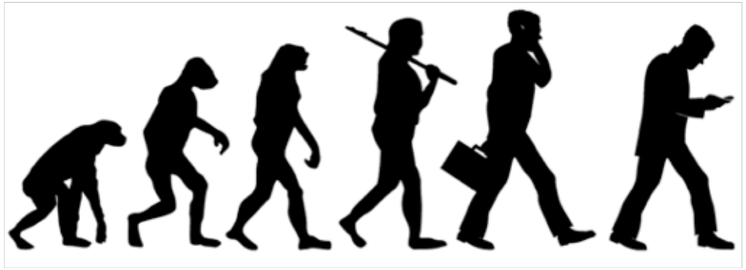
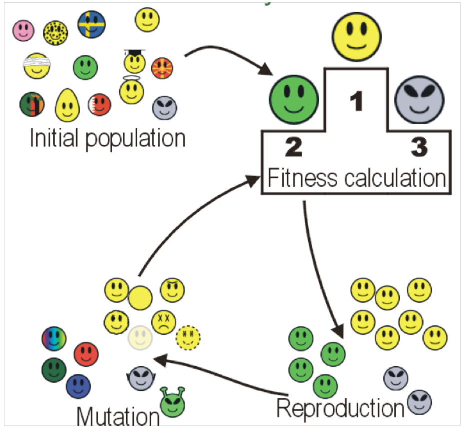
NTNU

Norwegian University of Science and Technology

Lecture 2



INTRODUCTION TO EVOLUTIONARY ALGORITHMS



Håken Jevne,
Kazi Ripon and Pauline Haddow

Outline

- Inspiration
- Biological Evolution
- Theory Behind
  - Darwinian Evolution
  - Influence by Malthus
- Evolutionary Cycle
- Classic Example: Genetic Algorithms (GAs)
- Components of GA
- Example and Simulation

Outline

- **Inspiration**
- Biological Evolution
- Theory Behind
  - Darwinian Evolution
  - Influence by Malthus
- Evolutionary Cycle
- Classic Example: Genetic Algorithms (GAs)
- Components of GA
- Example and Simulation

Start with a Dream…

- Suppose you have a problem.
- You don’t know how to solve it.
- What can you do?
- Can you use a computer to somehow find a solution for you?
- This would be nice! *Can it be done?*

Basic Idea (A Dumb Solution)

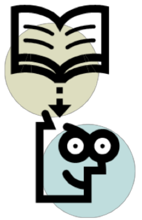
- A "blind generate and test" algorithm:

Repeat

Generate a random possible solution

Test the solution and see how good it is

Until solution is good enough



Can We Use this Dumb Idea?

- Sometimes - yes:
  - if there are only a few possible solutions,
  - and you have enough time,
  - then such a method *could* be used.

- For most problems - no:
  - many possible solutions,
  - with no time to try them all,
  - so this method *can not* be used.

NTNU

A “less-dumb” Idea (EA)

Generate a *set* of random solutions

Repeat

Test each solution in the set (rank them)

Remove some bad solutions from set

Duplicate some good solutions

make small changes to some of them

Until best solution is good enough

Basic Idea of Principle of Natural Selection

"Select The Best, Discard The Rest"

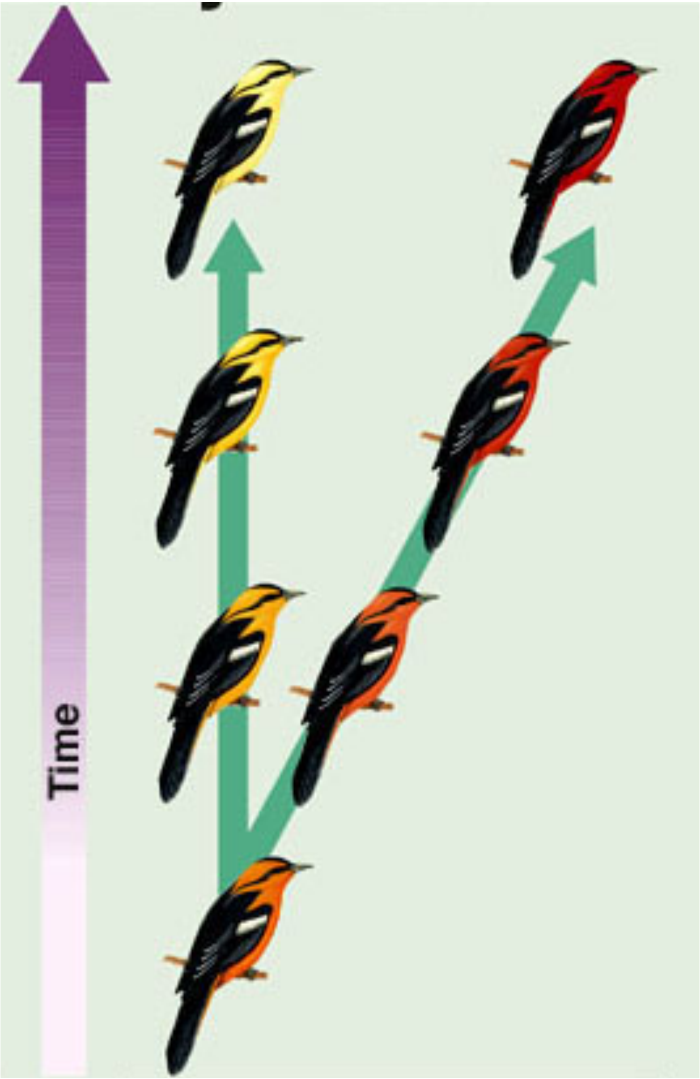
- The main principle of evolution used in GA is "*survival of the fittest*".
- The good solution survive, while bad ones die.

Outline

- Inspiration
- **Biological Evolution**
- Theory Behind
  - Darwinian Evolution
  - Influence by Malthus
- Evolutionary Cycle
- Classic Example: Genetic Algorithms (GAs)
- Components of GA
- Example and Simulation

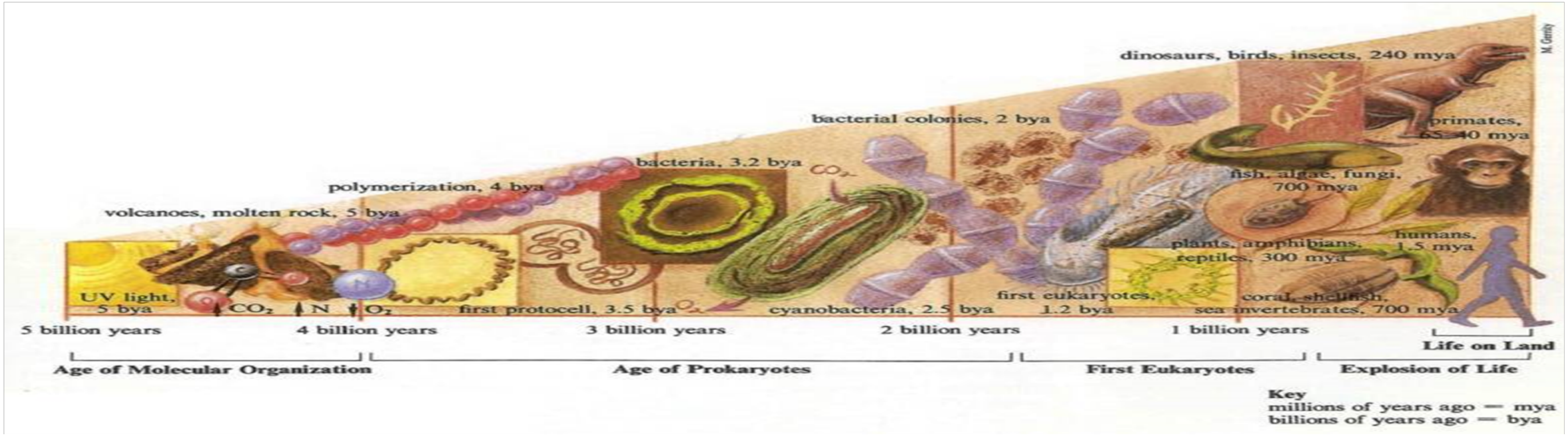
What is Evolution?

- Change over time.
- Common descent with modification.



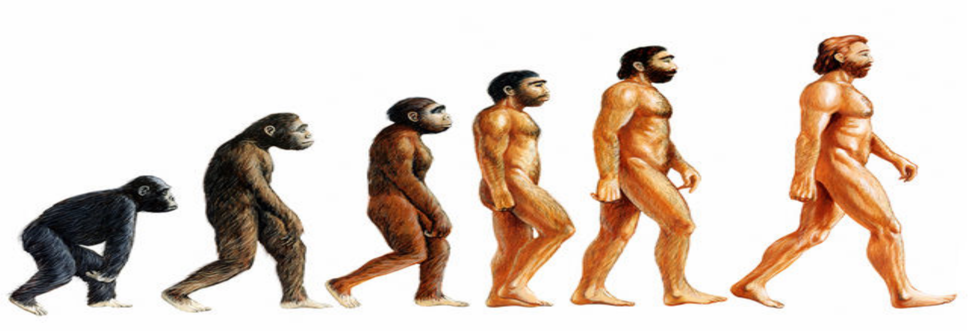
What is Evolution?

- Evolution is the change in the **genetic** make up of populations over time.
- All living things change. Populations evolve, not individuals.
- The mechanism for evolution (how it happens) is a theory. The theory of **natural selection** is a well supported, testable explanation of how evolution occurs.



Darwinian Evolution

- Natural Selection
  - Darwin's theory of evolution: *only the organisms best adapted to their environment tend to survive* and *transmit their genetic characteristics in increasing numbers to succeeding generations* *while those less adapted tend to be eliminated*.

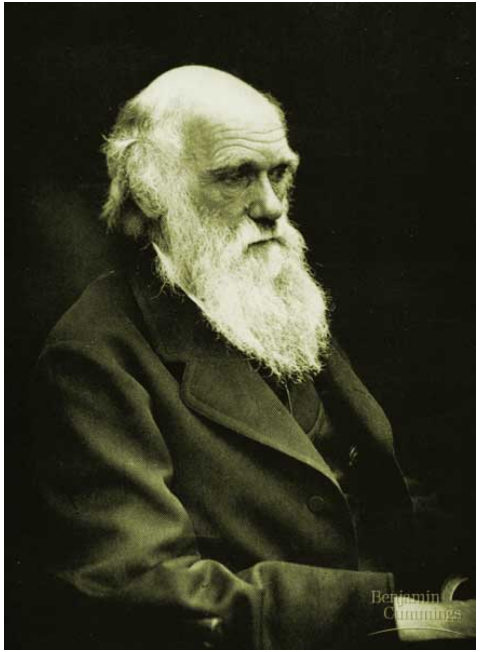


Source: http://www.bbc.co.uk/programmes/p0022nyy

Outline

- Inspiration
- Biological Evolution
- **Theory Behind**
  - **Darwinian Evolution**
  - **Influence by Malthus**
- Evolutionary Cycle
- Classic Example: Genetic Algorithms (GAs)
- Components of GA
- Example and Simulation

Who was Charles Darwin? (1809- 1882)



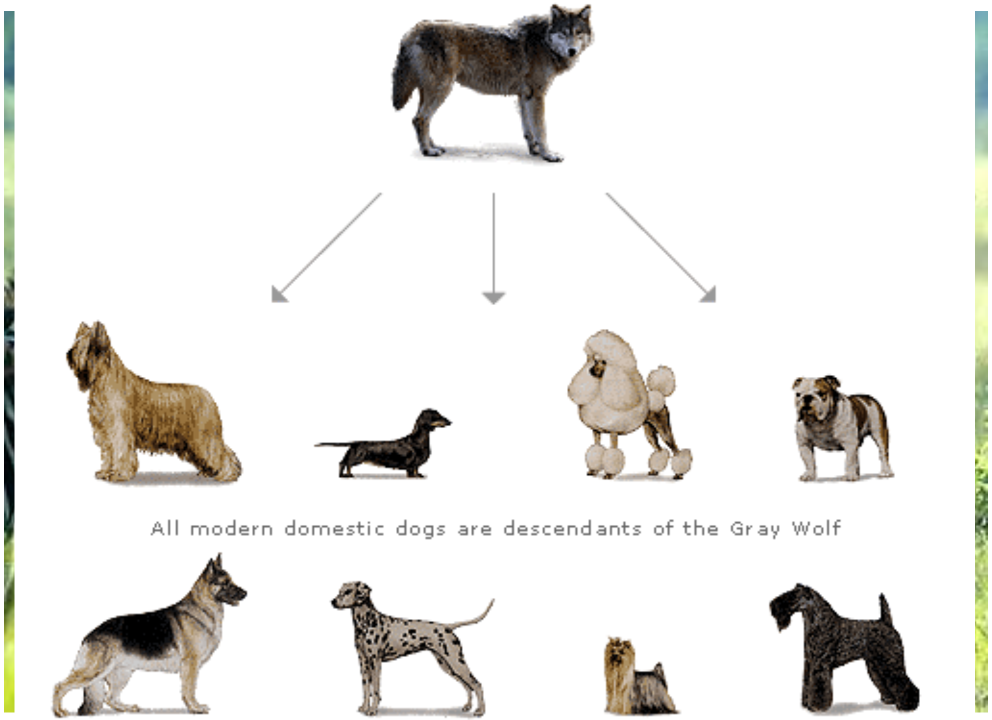
- English naturalist
- Traveled around the world on the Beagle (1831)—Famous in the Galapagos Islands
- Observed many species and fossils
- Devised his theory of evolution.

Voyage of HMS Beagle

Galapagos Island

All populations have variation

Darwin knew many farmers and animal breeders. From them and his own research he knew all individuals in a population are different.



DARWIN - After the Voyage

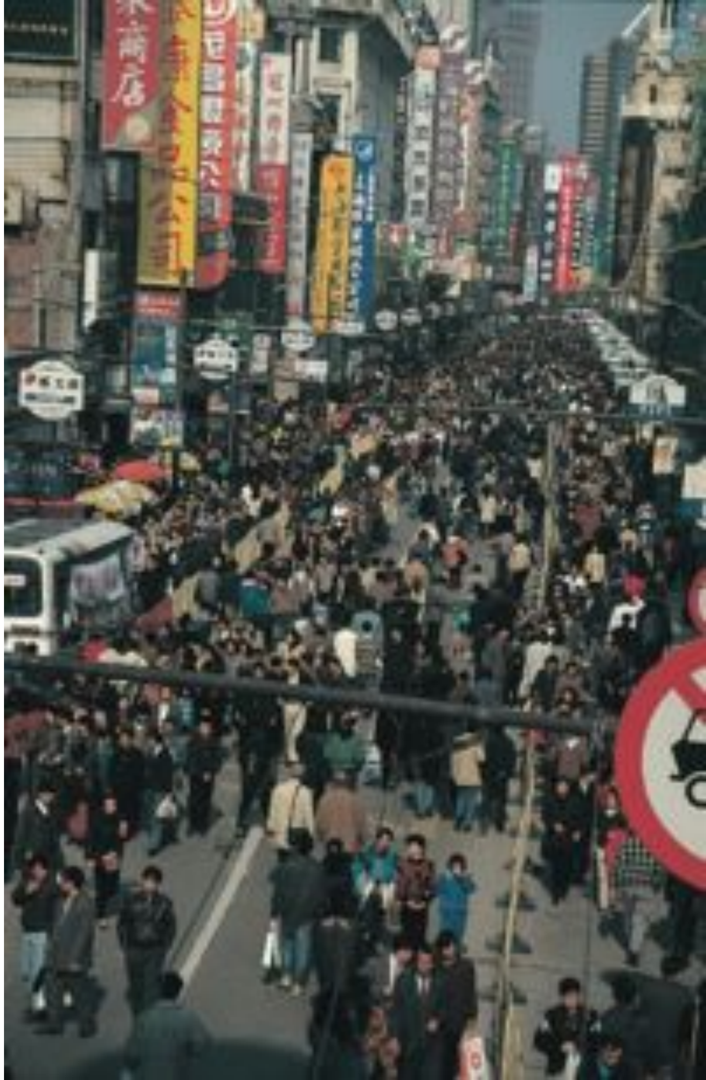
- Darwin developed his theory of Natural Selection.

- What inspired him?
  - James Hutton and Charles Lyell – Geological Record
  - Farmers/Animal Breeder - Variation in populations.
  - Malthus - Populations grow rapidly.
    - Not enough resources for all offspring.

http://www.pbs.org/wgbh/evolution/darwin/diary/

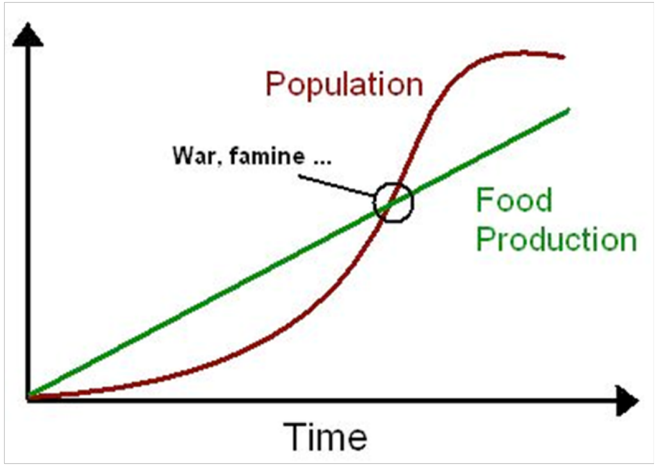
Thomas Malthus

- Thomas Malthus – English economist.
- *Essay on the Principle of Population* (1798).
- He predicted that the human <u>population</u> would grow faster than the space and food supplies needed to sustain it.



Influence of Malthus (1838)

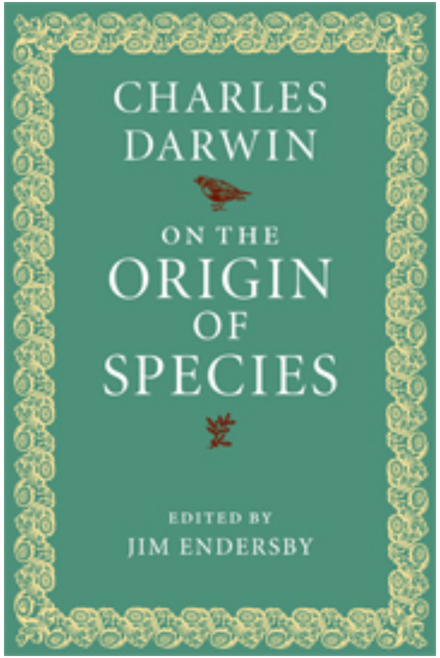
- In 1838, Darwin read for amusement Malthus's book *Population.*
- In nature, animals and plants produce more offspring than can survive.
    - This leads to a struggle for existence.
- Darwin saw that favourable variations in a population would tend to be preserved, and unfavourable ones would be destroyed.
- Darwin wondered, what determines which individuals survive and reproduce?
- He at last had a theory by which to work.



- **exponential population growth.**
- **arithmethical food growth.**

Origins of Species

- Finally published on 24 November 1859



Darwin's Theory at a Glance

- Darwin's Theory of Evolution actually contains two major ideas:

  1. organisms change over time.
  2. evolution occurs by natural selection.

“Survival of the Fittest”

- Fitness is the ability of an organism to **survive** and **reproduce** in its environment.
- Individuals in nature with characteristics best suited to their environment survive the struggle for existence.
- This principle is called **survival of the fittest**.

Natural Selection: formal definition

- The process by which **nature** allows only the organisms best suited to their environment to **reproduce** is called **natural selection**.

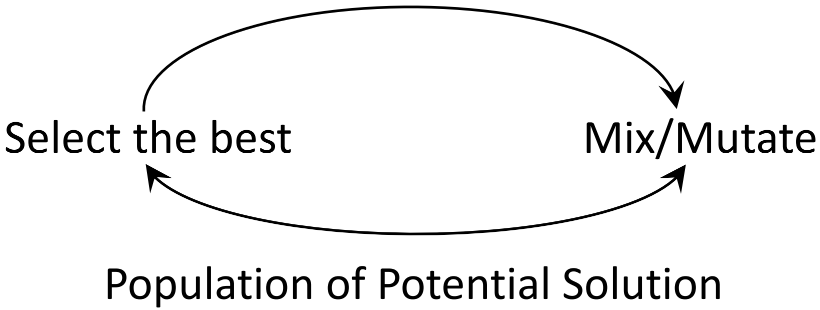
I have called this principle, by which each slight variation, if useful, is preserved, by the term Natural Selection.

—Charles Darwin from "The Origin of Species"

Outline

- Inspiration
- Biological Evolution
- Theory Behind
  - Darwinian Evolution
  - Influence by Malthus
- **Evolutionary Cycle**
- Classic Example: Genetic Algorithms (GAs)
- Components of GA
- Example and Simulation

Evolutionary Algorithms (EAs)



EAs are ***stochastic***, ***population-based*** algorithms.

They fall into the category of *"**generate and test**"* algorithms.

Generate

Mutate and/or recombine individuals in a population.

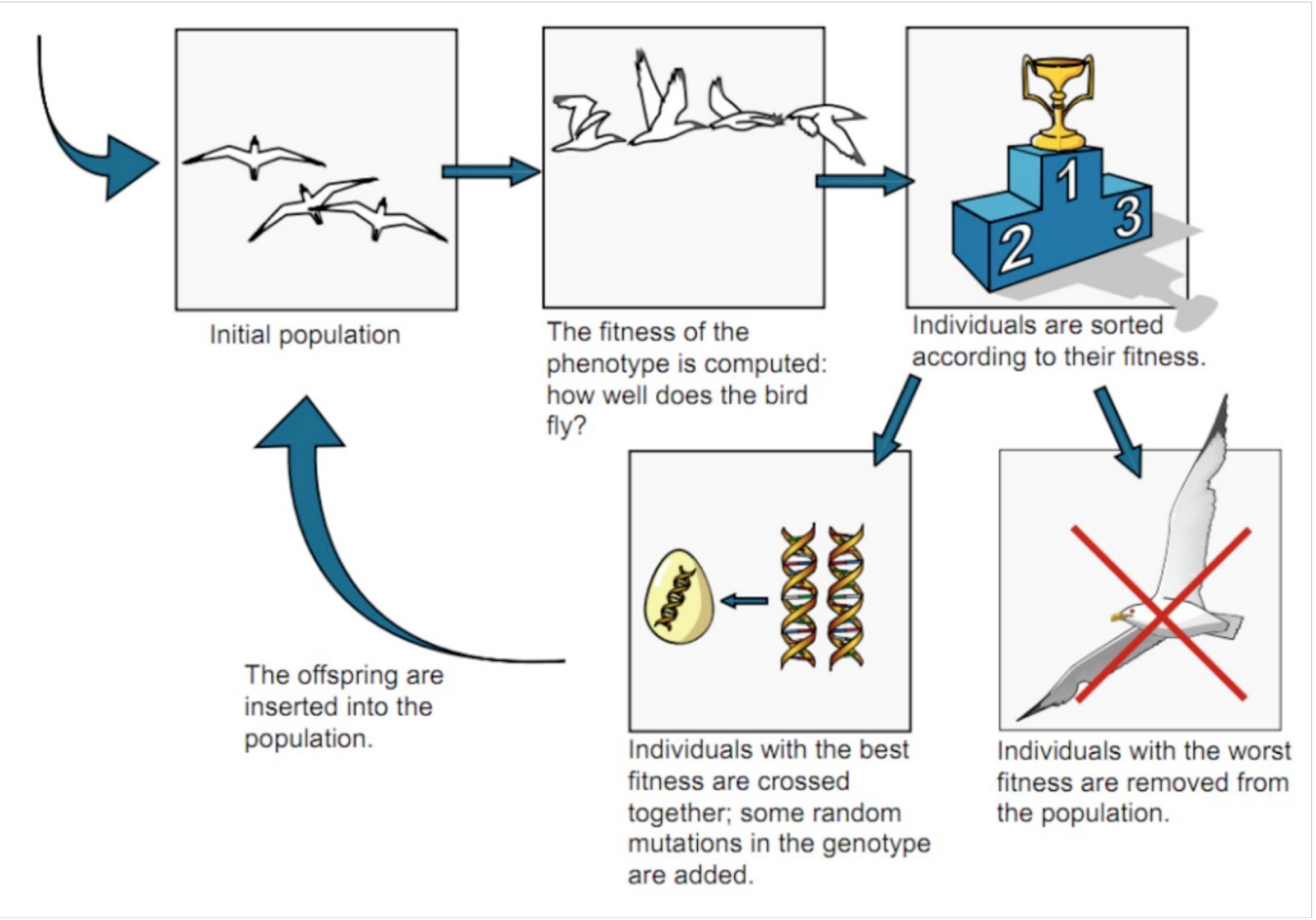
Create the necessary variation and thereby facilitate novelty.

Test

Select the next generation from the parents and offsprings.

Selection reduces variation and acts as a force pushing quality.

Evolutionary Cycle



Evolutionary Cycle

1. $t := 0$;
2. Generate initial Population $P^{(t)}$ at random;
3. Evaluate the fitness of each individual in $P^{(t)}$;
4. while (not termination condition) do
    5. Select parents, $Pa^{(t)}$ from $P^{(t)}$ based on their fitness in $P^{(t)}$;
    6. Apply crossover to create offspring from parents: $Pa^{(t)} \rightarrow O^{(t)}$
    7. Apply mutation to the offspring: $O^{(t)} \rightarrow O^{(t)}$
    8. Evaluate the fitness of each individual in $O^{(t)}$;
    9. Select population $P^{(t+1)}$ from current offspring $O^{(t)}$ and parents $P^{(t)}$;
    10. $t := t+1$;
11. end-do

EA

Some Classical EAs

- Genetic Algorithm (GA)
- Evolutionary Strategies (ES)
- Genetic Programming (GP)
- Evolutionary Programming (EP)

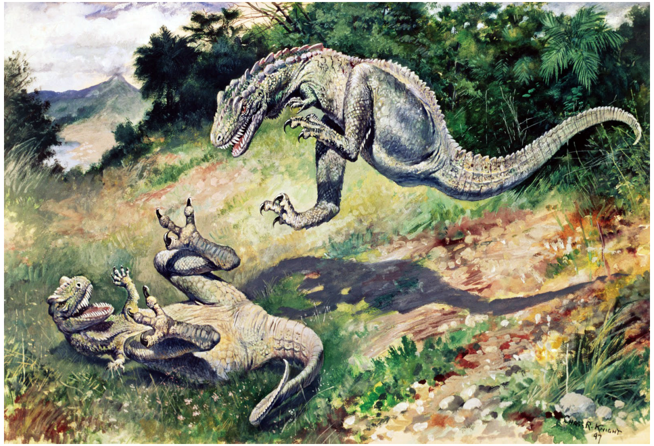
Outline

- Inspiration
- Biological Evolution
- Theory Behind
  - Darwinian Evolution
  - Influence by Malthus
- Evolutionary Cycle
- **Classic Example: Genetic Algorithms (GAs)**
- Components of GA
- Example and Simulation

Genetic Algorithm (GA)

- *Directed search algorithms* inspired by biological evolution
  - *Darwin's survival of the fittest*
  - reproduction through cross-breeding

- GA maintains *a population of candidate solutions* for the problem at hand, and makes it evolve by iteratively *applying a set of stochastic operators* .



Outline

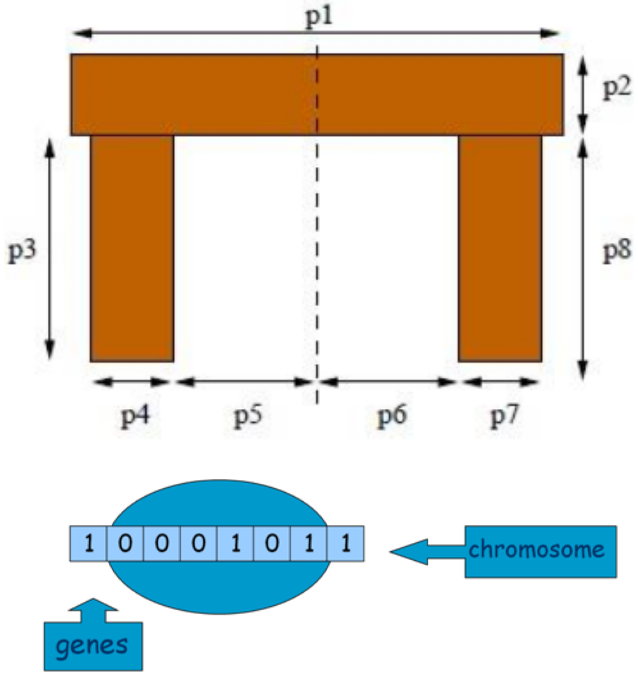
- Inspiration
- Biological Evolution
- Theory Behind
  - Darwinian Evolution
  - Influence by Malthus
- Evolutionary Cycle
- Classic Example: Genetic Algorithms (GAs)
- **Components of GA**
- Example and Simulation

Components of GA

- Representation (definition of individuals)
- Evaluation function/fitness function
- Population
- Parent selection mechanism
- Variation operators
    - Recombination (Crossover), and
    - Mutation.
- Survivor selection mechanism (replacement)

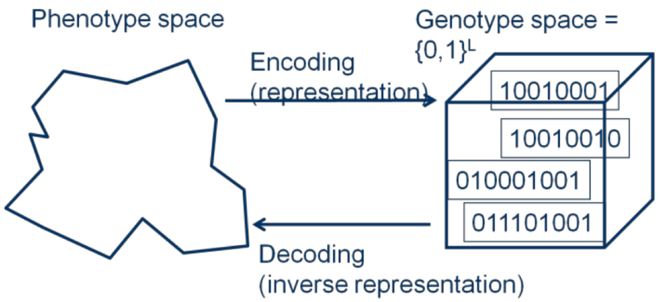
Representations

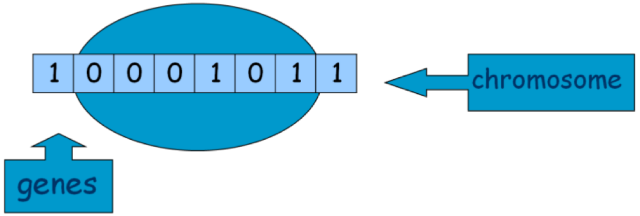
- It is important to choose the **right representation** for the problem being solved.
- When choosing a representation, we have to bear in mind how the genotypes will be evaluated and what the genetic operators might be.
- Getting the representation right is one of the most difficult parts of designing a EA.
- Often this only comes with **practice** and **a good knowledge of the application domain**.



Representations

- Candidate solutions (*individuals*) exist in *phenotype* space
- They are encoded in *chromosomes*, which exist in *genotype* space
  - Encoding : phenotype => genotype (not necessarily one to one).
  - Decoding : genotype => phenotype (must be one to one).
- Chromosomes contain *genes*, which are in (usually fixed) positions called *loci* (sing. locus) and have a value (*allele*).

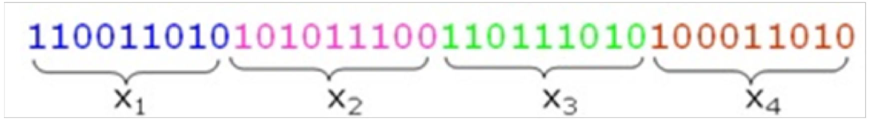




Representations

Chromosomes could be:

- Bit strings



- Real numbers

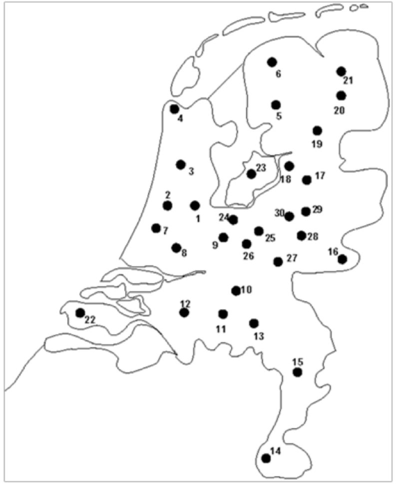
2	1.5	3.1	4.2	5.1	6.3	4.6	3.1	2.5	1.2	3.4	5.3

- Permutations of element

1/a) London	3/c) Dunedin	5/e) Beijing	7/g) Tokyo
2/b) Venice	4/d) Singapore	6/f) Phoenix	8/h) Victoria

Chromosome (3 5 7 2 1 6 4 8)

Chromosome (b e g f h a c d)



Representations

Chromosomes could be:

- Lists of rules (R1 R2 R3 ... R22 R23)
- Program elements (genetic programming)
- ... any data structure ...

Hamburger Restaurant Problem

- Price

  1 = $ 0.50 price

  0 = $10.00 price

- Drink

  1 = Coca Cola

  0 = Wine

- Ambiance

  1 = Fast snappy service

  0 = Leisurely service with tuxedoed waiter

Chromosome

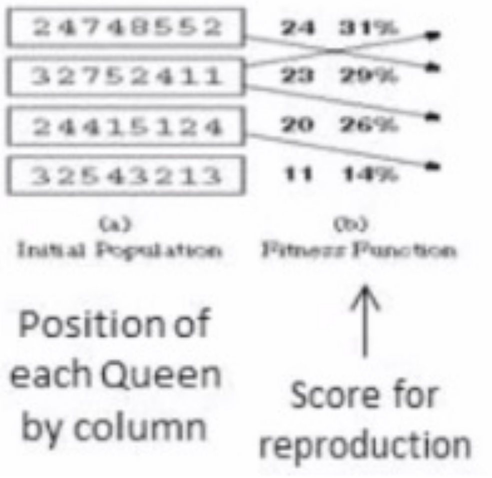
1	1	1

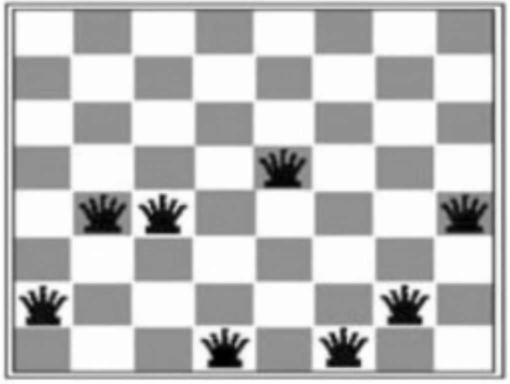
1	0	1

0	0	1

Evaluation (Fitness) Function

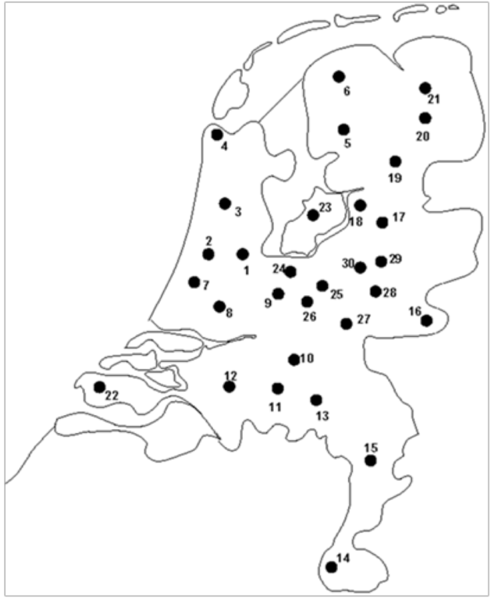
- The selection probability for reproduction is based on fitness function.
- a.k.a. *quality* function or *objective* function.
- Represents the “fitness to the environment” or “ability” of a chromosome’s.
- Assigns a single real-valued fitness to each individual which forms the basis for selection.
- Typically we talk about fitness being maximised.
  - Some problems may be best posed as minimisation problems, but conversion is trivial.





Search Space: TSP

# of cities n	possible solutions $(n\text{-}1)!$ = # of cyclic permutations
10	≈ 181,000
20	≈ 10,000,000,000,000,000 $= 10^{16}$
50	≈ 100,000,000,000,000,000,000,000,000,000, 000, ,000,000,000,000,000,000,000,000,000,000,000 $= 10^{62}$



- 81-bit problems are very small for GA.
- However, even as small as 81, $2^{81} \sim 10^{27}$ = number of nanoseconds since the beginning of the universe 15 billion years ago.

Population

- Holds (representations of) possible solutions.
- Usually has a fixed size and is a *multiset* of genotypes.
- *Diversity* of a population refers to the number of different finesses / phenotypes / genotypes present (note not the same thing).

1111010101
0111000101
1110110101
0100010011
1110111101
0100110000

Selection Mechanism

- Candidates are selected at random.
- They are selected based on their fitness function score.
- One MAY be selected more than once, where as one may NOT be selected at all.

Selection Mechanism

- Usually probabilistic.
  - high quality solutions more likely to become parents than low quality .
    - but not guaranteed.
  - even worst in current population usually has non-zero probability of becoming a parent.
- This *stochastic* nature can aid escape from local optima.

Selection Scheme

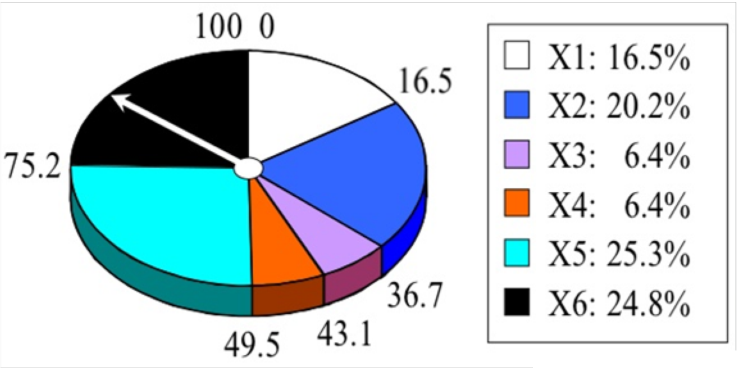
Fitness-Proportionate:

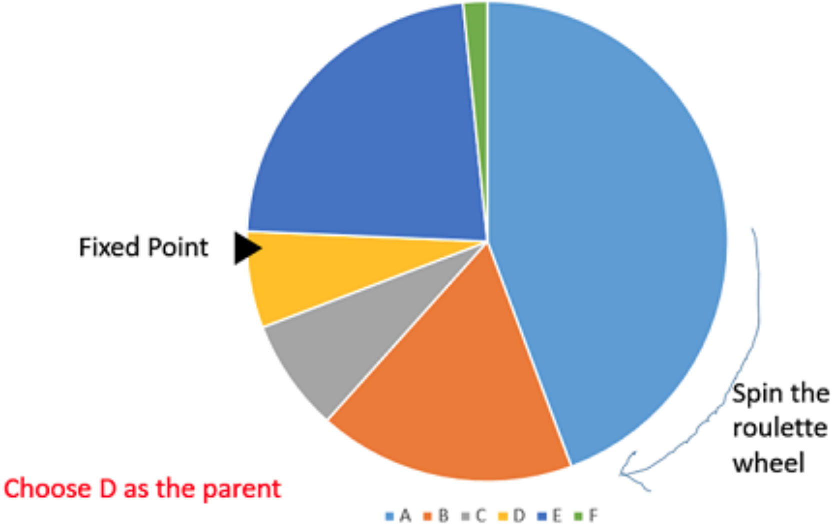
- Every individual can become a parent with a probability which is proportional to its fitness.
- Applies a selection pressure to the more fit individuals in the population, evolving better individuals over time.
- It is generally more sensitive to selection pressure
  - Scaling function

Ordinal based:

- Selects individuals not upon their raw fitness,
  - but upon their rank within the population.
- Selection pressure is independent of the fitness distribution of the population and solely based upon the relative ordering.

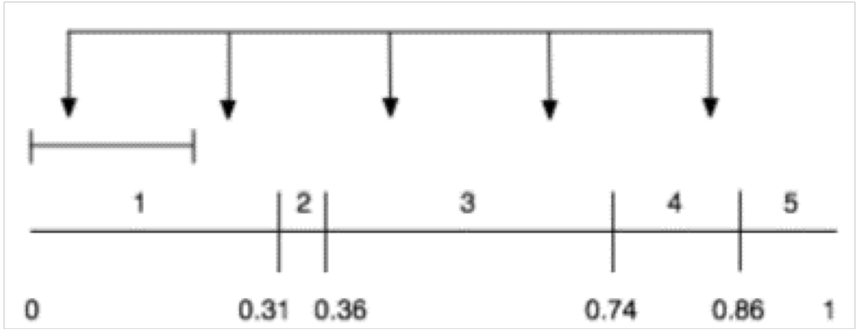
Fitness-Proportionate: Roulette Wheel





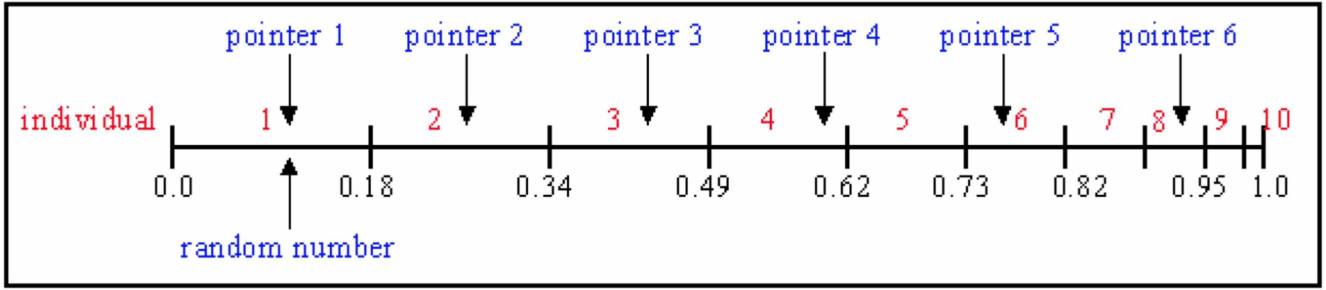
Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Fitness-Proportionate: Stochastic Universal Sampling (SUS)



- An elaborately-named variation of roulette wheel selection.
- It is a development of fitness proportionate selection (FPS) with **minimum spread and zero bias**.
- This gives weaker members of the population (according to their fitness) a chance to be chosen and thus reduces the unfair nature of FPS.
- Ensures that the observed selection frequencies of each individual are in line with the expected frequencies.
  - Standard roulette wheel selection does not make this guarantee.

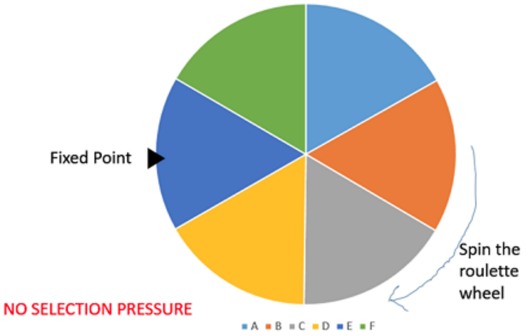
Fitness-Proportionate: Stochastic Universal Sampling (SUS)



- Instead of a single selection pointer employed in roulette wheel methods, SUS uses N equally spaced pointers, where N is the number of selections required.
- Works by making a single spin of the roulette wheel.
- The population is shuffled randomly and a single random number *pointer1* in the range [$0, 1/N$] is generated.
- The N individuals are then chosen by generating the N pointers, starting with *pointer1* and spaced by $1/N$, and selecting the individuals whose fitness spans the positions of the pointers.

Ordinal based: Raking Selection

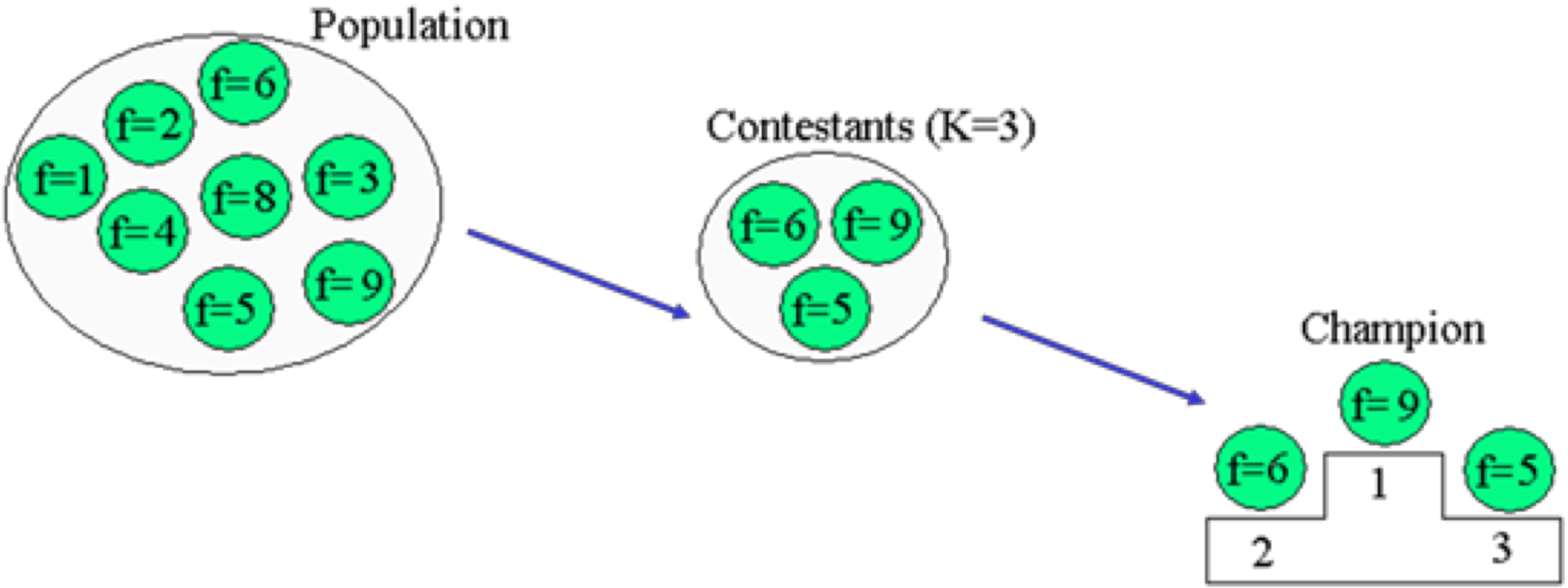
- Mostly used when the individuals in the population have very close fitness values (usually at the end of the run).
  - Loss in the selection pressure towards fitter individuals
- Remove the concept of a fitness value while selecting a parent.
- However, every individual in the population is ranked according to their fitness.
- The selection of the parents depends on the rank of each individual and not the fitness.



Chromosome	Fitness Value
A	8.1
B	8.0
C	8.05
D	7.95
E	8.02
F	7.99

Chromosome	Fitness Value	Rank
A	8.1	1
B	8.0	4
C	8.05	2
D	7.95	6
E	8.02	3
F	7.99	5

Ordinal based: Tournament Selection



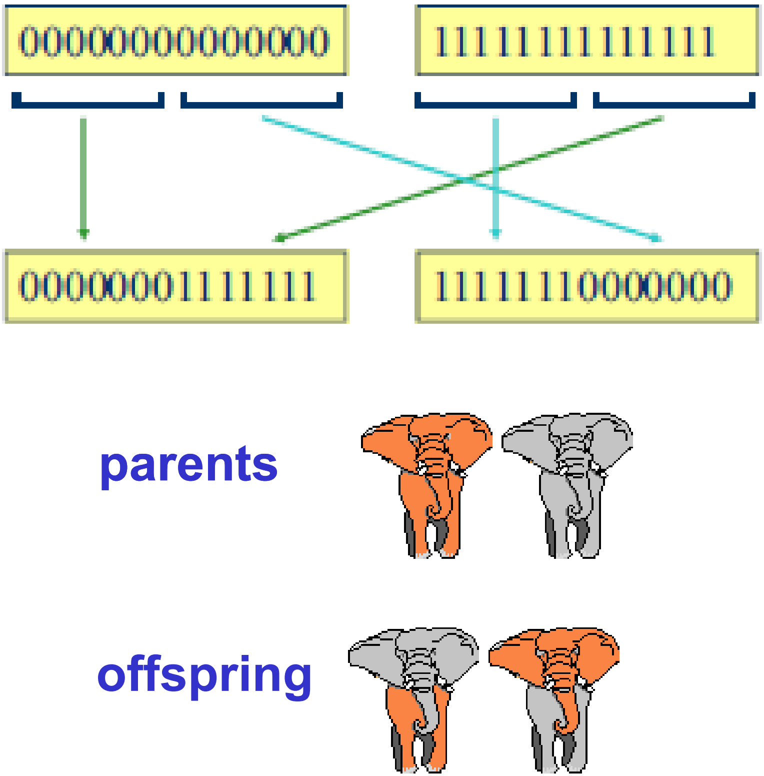
Variation Operators

- Role is to generate new candidate solutions.
- Usually divided into two types according to their *arity* (number of inputs):
  - Arity 1 : mutation operators.
  - Arity >1 : Recombination operators.
  - Arity = 2 typically called *crossover*.

- *Choice of particular variation operators is representation dependent.*

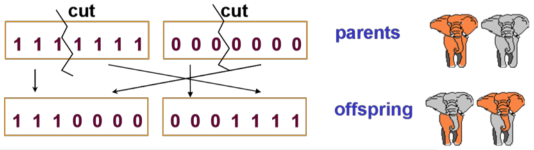
Crossover

- Population is diverse early in the process, this causes the crossover to be large in the beginning.
- However it will settle down in future generations.

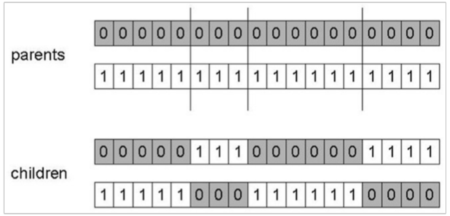


Crossover

- 1-point Crossover



- *n*-Point Crossover



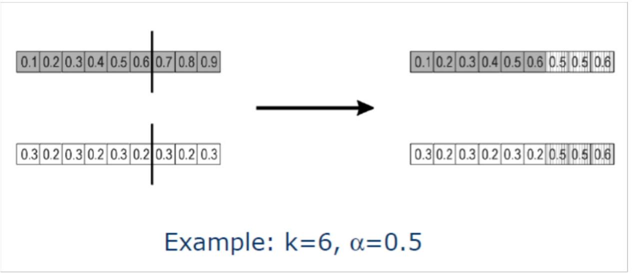
- Simple Arithmetic Crossover

Child 1:

$\langle x_1, \ldots, x_k, \ \alpha y_{k+1} + (1-\alpha) x_{k+1}, \ldots, \alpha y_n + (1-\alpha) x_n \rangle$

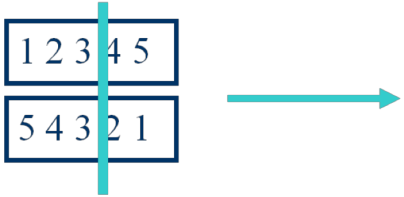
Child 2:

$\langle y_1, \ldots, y_k, \ \alpha x_{k+1} + (1-\alpha) y_{k+1}, \ldots, \ \alpha x_n + (1-\alpha) y_n \rangle$



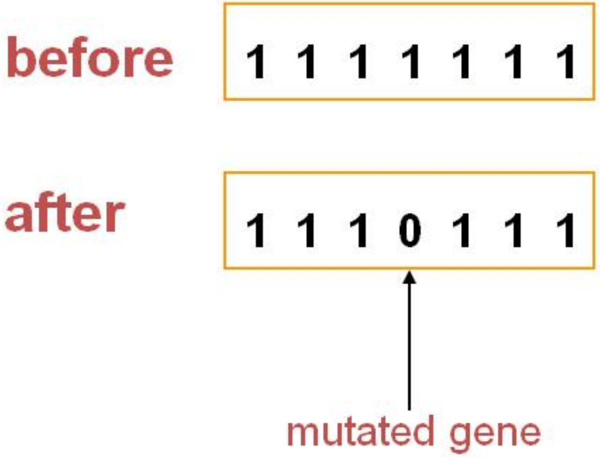
Crossover for Permutations

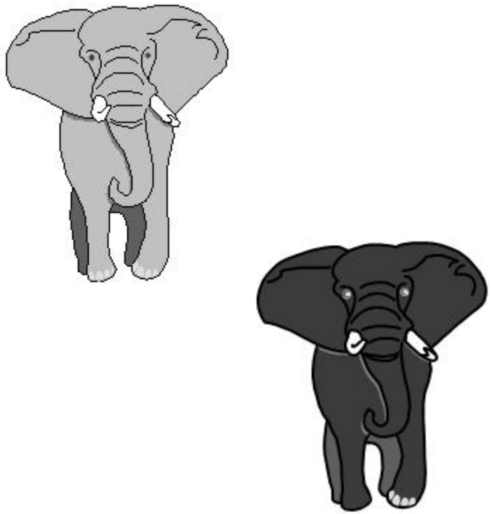
- "Normal" crossover operators will often lead to inadmissible solutions.



- Many specialised operators have been devised which focus on combining ***order*** or ***adjacency*** information from the two parents.
- Most commonly used operators:
  - For Adjacency-type Problems (e.g. TSP)
    - Partially Mapped Crossover (PMX)
    - Edge Crossover
  - For Order-type Problems (e.g. Job Shop Scheduling)
    - Order Crossover
    - Cycle Crossover

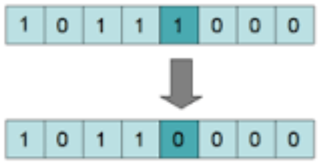
Mutation



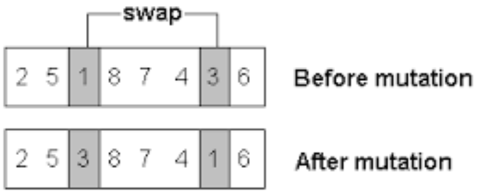


Mutation

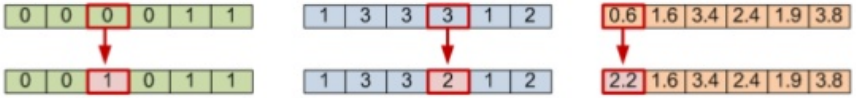
- Flip Mutation



- Swap Mutation



- Uniform Mutation



changes the value of chosen gene with uniform random value selected between upper and lower bound for that gene

- Gaussian Mutation (real coding)

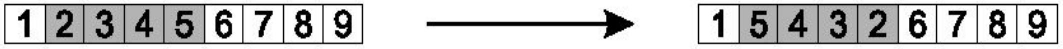
$$x' = x + N(0, \sigma)$$

Mutation for Permutations

- Insert Mutation

- Scramble Mutation

- Inversion Mutation



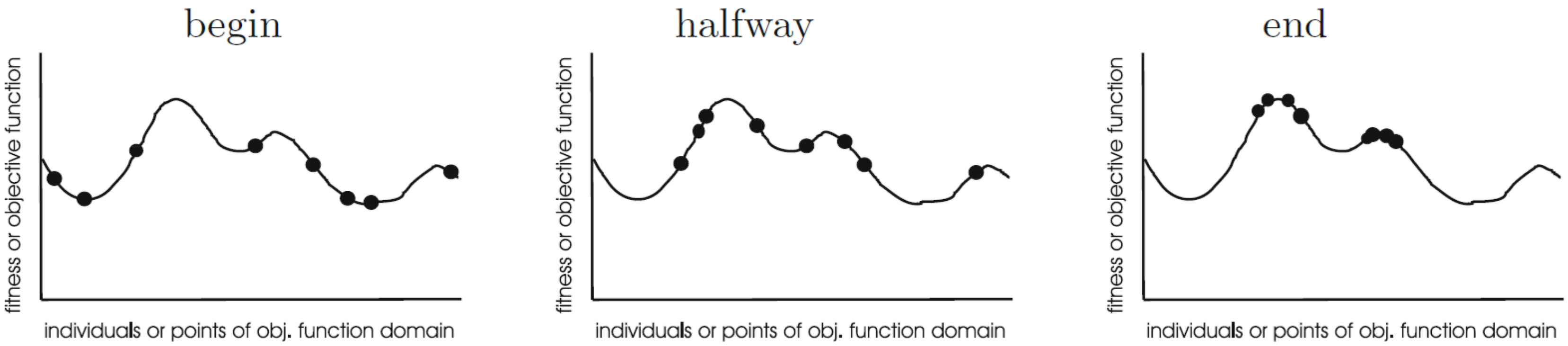
Initialization

- Initialization is kept simple in most EA applications.
  - the first population is seeded by randomly generated individuals.
- In principle, problem-specific heuristics can be used in this step, to create an initial population with higher fitness.
- Whether this is worth the extra computational effort, or not, very much depends on the application at hand.

Termination

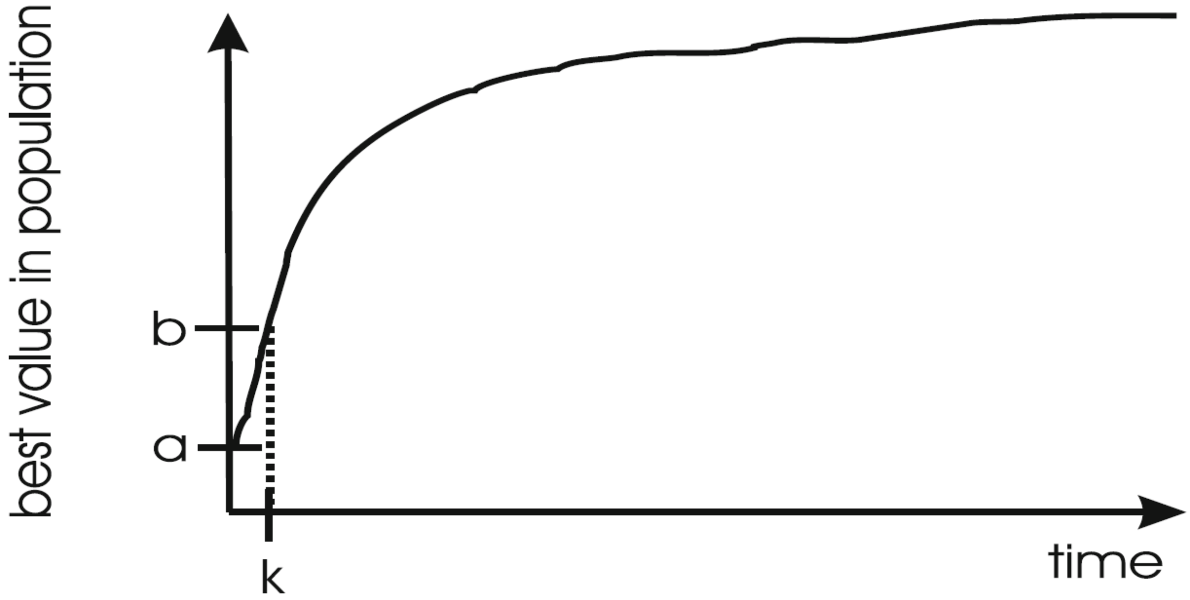
- Termination condition checked every generation
  - Reaching some (known/hoped for) fitness.
  - Reaching some maximum allowed number of generations.
  - Reaching some minimum level of diversity.
  - Reaching some specified number of generations without fitness improvement.

How EA Works?



Typical progress of an EA illustrated in terms of population distribution.

Anytime Behaviour of EA

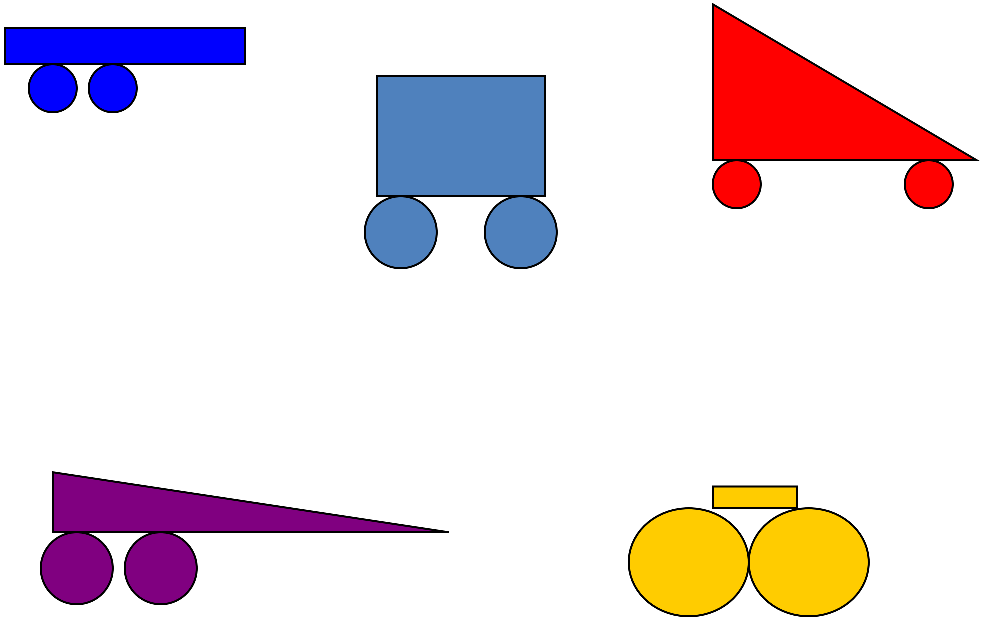


Why heuristic initialization might not be worth additional effort? Level a shows the best fitness in a randomly initialized population; level b belongs to heuristic initialization?

Outline

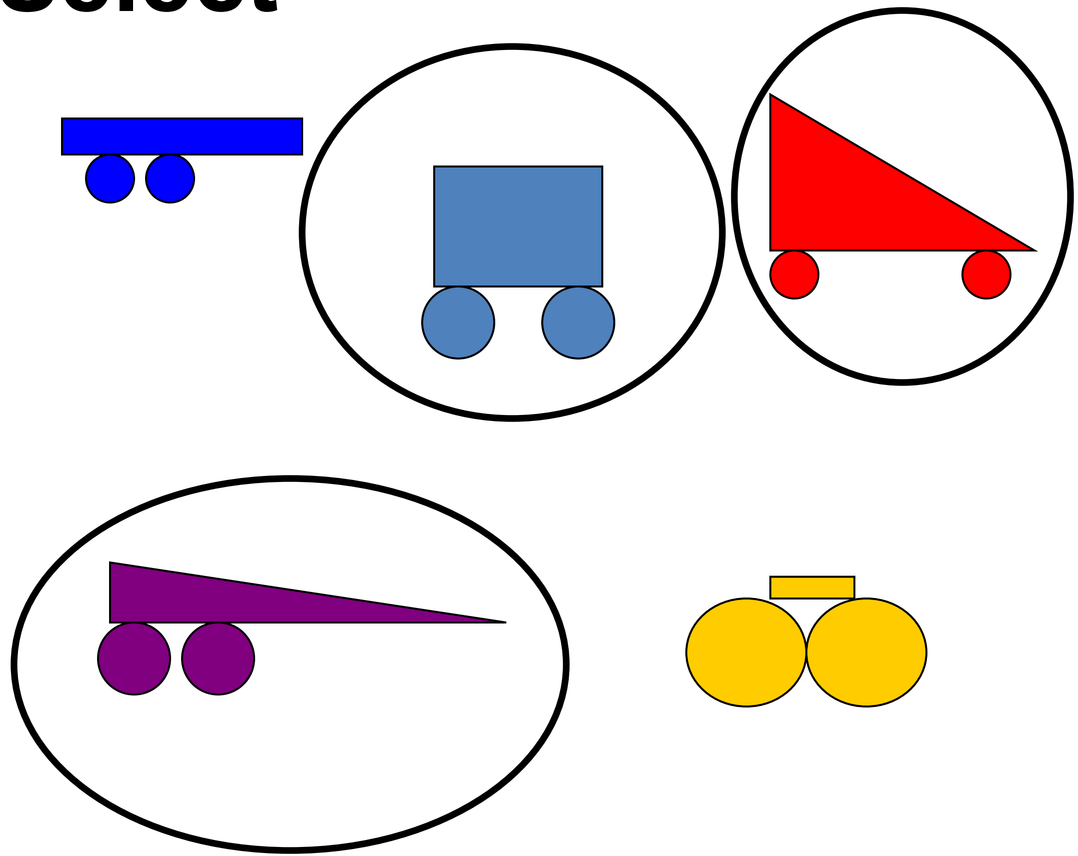
- Inspiration
- Biological Evolution
- Theory Behind
  - Darwinian Evolution
  - Influence by Malthus
- Evolutionary Cycle
- Classic Example: Genetic Algorithms (GAs)
- Components of GA
- **Example and Simulation**

Initial Population

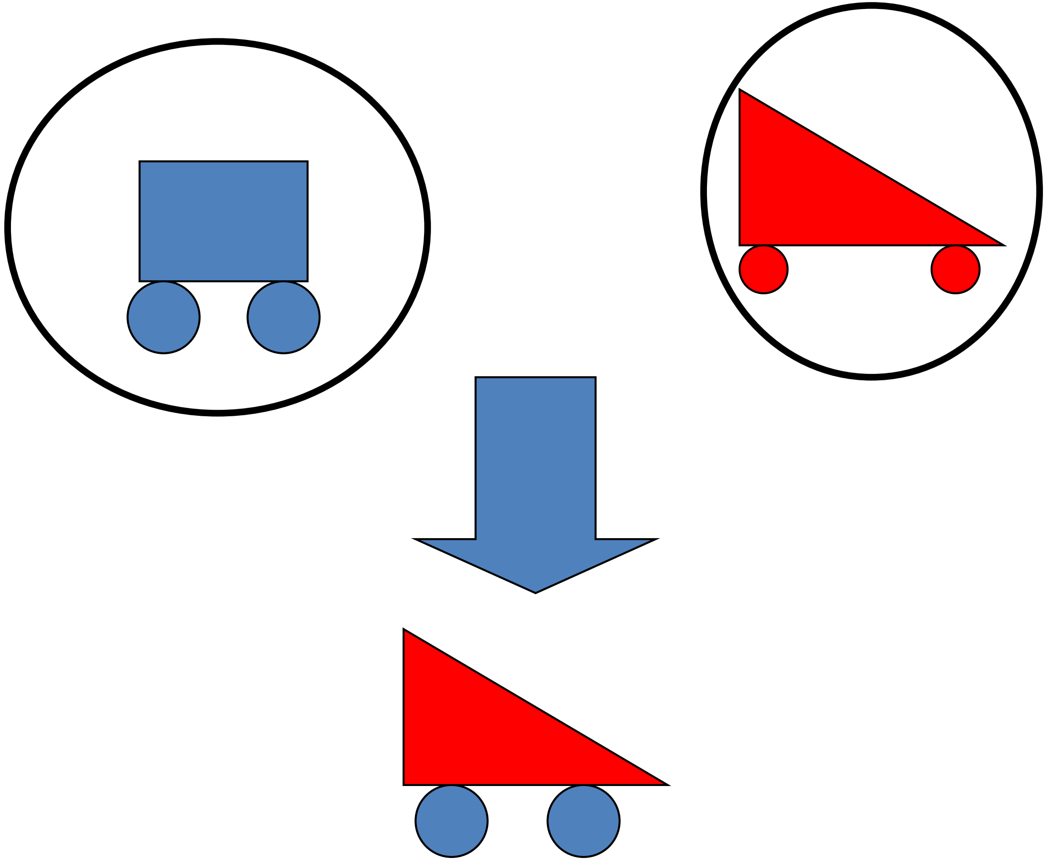


Source: http://www.macs.hw.ac.uk/~dwcorne/Teaching/bic.html

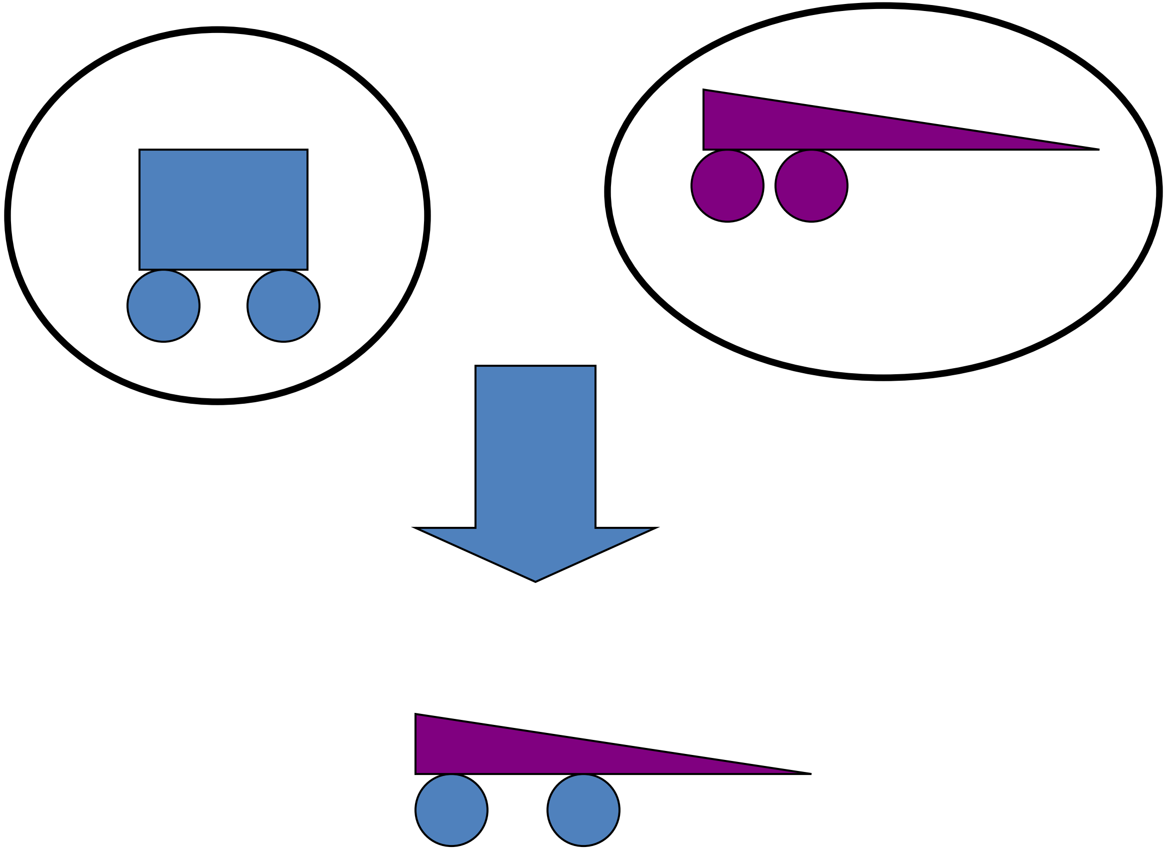
Select



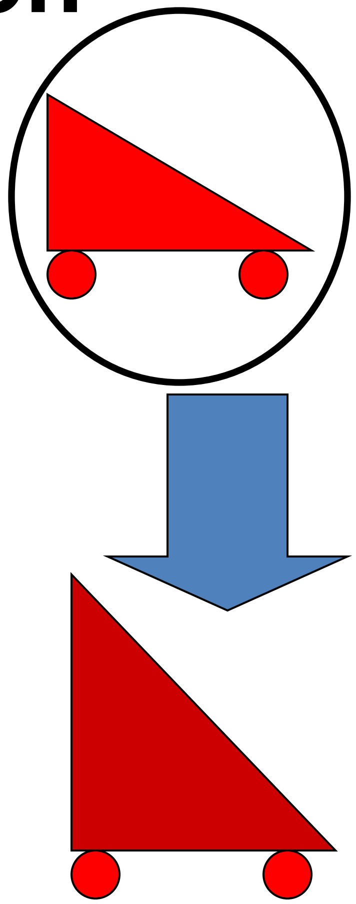
Crossover



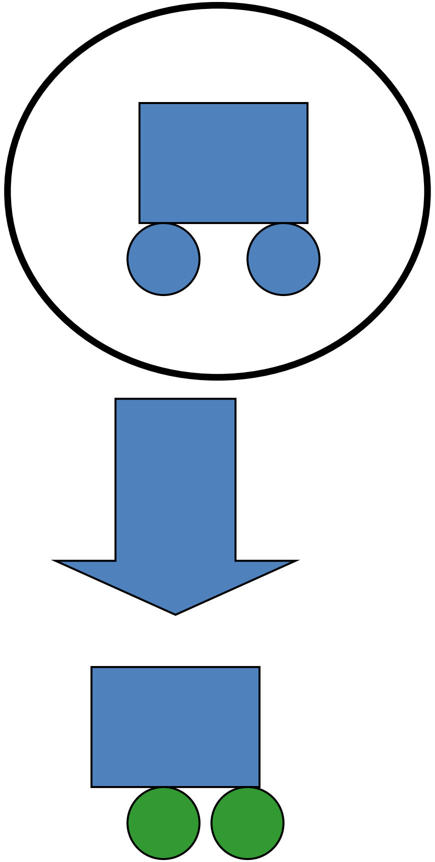
Another Crossover



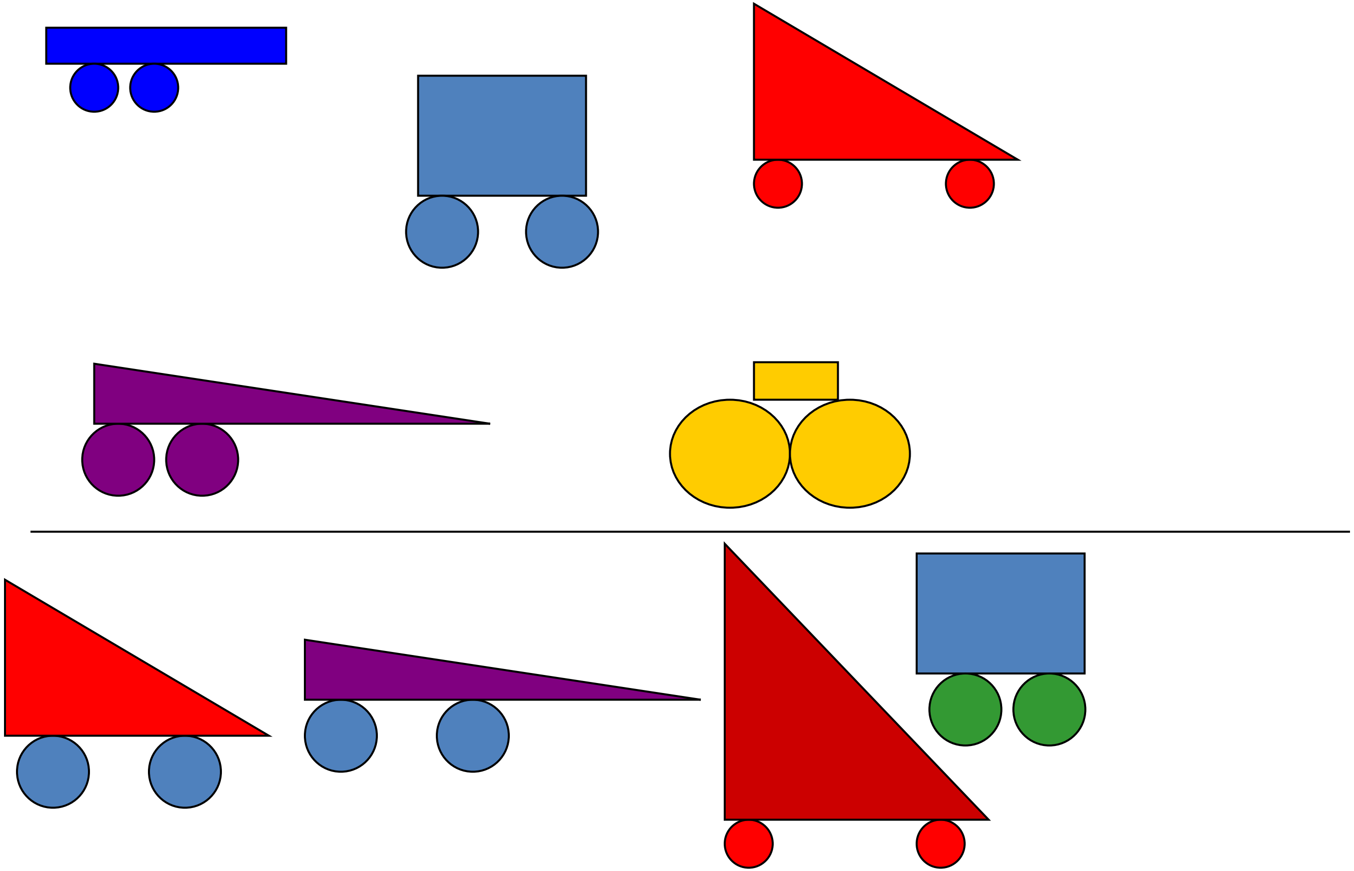
A Mutation



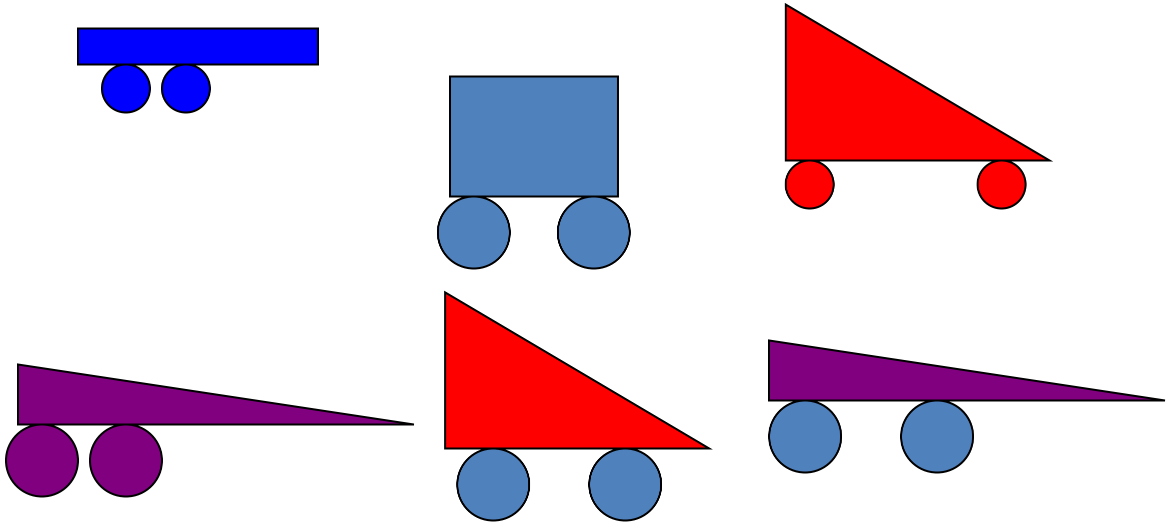
Another Mutation



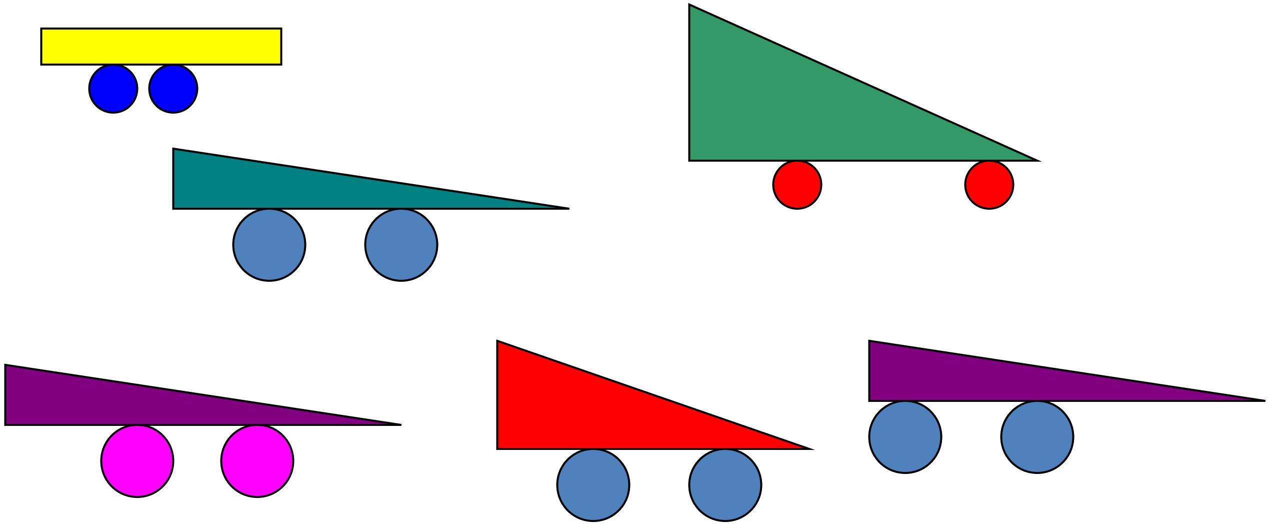
Old Population + Children



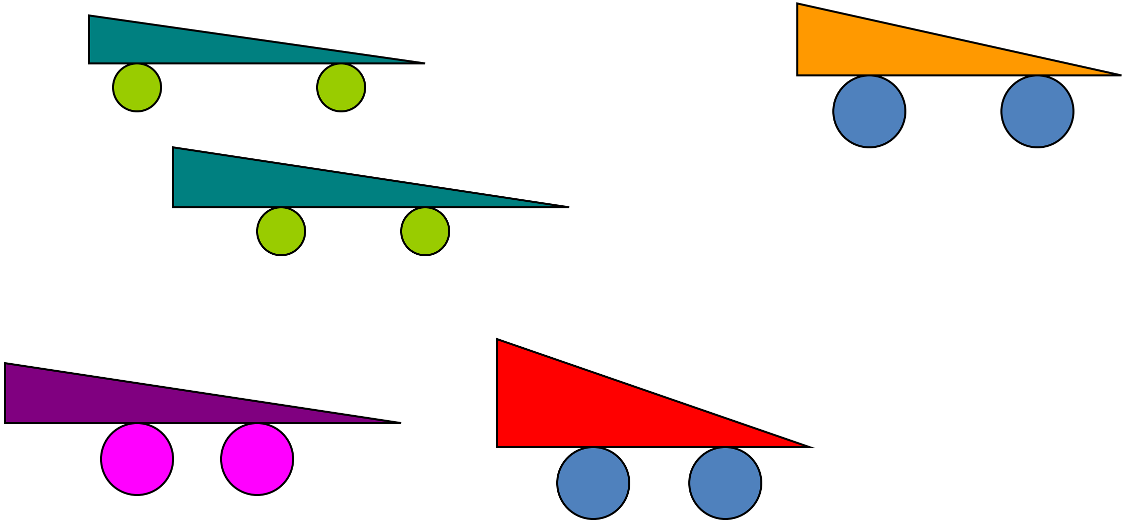
New Population: Generation 2



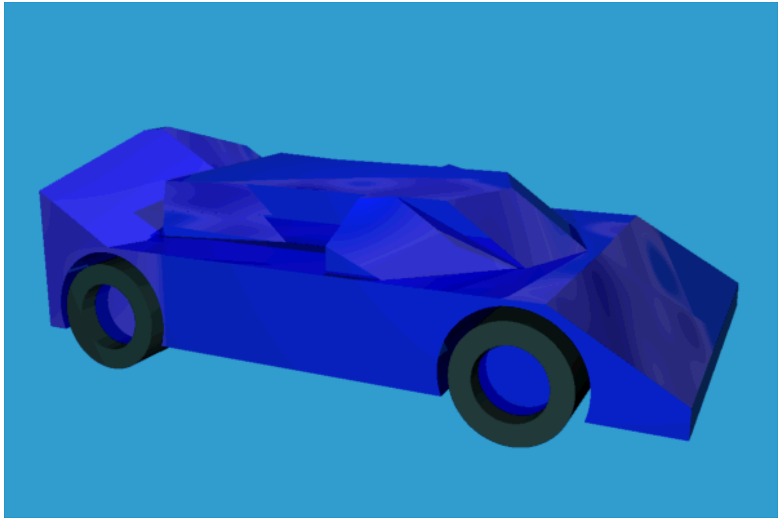
Generation 3



Generation 4, etc …

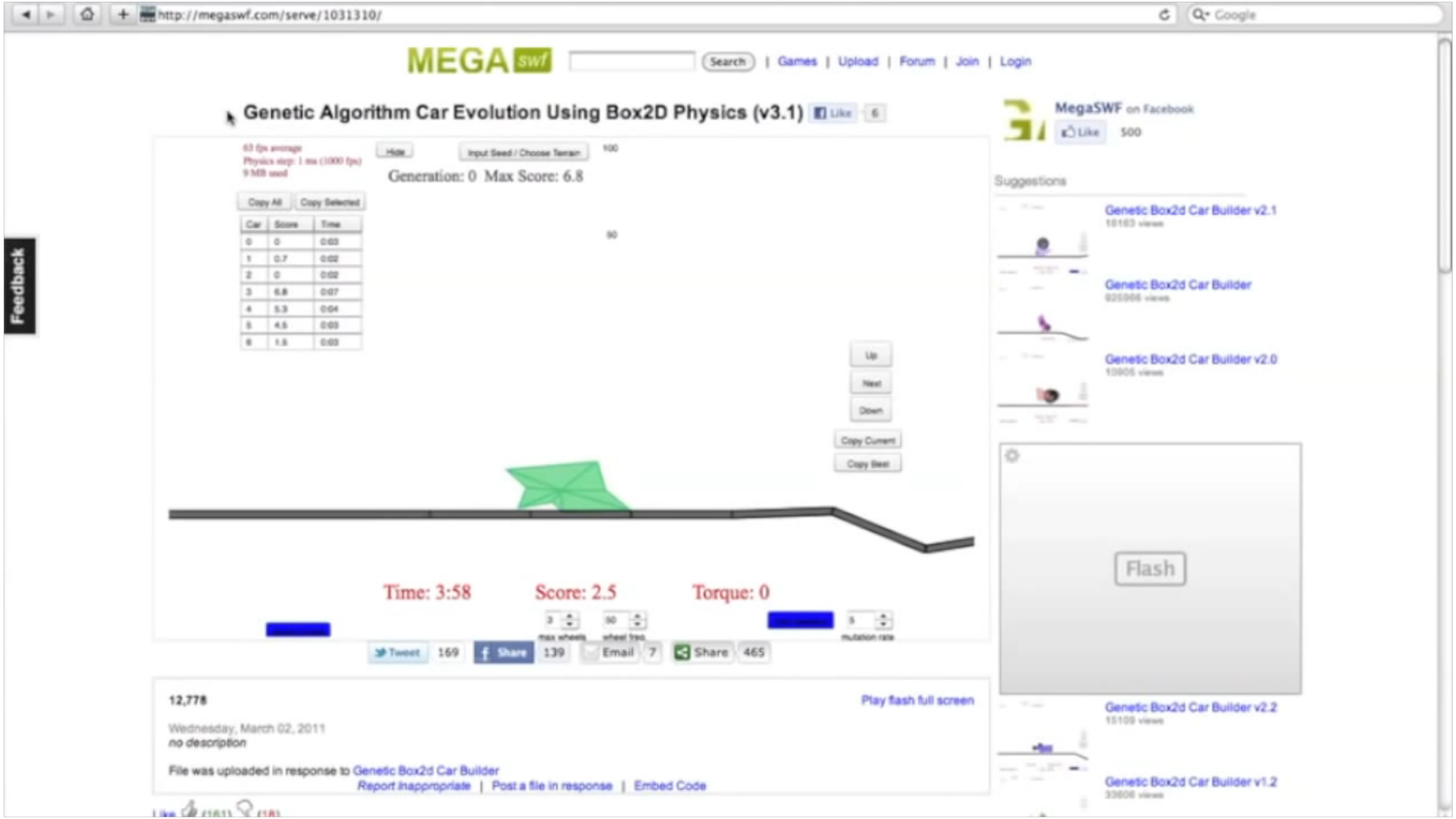


Bentley's Thesis Work



- Fixed wheel positions, constrained bounding area
- Chromosome is a series of slices
- Fitnesses evaluated via a simple airflow simulation

Simulation GA



Required Reading:

- D. Floreano and C. Mattiussi Bio-inspired Artficial Intelligence, Theories, Methods and Technologies,, MIT Press.
  - Chapter 1: 1.1 - 1.9

- A.E. Eiben and J.E. Smith, *Introduction to Evolutionary Computing*, Second Ed. (2015), Natural Computing Series, DOI *10.1007/978-3-662-44874-8_1*
  - Chapter 3: 3.3, 3,6
  - Chapter 4: 4.2, 4.3, 4.4, 4.5, 4.6