
Design options for evolutionary algorithms

- Option for recombination: take the first part from one parent, up to a **cross-over point**, and the second part from the other
- The **mutation rate**: probability that random changes are applied to an individual's representation



https://visualstudiomagazine.com/Articles/2022/12/20/traveling-salesman-problem.aspx?Page=1

Genetic algorithm



https://www.researchgate.net/publication/287290332_Recent_trends_in_advanced_control_of_wastewater_systems

Pseudo code of a genetic algorithm

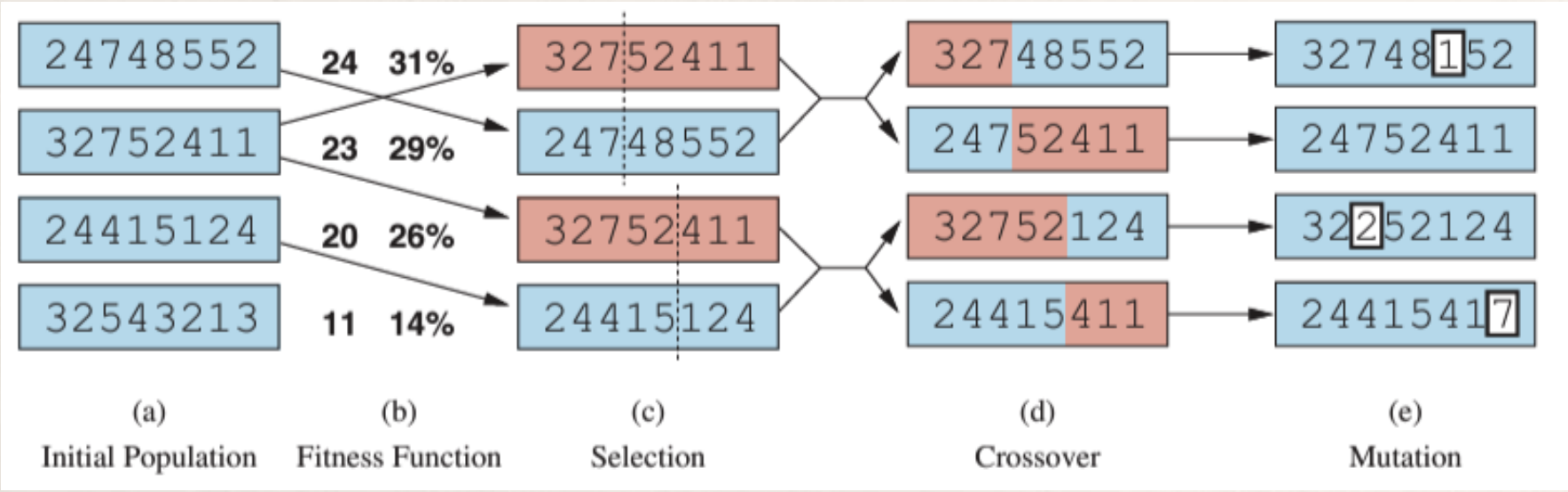
```
1   INITIALIZE population w/ random individuals
2   REPEAT UNTIL (termination condition)
3     EVALUATE population / individual fitness
4     SELECT parents with high fitness
5     COMBINE parents to form offspring
6     MUTATE resulting offspring
7     NEXT POPULATION = offspring
```

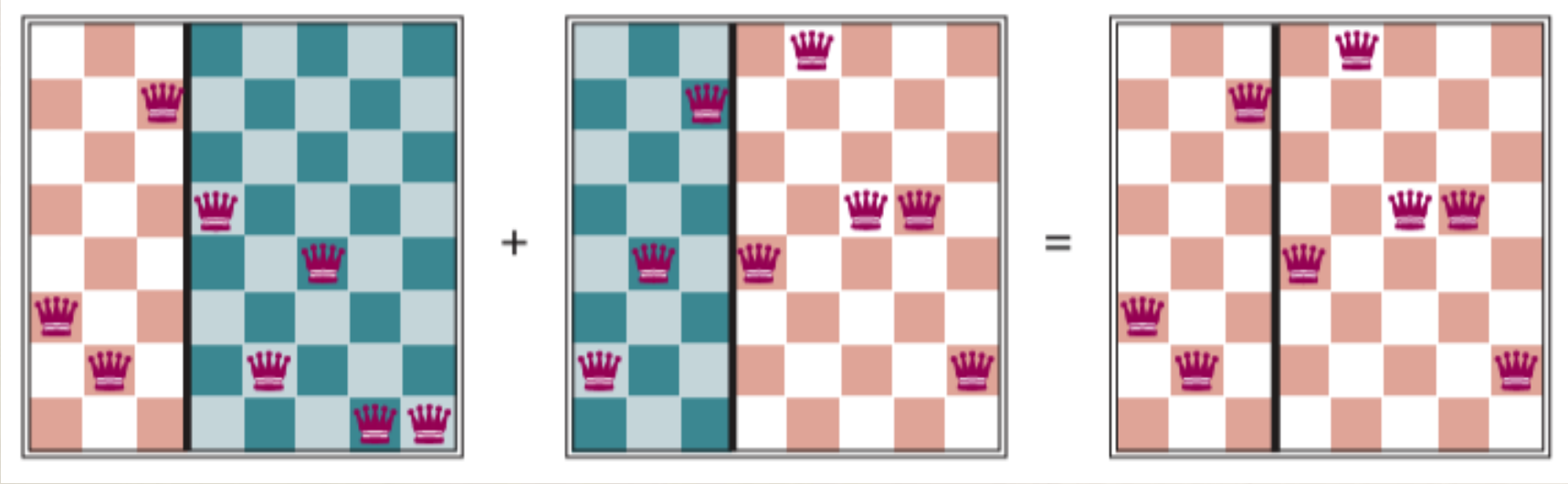
Pseudo code of a genetic algorithm

```
function GENETIC-ALGORITHM(population, fitness) returns an individual
  repeat
    weights ← WEIGHTED-BY(population, fitness)
    population2 ← empty list
    for i = 1 to SIZE(population) do
      parent1, parent2 ← WEIGHTED-RANDOM-CHOICES(population, weights, 2)
      child ← REPRODUCE(parent1, parent2)
      if (small random probability) then child ← MUTATE(child)
      add child to population2
    population ← population2
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to fitness

function REPRODUCE(parent1, parent2) returns an individual
  n ← LENGTH(parent1)
  c ← random number from 1 to n
  return APPEND(SUBSTRING(parent1, 1, c), SUBSTRING(parent2, c + 1, n))
```

A genetic algorithm for the 8-queens problem





Theoretical considerations

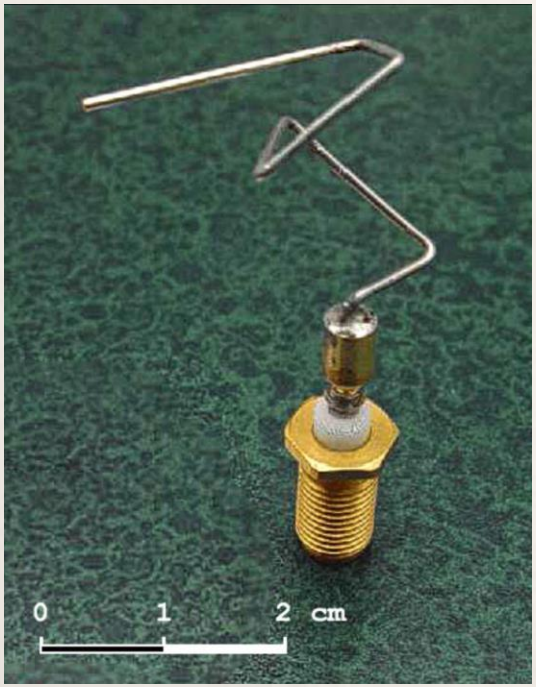
- Theory of genetic algorithms has been developed using **schema**: patterns with wildcards (e.g., 246*****)
- Strings that match a schema are called instances
- Finding: if the average fitness of a schema's instances is above the mean, the number of instances of that schema grows over time
- Genetic algorithms are best when "good" schemas correspond to contiguous blocks that are likely to be preserved in cross-over
- The "good" schemas are those that correspond to meaningful components of fit individuals
- Implication: creating a suitable representation of individuals, where such components may be present, is key

Genetic algorithms vs. evolution in biology

Aspect	Biological Evolution	Genetic Algorithms
Purpose	Adaptation to the environment over many generations.	Find an optimal or near-optimal solution to a problem.
Representation	DNA sequences represent genetic traits.	Chromosomes represent solutions (e.g., binary strings).
Timescale	Millions of years	Seconds to hours
Environment	Real, dynamic, unpredictable	Simulated, controlled, deterministic
Reproduction	Sexual/asexual reproduction	Algorithmic crossover/mutation
Fitness	Fitness is based on how well a solution solves the problem.	Fitness is based on survival and reproduction capabilities.
Constraints	Physical/biological laws	Programmer-defined
Outcome	Diverse life forms	Problem-specific solutions

An application: evolved antenna

- 2006 NASA ST5 spacecraft antenna: an evolved antenna that made it into space



- Link to PDF with publication describing the process: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1426&context=smallsat

Genetic algorithm examples

- https://rednuht.org/genetic_cars_2/
- https://www.geeksforgeeks.org/genetic-algorithms/