Markov chains and genetic algorithms

Margareta Ackerman

Markov chains

Markov process

- A process in which the next state (“the future”) depends only on the current state
- *Memoryless*: It has no memory of how it got to the current state

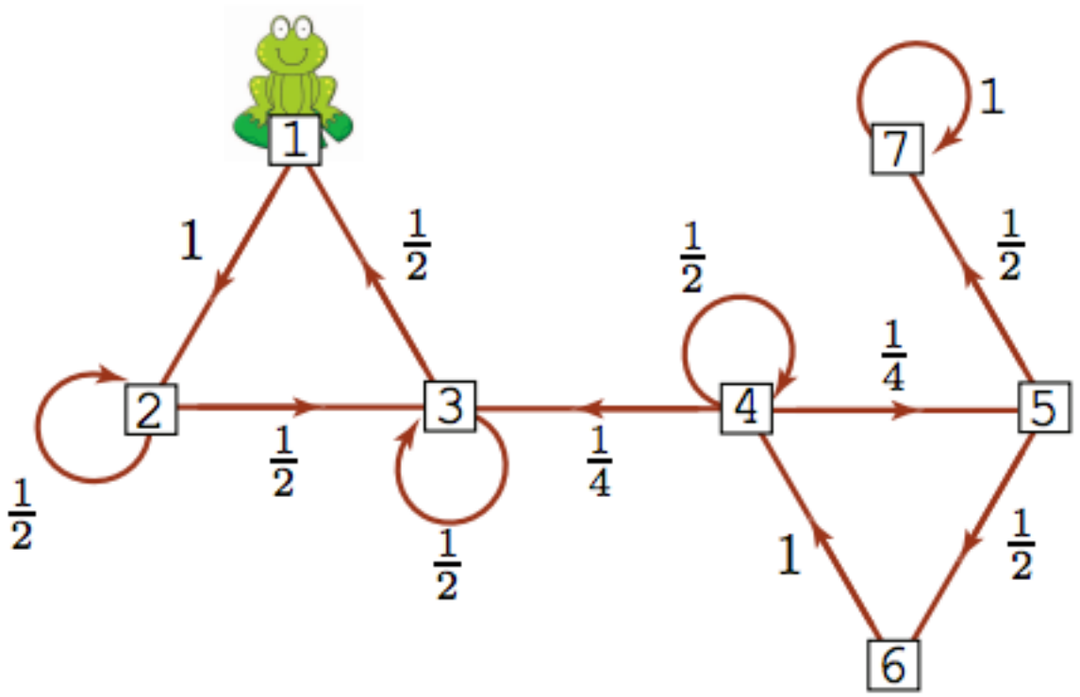
Markov chains

- Andrei Andreyevich Markov 1906
- One of the most powerful analytical techniques

Formal Definition

- A Markov chain is a discrete-state random process where the evolution of a state depends only on the current state and not how the current state was reached.

$$\Pr\left[X_{n+1} = s_i \mid \left(X_n = s_j\right) \wedge \left(X_{n-1} = s_k\right) \wedge \ldots \wedge \left(X_1 = s_l\right)\right] = \Pr\left[X_{n+1} = s_i \mid X_n = s_j\right]$$



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

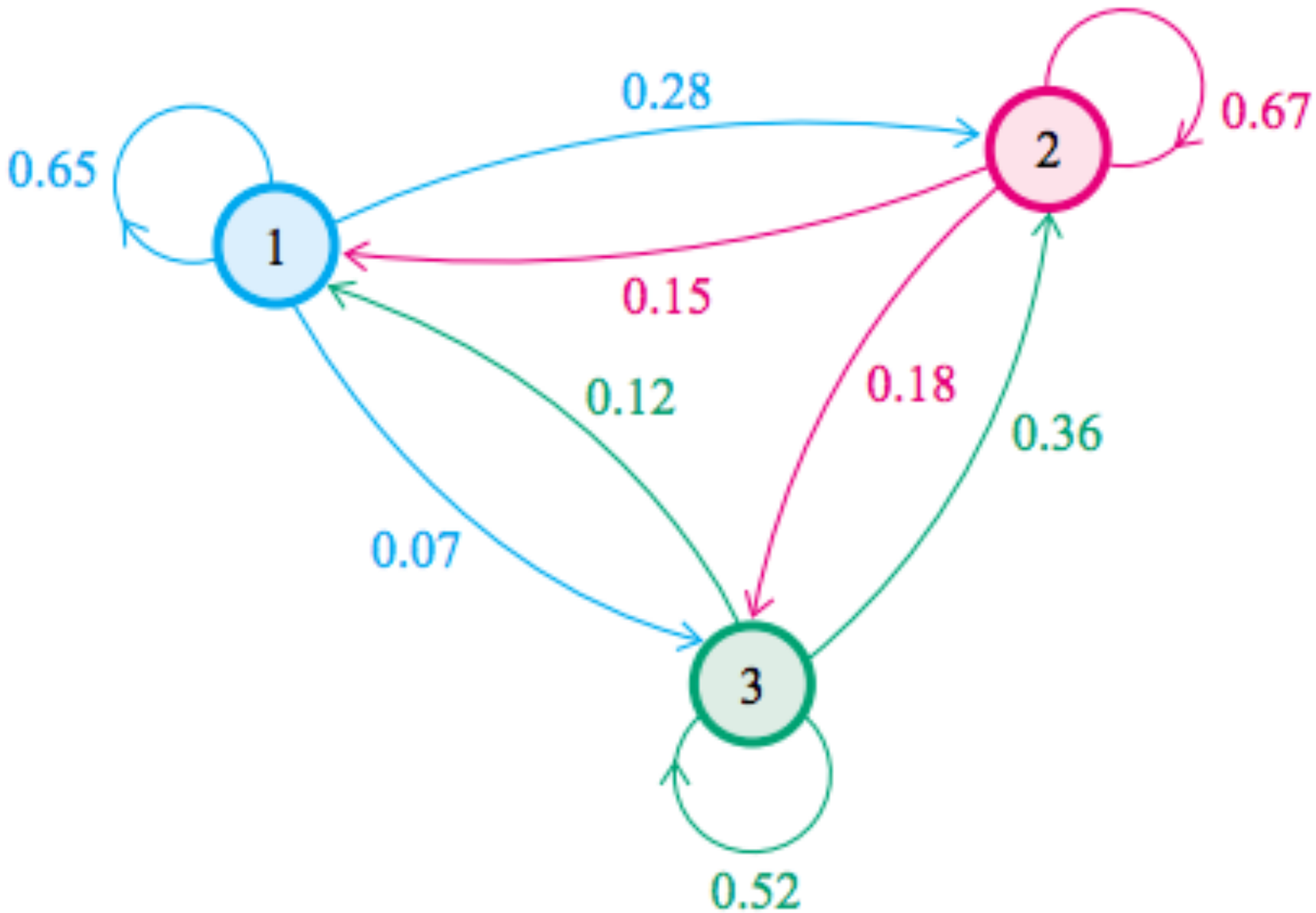
Example from Sociology

- Sociologists found that the strongest determinant in one's income is their parents' income.

Table 1

		Next Generation		
	State	1	2	3
Current Generation	1	0.65	0.28	0.07
	2	0.15	0.67	0.18
	3	0.12	0.36	0.52

1: lower 2: middle 3: upper


0.65
0.28
0.67
1
2
0.15
0.12
0.18
0.36
0.07
3
0.52

Simple text generation

- Let the words represent states
- Given some text, count how often each word is followed by another word
- Create the corresponding probability matrix
- Use to generate new sequences of words

Simple text generation

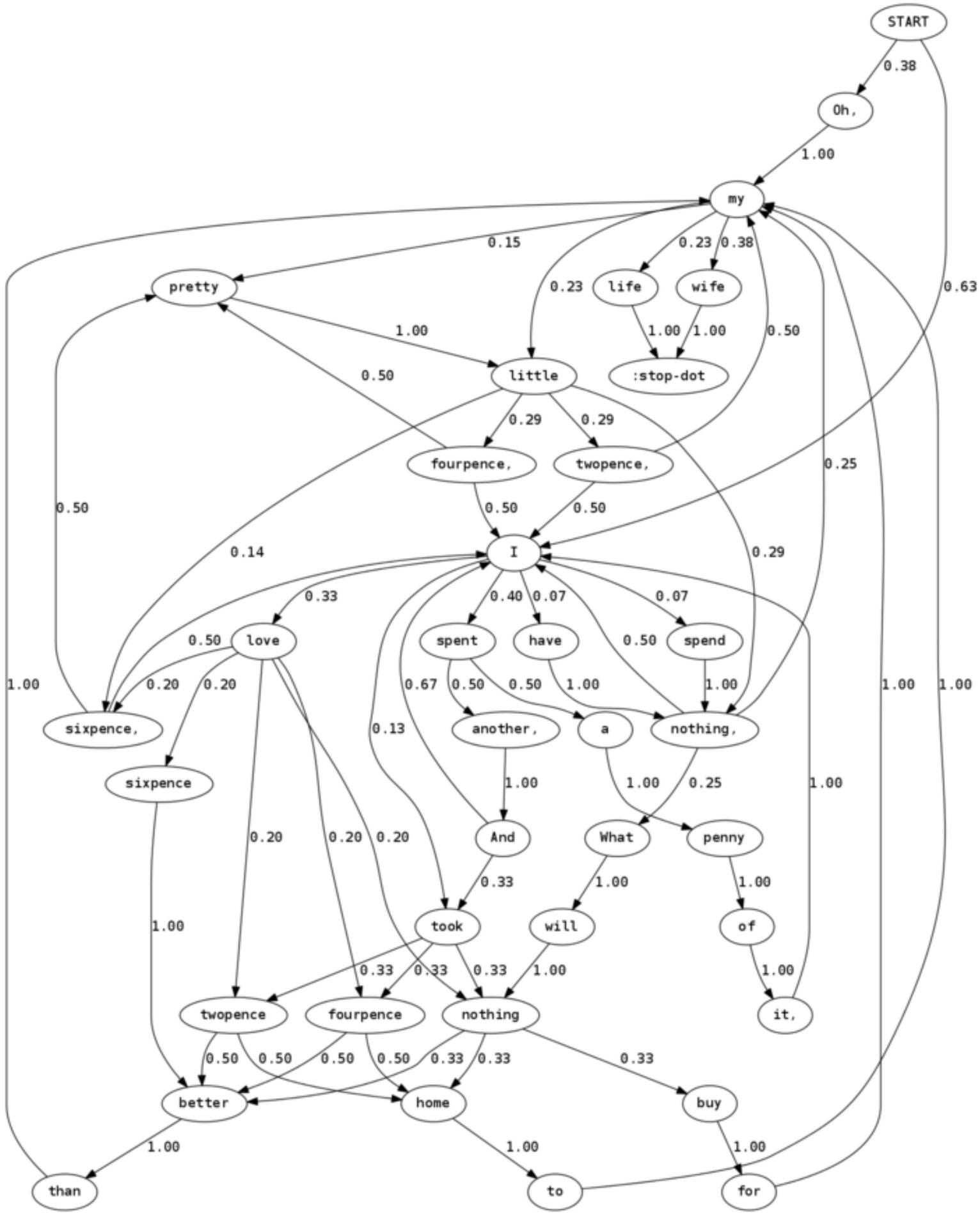
I love sixpence, pretty little sixpence,
I love sixpence better than my life.
I spent a penny of it, I spent another,
And took fourpence home to my wife.

Oh, my little fourpence, pretty little fourpence,
I love fourpence better than my life.
I spent a penny of it, I spent another,
And I took twopence home to my wife.

Oh, my little twopence, my pretty little twopence,
I love twopence better than my life.
I spent a penny of it, I spent another,
And I took nothing home to my wife.

Oh, my little nothing, my pretty little nothing,
What will nothing buy for my wife.
I have nothing, I spend nothing,
I love nothing better than my wife."

Consider as 4 separate texts


START
0.38
Oh,
1.00
my
0.15
0.23
0.38
0.63
pretty
0.23
life
wife
1.00
1.00
0.50
1.00
0.50
little
:stop-dot
0.29
0.29
fourpence,
twopence,
0.25
0.50
0.50
0.50
0.14
I
0.29
0.33
0.40
0.07
0.07
0.50
love
spent
have
0.50
spend
1.00
0.20
0.20
0.67
0.50
0.50
1.00
1.00
1.00
1.00
sixpence,
0.13
another,
a
nothing,
sixpence
1.00
1.00
0.25
1.00
0.20
0.20
0.20
And
What
penny
0.33
1.00
1.00
1.00
took
will
of
0.33
0.33
0.33
1.00
1.00
twopence
fourpence
nothing
it,
0.50
0.50
0.50
0.50
0.33
0.33
0.33
better
home
buy
1.00
1.00
1.00
than
to
for

What does it generate

- Often meaningless phrases
- But it can surprise us!
  - "Oh, my life."
  - "I spent another, And took fourpence home to my wife."
  - "I love fourpence better than my wife."

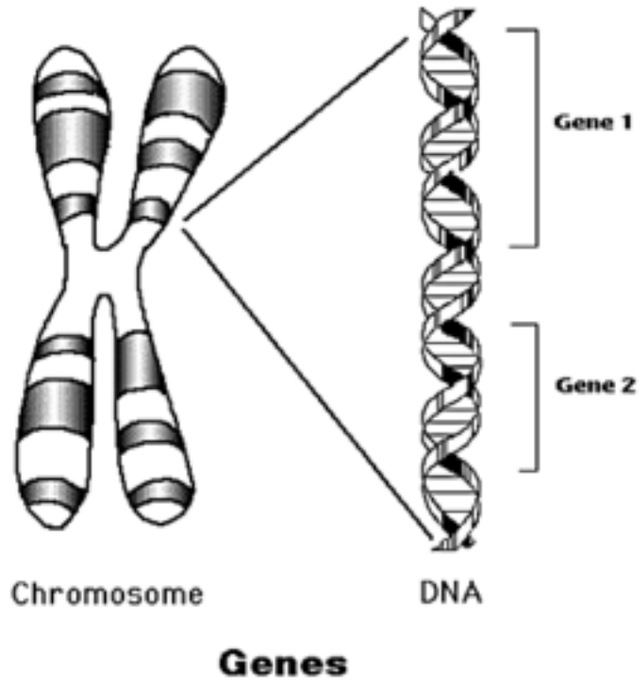
Setting the probabilities

- Best way: Analyze a corpus to determine the probabilities
- Alternative: set probabilities based on domain expertise

Genetic algorithms

Motivation

- Mimic evolution in nature
- A chromosome is a string of DNA
- A gene is a string of DNA that determines some feature of the organism



Motivation

- Reproduction:
  - Recombination of genes from parents
    - Small number of random mutations
- Survival of the fittest: survive *and* reproduce

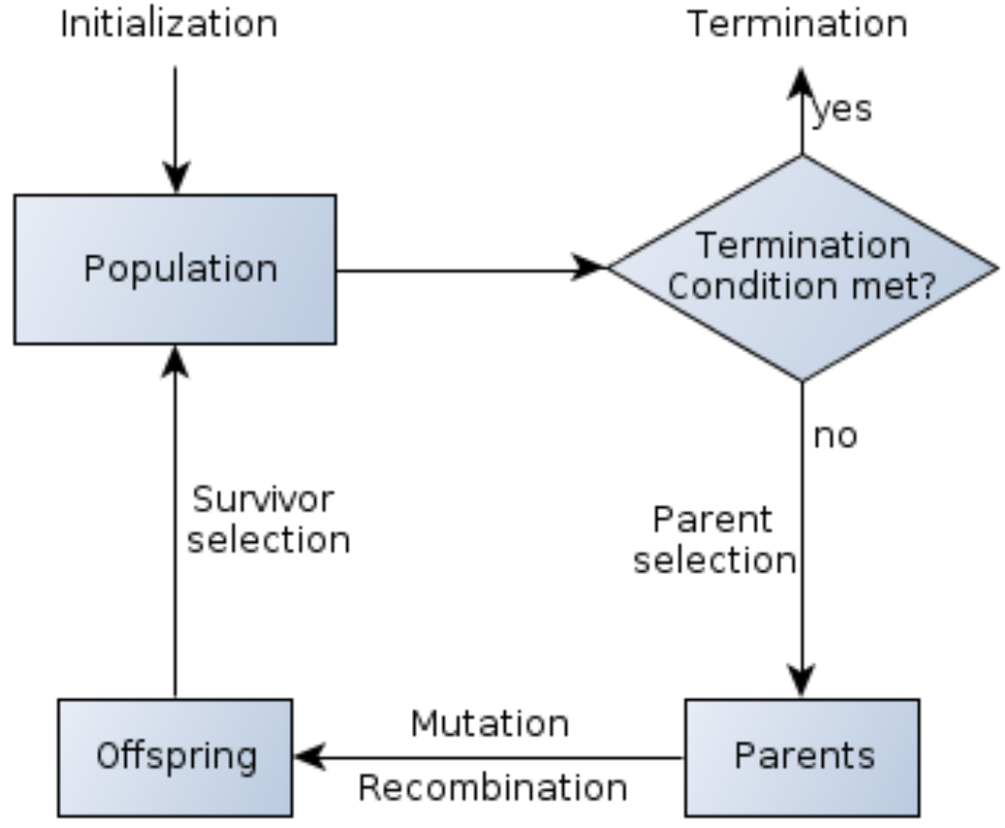
Genetic algorithms

- **Generation**: A set of individuals (classically binary strings)

- **Fitness function**: A formula that determines whether an individual becomes a parent

Genetic algorithms

- Imitate evolution



Genetic algorithms

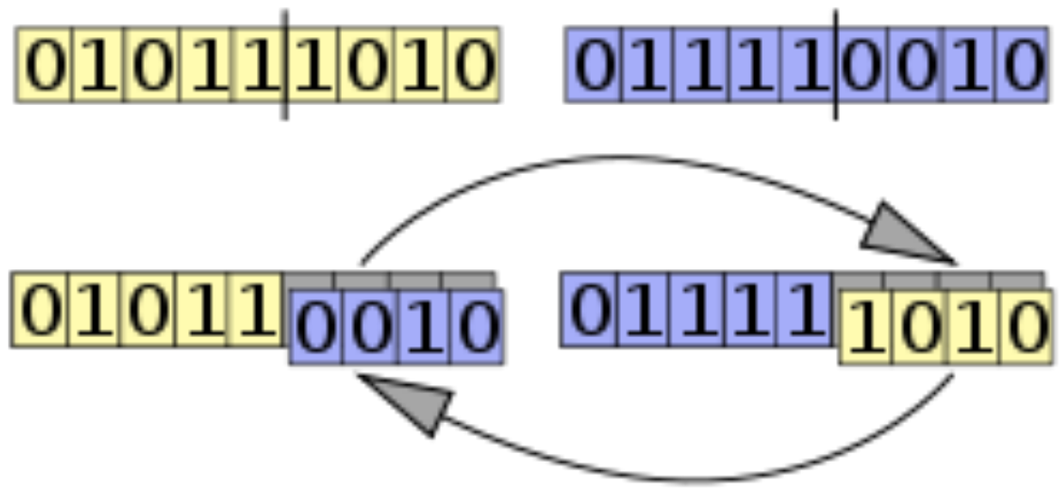
- **Initialization** - Typically randomly generated initial population.
- **Evaluation** - Find the fitness for each member.
- **Selection** - Select who will get to breed.
- **Crossover** - Create new members by combining traits of their parents.
- **Mutation** - Add a small number of random changes in individual members.
- **Repeat** - start over at evaluation with the new generation

Fitness Function

- One of the most important choices
- Can vary radically, including:
  - Mathematical formulae
  - Simulations
  - User input (inefficient)

Crossover

- A random crossover section is chosen to create a child from a pair of parents



Example: GA for design

- Assume you wish to build a physical object (car, plane, parachute etc)
- Isolate important features (weight, surface area, …)
- Represent a member through a string of feature settings
- Evaluate fitness through a simulation

Mutation

- For each part of the new child, change it with a small probability

When to use genetic algorithms?

- Effective in a wide range of applications, from the arts to engineering, hardware, automobile design
- Warning: Genetic methods are NOT efficient

Fitness

- The ability to survive and reproduce
- Parents are chosen to mate with probability proportional to their fitness
- Children replace their parents
- There are many variations

Genetic algorithms and Computational creativity

- One of the primary techniques of algorithmic compositions
- Can be used to create generative art

Resources

- http://www.egr.msu.edu/~goodman/GECSummitIntroToGA_Tutorial-goodman.pdf

- Evolutionary art: http://www.cs.bham.ac.uk/~rjh/courses/NatureInspiredDesign/2009-10/StudentWork/Group4/EvoArt.pdf

- Another source: http://ir.nmu.org.ua/bitstream/handle/123456789/115124/fb354ca780d35ffcf82cc1b44a5a6c35.pdf?sequence=1