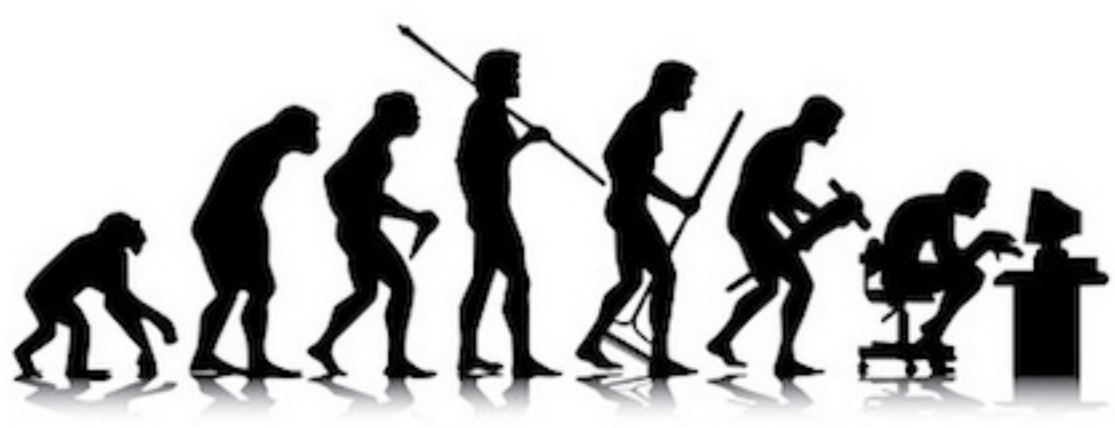
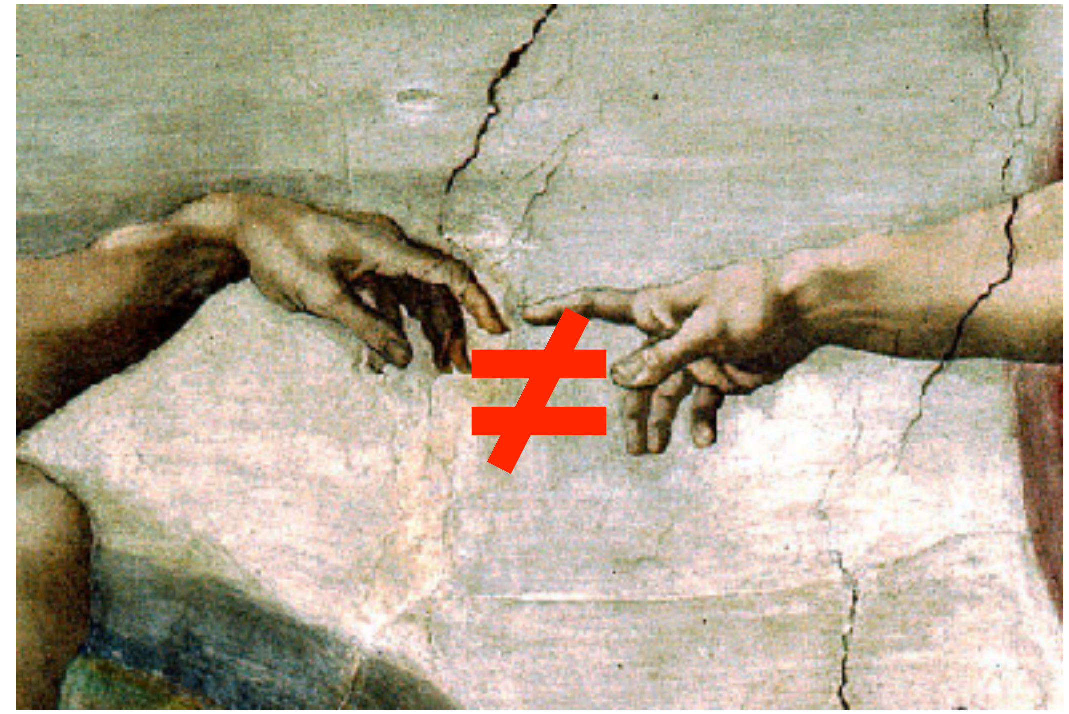
Evolutionary Computation

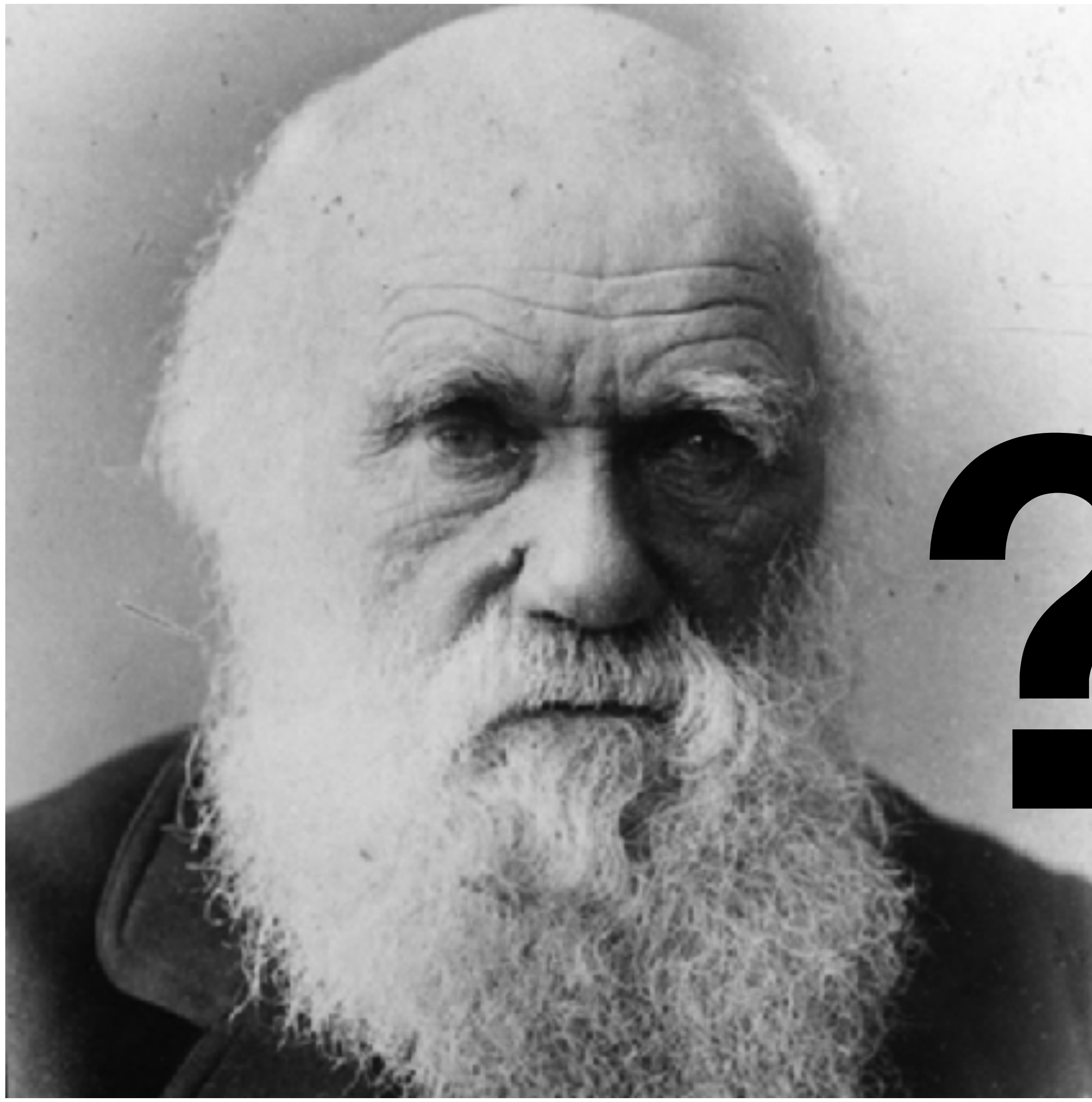
School of Computing, KAIST
Shin Yoo

(with slides borrowed from Jinsuk Lim @ COINSE)

What is **evolution**?



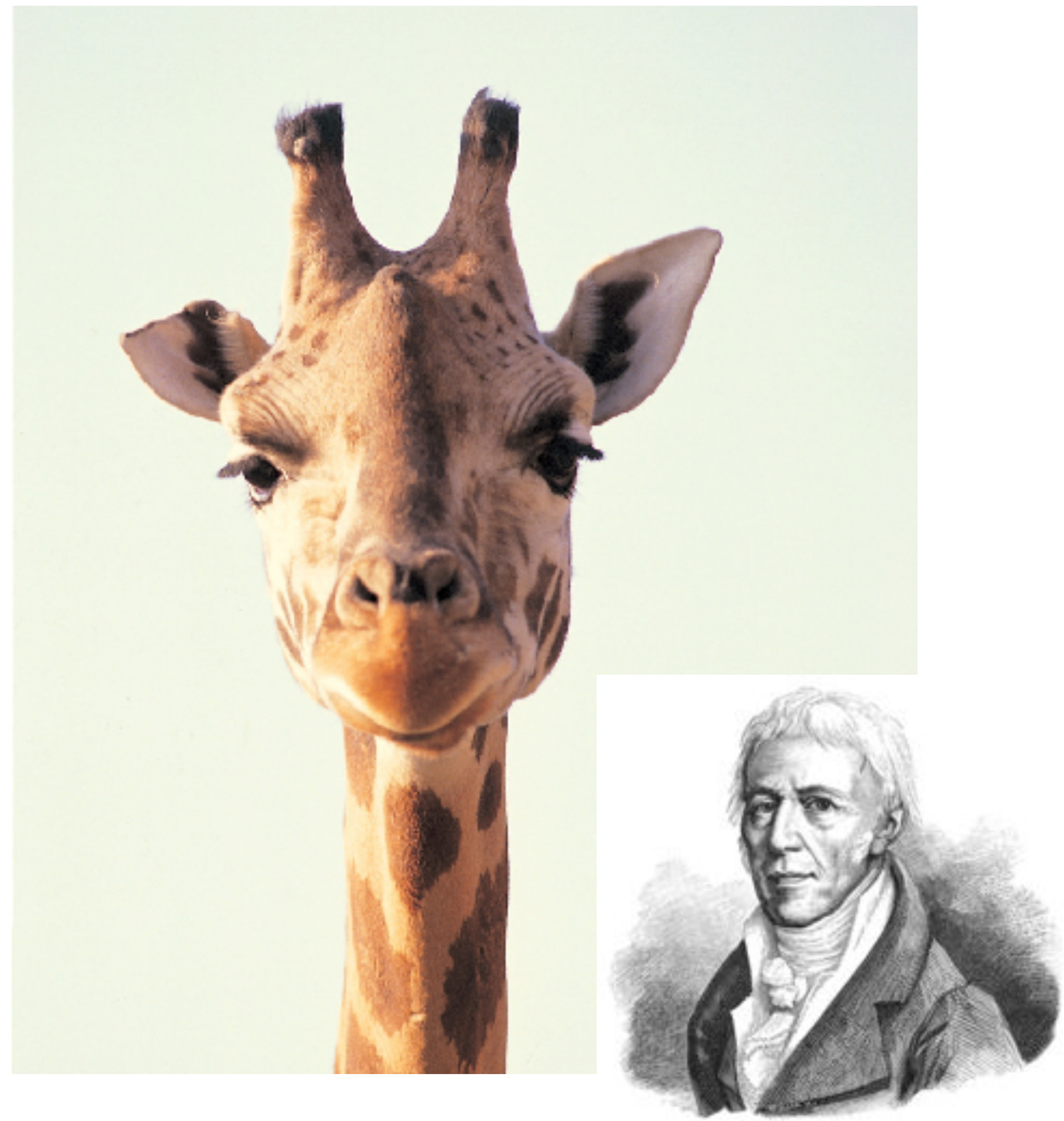




??

Lamarckism (用不用說)

- “Heritability of acquired characteristics”
- During lifetime, an organism will adapt to its environment and acquire certain traits.
- These traits are inherited to the offspring.
- Eventually, the species changes in the direction of adaptation.



Is it correct?

- Does **NOT** explain the majority of what we call evolution; has been criticised for a long time.

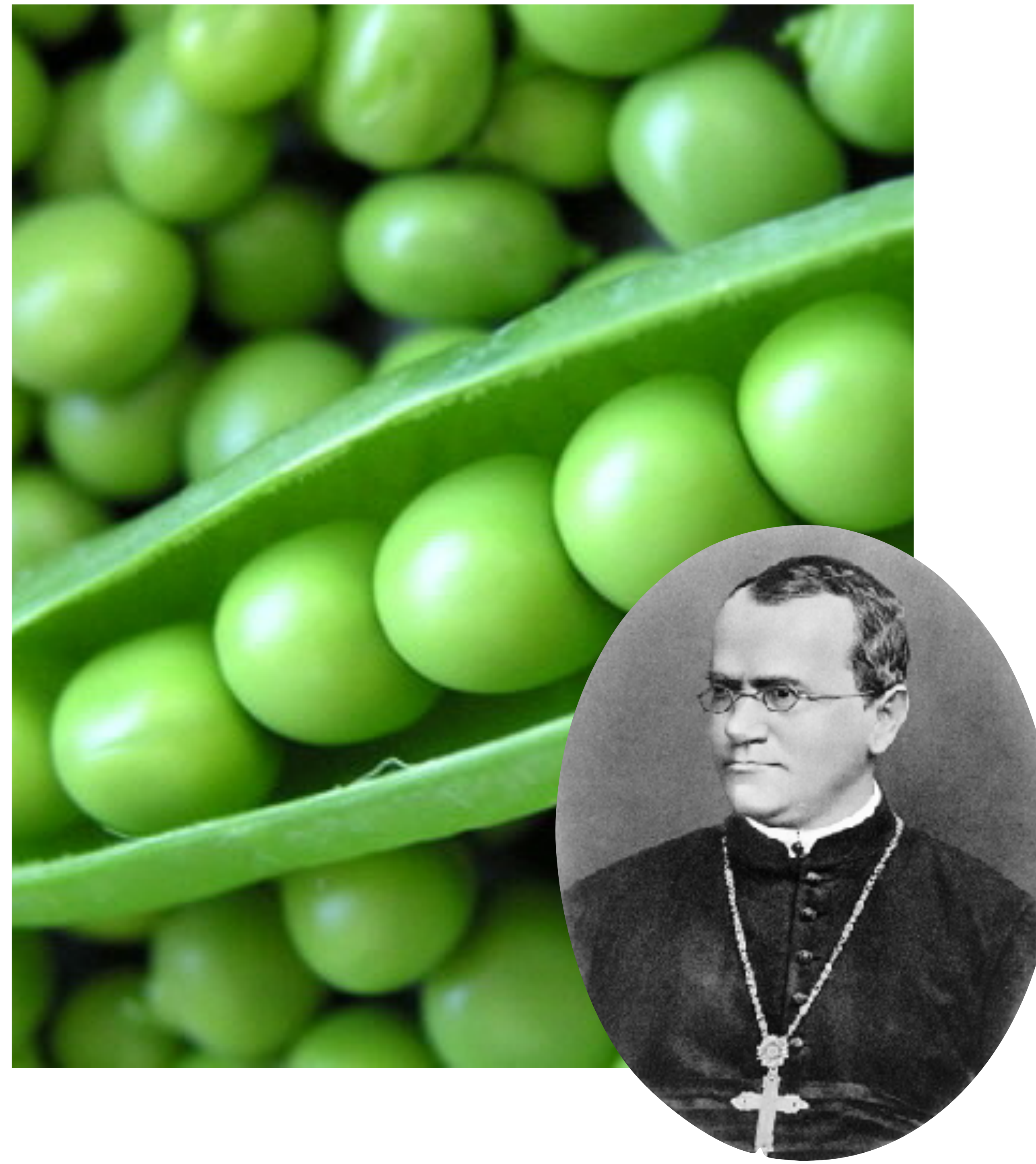
- Interestingly, some people - such as George Bernard Shaw - thought that Lamarckism was more humane and optimistic than Darwinism: individuals can try to **develop a new habit** that are beneficial!

- Epigenetics: trait variations that are caused by environments (!)

    - Renewed interest, but still a topic of a big debate.

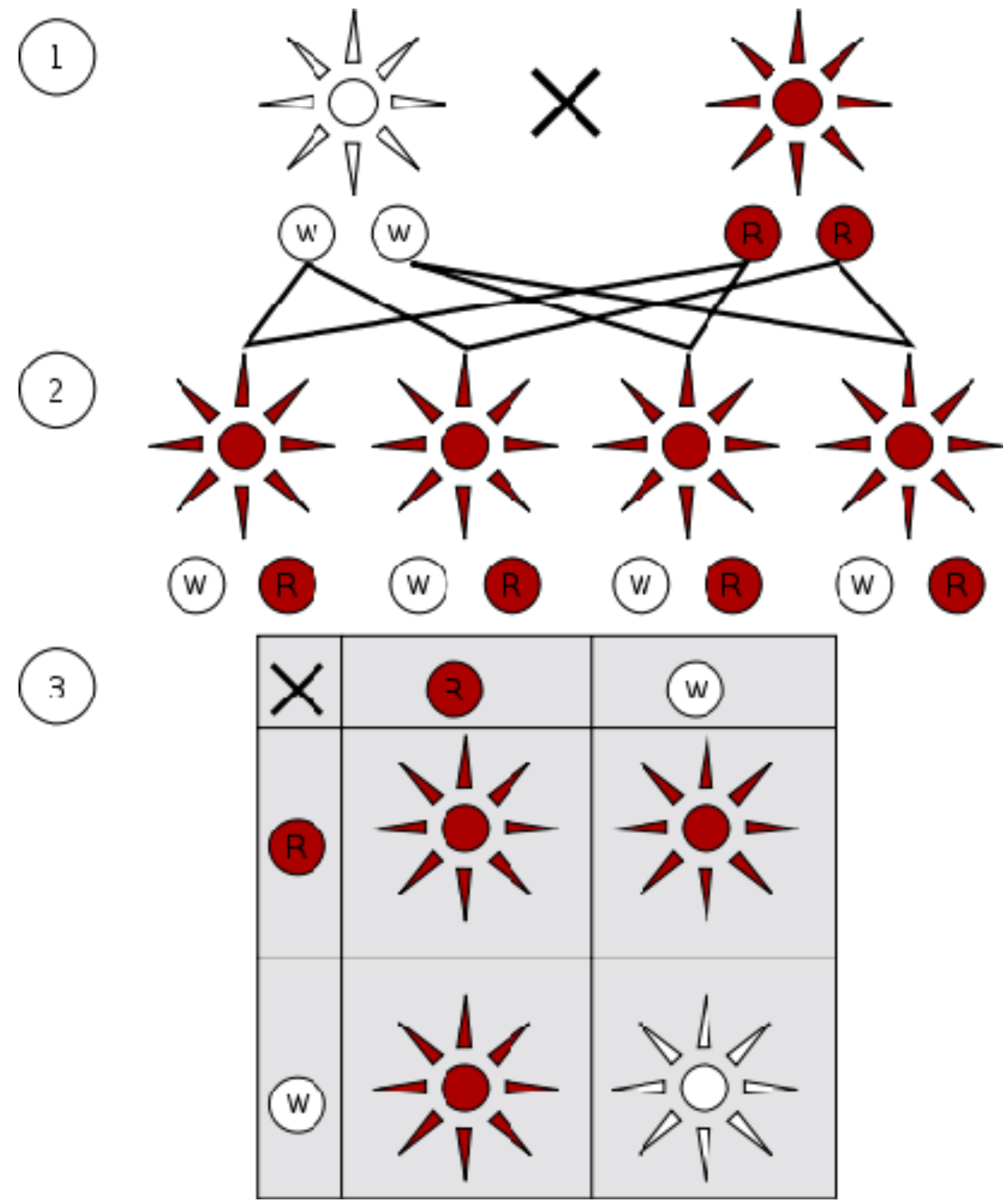
Mendelism

- Hereditary “unit” (he called them “factors”, now we know them as “genes”)
- Explained the mechanism of inheritance.



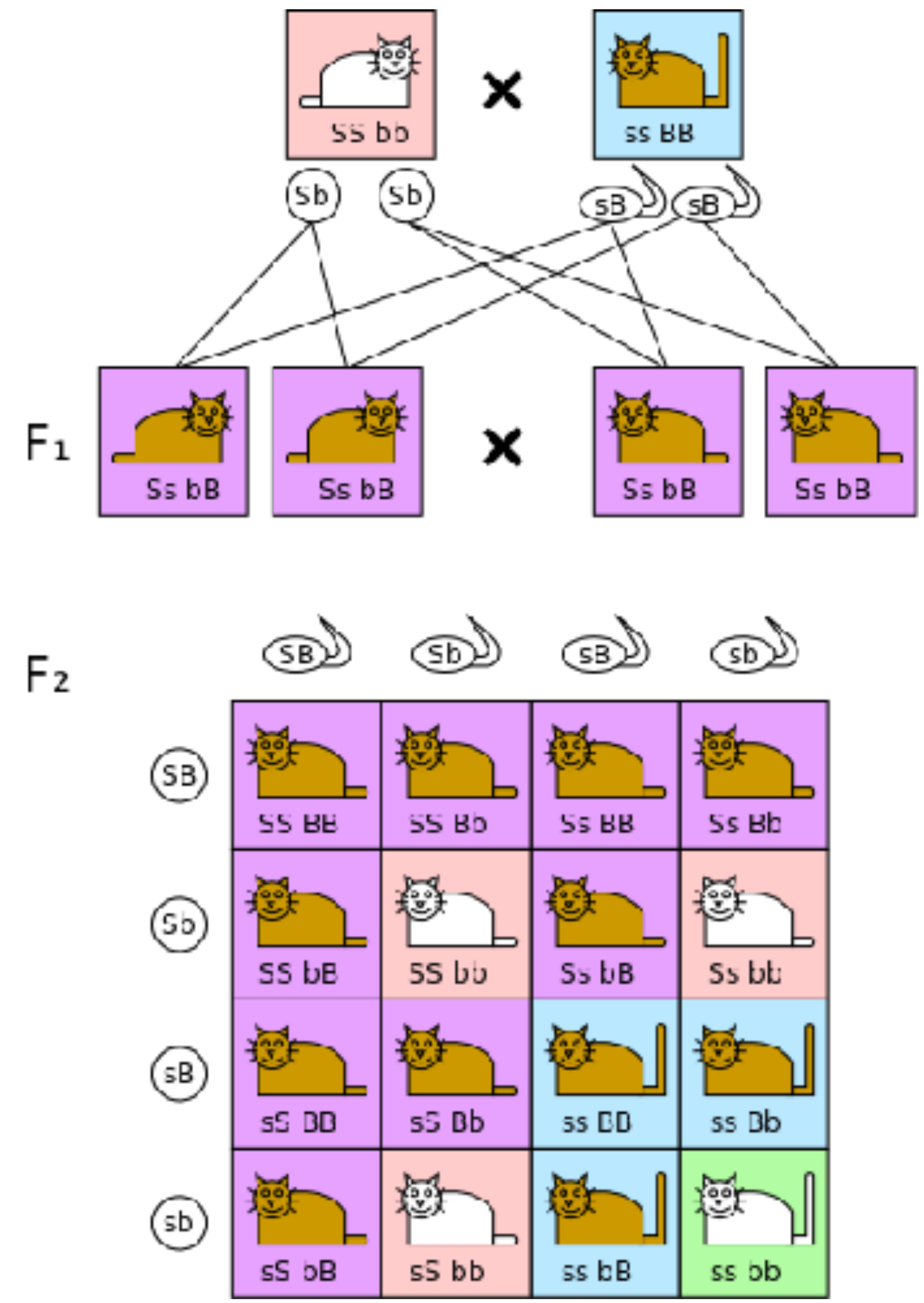
Law of Segregation

- Individuals contain a pair of alleles. During reproduction, the pair is separated; a child inherits one of these alleles, randomly chosen.



Law of Independent Assortment

- Informally: separate genes for separate traits are passed independently from parents to offsprings.
    - Colour and tail length are independent; any combination is possible.



Law of Dominance

- Recessive alleles will be masked by dominant alleles.
- Little evidence that tongue-rolling is a dominant Mendelian trait though.
  - Martin, N. G. No evidence for a genetic basis of tongue rolling or hand clasping. J. Hered. 66: 179-180, 1975.

Darwinism

- An attempt to theorise the emergence of new species.
- It should be noted that Alfred Wallace independently arrived at a very similar conclusion at the same time. Wallace's paper prompted Darwin to publish "On the Origin of Species".

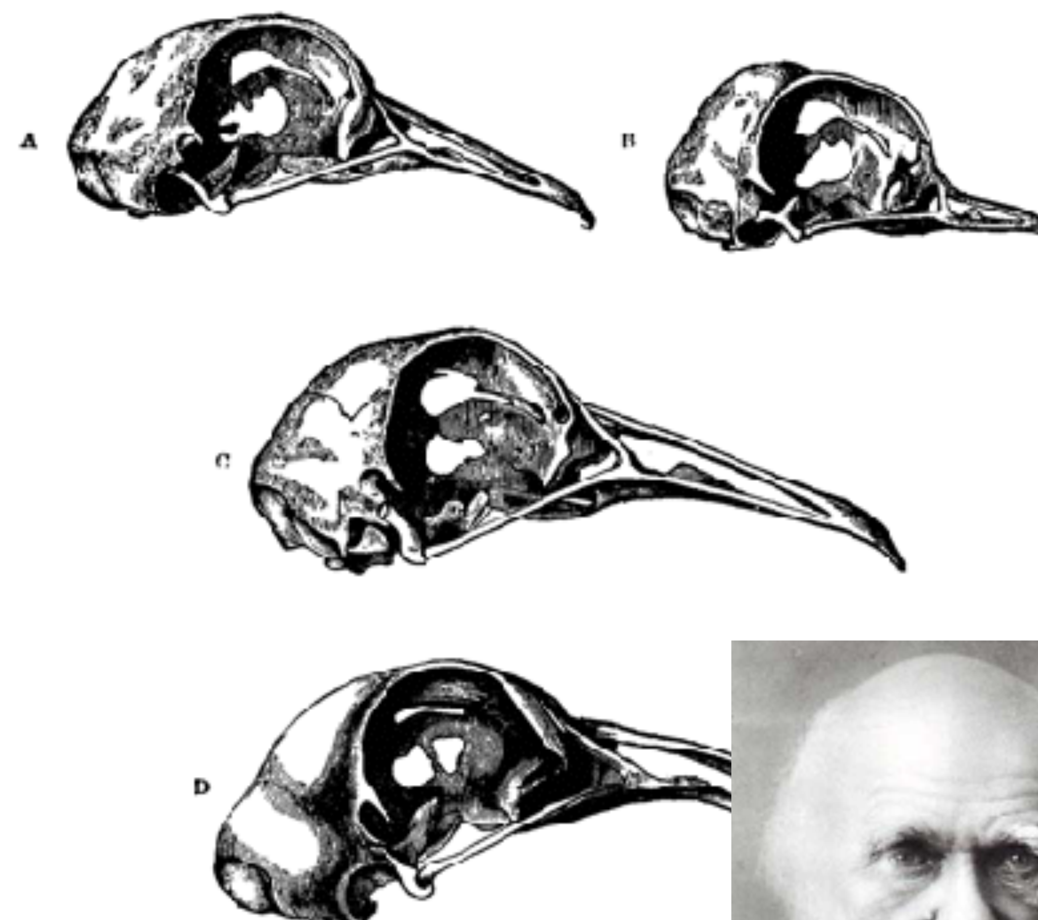
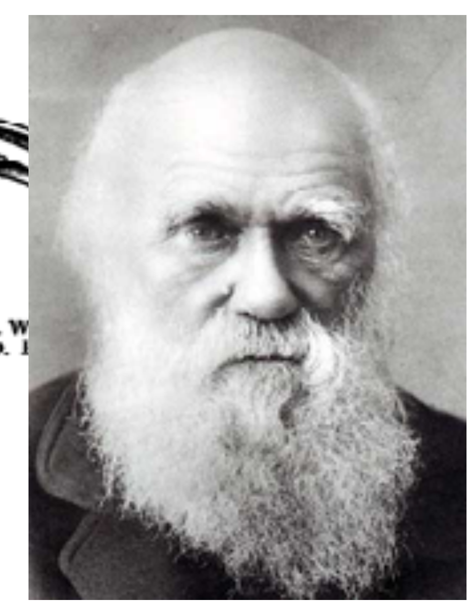


Fig. 24.—Skulls of Pigeons viewed laterally, of natural size. A. W
livia. B. Short-faced Tumbler. C. English Carrier. D. I



What is it exactly?

- If all offspring survived to reproduce the population would grow (fact).
- Despite periodic fluctuations, populations remain roughly the same size (fact).
- Resources are limited and are relatively stable over time (fact).
- A struggle for survival ensues (inference).
- Individuals in a population vary significantly from one another (fact).
- Much of this variation is heritable (fact).
- Individuals less suited to the environment are less likely to survive and less likely to reproduce; individuals more suited to the environment are more likely to survive and more likely to reproduce and leave their heritable traits to future generations, which produces the process of **natural selection** (inference).
- This slowly effected process results in populations changing to adapt to their environments, and ultimately, these variations accumulate over time to form new species (inference).

Does it explain everything?

- Genetic drift: nature does not select, it merely samples randomly, resulting in frequency of specific alleles.
  - A much more neutral view on evolution.
- Both work at the same time.

Genotype vs. Phenotype

- **Genotype**: that part of the genetic material that determines a specific characteristic of an individual
- **Phenotype**: the characteristic manifested by a specific genotype

Genotype vs. Phenotype

- For example, 0-1 knapsack problem: given N items with individual weights and values, fill a knapsack that can hold X kilograms with the maximum value possible.
- Genotype: a bit string of length N; 1 if corresponding item is chosen, 0 if not.
- Phenotype: the weight and the value of the filled knapsack.

Evolutionary Pressure

- Also known as selection pressure: anything that affects the reproductive success rate exerts **evolutionary pressure**.
- One critical link in Darwinian evolution: fitter individuals are assumed to have better reproductive success rate.

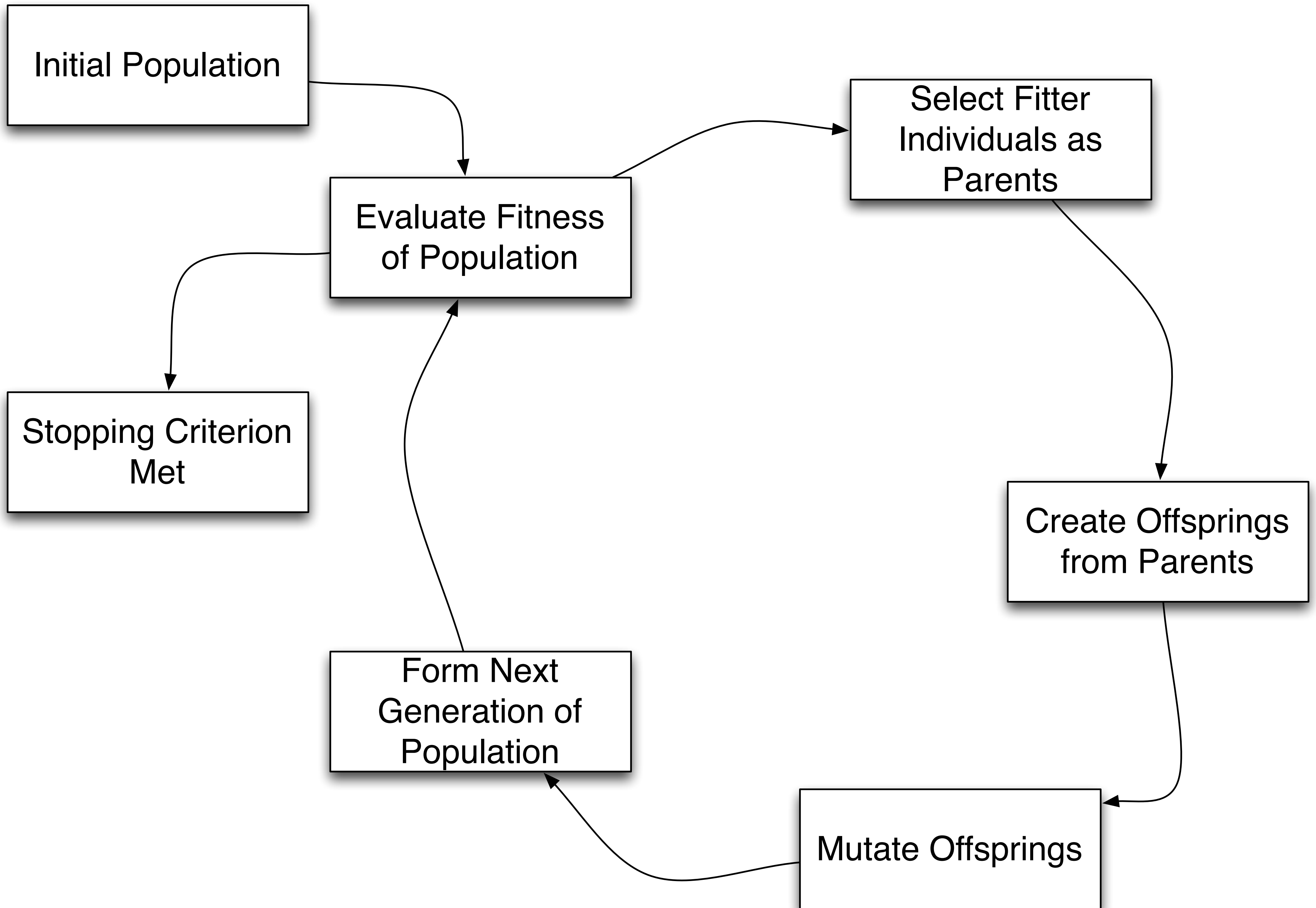
Adaptation vs. Optimisation

- Does nature adapt or optimise?
- Experiments with artificial co-evolution often result in stale and stagnant populations: they co-adapt, rather than doing arms-race.
- Optimisation through Darwinian evolution may be a purely artificial concept.

Okay, now back to algorithms. Let’s start with Genetic Algorithms.

Artificial Evolution

- Artificial evolution as computation and, especially, as optimisation.
- Apply selection pressure so that a species (a population) evolves towards better fitness values.
- We have to emulate the entire evolutionary loop.
- Remember: exploitation vs. exploration.
    - Too much pressure: premature convergence.
    - Too little pressure: search goes nowhere.


Initial Population
Evaluate Fitness of Population
Select Fitter Individuals as Parents
Create Offsprings from Parents
Mutate Offsprings
Form Next Generation of Population
Stopping Criterion Met

Suppose we break a 6 digit numeric password with GA

- Let's assume that we have a tool that tells us how many digits are correct #yessomewhatunrealistic

Password: 893714

1<u>93</u>562

24<u>3</u>690

12<u>3</u>456

1212<u>14</u>

Randomly Generated Initial Population

Suppose we break a 6 digit numeric password with GA

- Let's assume that we have a tool that tells us how many digits are correct #yessomewhatunrealistic

Password: 893714

193562	Score: 2
121214	Score: 2
243690	Score: 1
123456	Score: 1

Evaluation

Suppose we break a 6 digit numeric password with GA

- Let's assume that we have a tool that tells us how many digits are correct #yessomewhatunrealistic

Password: 893714

243690

123456

Choose Parents		Crossover
193562	→	193514
121214	→	121262

Suppose we break a 6 digit numeric password with GA

- Let’s assume that we have a tool that tells us how many digits are correct #yessomewhatunrealistic

Password: 893714

	Choose Parents	Mutation
	193562	**8**93514
243690		
	121214	12**3**262
123456		

Choose Parents

Mutation

Initial Population

- Usually initialised randomly: this introduces the **variance** among individuals.
- We mean phenotype variance. Depending on problems, genotype variance may not always result in phenotype variance.

Selection Operators

- We apply selection operators to the population, to choose two parent individuals.

- This is one of two places where we apply evolutionary pressure: we should make sure that the fitter you are, the more successful you are in terms of reproduction.

- This is also relatively universal - i.e. not dependent on the choice of representation

Fitness Proportional Selection (FPS)

- The probability of selecting an individual is proportional to its **absolute fitness** over the rest of the population.
- Given an individual *i,* its fitness f_i and population size μ,

$$P_{FPS}(i) = \frac{f_i}{sum_{j=1}^{\mu} f_j}$$

Issues with FPS

- Outstanding individuals tend to take over the population quickly, leading to **premature convergence.**
- When fitness values are close together, there is **almost no selection pressure.**

Individual	Fitness for f	Sel. prob. for f	Fitness for $f + 10$	Sel. prob. for $f + 10$	Fitness for $f + 100$	Sel. prob. for $f + 100$
A	1	0.1	11	0.275	101	0.326
B	4	0.4	14	0.35	104	0.335
C	5	0.5	15	0.375	105	0.339
Sum	10	1.0	40	1.0	310	1.0

Selection pressure rapidly falls as fitness is linearly translated...

Improving FPS

- **Windowing:** At each generation, fitness is transformed by subtracting $\beta(t)$ from the raw fitness.
- Usually, $\beta(t)$ is the minimum fitness of the current population, *i.e.,*

$$\beta(t) = min_{y \in P} f(y)$$

- **Sigma scaling:** $f'(x) = max(f(x) - (\bar{f} - c \cdot \sigma_f), 0)$

$\bar{f}, \sigma_f, c$ are mean, standard deviation and hyperparameter (usually 2)

Ranking Selection

- Sort the population by fitness and allocate selection probabilities **according to the individuals' rank.**
- Maintains constant selection pressure, as opposed to FPS.
- Given a population of μ, the best individual is ranked $\mu - 1$ and the worst *0*.
- **Linear ranking** vs **Exponential ranking**

Linear ranking

- Parameterised by a value $s \ (1 \leq s \leq 2)$

$$P_{linear}(i) = \frac{2-s}{\mu} + \frac{i(s-1)}{\sum_{j=0}^{\mu} j}$$

Individual	Fitness	Rank	P_{selFP}	$P_{selLR} \quad (s=2)$	$P_{selLR} \quad (s=1.5)$
A	1	0	0.1	0	0.167
B	4	1	0.4	0.33	0.33
C	5	2	0.5	0.67	0.5
Sum	10		1.0	1.0	1.0

FPS versus Linear Ranking

Exponential Ranking

- Exponential ranking is used for greater selection pressure.

$$P_{exp}(i) = \frac{1 - e^{-i}}{\sum_{j=0}^{\mu} 1 - e^{-j}}$$

Sampling from the selection probabilities

- How to sample individuals according to the selection probabilities? (FPS or ranking selection)

- **Roulette Wheel Sampling**
- **Stochastic Universal Sampling (SUS)**
- **Tournament Selection**
- **Overselection** (refer to book)

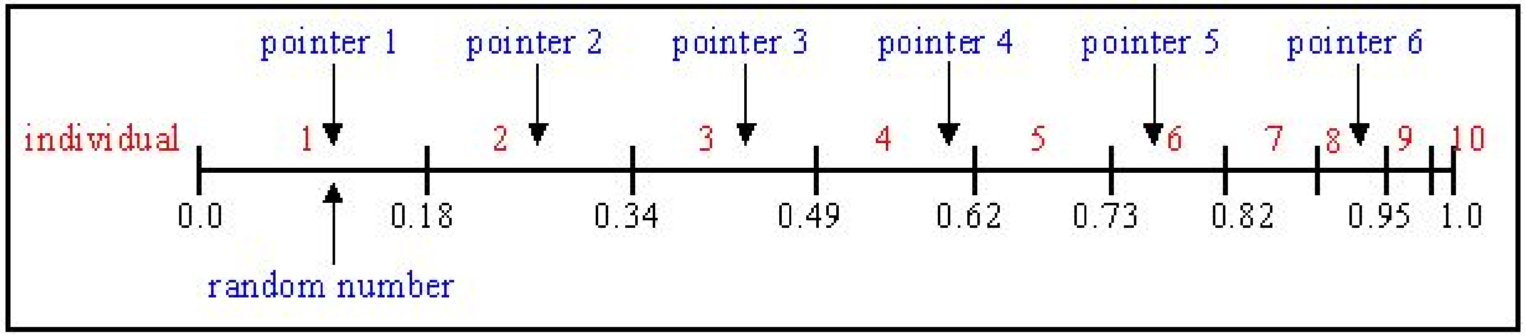
Roulette Wheel Sampling

- Given some order over the population from *1* to μ, compute the **cumulative probabilities** $[a_1, a_2, ..., a_\mu]$ and:

```
BEGIN
  /*  Given the cumulative probability distribution a */
  /*  and assuming we wish to select λ members of the mating pool */
  set current_member = 1;
  WHILE ( current_member ≤ λ ) DO
    Pick a random value r uniformly from [0, 1];
    set i = 1;
    WHILE (  a_i < r ) DO
      set i = i + 1;
    OD
    set mating_pool[current_member] = parents[i];
    set current_member = current_member + 1;
  OD
END
```

Intuitively, each individual is assigned with roulette area whose size corresponds to its selection probability: then spin the roulette to select one sample.

Stochastic Universal Sampling




```
BEGIN
  /*  Given the cumulative probability distribution a */
  /*  and assuming we wish to select λ members of the mating pool */
  set current_member = i = 1;
  Pick a random value r uniformly from [0, 1/λ];
  WHILE ( current_member ≤ λ ) DO
    WHILE (  r ≤ a[i] ) DO
      set mating_pool[current_member] = parents[i];
      set r = r + 1/λ;
      set current_member = current_member + 1;
    OD
    set i = i + 1;
  OD
END
```

When more than one sample is required, SUS is preferred. If we are sampling N individuals, think of this as a roulette wheel with N arms.

Tournament selection

- What if fitnesses cannot be measured on an absolute scale?
- e.g. On evolving game strategies, fitnesses of two strategies can be evaluated only by playing against each other.

- Or if computing selection probabilities is impossible?
- e.g. On a distributed setting, acquiring global knowledge of the fitnesses may not be possible.
- **Tournament selection** solves these problems.

Tournament selection

- Select **k** random individuals from the population (*with* or *without replacement)* and pick the best out of them.
- Add it to the mating pool until full.
- Increasing **k** increases selection pressure.
- The simplest, most widely used selection mechanism.

Crossover Operators

- Offsprings inherit genes from their parents, but not in identical forms.
- Think Mendelian recombination of alleles; since we don't have alleles, we actually recombine the whole genotype.

Crossover Operators

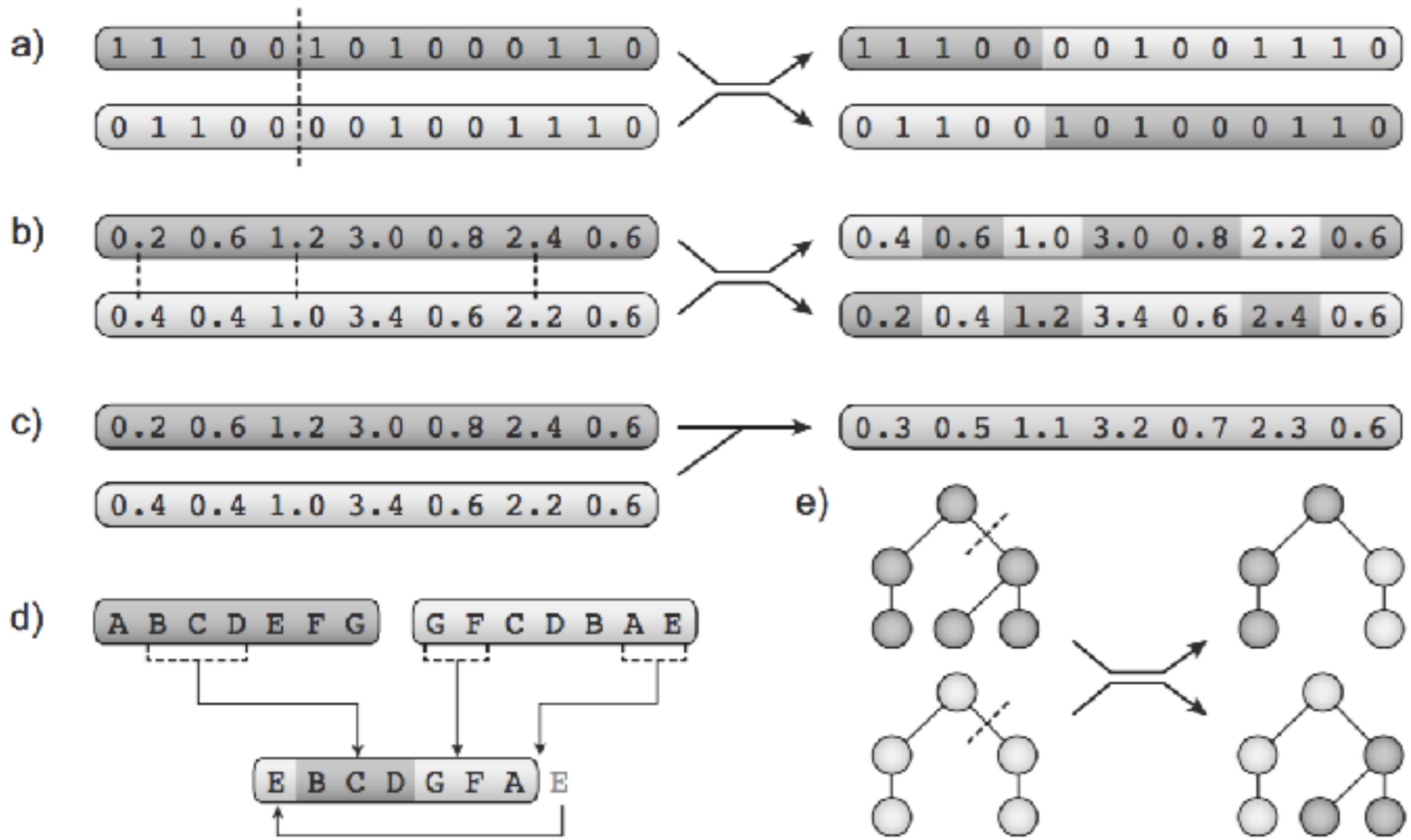


Figure 1.11 Examples of crossover operators. *a*) one-point; *b*) uniform; *c*) arithmetic; *d*) for sequences; *e*) for trees.

(from "Bio-inspired Artificial Intelligence: Theories, Methods, and Technologies" by Dario Floreano and Claudio Mattiussi)

Mutation Operators

- This is, usually, the **only** way **new genetic material** is introduced into the population; without mutation, all we do is recombining the initial population (which was randomly generated).

Mutation Operator

- Small, local modifications to genotypes:
  - single bit-flip
  - adding/subtracting small amount to integers
  - swapping two elements in permutations
  - replacing one node in a tree with a different, compatible type

Generational Selection

- Generational Replacement: the offsprings become the new current population (no parent survives)

- Elitism: maintain M best individuals from the parents' generation (reasons: noisy fitness, too strong mutations, too complicated search space…)

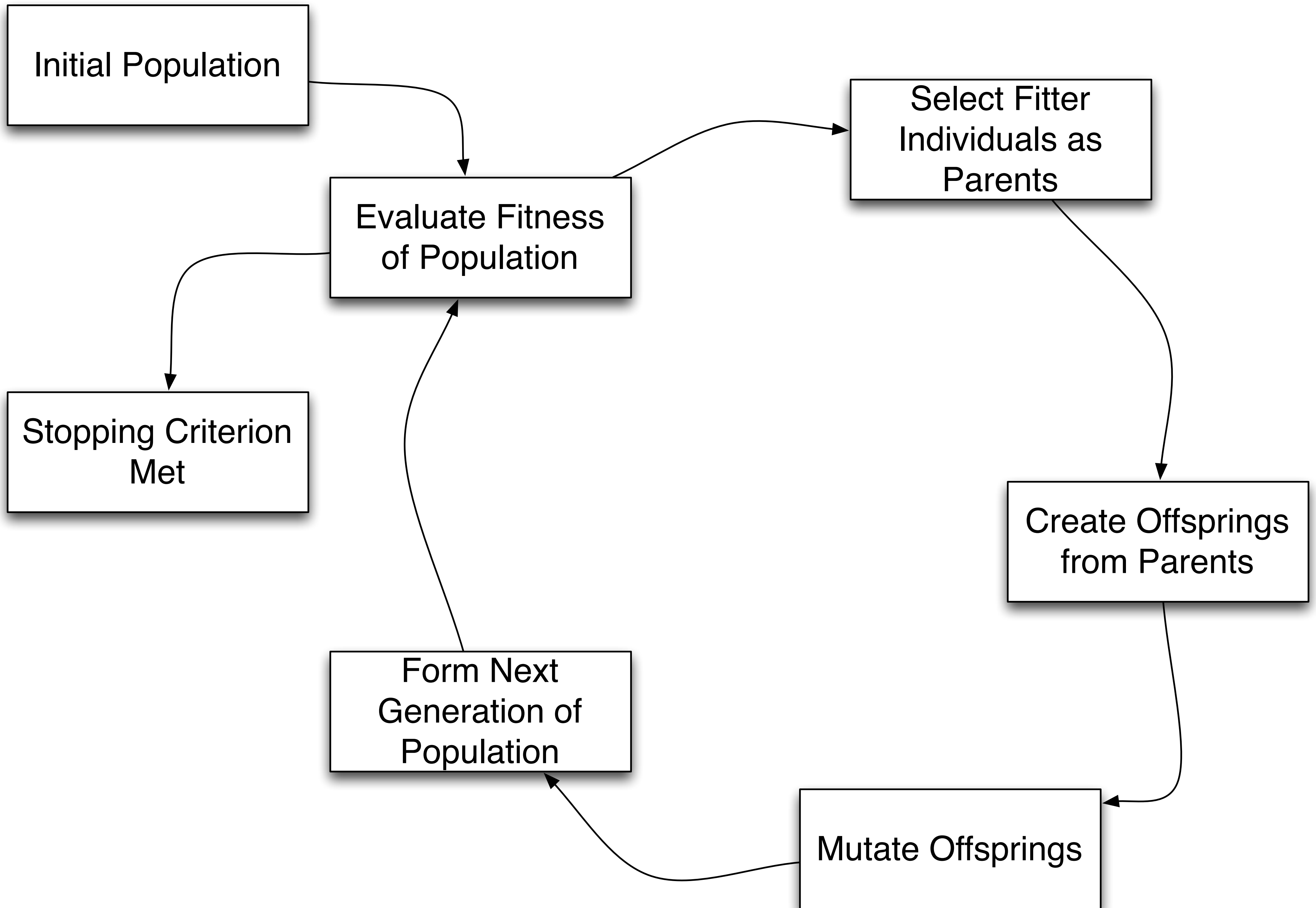
- Gradual Replacement: replace M worst individuals from the parents' generation with M best individuals from the offsprings.

Stopping Criterion

- Deciding one can be hard: these are stochastic algorithms, and you don't know what the global optimum is.
- In reality, one of the following two:
  - Fixed number of fitness evaluations, or
  - When a good enough solution has been found

Parameters

- One weakness of GAs: many parameters to tune, no fixed guideline.
  - Population Size
  - Crossover Rate (usually high, we do want to crossover)
  - Mutation rate (usually low: e.g. `1/N` for 1 bit flip for each bit of length `N` bit string)
  - Elitism: the proportion of parent generation to preserve


Initial Population
Evaluate Fitness of Population
Select Fitter Individuals as Parents
Create Offsprings from Parents
Mutate Offsprings
Form Next Generation of Population
Stopping Criterion Met

Why (or when) does it work?

- Not much theoretic foundation.
- Schema Theory (John Holland, 1975): given genotypes of k symbols with length l, the schemata set is $\{s_0, \ldots, s_k, *\}$ where $*$ means "don't care". There are $(k+1)^l$ schemas.
  - Intuitively, schemas can be thought of as non-consecutive building blocks to the solution.
  - Holland mathematically proved that selective reproduction allows exponentially increasing number of samples of schemas with better-than-average fitness, and exponentially decreasing number of schemas with lower-than-average fitness.

01

Schema Theorem (Holland)

Schema Theorem

Schema :

Hyper place in the search space.

Schema Theorem

Schema :

Hyper place in the search space.

11###

: *The “don’t care” symbol.*

Schema Theorem

Schema :

Hyper place in the search space.

$$\boxed{2^3 = 8}$$

$$11\#\#\#$$

: *The "don't care" symbol.*

Schema Theorem

Instances :

All strings meeting this criterion.

$$\boxed{2^3 = 8}$$

11###

Schema Theorem

Instances :

All strings meeting this criterion.

11000

Schema Theorem

Instances :

All strings meeting this criterion.

11111

Schema Theorem

Fitness of a schema :

Mean fitness of all string instances.

11###

Schema Theorem

Global optimisation :

Highest fitness schema with zero “don’t care” symbols.

11###

Schema Theorem

Holland showed that the analysis of GA behavior was far simpler if carried out in terms of schemata.

Schema Theorem

Holland showed that the analysis of GA behavior was far simpler if carried out in terms of schemata.

Aggregation :

Rather than model the evolution of all possible strings, group together in some way and model the evolution of the aggregated variables.

Schema Theorem

Two features to describe schemata.

H=1##0#1#0

Schema Theorem

Order of schemata :

Number of positions in the schemata that do not have the "don't care" sign.

H=1##0#1#0

$$o(H) = 4$$

Schema Theorem

Defining length of schemata :

Distance between the outermost defined position (which equals the number of possible crossover points between them).

H=1##0#1#0

$$d(H) = 8 - 1 = 7$$

Schema Theorem

Standard genetic algorithm (SGA)

- *Fitness proportionate parent selection,*
- *One-point crossover (1X),*
- *Bitwise mutation,*
- *Generational survivor selection*

Schema Theorem

$$E(m(H, t+1)) \geq \frac{m(H,t)f(H)}{a_t}(1-p)$$

$$p = \frac{\delta(H)}{l-1}p_c + o(H)p_m$$

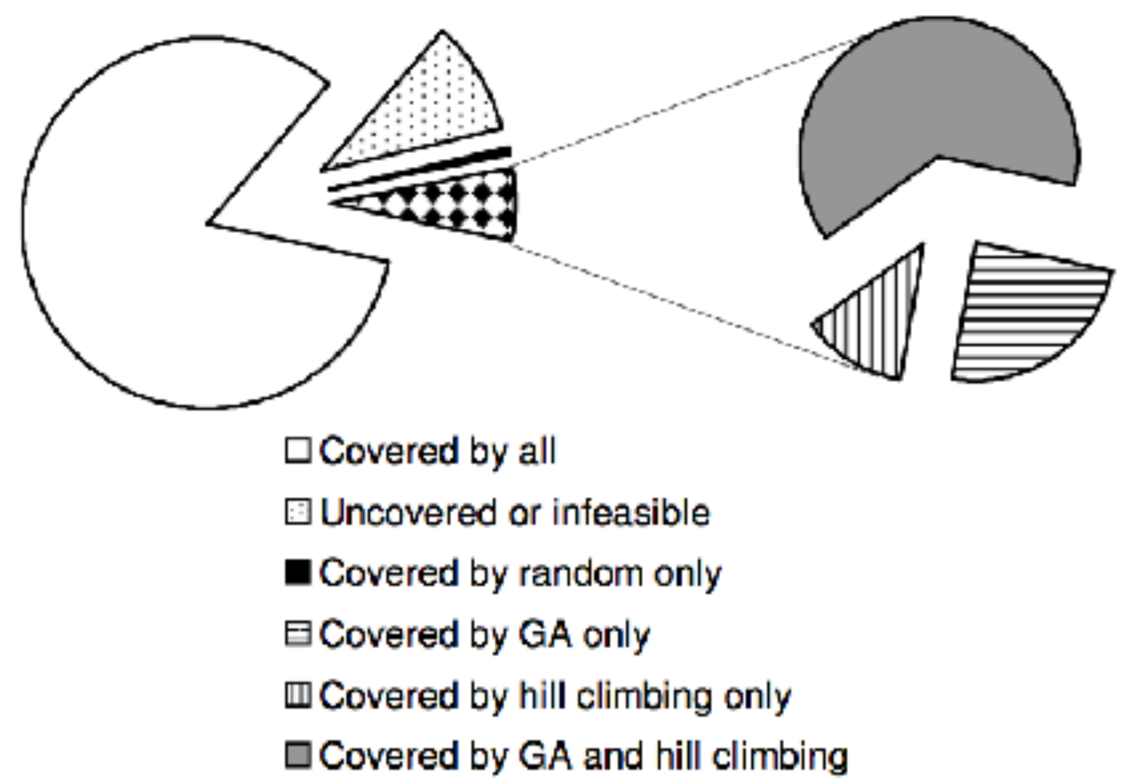
- $m(H, t)$: number of instances of schema H at generation t
- a_t: average fitness of population at t
- l: length of chromosomes
- $o(H)$: order of H, $\delta(H)$: defining length of H
- p_c: crossover rate , p_m: mutation rate

Case Study: Search-Based Software Testing

- Traditionally, GAs have been very popular with researchers: it appears fancy :)
- Is it really grounded on facts?
- Harman and McMinn (2007) compared the performance of HC and GA for automated test data generation for branch coverage for C programs.
  - `bibclean`, `eurocheck`, `gimp`, `space`, `spice`, `tiff`

Comparison of HC and GA

- There are branches that can only be covered by HC and GA respectively.
- Branches easier for GA: **`bibclean`**, especially in functions **`check_ISBN()`** and **`check_ISSN()`**.
- Why?



Proportion of branches covered by different algorithms (Harman & McMinn, 2007)

Schema Theory in Work

- "…Registration group identifiers have primarily been allocated within the **978 prefix element**. The single-digit group identifiers within the **978** prefix element are: 0 or 1 for English-speaking countries; 2 for French-speaking countries; 3 for German-speaking countries; 4 for Japan; 5 for Russian-speaking countries; and 7 for People's Republic of China. An example 5-digit group identifier is **99936**, for Bhutan. The allocated group IDs are: **0–5**, **600–621**, **7**, **80–94**, **950–989**, **9926–9989**, and **99901–99976**." (from Wikipedia entry for ISBN)

Schema Theory in Work

- Once a small schema is formed (e.g., `9*`), it can be used as a building block for a larger schema (e.g. `99*`). Crossover allows assembly of different building blocks.

- This is also called **Building Block Hypothesis**: GAs work best for problems with building block structure in their solutions.

Computational Complexity

- !!?@#?!@#!!??!?
- Can only be considered in relation to a specific problem; often, analysis is done to problems with well defined structure, using probabilistic approach.

Population Diversity

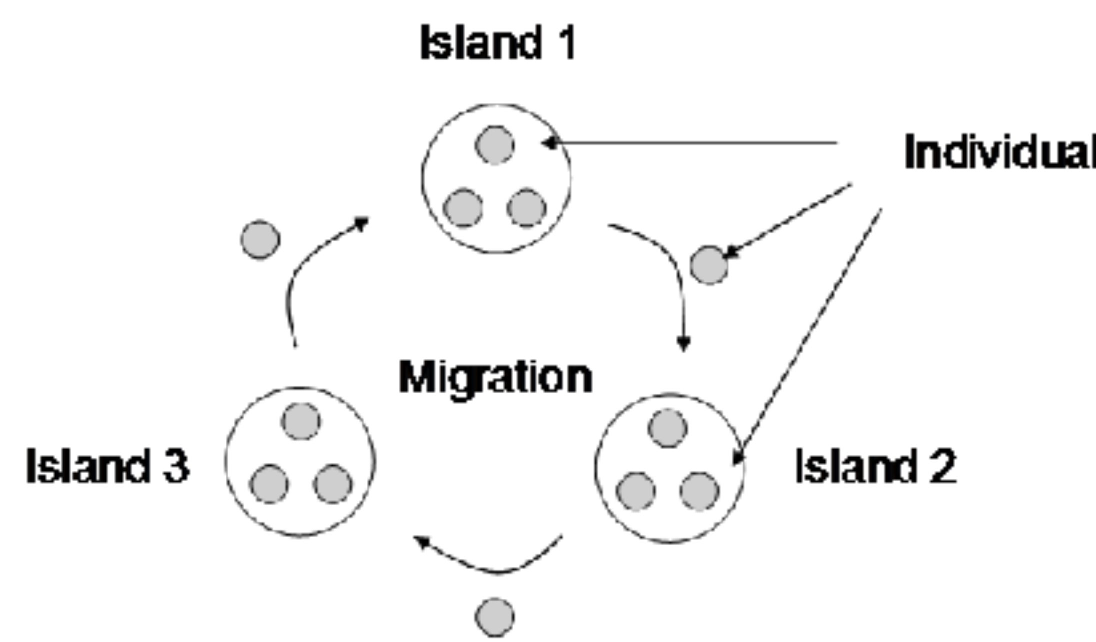
- Just like biodiversity, population diversity is important for GA. Even solutions with worst fitness may still contain valuable schemas.

- Various auxiliary mechanisms have been developed to preserve and promote population diversity.

Island GA

- Let multiple populations evolve in separation; every now and then, move individuals between islands.

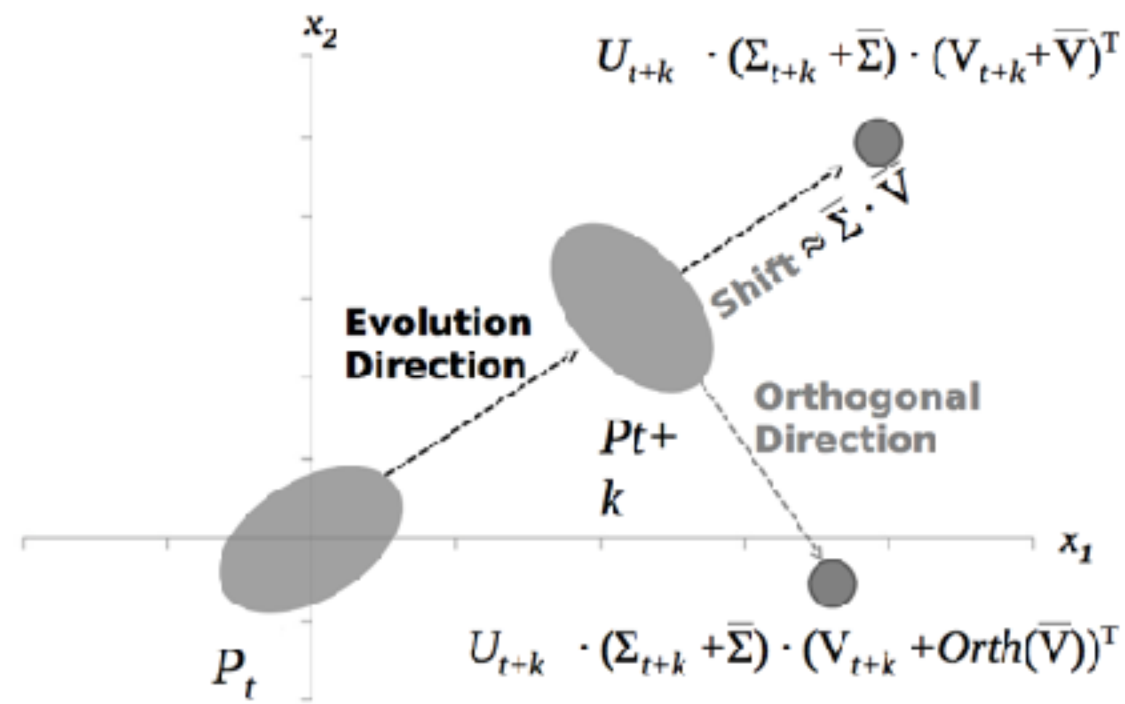
- The Island Model Genetic Algorithm: On Separability, Population Size and Convergence, *Darrell Whitley, Soraya Rana, Robert B. Heckendorn,* Journal of Computing and Information Technology, Vol. 7 (1999), pp. 33-47



Orthogonal Exploration

- Determine the direction of evolution; forcefully replace worst solutions with generated solutions that explore orthogonal direction.

- Orthogonal exploration of the search space in evolutionary test case generation, *F. M. Kifetew, A. Panichella, A. De Lucia, R. Oliveto, and P. Tonella,* in Proceedings of the 2013 International Symposium on Software Testing and Analysis, ISSTA 2013



Real Applications

- GA is a **BIG** toolbox, full of specialised operators, representation, and other assorted tricks.
- Just like any other AI technique, the more domain knowledge you have, the better your optimisation will be.

Summary

- Understand the framework of Darwinian evolution.
- Optimisation using evolution works, based on:
  - Selection pressure
  - Schema theory (one possible explanation)
- Understand various genetic operators.
- Understand the importance of population diversity.