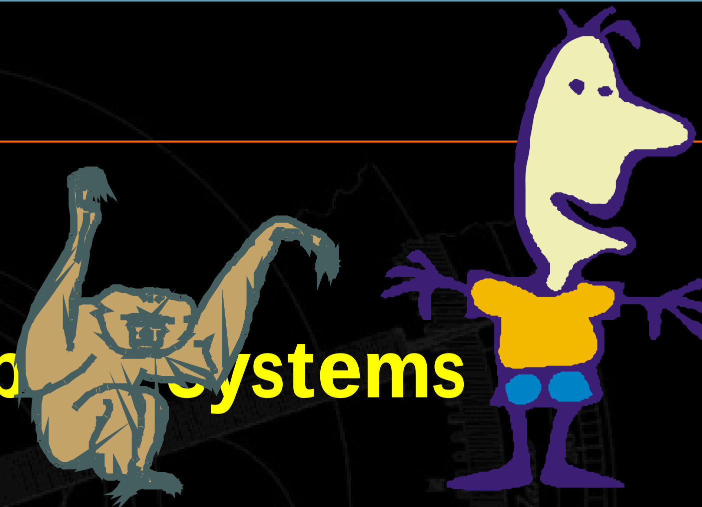
Artificial Life & Complex Systems

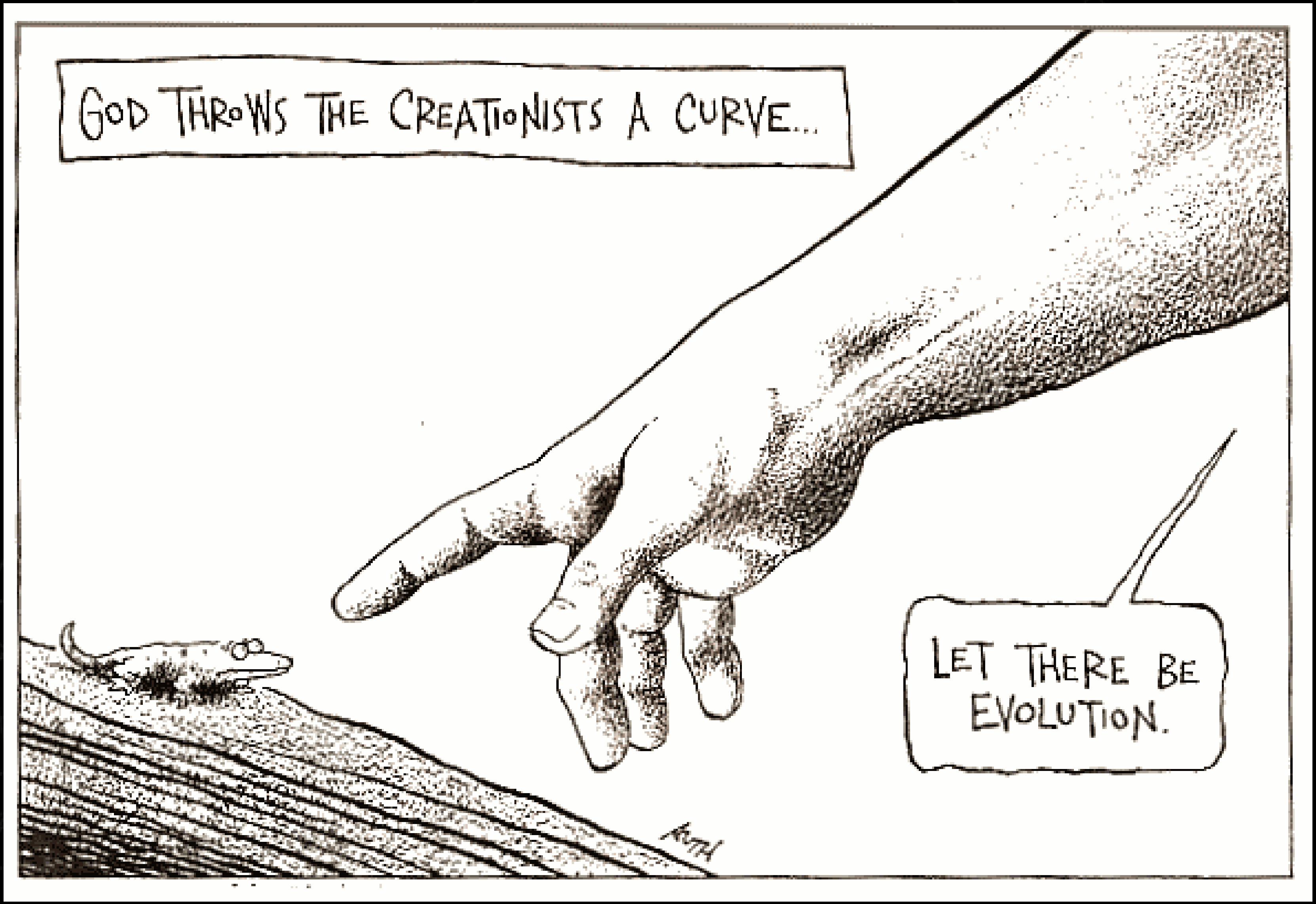


Lecture 11

Artificial Evolution I

Max Lungarella

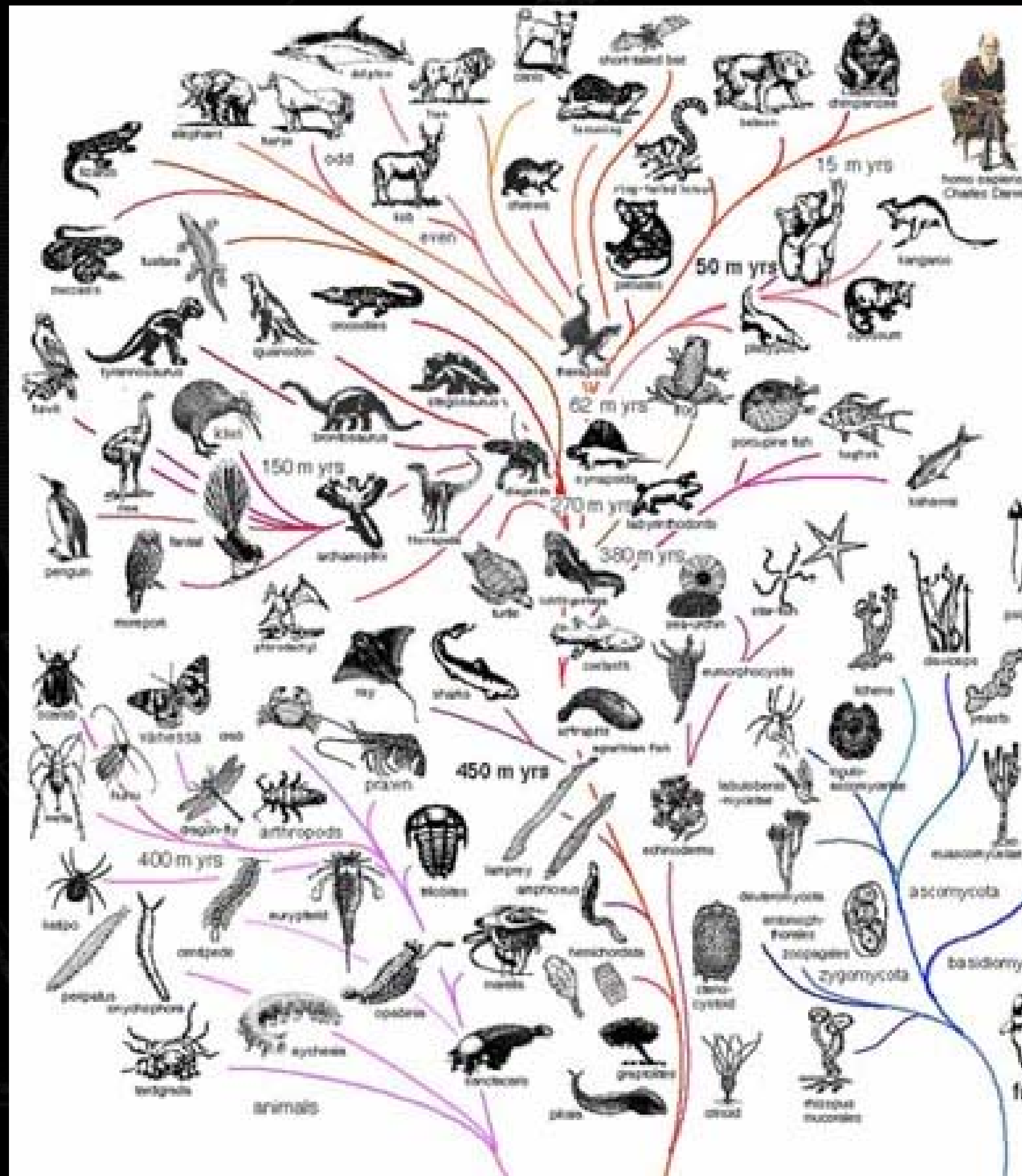
Today's Topic



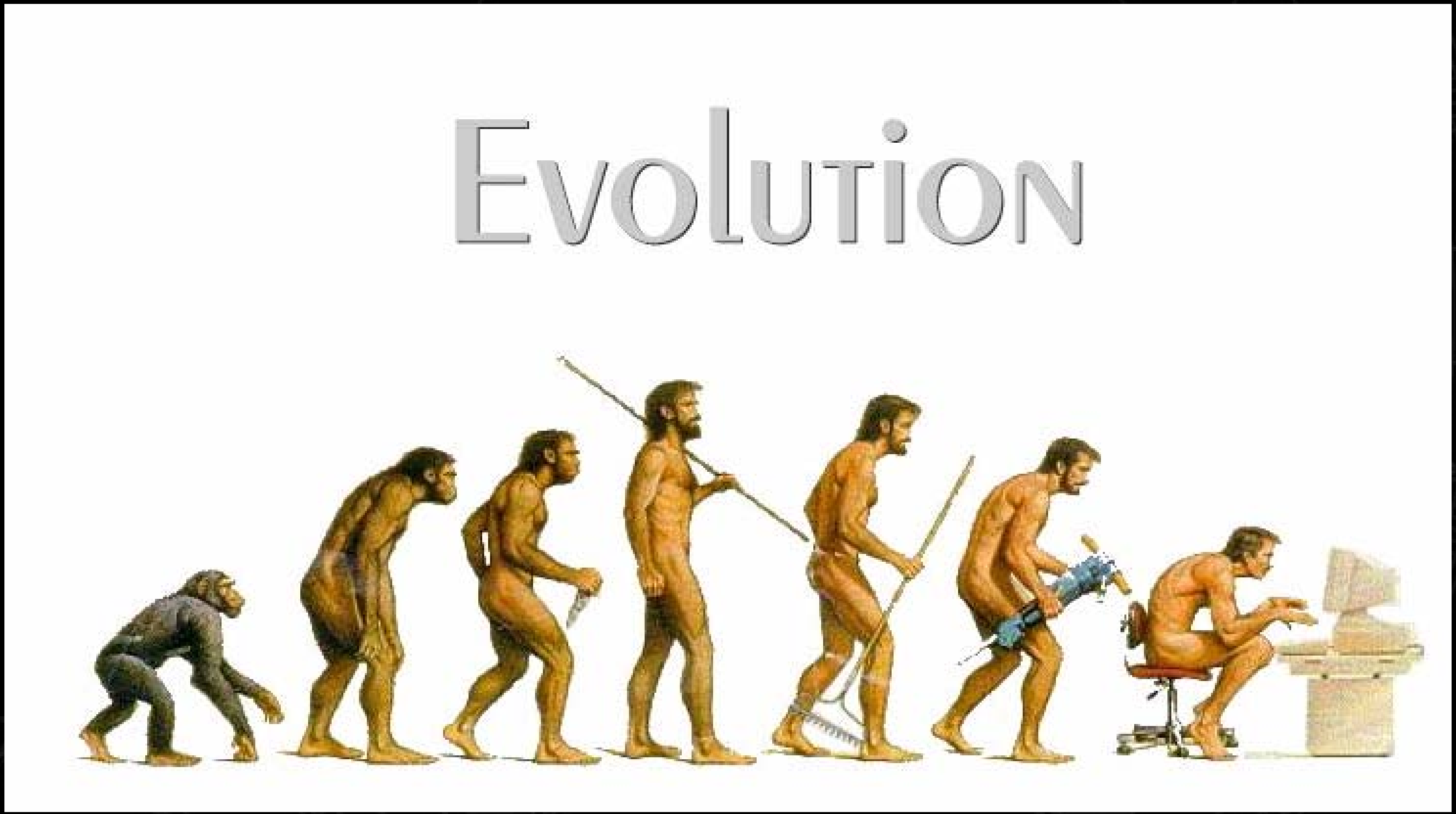
Contents

- Evolution and Darwinism
- Evolutionary Programming
- GA, EP, ES, GP
- Co-evolution
- Baldwin effect
- Punctuated equilibrium

Natural Evolution: What Do We Observe?

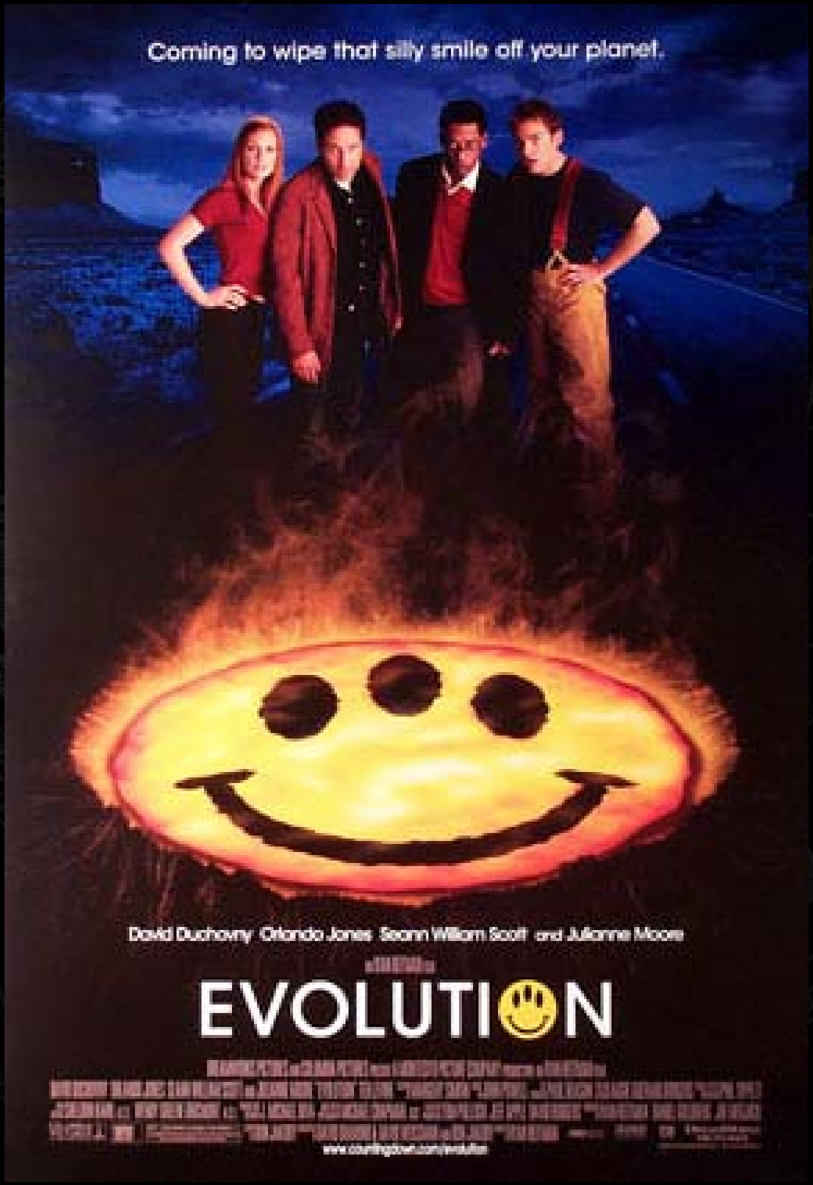


Evolution → Devolution?



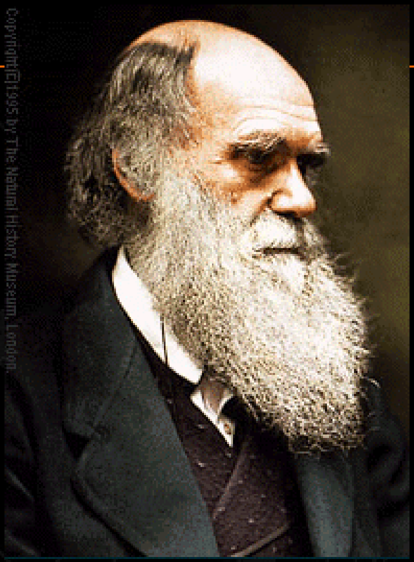
Another Popular View

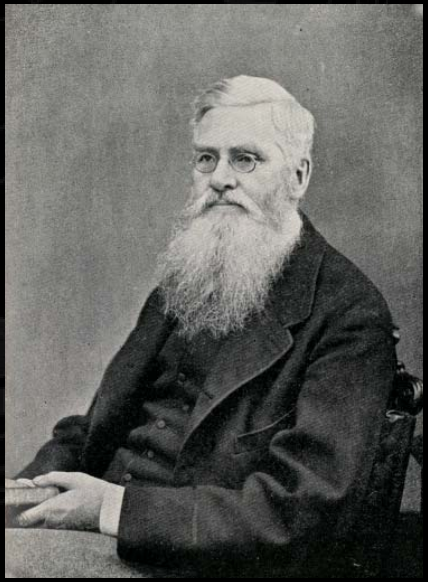
Hollywood's View



Darwin and Wallace

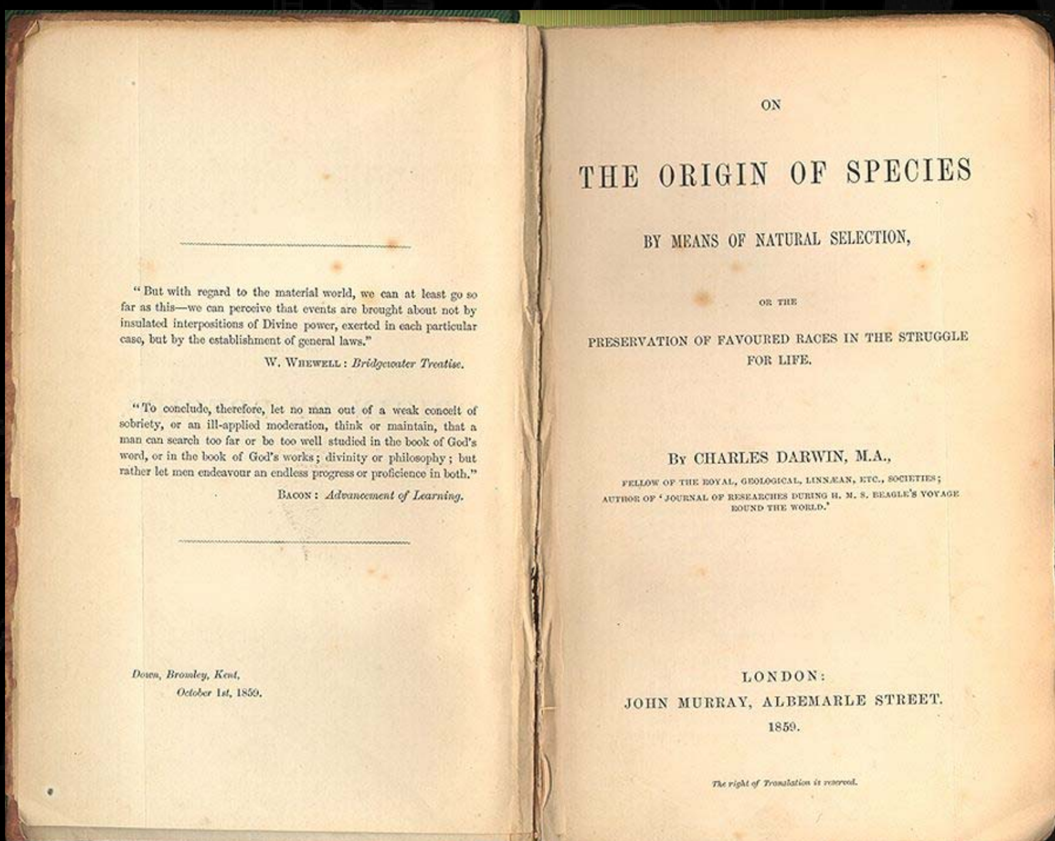
- The first public announcement of natural selection by Darwin and Wallace 1858
- “On the tendency of species to form varieties: and on the perpetuation of varieties and species by natural means of selection.” *(J. of the Proc. of the Linnean Society, Zoology, 3:45-62, 1858)*
- Note: Up till then, most people believe that life on earth had originated by heavenly decree





The Origin of Species (1859)

Revolutionized 19th century natural science revealing a logical and evincible mechanism, called natural selection or survival of the fittest, by which all plants and animals had slowly evolved from earlier forms



"But with regard to the material world, we can at least go so far as this—we can perceive that events are brought about not by insulated interpositions of Divine power, exerted in each particular case, but by the establishment of general laws."

W. WHEWELL: *Bridgewater Treatise.*

"To conclude, therefore, let no man out of a weak conceit of sobriety, or an ill-applied moderation, think or maintain, that a man can search too far or be too well studied in the book of God's word, or in the book of God's works; divinity or philosophy; but rather let men endeavour an endless progress or proficience in both."

BACON: *Advancement of Learning.*

Down, Bromley, Kent,
October 1st, 1859.

ON

THE ORIGIN OF SPECIES

BY MEANS OF NATURAL SELECTION,

OR THE

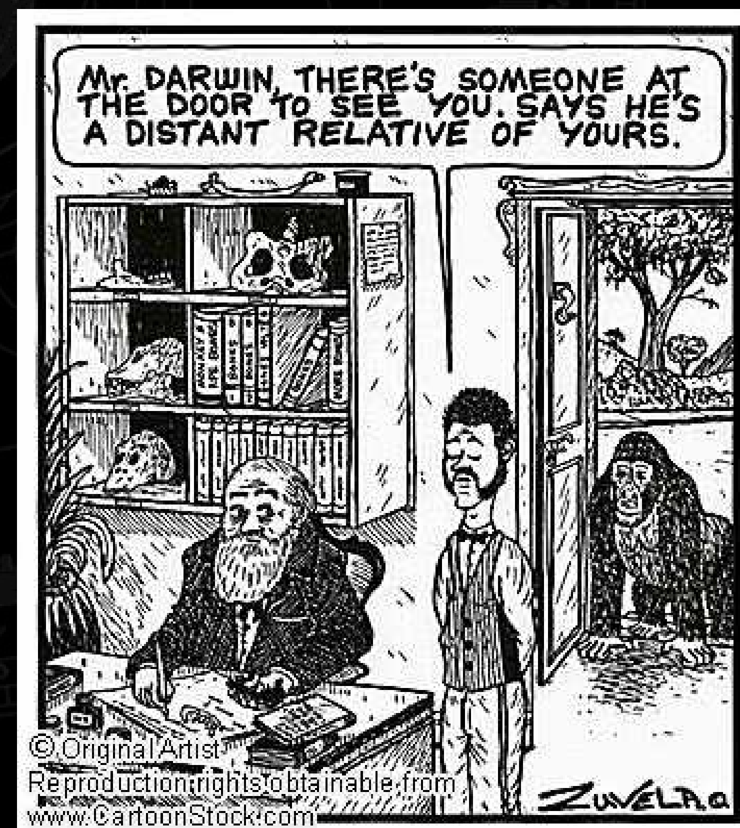
PRESERVATION OF FAVOURED RACES IN THE STRUGGLE FOR LIFE.

BY CHARLES DARWIN, M.A.,

FELLOW OF THE ROYAL, GEOLOGICAL, LINNÆAN, ETC., SOCIETIES;
AUTHOR OF 'JOURNAL OF RESEARCHES DURING H. M. S. BEAGLE'S VOYAGE ROUND THE WORLD.'

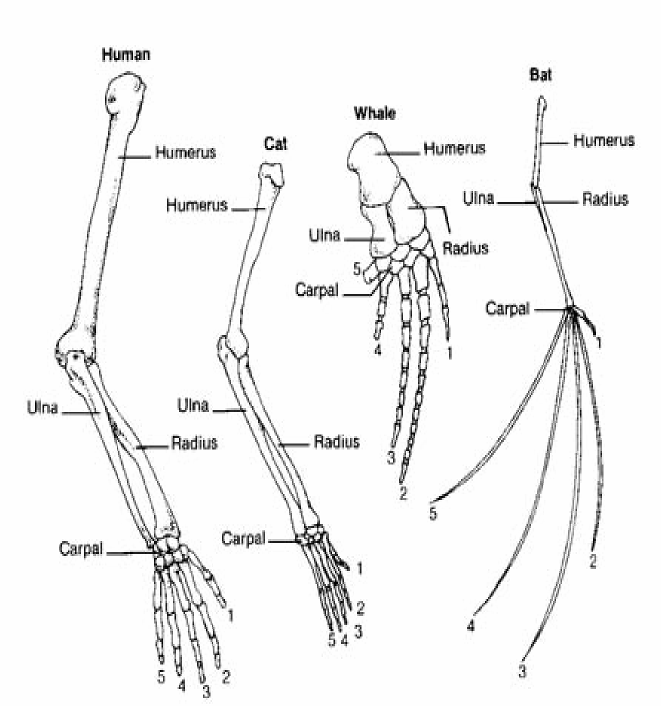
LONDON:
JOHN MURRAY, ALBEMARLE STREET.
1859.

The right of Translation is reserved.

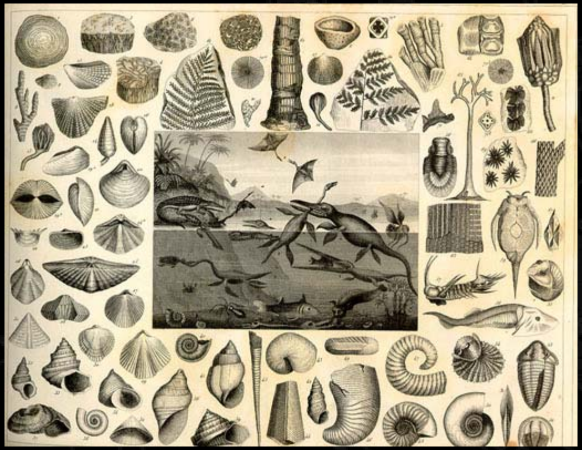


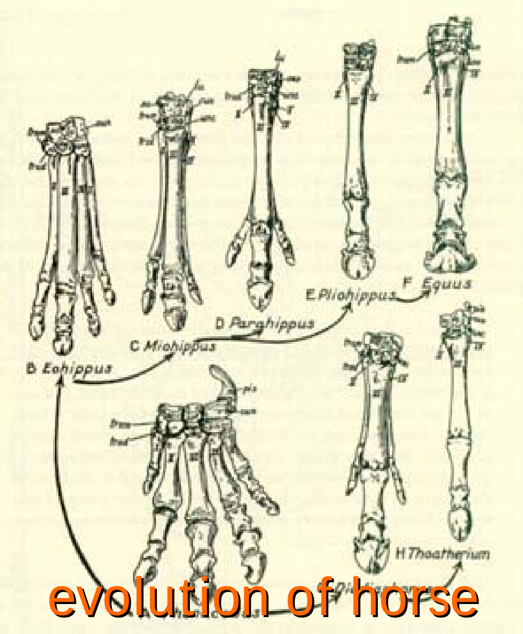
The Origin of Species (1859)

The empirical evidence supporting the Theory – as manifest in the fossil record – was compelling and made the idea irrefutable

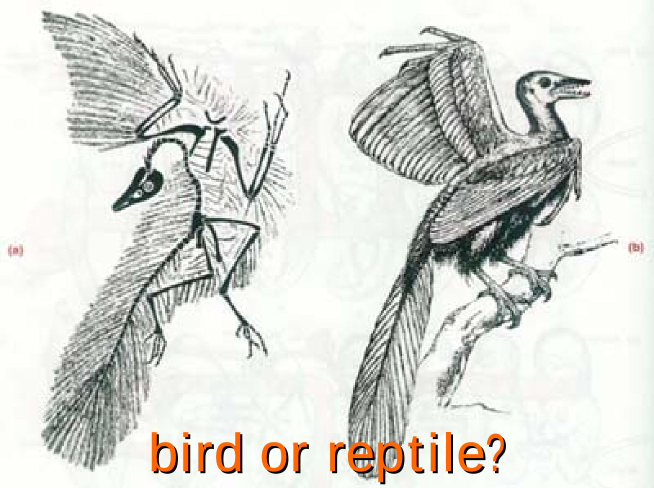


comparative anatomy





evolution of horse



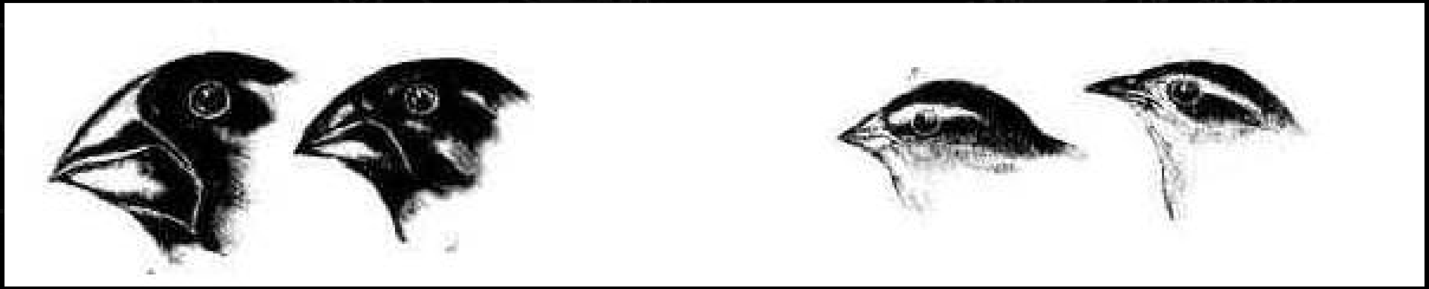
bird or reptile?

Darwinian (Natural or Biological) Evolution

Four essential preconditions for the occurence of evolution:

1. *Variability:* Individuals within species are variable
2. *Heredity:* Some of the variations are passed on to offspring
3. *Reproduction:* In every generation more offspring are produced than can survive
4. *Selection:* The survival and reproduction of individuals are not random. The individuals who survive and go on to reproduce, or who reproduce the most, are those with the most favourable variations. They are naturally selected.

(“On the Origin of Species by Means of Natural Selection”, Darwin 1859)



Neo-Darwinism

- Darwin invented the theory of evolution without any modern notion of genetics (Darwin knew very little about processes of variation)
- Neo-Darwinism combines Darwin's theory of evolution by natural selection and Gregor Mendel's theory of genetics with random mutations as the source of variation (Neo-Darwinism = Darwin + Mendel + some maths)
- Neo-Darwinism postulates that natural selection acts on the heritable (genetic) variations within individuals in populations and that mutations (especially random copying errors in DNA) provide the main source of these genetic variations. Because positive mutations seem to be rare, Neo-Darwinism contends that evolution will be a slow, gradual process

Literature: Biological Evolution

Charles Darwin:

- "On the Origin of Species"

Mark Ridley:

- "Evolution"

John Maynard Smith:

- "The Theory of Evolution"

Richard Dawkins:

- "The Selfish Gene"
- "The Blind Watchmaker"
- "Climbing Mount Improbable"

Some Terms from Genetics

DNA

- *Very large linear <u>self-replicating molecules</u> found in all living cells, the physical carrier of Genetic Information (<u>D</u>eoxyribo<u>n</u>ucleic <u>A</u>cid)*

Chromosome

- *A single, very long molecule of DNA*

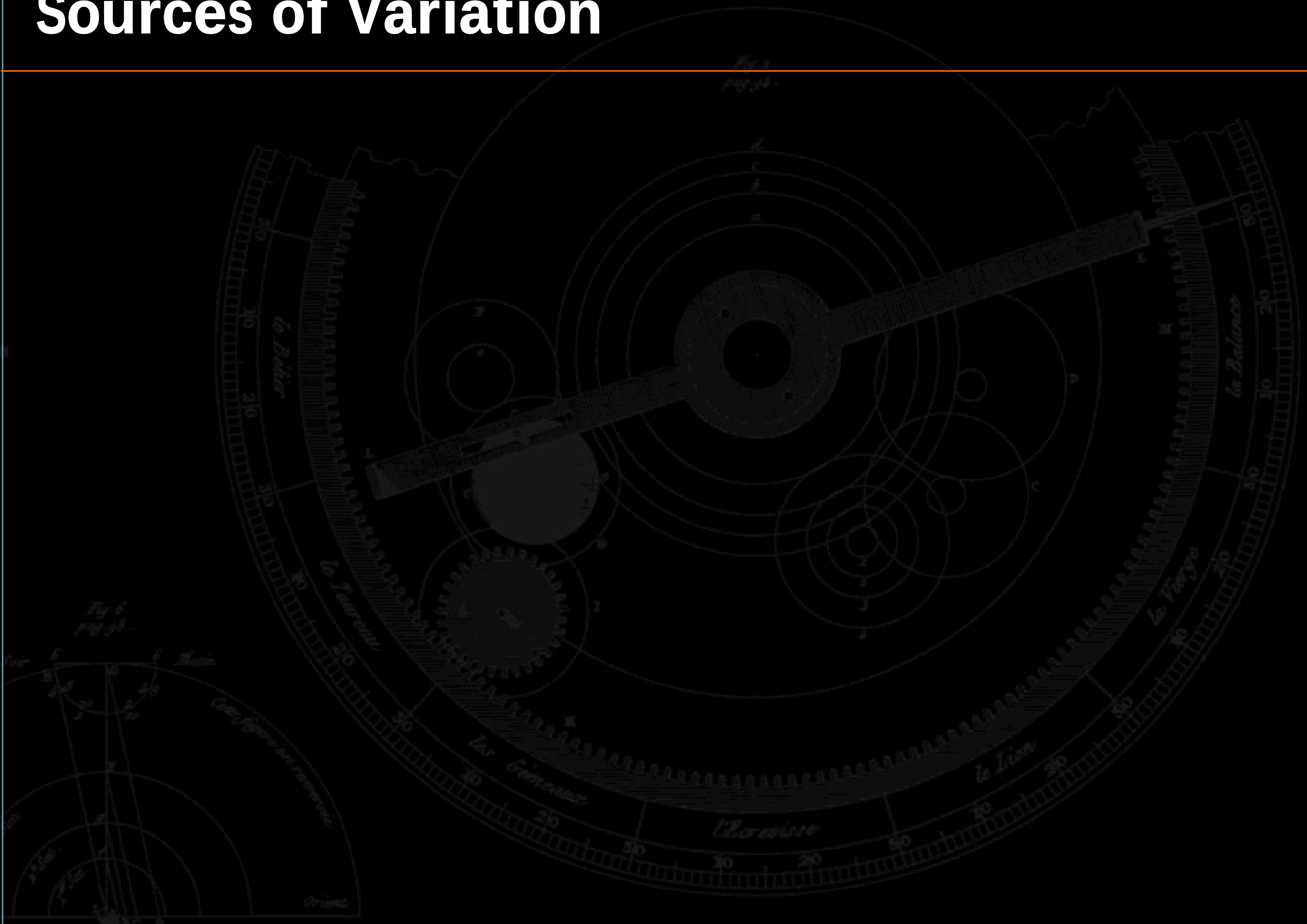
Gene

- *The basic unit of inheritance, (...) a <u>segment of DNA</u> which exerts its influence on an organisms form and function by encoding and directing the synthesis of a protein (...)*

Allele

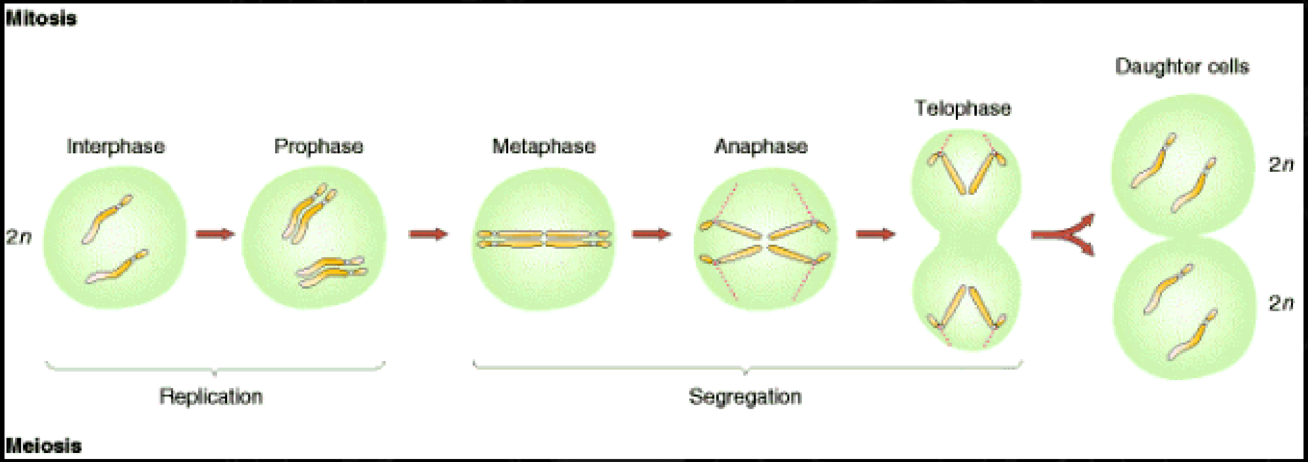
- *One of a <u>number of alternative forms</u> of a gene that can occupy a given genetic locus on a chromosome*

Sources of Variation



Mutation as a Source of Variation

Mitosis: nuclear division in cells



Mutations: errors during mitosis

- Point mutations: simple copy errors - create new alleles
- Duplication: duplicate stretch of DNA - creates extra genetic material

Most mutations are neutral !

Recombination as a Source of Variation

Mixing of genetic material

- Mixing genes on single chromosomes (crossover) → creates new combination of existing alleles

This is why…

- …you can inherit your mother's eyes, and your father's nose

Sexual Reproduction

- Can combine beneficial mutations that arise in different individuals → can eliminate disadvantageous mutations quickly

Evolution and Artificial Life

The original definition of Artificial Life (by Langton) concentrated on what counted as a synthesis of living artefacts, <u>without regard to origins or evolution</u>

Despite this, very quickly a high proportion of ALife work came to be dependent on some form of evolutionary ideas

Evolution and Artificial Life

Why are “Alifers” interested in evolution?

- “Results” of evolution are
  - ”creative”, “surprising”, “unexpected”
  - highly adapted to environmental niches
- Learn more about natural evolution (making abstractions)
- Exploiting the power of evolution for designing intelligent systems and for design “in general” (fully automated design)
- Can a program create intelligent agents “from scratch” ?
- If used with care, evolution does not suffer from designer bias
- Breaking the complexity barrier

Evolution and Artificial Life

Two „broad“ goals:

a) In ALife simulations one can study the process of evolution by
   1) creating an artificial world
   2) populating it with organisms
   3) giving those organisms a goal to achieve

   [see computational ecologies; synthetic methodology]

b) Using algorithms inspired by the evolutionary processes, one can teach artificial organisms (and systems) how to achieve their goal(s) through a crude form of evolution („design of adaptive – life-like – systems“)

Evolution and Artificial Life

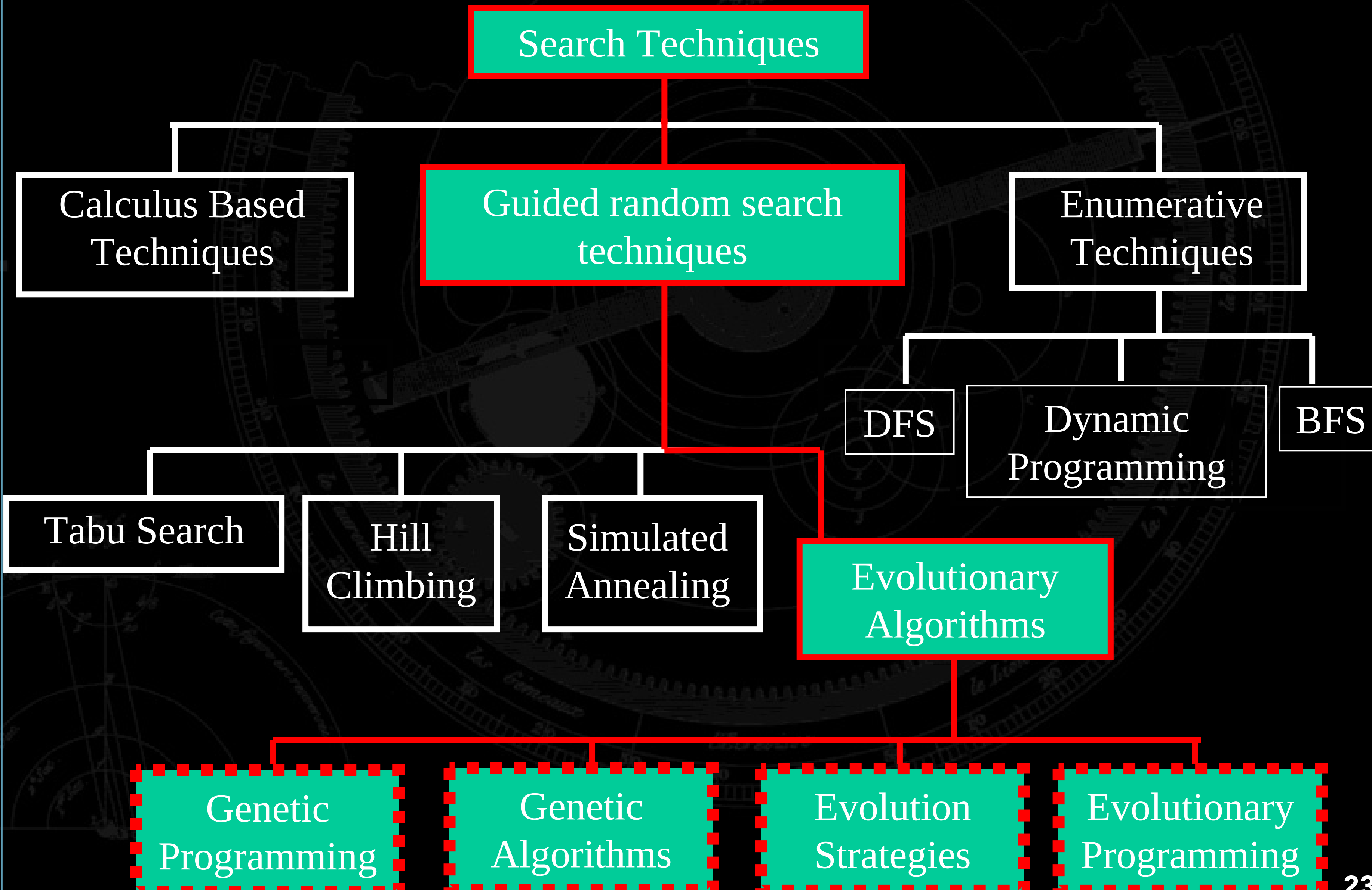
- The context of evolution is a population (organisms, objects, agents, etc.)
- The individual composing the population survive for a limited time and then die
- Some produce offspring, the „fitter“ produce more
- Over time populations change (usually) adapting to (new) conditions

Evolutionary Algorithm

What is an evolutionary algorithm (EA)?

- An evolutionary algorithm is a problem-solving strategy that uses genetics as its model (optimization tool)
- EAs apply the rules of reproduction, gene crossover, and mutation to artificial organisms so those organisms can pass traits to a new generation

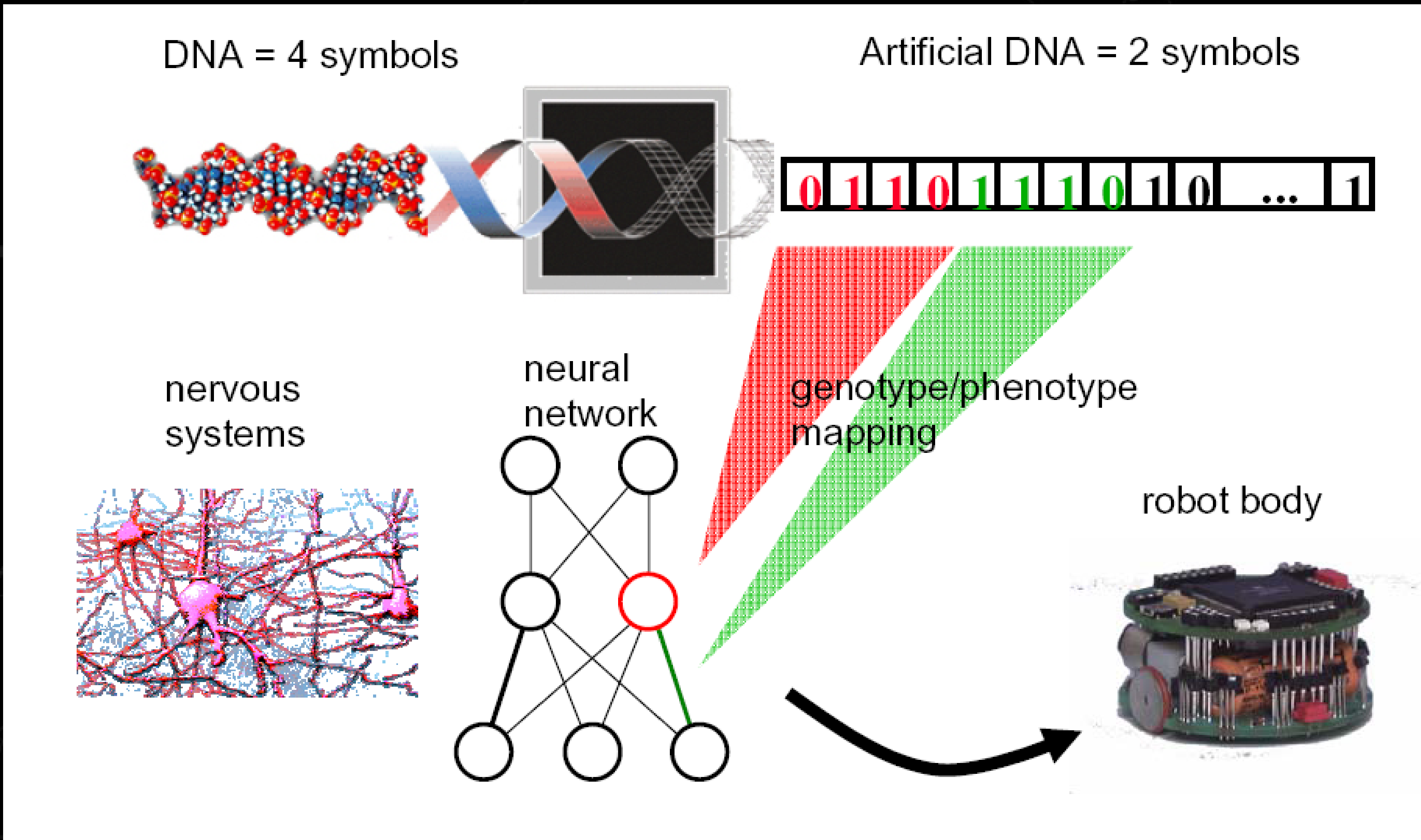
Classes of Search Techniques



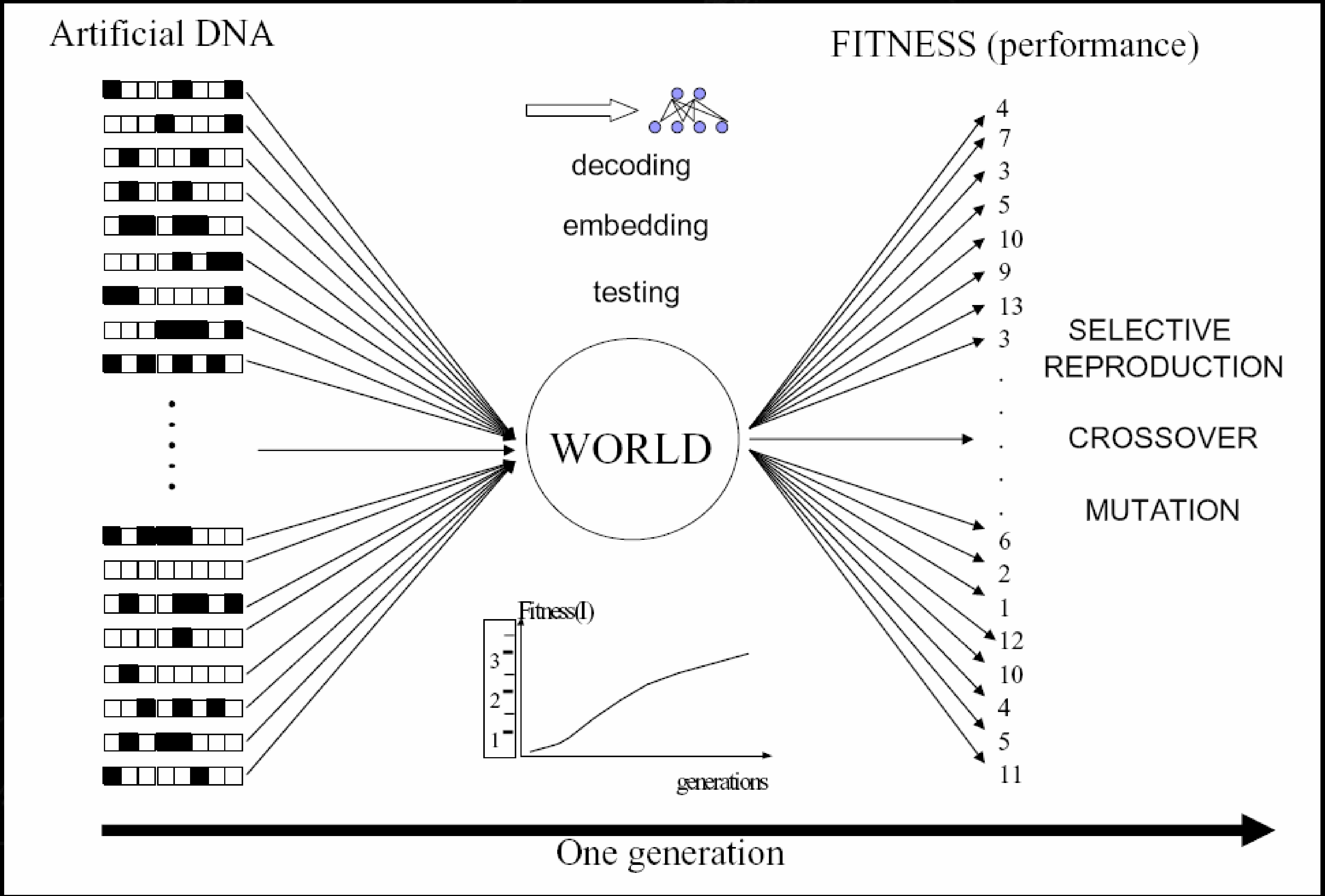
What is an Evolutionary Algorithm?

Very very briefly …

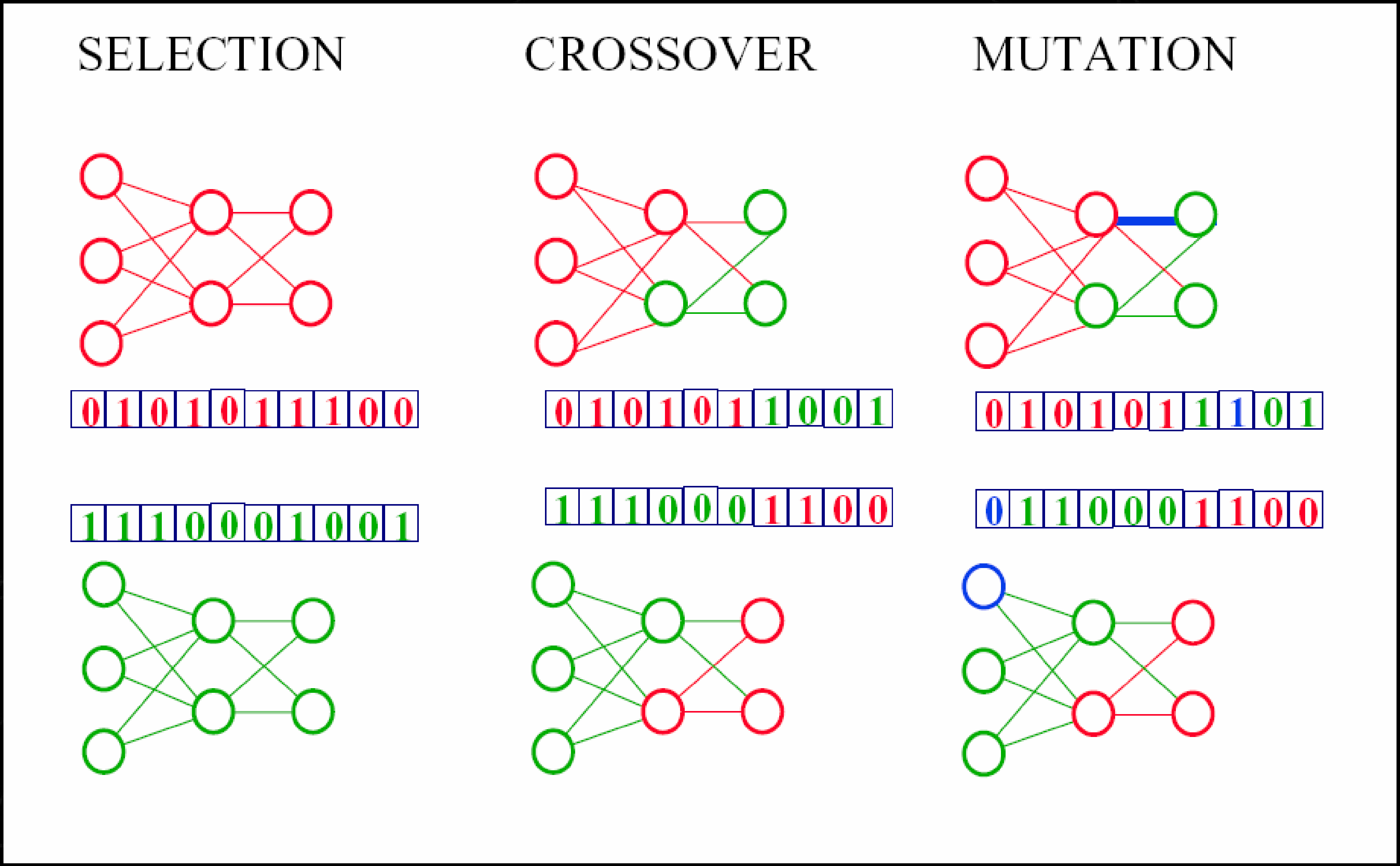
From Real DNA to Artificial DNA



Survival of the Fittest



Genetic Operators: Selection and Variation



Artificial Evolution: Origins

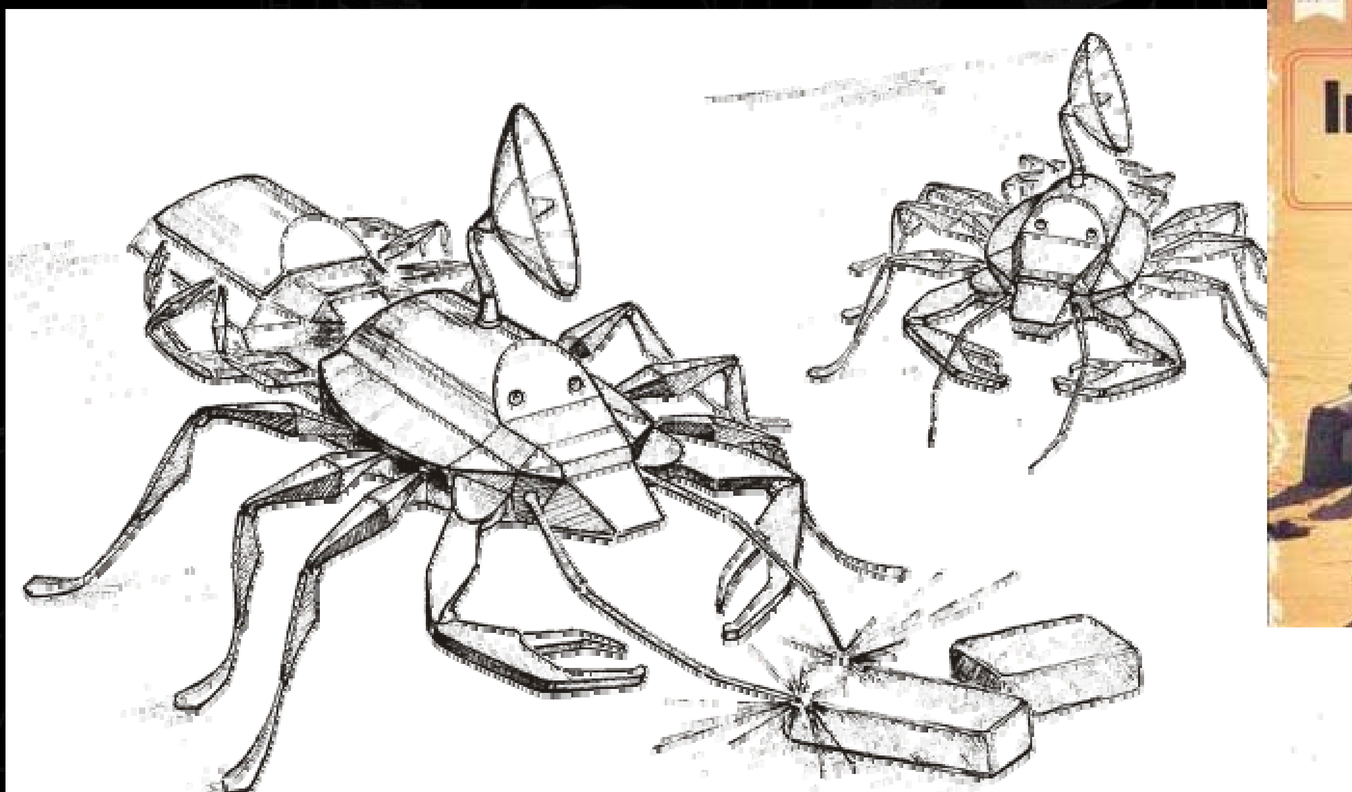
Evolution

Competition

More and more crabs

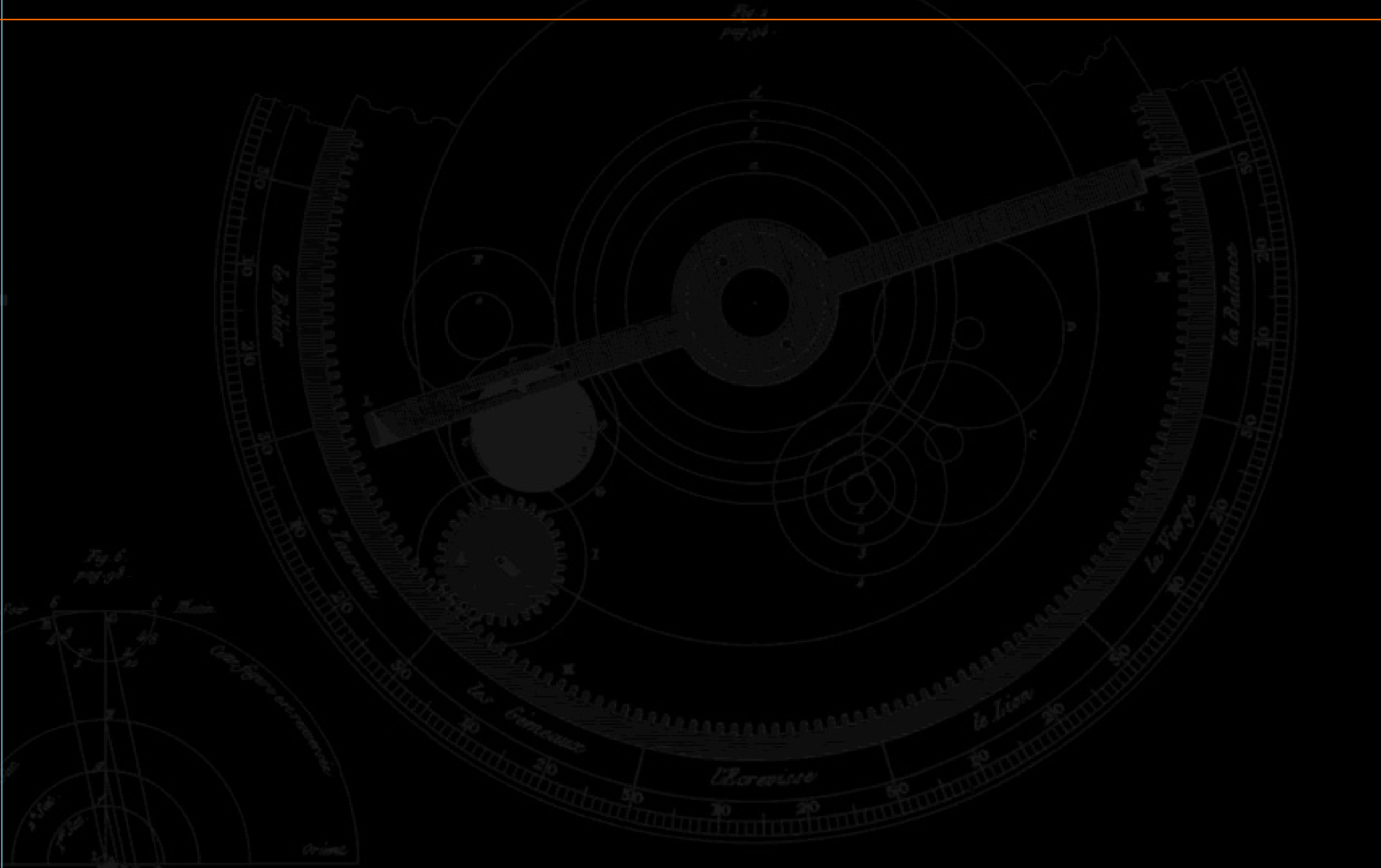
!

Making copies with errors



Inspiration for Schwefel and Rechenberg!

Two “Motivating” Examples



"Methinks it is like a weasel"

Hamlet: Do you see yonder cloud that's almost in shape of a camel?

Polonius: By the mass, and 'tis like a camel, indeed.

Hamlet: Methinks it is like a weasel.

"I don't know who it was first pointed out that, given enough time, a monkey bashing away at random on a typewriter could produce all the works of Shakespeare. The operative phrase is, of course, given enough time. Let us limit the task facing our monkey somewhat. Suppose that he has to produce, not the complete works of Shakespeare but just the short sentence 'METHINKS IT IS LIKE A WEASEL.', and we shall make it relatively easy by giving him a typewriter with a restricted keyboard, one with just the 26 (capital) letters, and a space bar. How long will he take to write this one little sentence?" *(from Richard Dawkins: "The Blind Watchmaker")*

The Power of Cumulative Selection

The sentence is composed of 29 characters.

Q: How many possible strings are there?

The Power of Cumulative Selection

- The sentence is composed of 29 characters (including space and punctuation)
- The number of possible combinations in a random sequence is 27^{29}, or about 3×10^{41} possible strings
- A computer program could be written to carry out the actions of the hypothetical monkey, continuously generating combinations of 26 letters and spaces at high speed. Even at the rate of millions of combinations per second, it is unlikely, even given the entire lifetime of the universe to run, that the program would ever produce the phrase "METHINKS IT IS LIKE A WEASEL."
- Is there a better way to search??

The Power of Cumulative Selection

Let's introduce a "goal function" and measure the distance of a particular string to the target sentence

1. Start with 10 random sentences
2. For each sentence measure distance from target sentence
3. Choose 5 highest scoring sentences (selection) and copy them while introducing error at random position (mutation)
4. Go to 2

The Power of Cumulative Selection

Q: How many generations does it take to find the correct sentence?

The Power of Cumulative Selection

Generation 1: WDLMNLT DTJBKWIRZREZLMQCO P
Generation 2: WDLTMNLT DTJBSWIRZREZLMQCO P
Generation 10: MDLDMNLS ITJISWHRZREZ MECS P
Generation 20: MELDINLS IT ISWPRKE Z WECSEL
Generation 30: METHINGS IT ISWLIKE B WECSEL
Generation 40: METHINKS IT IS LIKE I WEASEL
Generation 43: METHINKS IT IS LIKE A WEASEL

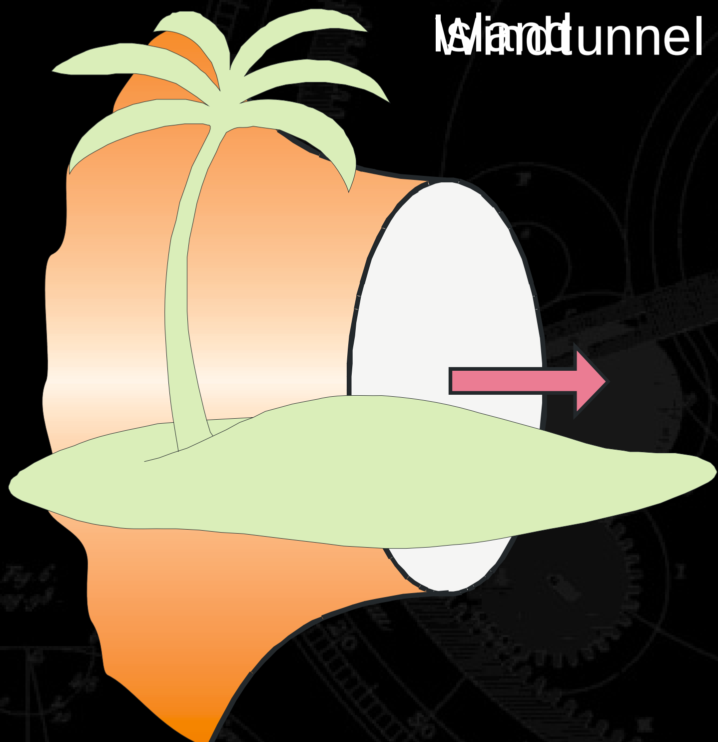
http://home.pacbell.net/s-max/scott/weasel.html

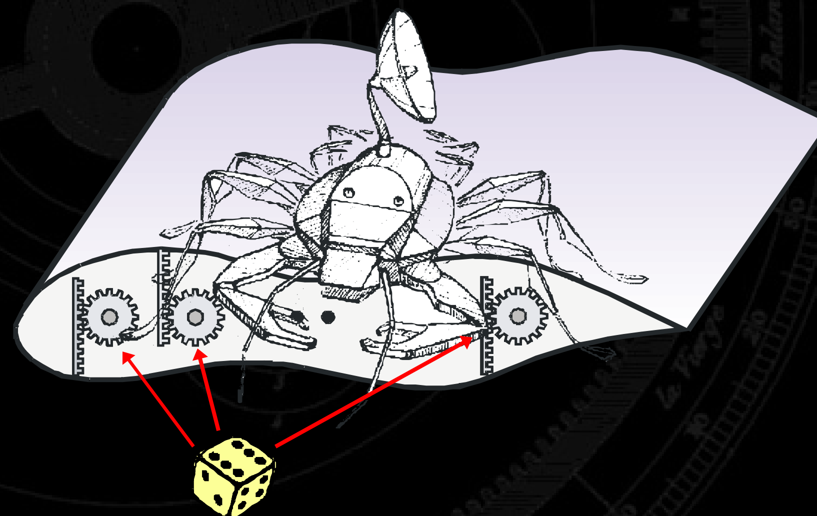
The Power of Cumulative Selection

Lessons learned from this example:

- a process of *cumulative* selection can take far fewer steps to reach any given target
- DNA sequences or organic compounds such as proteins can be the result of atoms “randomly” combining to form more complex structures (as long as there is selection)

Drag Minimization or The “Shaping” Problem

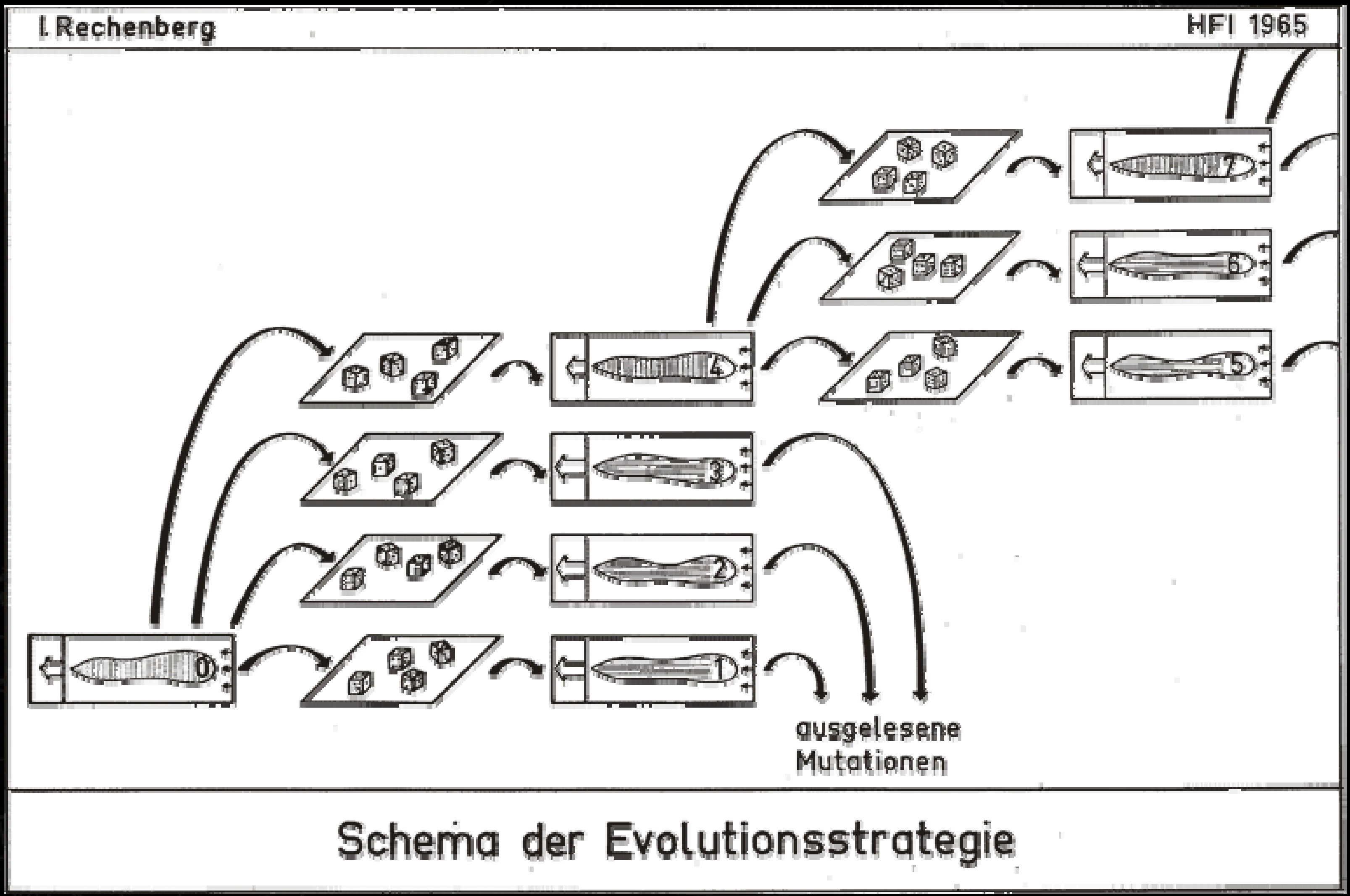




Adjustable gear instead
of making a copy

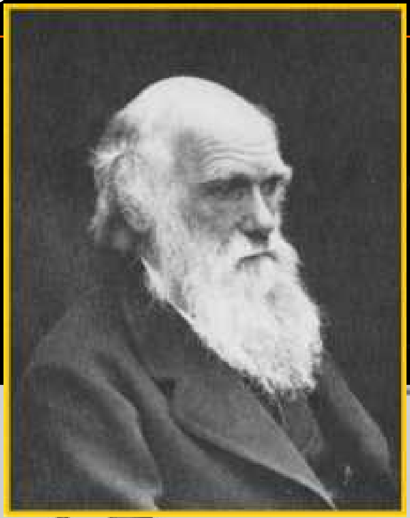
Schwefel and Rechenberg (1963)

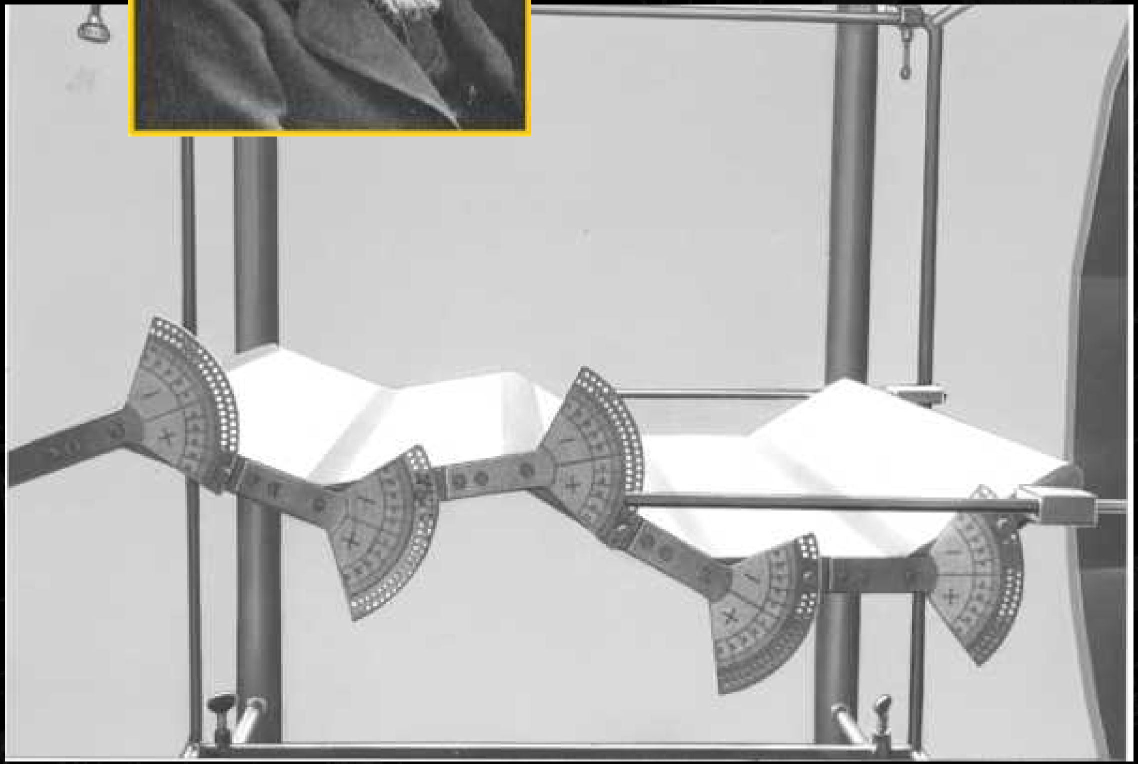
Drag Minimization



Schwefel and Rechenberg (1963) Idea for a mechanical evolution experiment

Drag Minimization



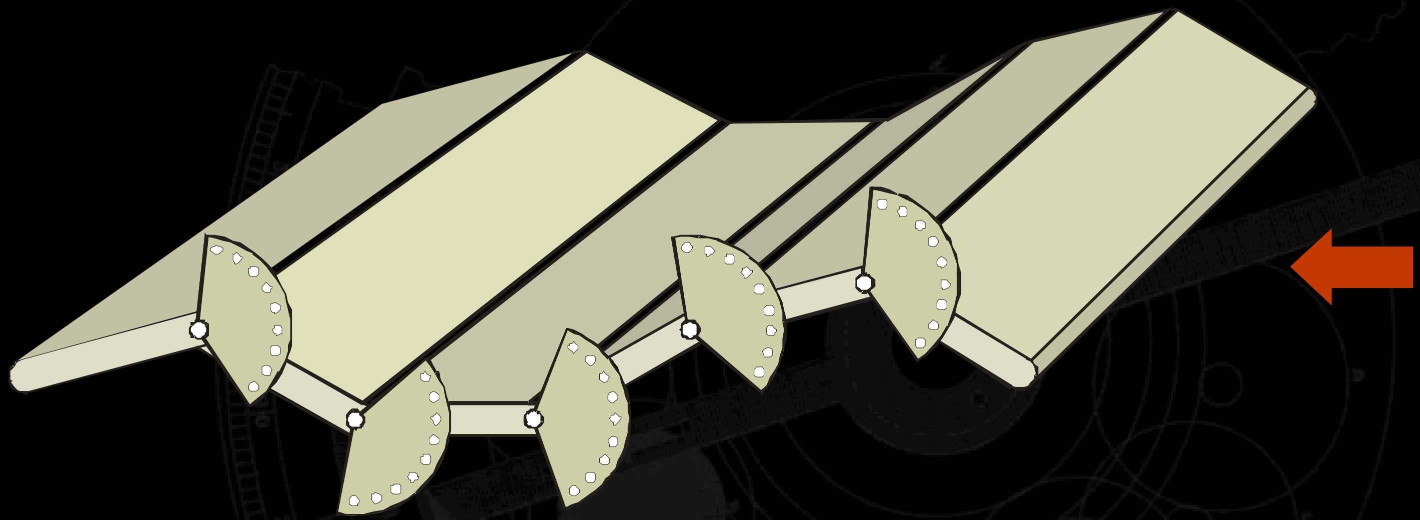


DARWIN in the windtunnel

The kink plate for the key experiment with the Evolution Strategy

Schwefel and Rechenberg (1963)

Drag Minimization

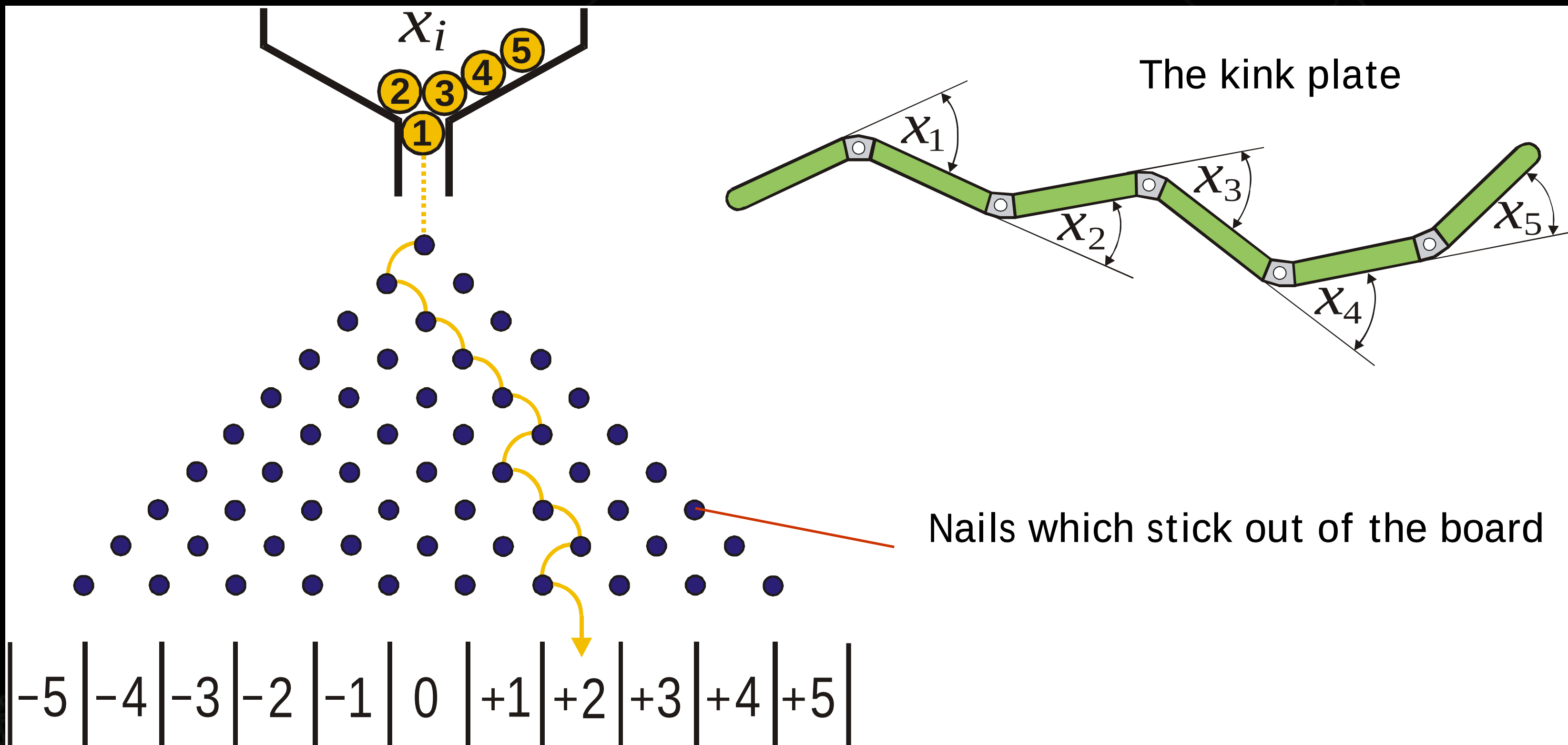


Number of possible adjustments?

$51^5 = 345\ 025\ 251$

Schwefel and Rechenberg (1963)

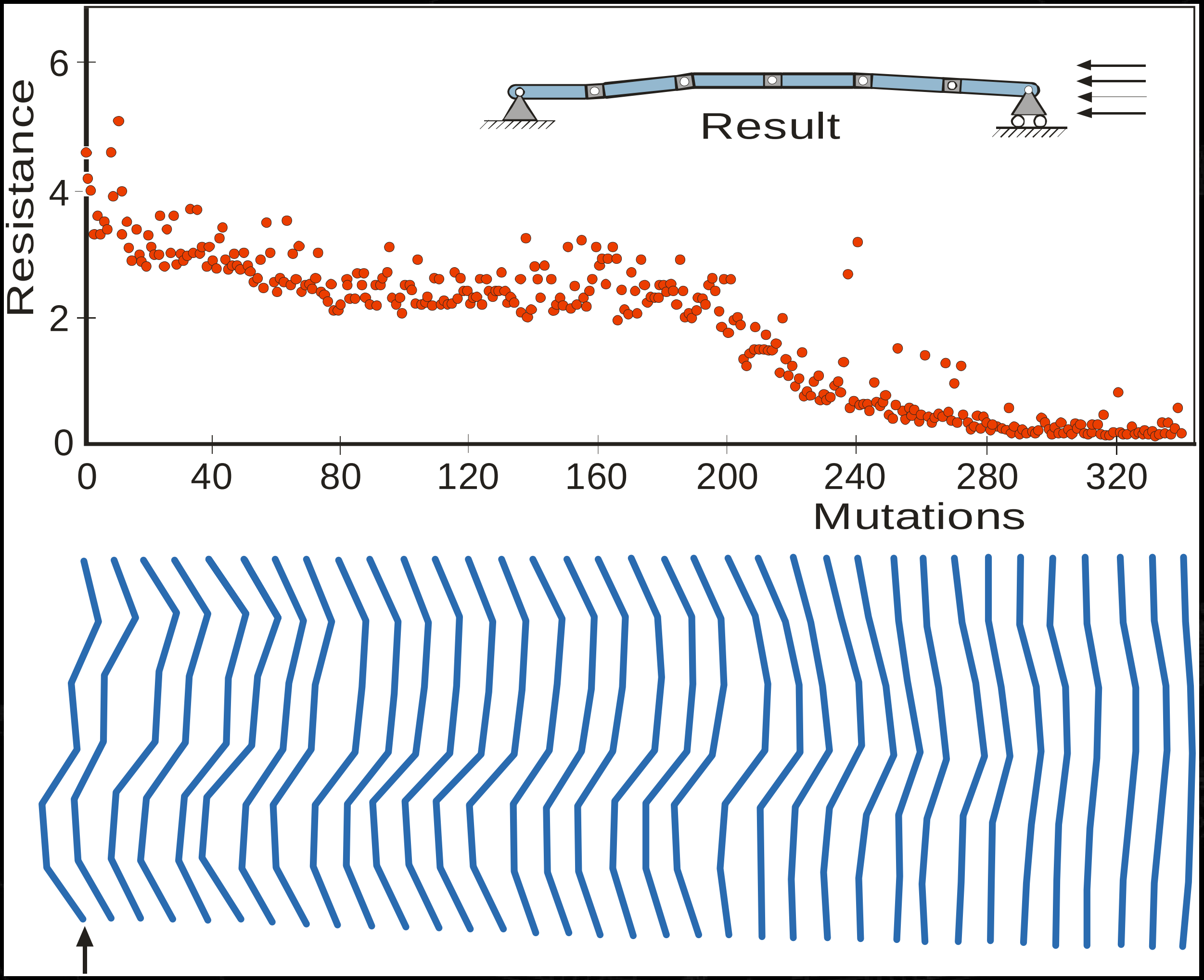
Drag Minimization



The mutation apparatus – GALTONs pin board

Schwefel and Rechenberg (1963)

Drag Minimization



The *experimentum crucis* – Drag minimization of the kink plate

Spiegel

Werkes — Proust an eine Bekannte: „Odette de Crécy (eine seiner Romanfiguren) ist nicht nur nicht Sie, sondern genau das Gegenteil von Ihnen" — sowie vom qualvollen Kampf um die endgültige Edition der „Recherche", den Proust unter Schmerzen und Atemnot vom Bett aus führen mußte, arbeitsfähig nur durch Koffeintabletten, getrieben von Todesangst und von Furcht, die Veröffentlichung des Werkes nicht mehr zu erleben oder den „endlosen Wälzer" (Proust) überhaupt nicht vollenden zu können.

Als sich nach Erscheinen der ersten Bände unerwarteter Erfolg einstellte — für den zweiten Band erhielt Proust 1919 den Prix Goncourt —, lebte der Moribunde für kurze Zeit noch einmal auf. Er ging wieder aus, meistens nachts, und dinierte im Hotel Ritz, wo er totenbleich, mit fiebrigen Augen erschien, in einem hocheleganten, aber derangierten Abendanzug, aus dessen Jackett wär-

FORSCHUNG

AERODYNAMIK

Zickzack nach Darwin

Der Eingebung und oftmals auch glücklichem Zufall verdanken Generationen von Flugzeugtechnikern zukunftweisende Lösungen. Aber ein Student der Technischen Universität in West-Berlin möchte den Fortschritt kalkulabel machen: Er fand für das Roulette-Spiel der Flugzeugingenieure ein System.

Zahllose aufwendige Versuchsreihen in mietshausgroßen Windkanälen, deren Bau Millionen Dollar kostet und in denen Mammut-Flügelräder leichte Brise ebenso wie heulenden Orkan oder mehrfach schallschnelle Luftströme erzeugen können, sind bei den großen Flugzeugfirmen nötig, um für ein neues

Student Rechenberg, Lehrer Wille*: Roulette in der Hochschule

Zigzag after DARWIN

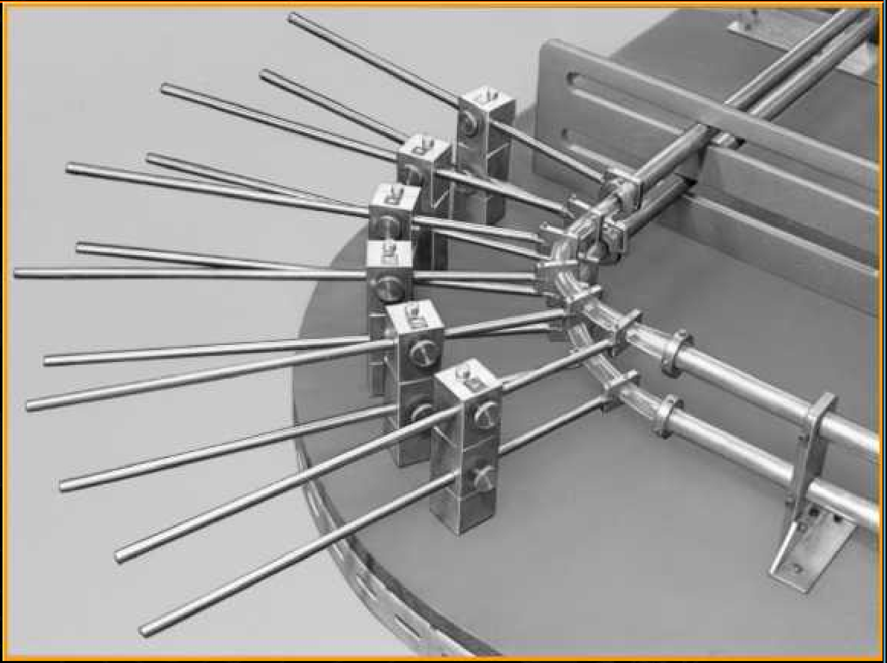
Story in the Magazin

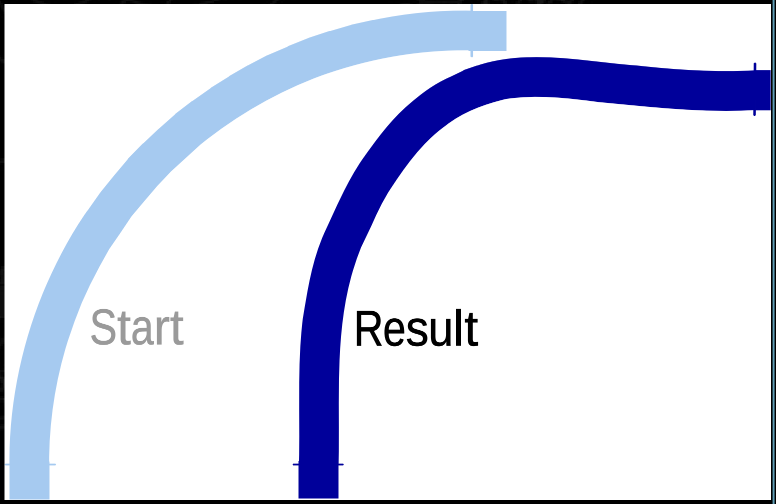
DER SPIEGEL

18th November 1964

The “Tubing Problem”

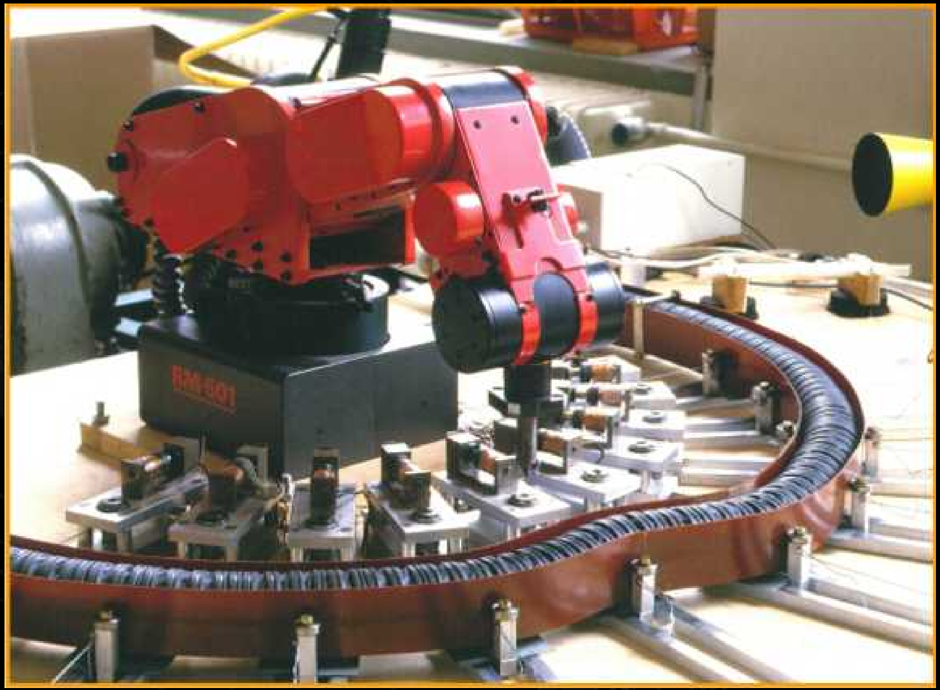
- Evolution of a 90 degree bend
- Fuel flows in from the left
- Fuel should leave the tube such that <u>resistance of the fluid is minimized</u>
- Six <u>manually adjustable shafts</u> determine the form of the pipe
- What is the shape of the connecting tube?

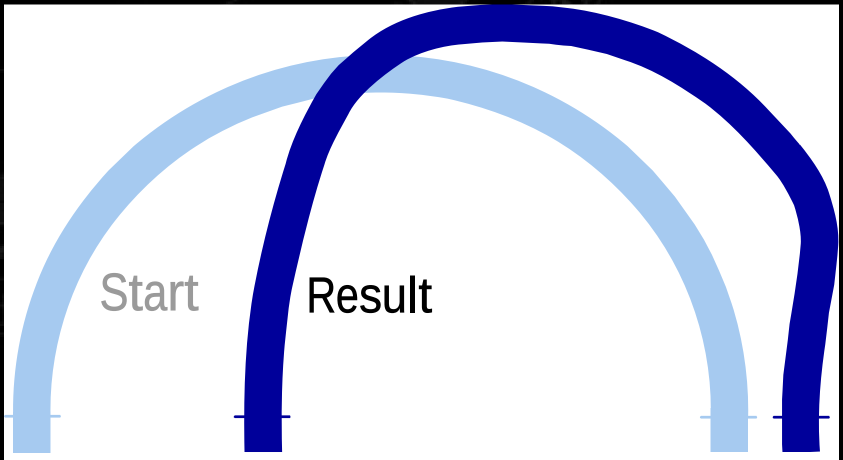




The “Tubing Problem”

- Evolution of a 180 degree bend
- 10 robot-controlled cable-drives alter the 180°pipe bend
- What is the shape of the connecting tube?





Brief History of Evolutionary Computation

The idea of using simulated evolution to solve engineering and design problems have been around since the 1950's:

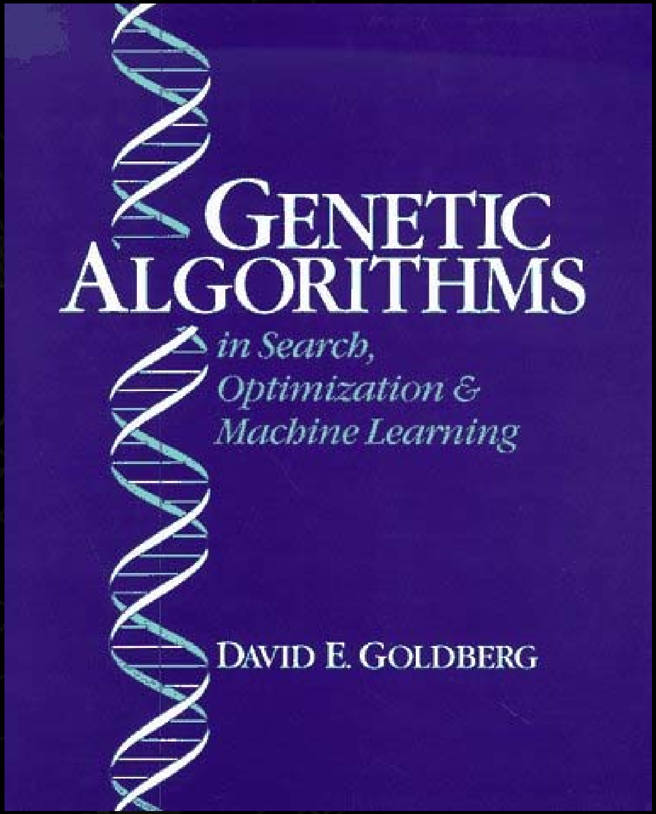
- Alex Fraser (1957): genomes as strings of binary numbers
- George Box (1957)
- Hans Bremermann (1958): virtual genomes producing offsprings
- Richard Friedberg (1958)

However, it wasn't until the early 1960's that we began to see three influential forms of EC emerge:

- Evolutionary Programming (Lawrence Fogel, 1962; Fogel, 1995)
- Genetic Algorithms (John Holland, 1962; Holland, 1975)
- Evolution Strategies (Ingo Rechenberg, 1965; Schwefel, 1968; Rechenberg, 1973)

Brief History of Evolutionary Computation

- Each of these researchers successfully developed appropriate ECs for their particular problems <u>independently</u>
- In the US, Genetic Algorithms have become the most popular EC technique due to a book by David E. Goldberg (1989) entitled, *"Genetic Algorithms in Search, Optimization & Machine Learning"*
- This book explained the concept of genetic search in such a way the a wide variety of engineers and scientist could understand and apply.



Brief History of Evolutionary Computation

A number of other books helped fuel growing interest in EC:

- Lawrence Davis (1991) *“Handbook of Genetic Algorithms”*
- Zbigniew Michalewicz (1992) *“Genetic Algorithms + Data Structures = Evolution Programs”*
- John R. Koza (1992) *“Genetic Programming”* (1992)
- D. B. Fogel (1995) *“Evolutionary Computation: Toward a New Philosophy of Machine Intelligence”*

These books not only fueled interest in EC but they also were instrumental in bringing together the EP, ES, and GA concepts <u>together</u> in a way that fostered unity and an explosion of new and exciting forms of EC

Brief History of Evolutionary Computation

Evolution of Evolutionary Computation

First Generation EC

- EP (Fogel)
- GA (Holland)
- ES (Rechenberg, Schwefel)

Second Generation EC

- Genetic Evolution of Data Structures (Michalewicz)
- Genetic Evolution of Programs (Koza)
- Hybrid Genetic Search (Davis)
- Tabu Search (Glover)

Brief History of Evolutionary Computation

Third Generation EC

- Artificial Immune Systems (Forrest)
- Cultural Algorithms (Reynolds)
- DNA Computing (Adleman)
- Ant Colony Optimization (Dorigo)
- Particle Swarm Optimization (Kennedy & Eberhart)
- Memetic Algorithms
- Estimation of Distribution Algorithms

Fourth Generation ????

EA Come in 4 Different Basic Flavors

In this course, we will look at first generation EC

Rest = variations on the theme

1) Genetic Algorithms (GA)
2) Evolutionary Programming (EP)
3) Evolution Strategy (ES)
4) Genetic Programming (GP)

Standard Evolutionary Algorithm

```
// pm = mutation rate, pc = probability of crossover, f = fitness function
procedure [P] = standard_EA(pc,pm)
    initialize P                    // GENERATE a population
    f ← eval(P)                     // EVALUATE each member of the population
    P ← select(P,f)                 // SELECT members which had good evaluation
    t ← 1
    while not_stopping_criterion do
        P ← reproduce(P,f,pc)
        P ← variate(P,pm)
        f ← eval(P)                 // EVALUATE each member of the population
        P ← select(P,f)
        t ← t + 1
    end while
end procedure
```

Genetic Algorithm

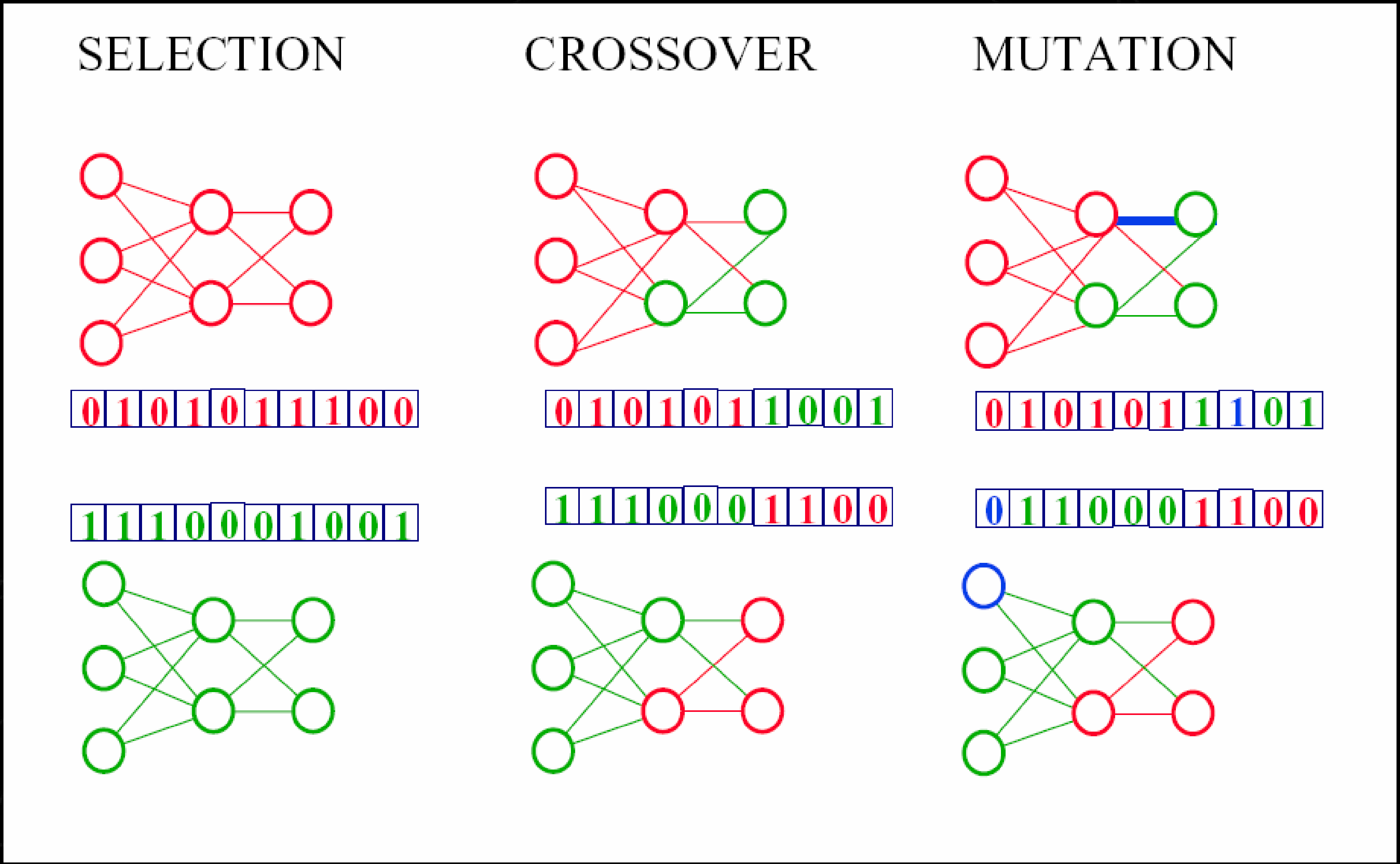
The *genetic algorithm* is a probabilistic search algorithm that iteratively transforms a set (called a population) of mathematical objects (typically fixed-length binary character strings), each with an associated fitness value, into a new population of offspring objects using the Darwinian principle of natural selection and using operations that are patterned after naturally occurring genetic operations, such as crossover (sexual recombination) and mutation

Genetic Algorithm

Four Main Elements

1. Group of individuals: one ore more populations which <u>reproduce</u> and whose individuals <u>develop</u>
2. <u>Selection</u> through reproductive fitness (survival of the fittest)
3. Source of variation: reproductive variability through genetic operators (mutuation and sexual reproduction)
4. Search process
   - Trial and error (+ selection of best variations)
   - Recipe for chosing next trial

Genetic Operators: Selection and Variation



Genetic Operators: Variation and Selection

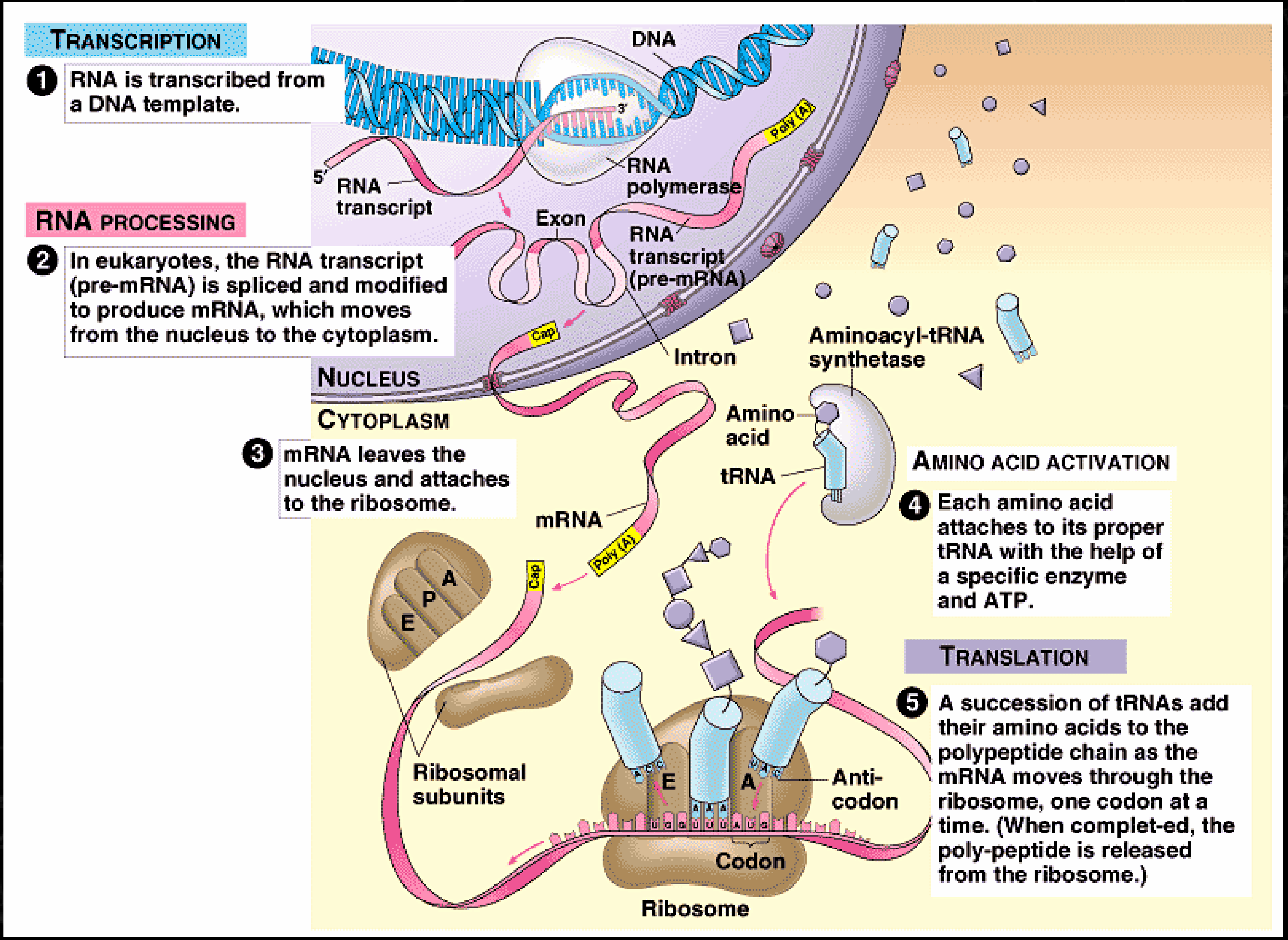
Sources of variation:

- Sexual reproduction (e.g. crossover or recombination): decomposes two distinct solutions (individuals) and then randomly mixes their parts to form novel solutions (individuals)
- Mutation: randomly perturbs a candidate solution

Selection:

Replicates the most successful solutions found in a population at a rate proportional to their relative quality

Information Processing in a Cell



Analogies With Biology

Genotype (set of chromosomes)

→ Structures and strings (usually strings of bits)

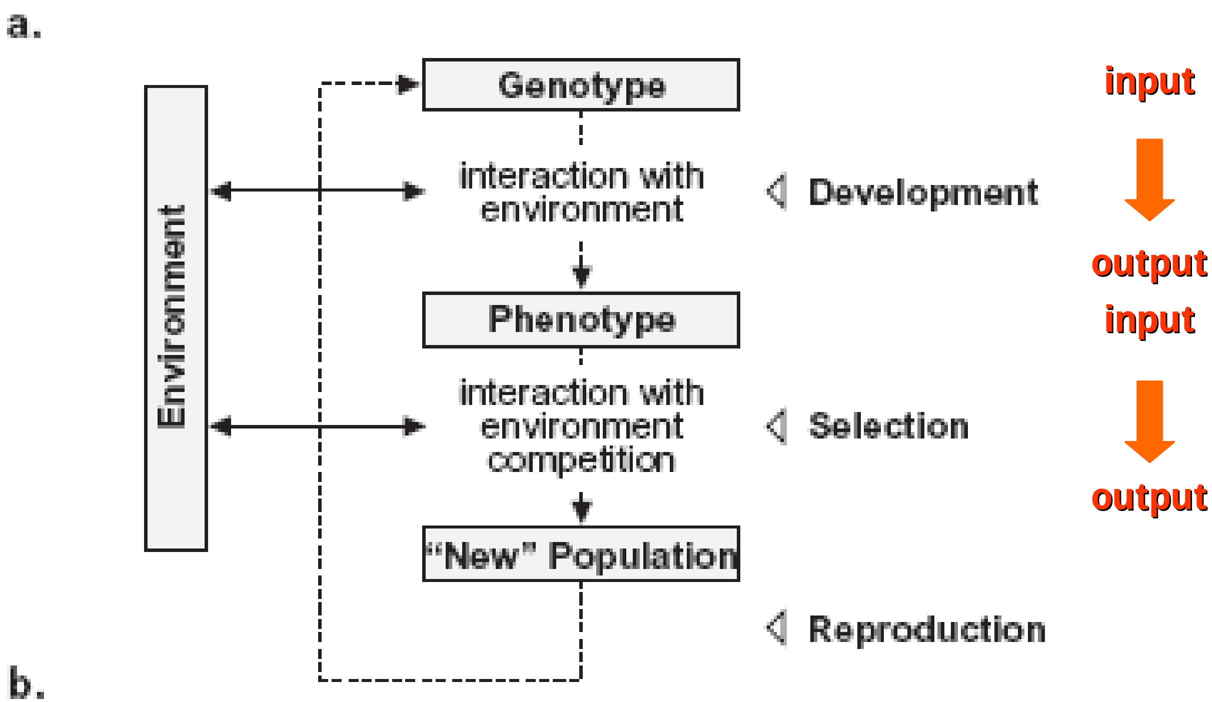
Phenotype (organism/individual formed with a certain fitness)

→ A solution (a set of parameter values)

Genetic Algorithm: Pseudo-Code

```
set population size and number of generations;
set generation counter g = 0;
initialize P(0);    // 0-th population (usually random)
repeat (until terminated) {
    evaluate each fitness of each individual of P(t);

    check for termination criteria (number of generations,
        amount of time, minimum fitness threshold satisfied,
        fitness has reached a plateau, other);

    select pairs to mate from best-ranked individuals (most
        fit individuals);

    apply genetic operators (crossover and mutation);

    g = g + 1;

    }
```

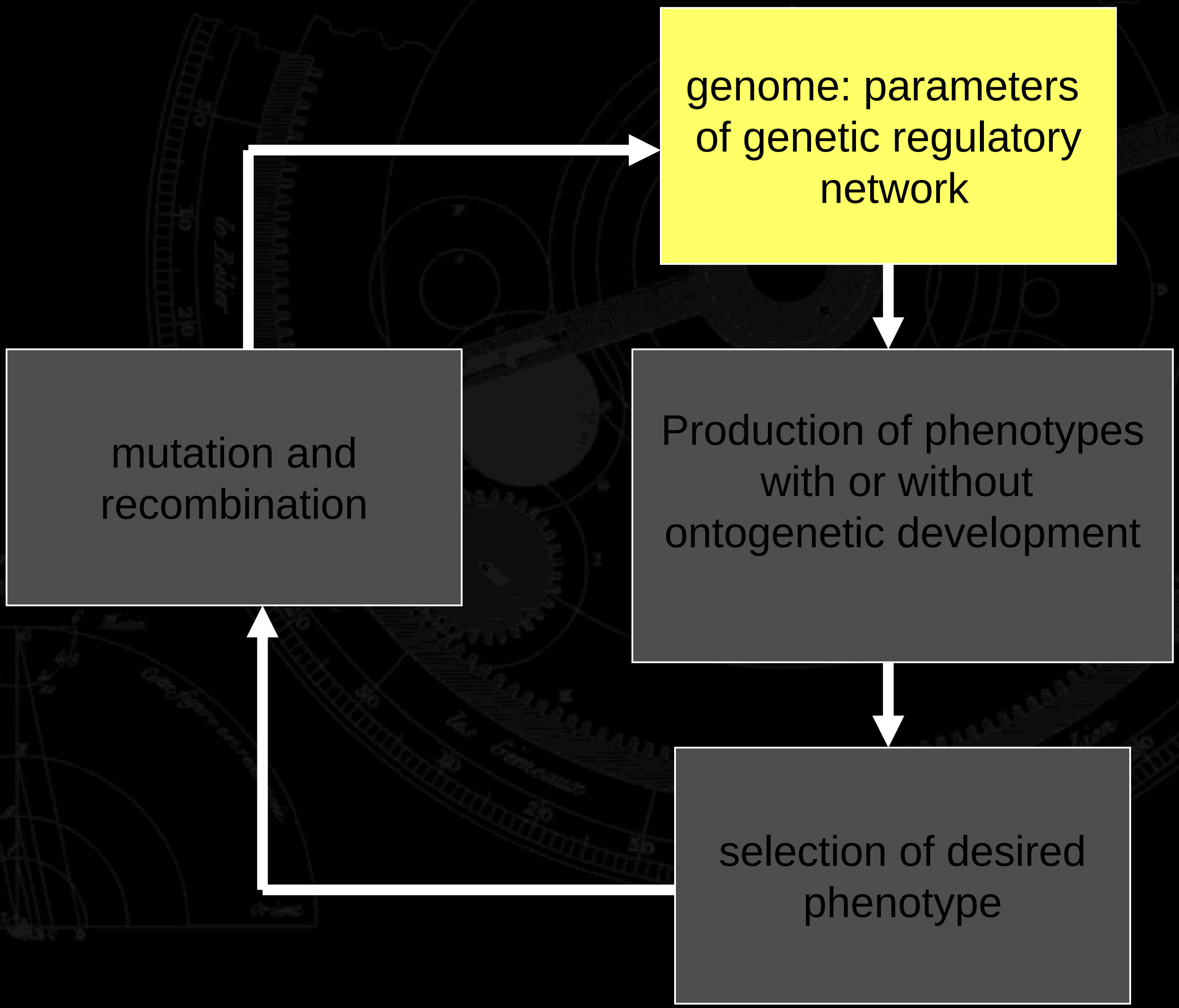
Genetic Algorithm: Basic Structure



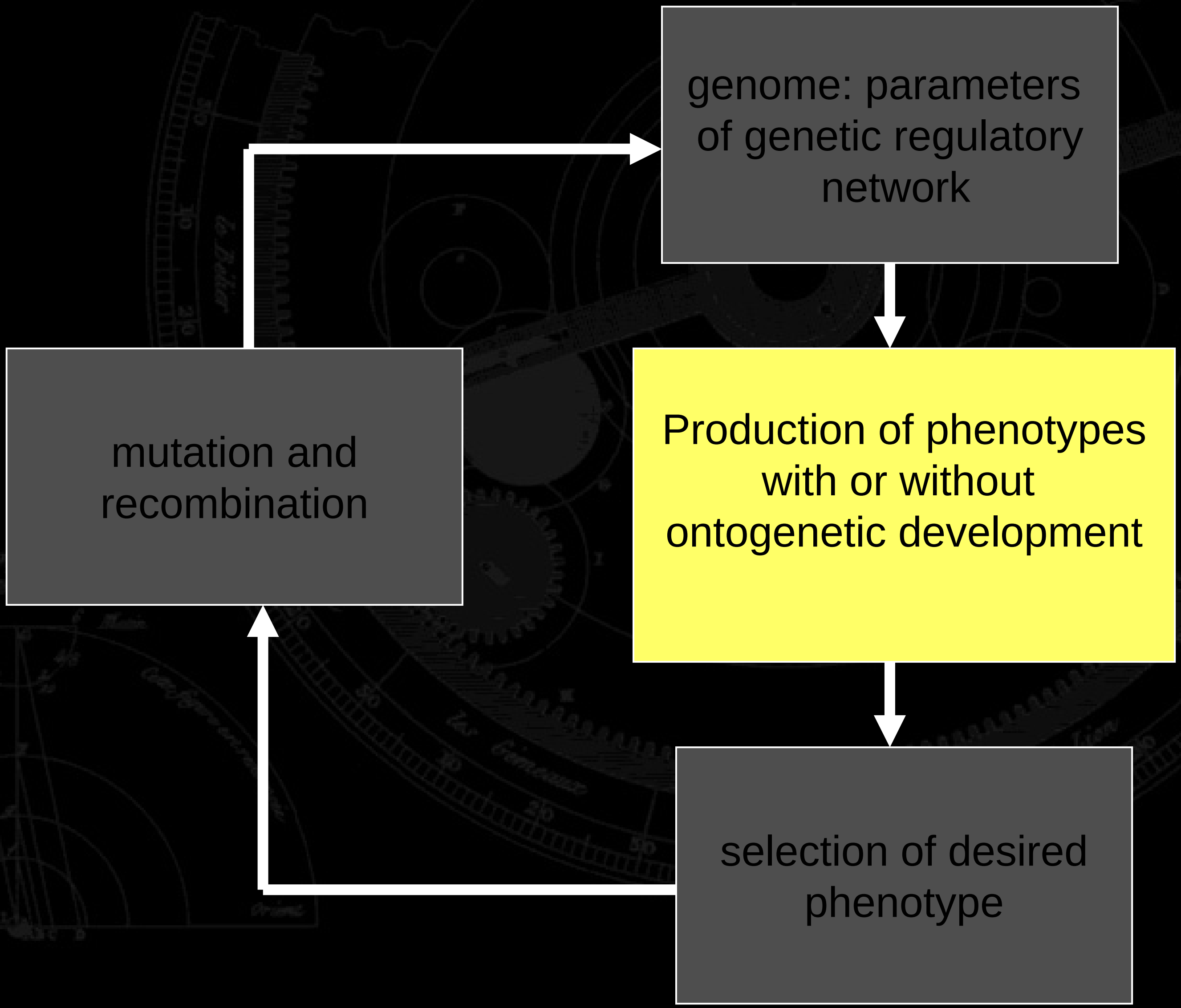
b.

encoding	development	selection	reproduction
• binary	• no development (phenotype = genotype)	• "roulette wheel"	• mutation
• many-character	• development with and without interaction with the environment	• elitism	• crossover
• real-valued		• rank selection	
		• tournament	
		• truncation	
		• steady-state	

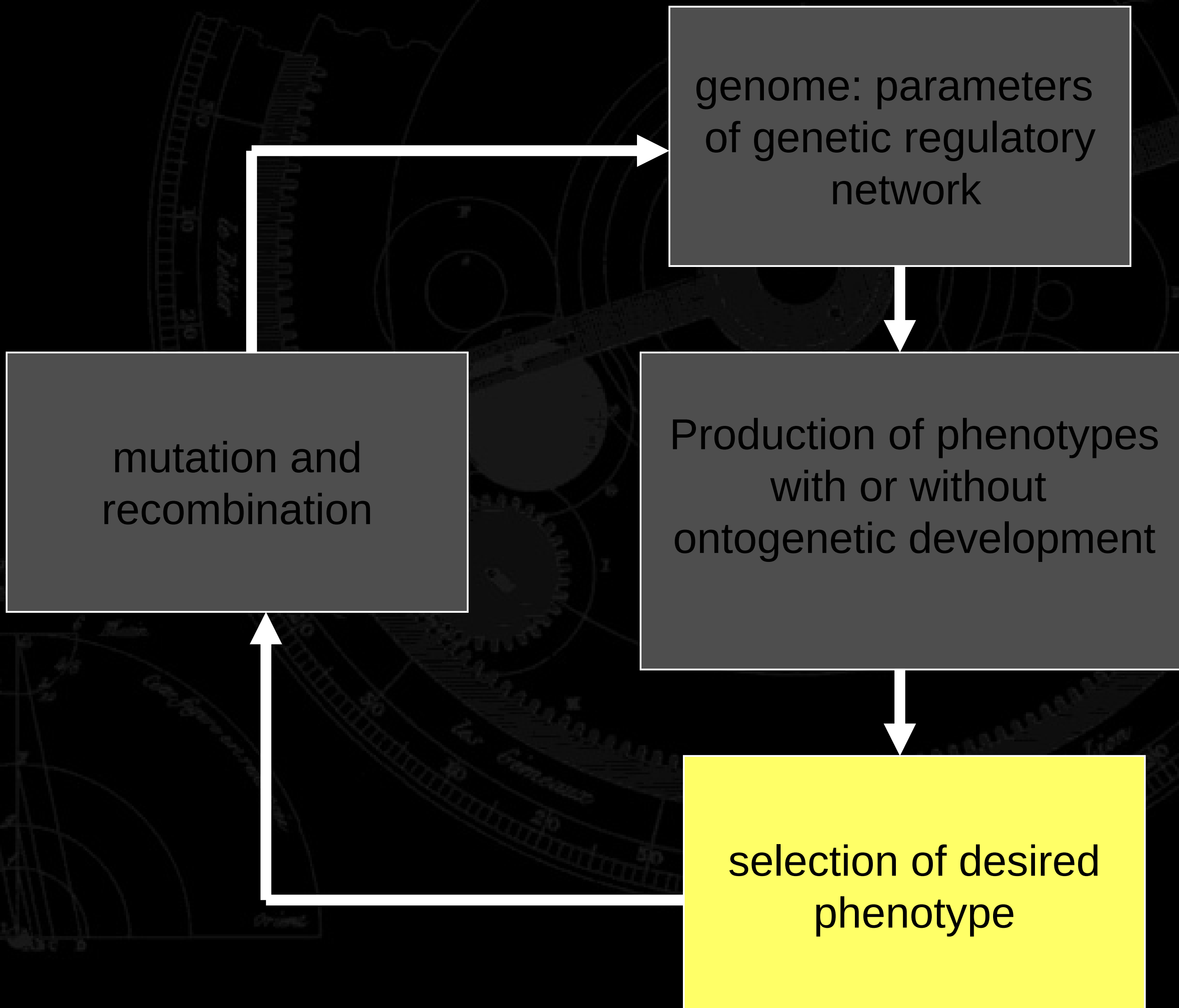
GA: Basic Algorithm



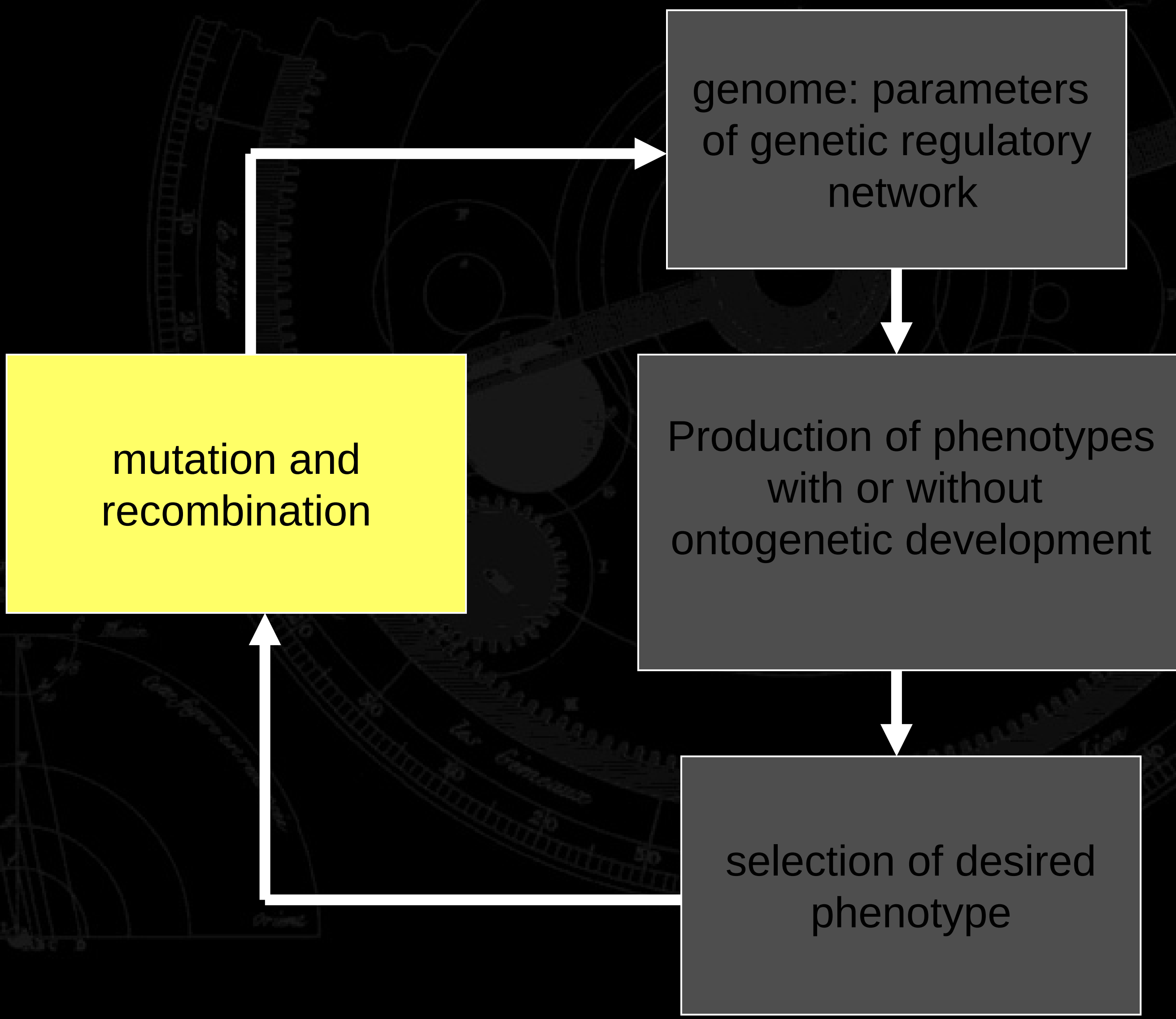
GA: Basic Algorithm



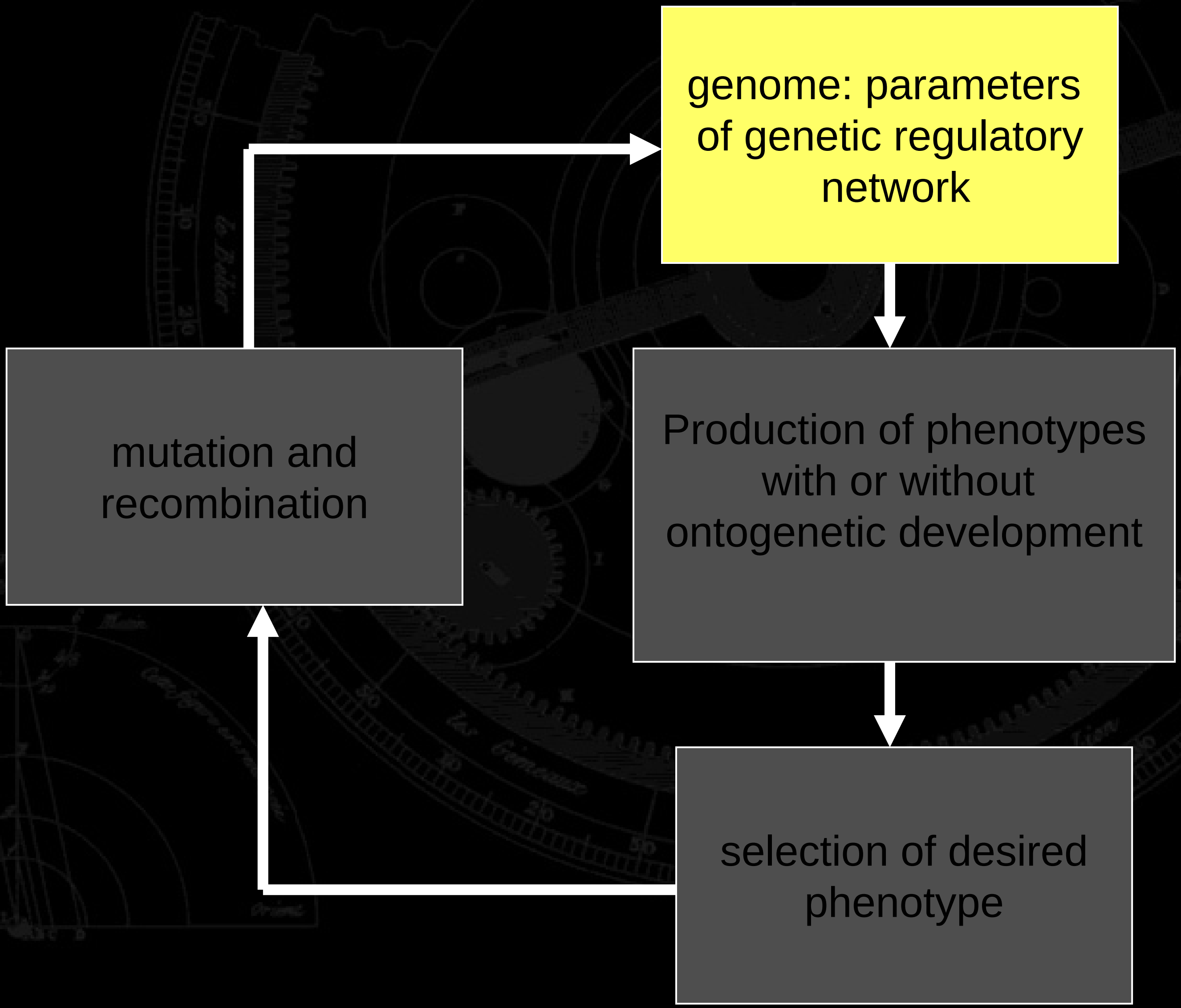
GA: Basic Algorithm



GA: Basic Algorithm



GA: Basic Algorithm

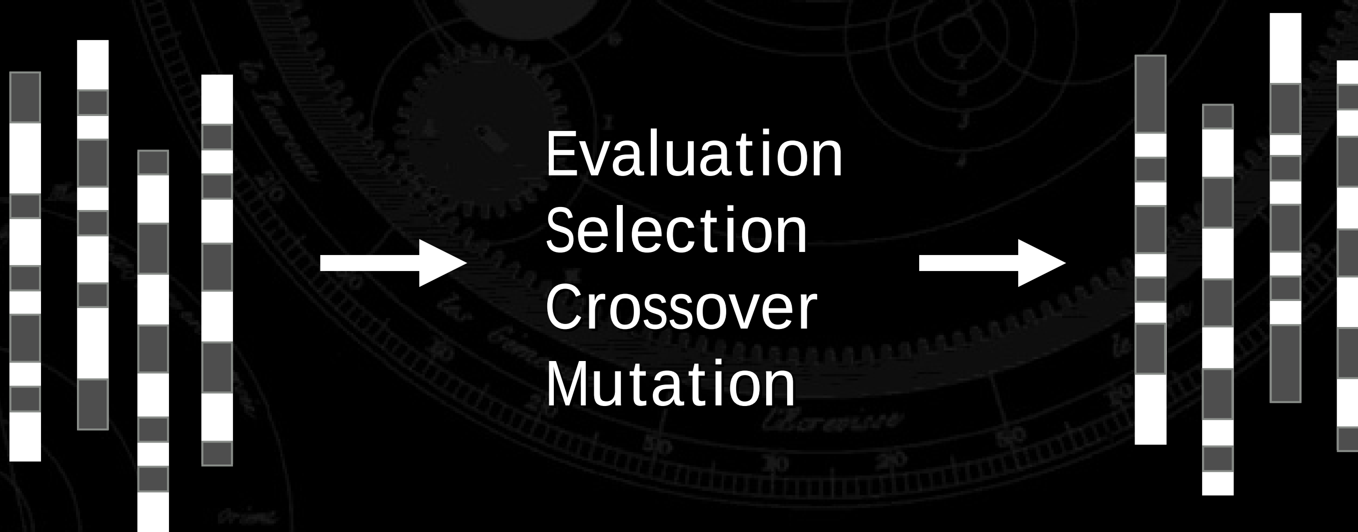


Genetic Algorithms: Encoding

Holland's original approach, and probably still the most common

Genome (or *chromosome*) is a string of bits

1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0



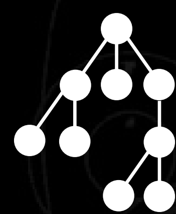
Information Processing in a GA

- Typically, in GAs the electronic genotype is nothing more than a long string of bits
- When a bit in the genotype is „on“ (has the value 1), a certain characteristic is apparent in the artificial organism
- When a bit is „off“ (has the value 0), that characteristic is missing

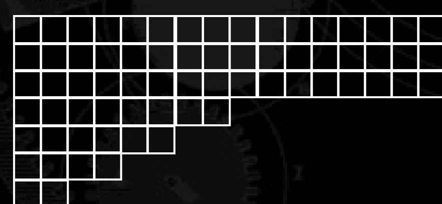
Other Encoding Schemes

The basic vary-and-select GA approach can be applied to almost any data structure

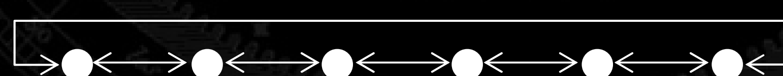
Trees



Arrays



Lists



Many-character encoding

A B C D A B B B A C C A

Genetic Algorithm: Development

Often, development is entirely neglected; in its absence selection is performed directly on the genotype

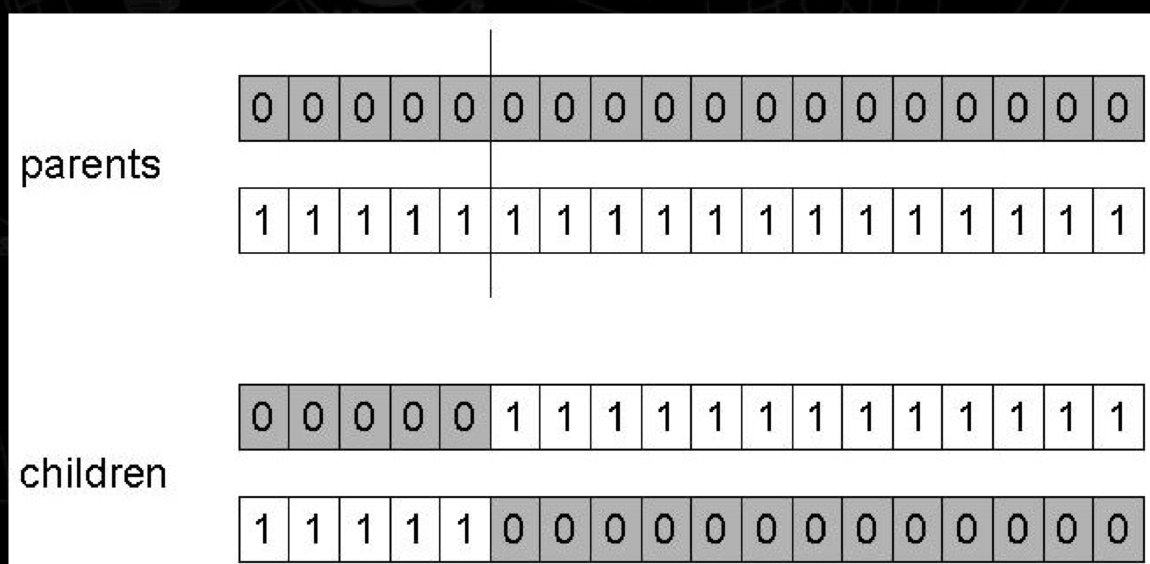
A trivial form of development the representation is mapped directly onto the organism's features without interaction with the environment

In more complex forms, there is strong interaction with the environment (e.g. Eggenberger's Artificial Evolutionary System and Bongard's Artificial Ontogeny)

Genetic Algorithm: 1-Point Crossover

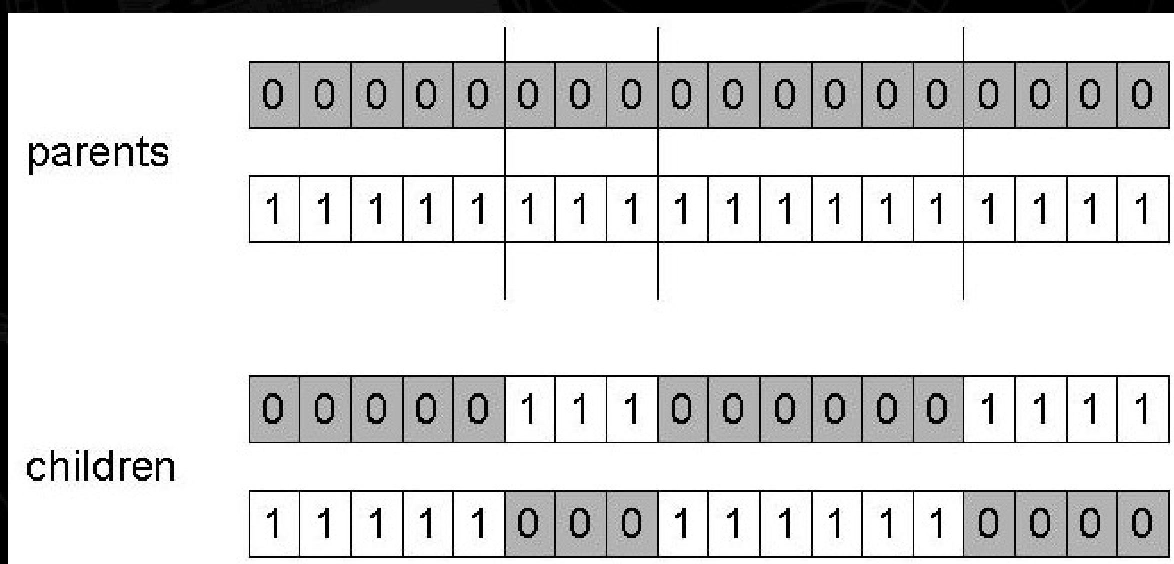
There are a number of techniques, but all involve swapping genes—sequences of bits in the strings—between two haploid individuals (or between two strands of a diploid individual)

1. Choose a random point on the two parents (1-point crossover)
2. Split parents at this crossover point
3. Create children by exchanging tails
4. Crossover probability P_c typically in range (0.6, 0.9)



Genetic Algorithm: n-Point Crossover

1. Choose n random crossover points
2. Split along those points
3. Glue parts, alternating between parents
4. Generalisation of 1 point (still some positional bias)



Genetic Algorithm: Mutation

This explores parts of the space that crossover might miss, and helps prevent premature convergence (generally, bits are flipped with a small probability)

1. Alter each gene independently with a probability p_m
2. p_m is called the mutation rate: Typically between 1/pop_size and 1/chromosome_length

parent	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
child	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1

Genetic Algorithm: Xover or Mutation?

Decade long debate: which one is better / necessary / main-background

Answer (at least, rather wide agreement):

- it depends on the problem, but in general, it is good to have both
- both have another role
- mutation-only-EA is possible, xover-only-EA would not work (see Evolution Strategy)

Genetic Algorithm: Xover or Mutation?

- Exploration: Discovering promising areas in the search space, i.e. gaining information on the problem
- Exploitation: Optimising within a promising area, i.e. using information
- There is <u>cooperation</u> AND <u>competition</u> between them
- Crossover is explorative, it makes a *big* jump to an area somewhere "in between" two (parent) areas
- Mutation is exploitative, it creates random *small* diversions, thereby staying near (in the area of) the parent

Genetic Algorithm: Selection

Main idea: more fit individuals get higher chance of survival

The *selection scheme* determines how individuals are chosen for mating, based on their fitness scores

Too great a bias towards the best individuals can result in premature convergence, so the best selection schemes are designed to maintain a diverse population (exploration-exploitation tradeoff)

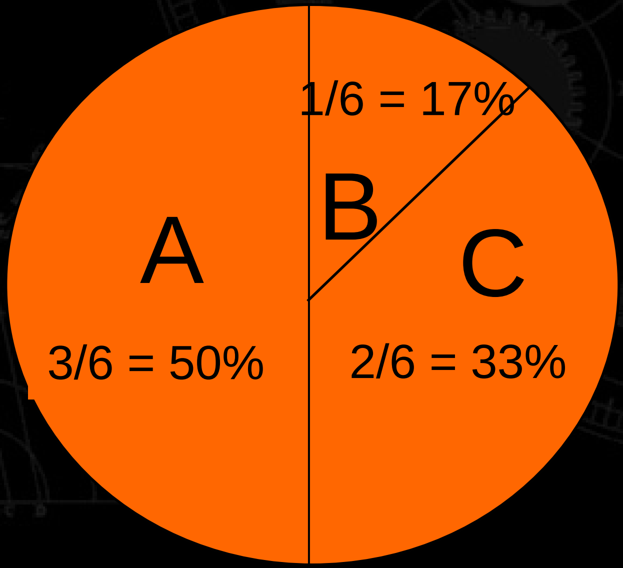
Genetic Algorithm: Selection

Popular selection schemes:

- *Roulette wheel* - probability of selection is proportional to fitness
- *Rank* - pick the best individuals every time (probability is proportional to rank)
- *Tournament* - initial large number of individuals are selected via roulette wheel or ranking, then the best ranked are chosen
- *Elitism* - in combination with other selection schemes, always keep the fittest individual around
- *Steady-state* selection - only small part of the population changes at any particular time, while the rest is preserved

Roulette Wheel Sel. (Fitness-Proportionate)

- Each individual is assigned a slice of circular "roulette wheel"; the size of the slice is proportional to the individual's fitness
- Wheel is spun as many times as the number of individuals
- On each spin, the individual under the wheel's marker is selected to be a parent for the next generation



fitness(A) = 3

fitness(B) = 1

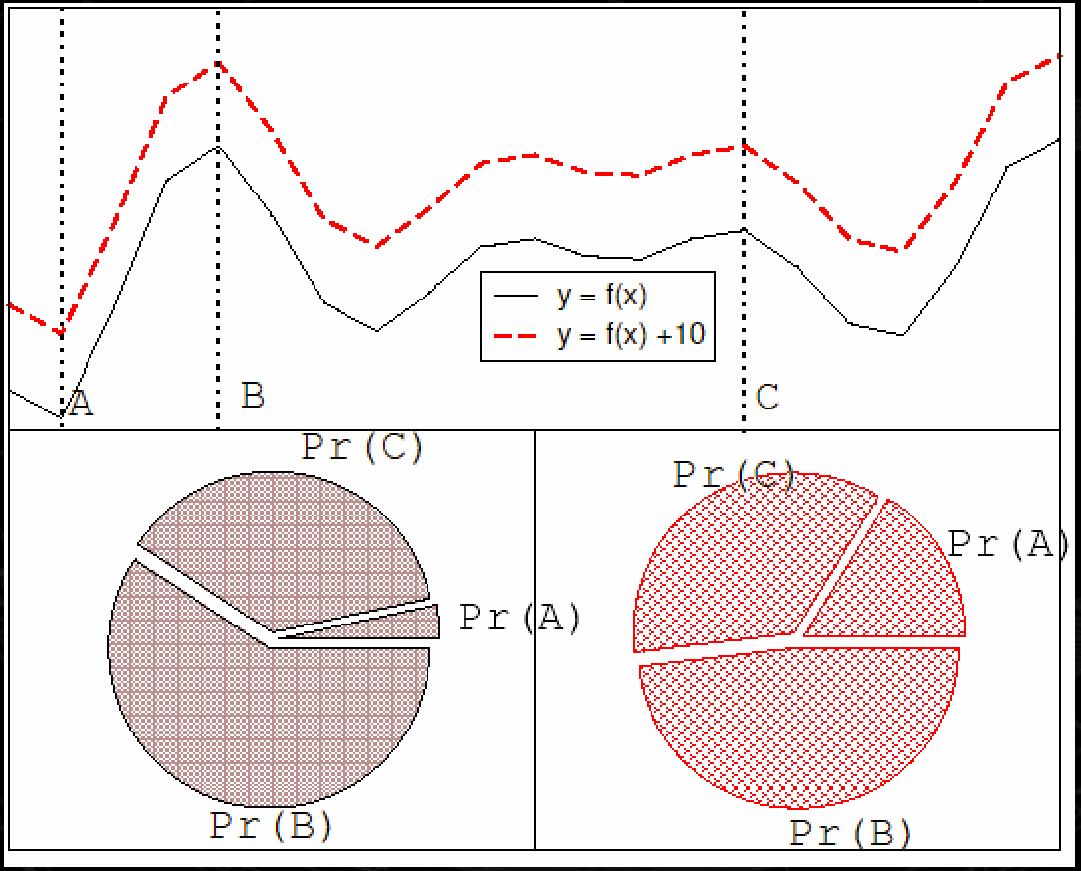
fitness(C) = 2

Roulette Wheel Selection

Problems include

- One highly fit member can rapidly take over if rest of population is much less fit: Premature convergence
- At end of runs when fitnesses are similar, lose selection pressure (weak convergence)
- Highly susceptible to function transposition (because slice of roulette is proportional to absolute value of fitness!)

Function Transportation for RWS



Slice of roulette is proportional to absolute value of fitness!

Rank-Based Selection

Attempt to remove problems of RWS by basing selection probabilities on *relative* rather than *absolute* fitness

Rank population according to fitness and then base selection probabilities on rank where fittest has rank μ and worst rank 1

This imposes a sorting overhead on the algorithm, but this is usually negligible compared to the fitness evaluation time

Linear Ranking

$$P_{lin-rank}(i) = \frac{(2-s)}{\mu} + \frac{2i(s-1)}{\mu(\mu-1)}$$

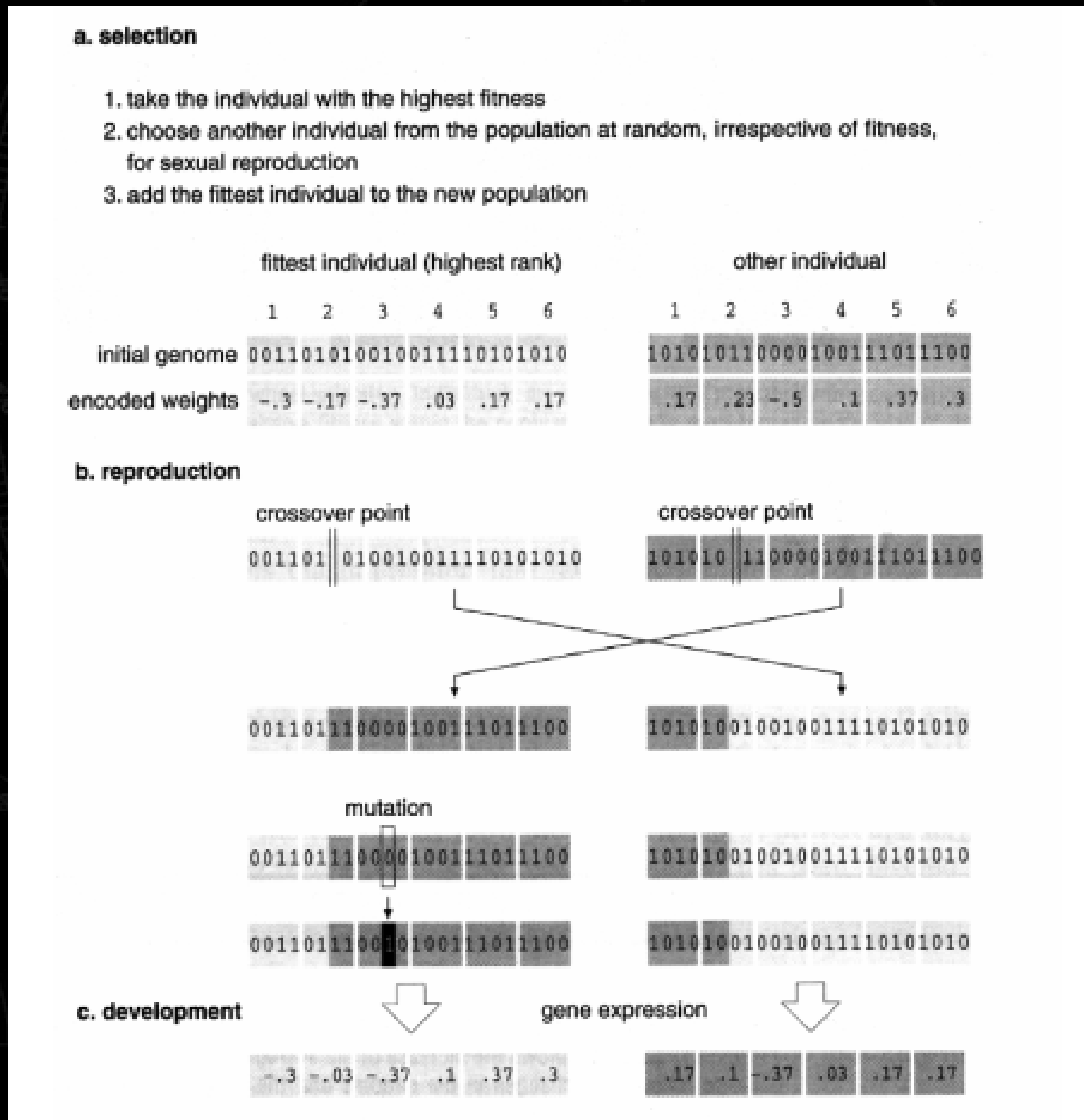
Parameterised by factor *s:* 1.0 < *s* ≤ 2.0

- measures advantage of best individual

Simple 3 member example

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

GA: Selection, Reproduction, Development



Evolution = Climbing Fitness Hills

Example after Goldberg (1989)

Simple maximization problem:

$f(x) = \max(x^2)$ over $\{0,1,\ldots,31\}$

GA approach:

- Encoding: binary code, e.g. 01101 $\leftrightarrow$ 13
- Population size: 4
- 1-point xover, bitwise mutation
- Roulette wheel selection
- Random initialization

X^2 Example: Selection

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

X^2 Example: Crossover

String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

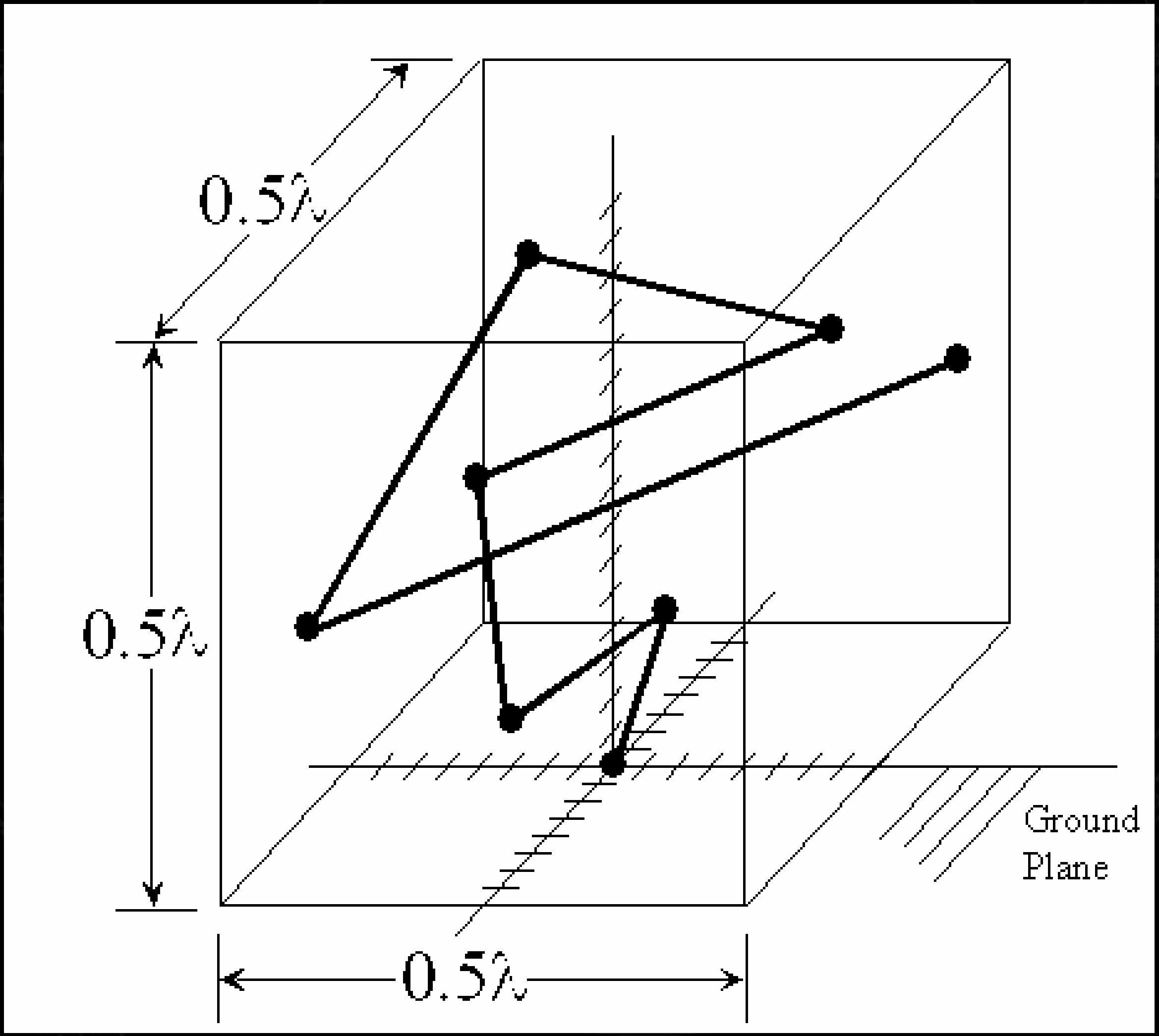
X^2 Example: Mutation

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	$\boxed{1}$ 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 $\boxed{1}$ 0 0	18	324
Sum				2354
Average				588.5
Max				729

X^2 Example: Mutation

Loop ...

Additional Example: Antenna



Antenna Design

The problem (Altshuler and Linden 1998) is to determine the *x-y-z* coordinates of the 3-dimensional position of the ends (*X1*, *Y1*, *Z1*, *X2*, *Y2*, *Z2*,… , *X7*, *Y7*, *Z7*) of 7 straight wires so that the resulting 7-wire antenna satisfies certain performance requirements

The first wire starts at feed point (0, 0, 0) in the middle of the ground plane

The antenna must fit inside the 0.5λ cube

Antenna Design: Genome

X_1	Y_1	Z_1	X_2	Y_2	Z_2	…
+0010	**-1110**	**+0001**	**+0011**	**-1011**	**+0011**	**…**

105-bit chromosome (genome)

Each *x-y-z* coordinate is represented by 5 bits (4-bit granularity for data plus a sign bit)

Total chromosome is $3 \times 7 \times 5 = 105$ bits

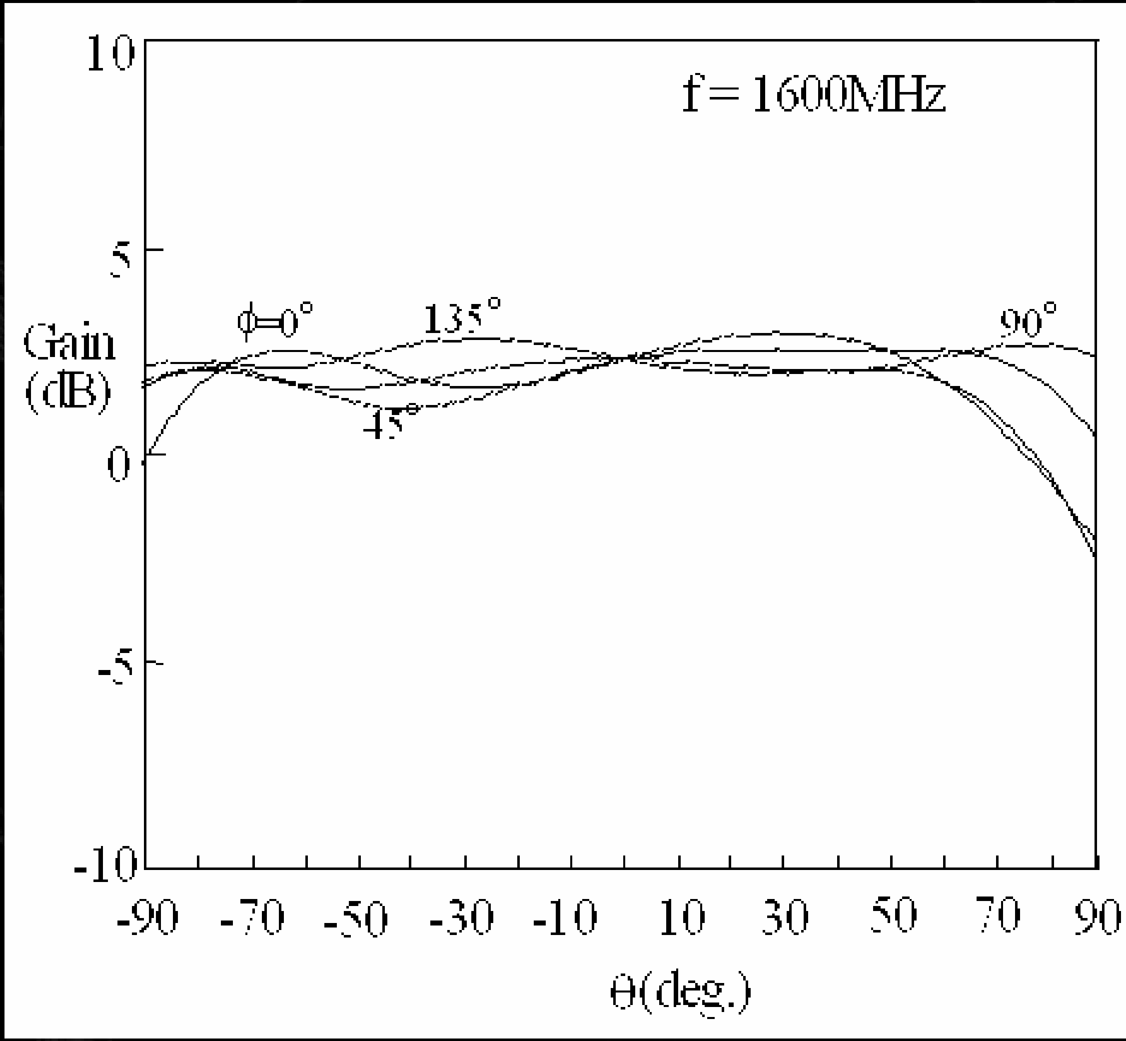
Antenna Design: Fitness

- Antenna is for ground-to-satellite communications for cars and handsets
- We desire near-uniform gain pattern 10° above the horizon
- Fitness is measured based on the antenna's radiation pattern. The radiation pattern is simulated by National Electromagnetics Code (NEC)

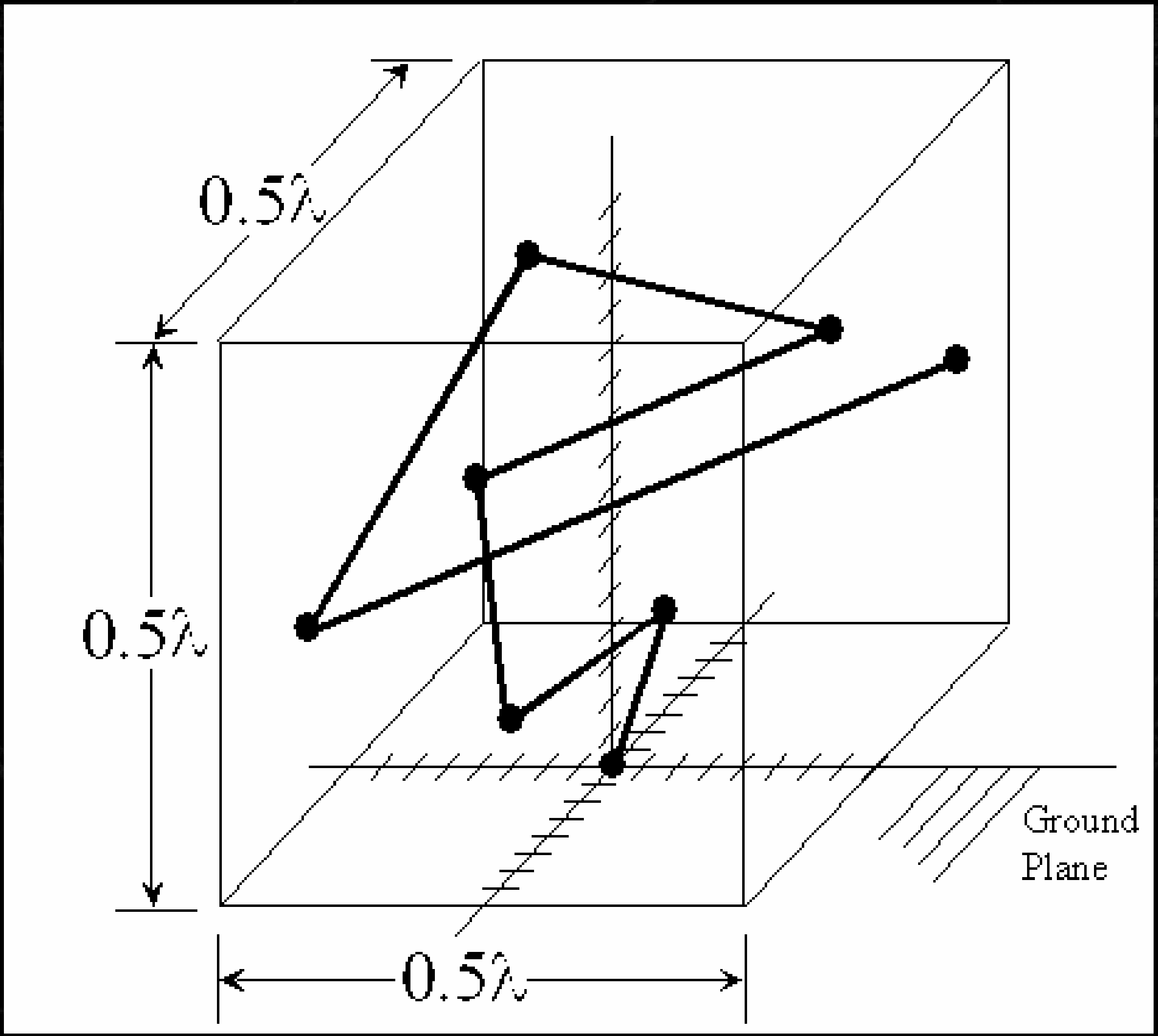
Antenna Design: Fitness

- Fitness is sum of the squares of the difference between the average gain and the antenna's gain
- Sum is taken for angles Θ between -90° and +90° and all azimuth angles Φ from 0° to 180°
- The smaller the value of fitness, the better

Graph of Antenna Fitness



U.S. Patent 5,719,794



Other Patents

For other instances in which genetic programming has produced human-competitive results see:

http://www.genetic-programming.com/humancompetitive.html

Additional Example: Robot

Example of evolutionary robotics

Robot has 3 sensors and 2 motors

ANN has 6 weights

EP and ES: Differences to GAs

1. The representation of a solution in an EP follows directly from the problem and is not constrained to be in the form of a string of bits/characters, as in the GA
2. To solve a multidimensional function approximation problem with an EP/ES, we can encode a solution as a vector of floating-point numbers
3. EP/ES do not attempt to closely model genetic operations
4. For the mutation operator it is typical to use multivariate Gaussian distributions instead of raw bit-flipping as in GAs
5. The rate of mutation is typically reduced as the optimal solution is approached

ES: Basic Algorithm (mutation-selection)

```
set t = 0;
create initial point x^t = ⟨ x_1^t,…,x_n^t ⟩;
repeat until (TERMIN.COND satisfied) do
    draw z_i from a normal distr. for all i = 1,…,n
    y_i^t = x_i^t + z_i;        // mutation
    if (  f(x^t) < f(y^t) )  // selection
        x^{t+1} = x^t;
    else
        x^{t+1} = y^t;
    fi
    set t = t+1;
od
```

ES: Basic Algorithm

z values drawn from normal distribution $N(\xi,\sigma)$

- mean ξ is set to 0
- variation σ is called mutation step size

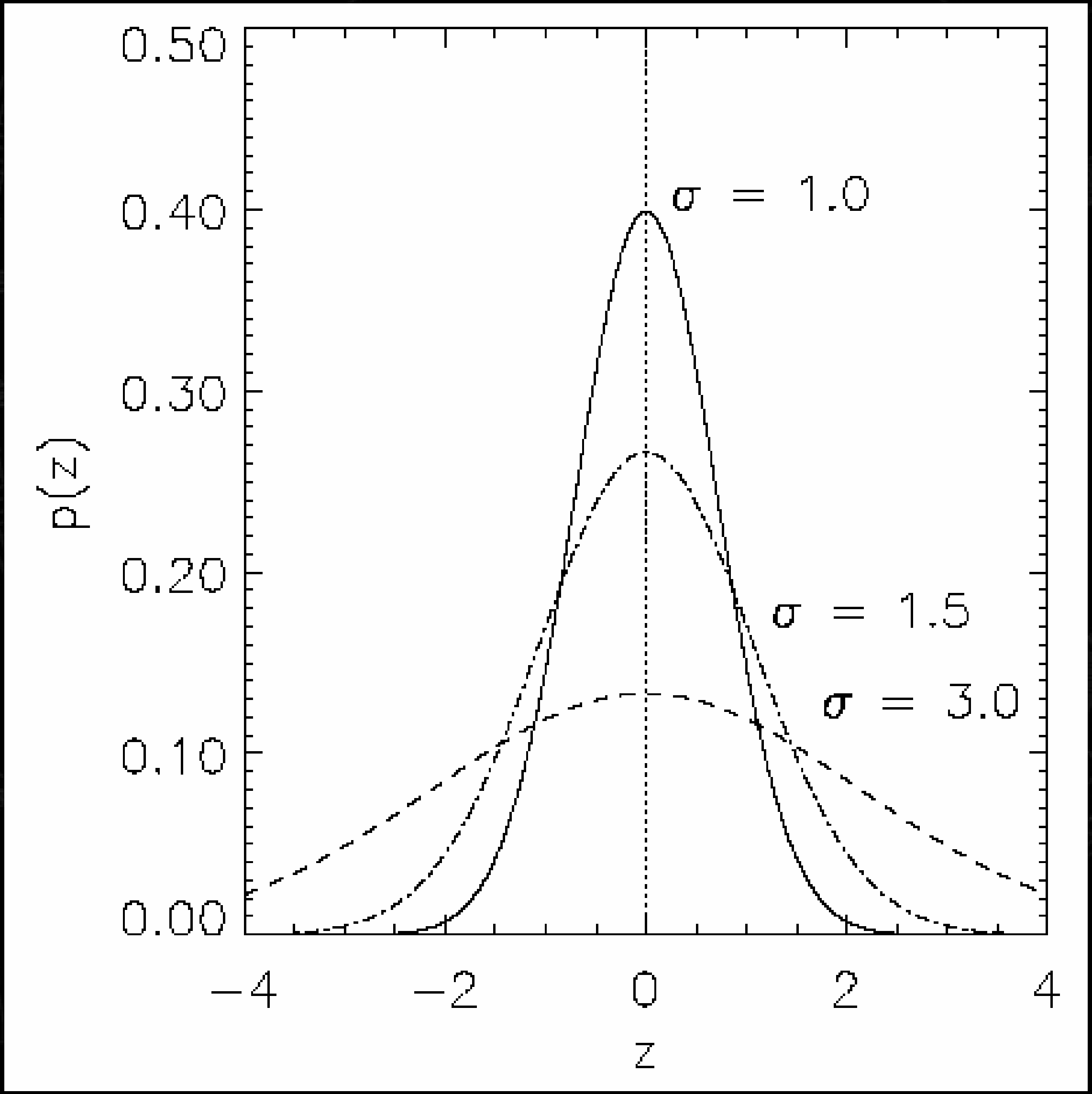
σ is varied on the fly by the "1/5 success rule":

This rule resets σ after every k iterations by

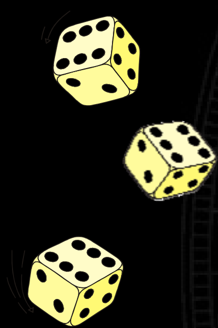
- $\sigma = \sigma / c$ if $p_s > 1/5$
- $\sigma = \sigma \cdot c$ if $p_s < 1/5$
- $\sigma = \sigma$ if $p_s = 1/5$

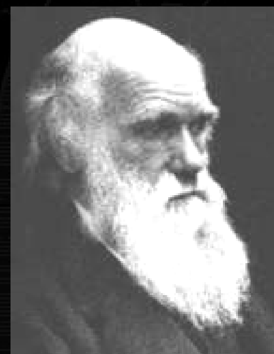
where p_s is the % of successful mutations, $0.8 \le c \le 1$

Illustration of Normal Distribution

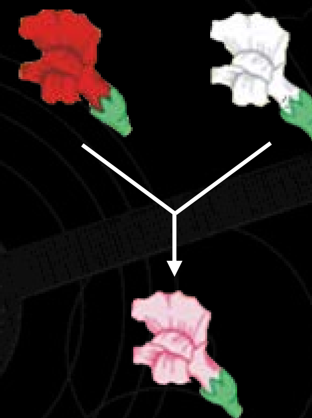


Evolution Strategy





+



+

Wright Haldane Fisher

$$\left[\mu'/\rho' \overset{+}{,} \lambda'(\mu/\rho \overset{+}{,} \lambda)^{\gamma}\right]^{\gamma'}\text{-ES}$$

μ' = Number of parental populations

ρ' = Mixing number for populations

λ' = Number of offspring populations

γ' = Number of population generations

μ = Number of parental individuals

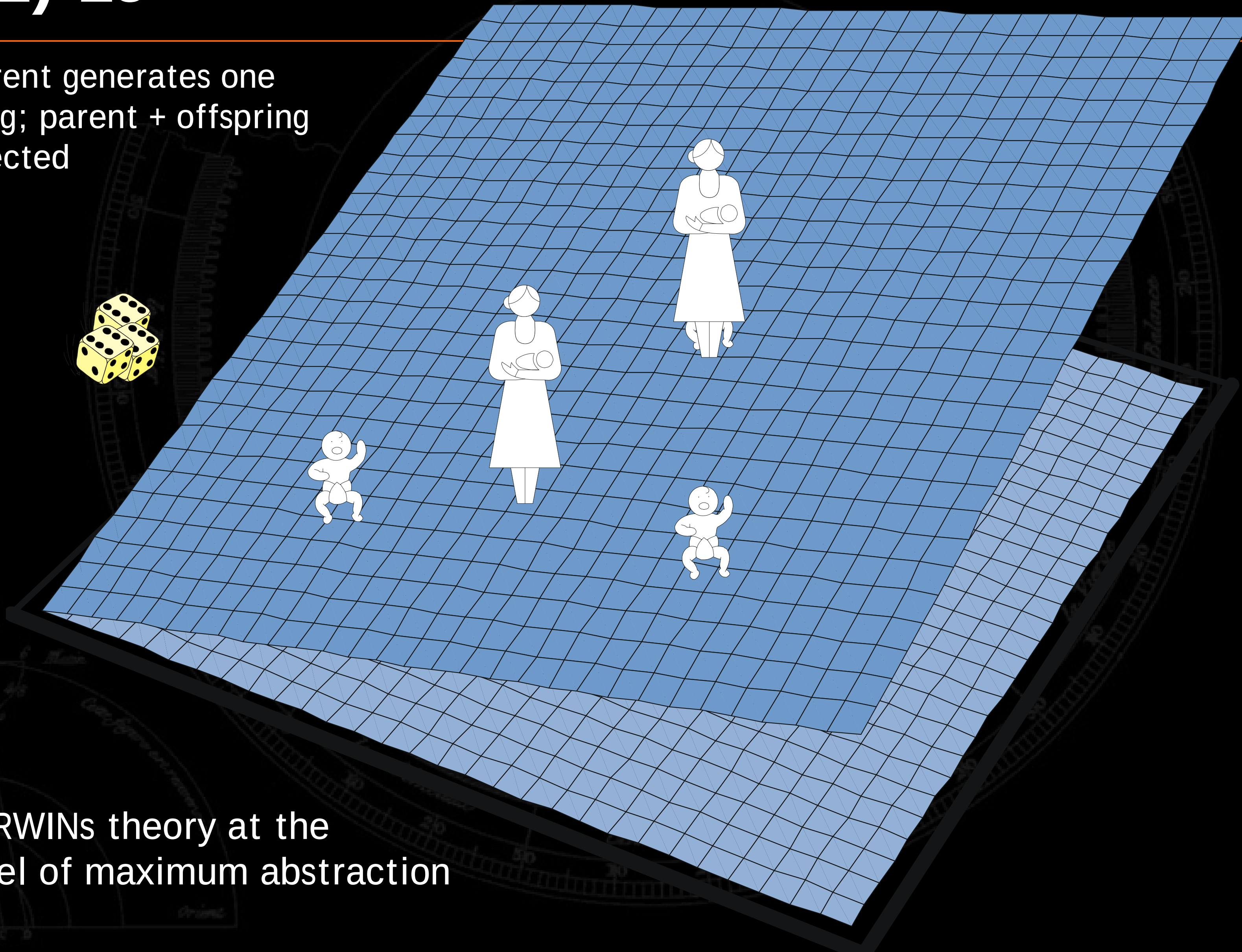
ρ = Mixing number for individuals

λ = Number of offspring individuals

γ = Generations of isolation

(1+1)-ES

One parent generates one offspring; parent + offspring are selected

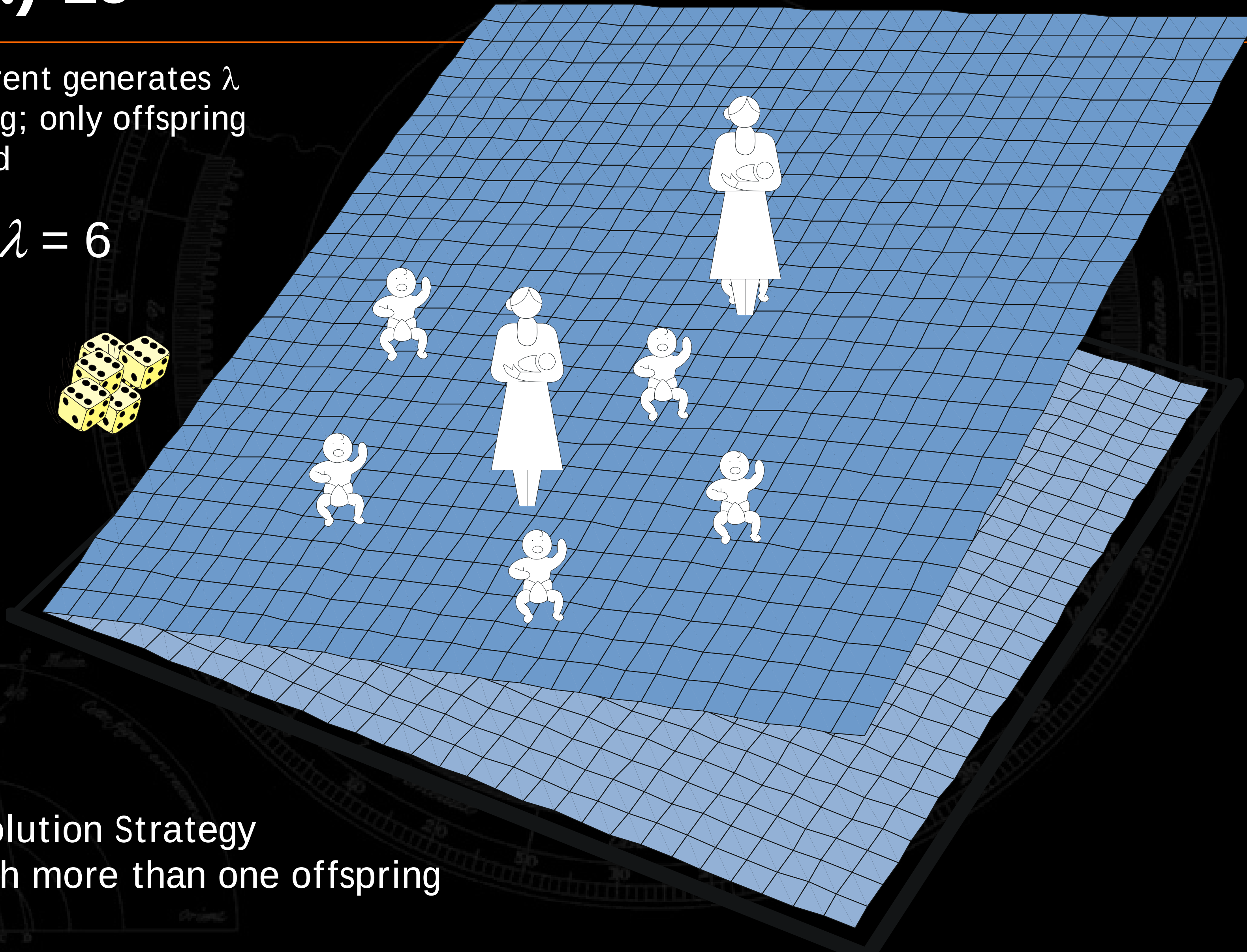


DARWINs theory at the level of maximum abstraction

(1,λ)-ES

One parent generates λ offspring; only offspring selected

$\lambda = 6$

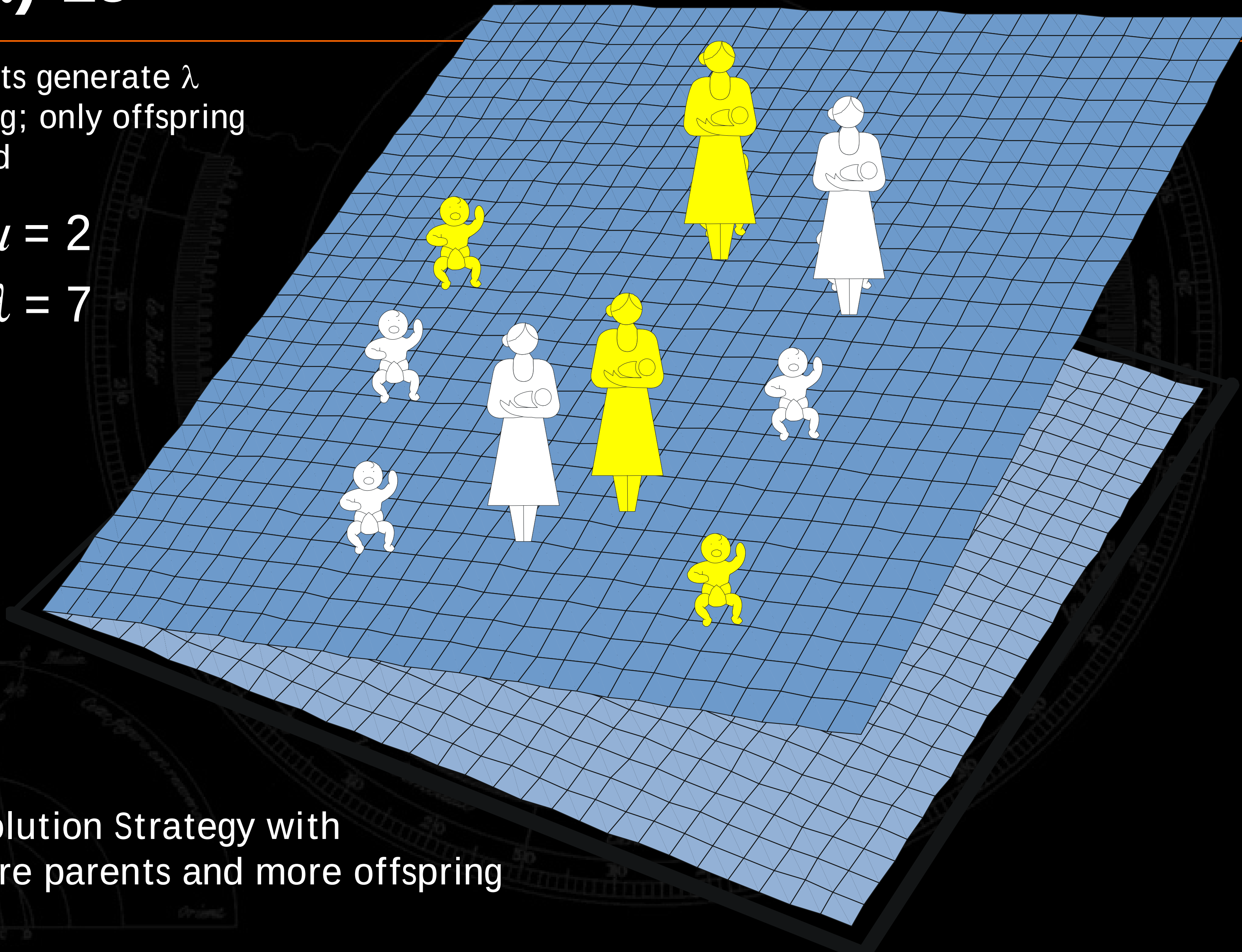


Evolution Strategy
with more than one offspring

(μ,λ)-ES

μ parents generate λ offspring; only offspring selected

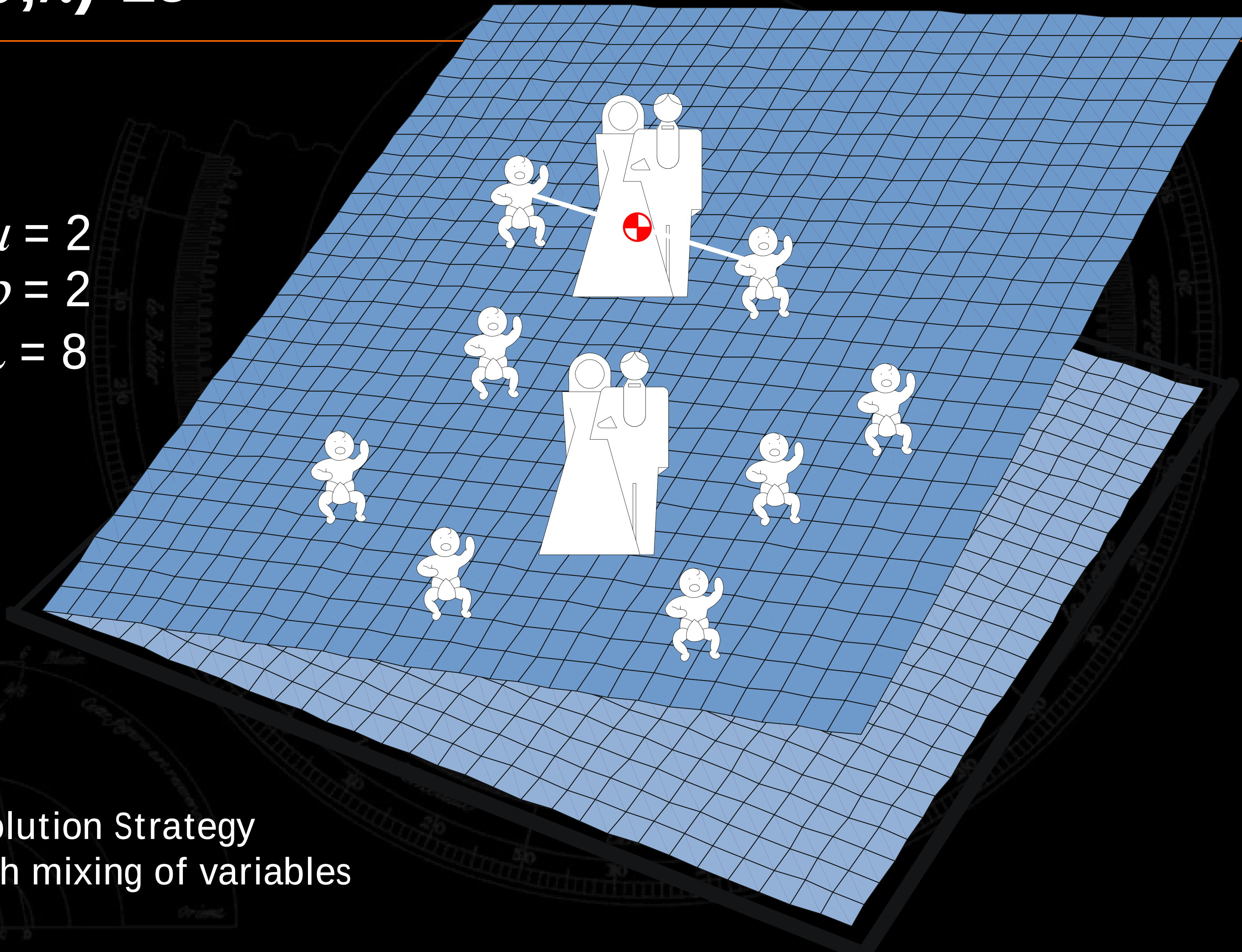
$$\mu = 2$$
$$\lambda = 7$$



Evolution Strategy with
more parents and more offspring

($\mu/\rho,\lambda$)-ES

$\mu = 2$
$\rho = 2$
$\lambda = 8$



Evolution Strategy
with mixing of variables

$$[\mu', \lambda'(\mu, \lambda)^{\gamma}] - \mathrm{ES}$$

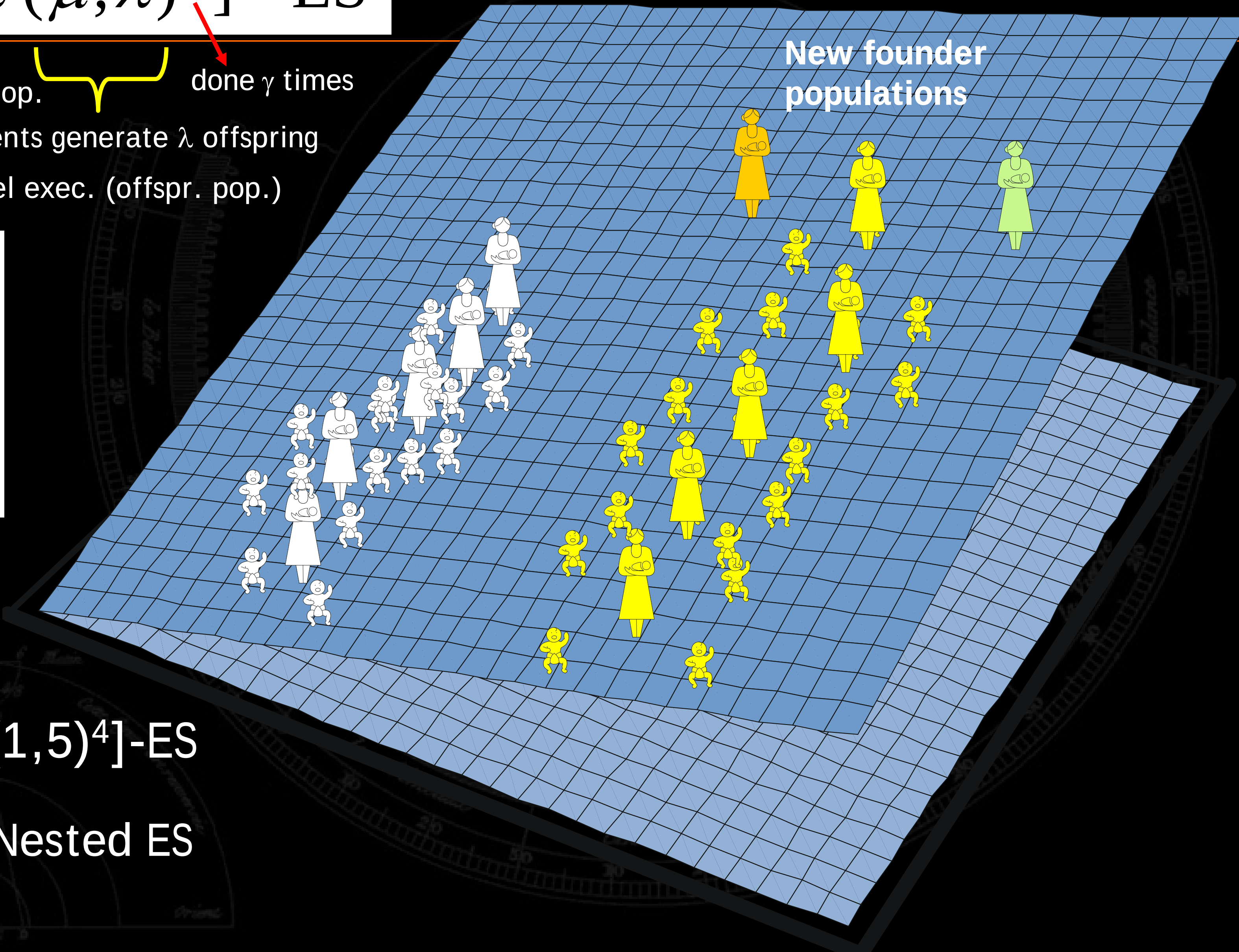
#parent pop.

done γ times

μ parents generate λ offspring

#parallel exec. (offspr. pop.)

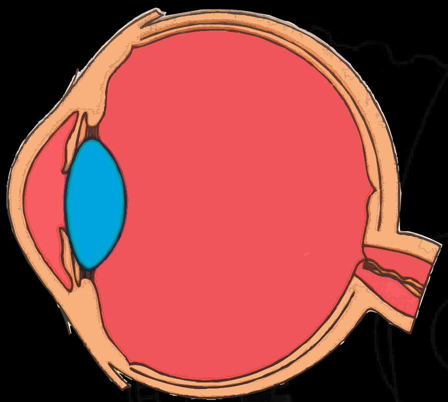
$$\begin{aligned} \mu' &= 1 \\ \lambda' &= 2 \\ \mu &= 1 \\ \lambda &= 5 \\ \gamma &= 4 \end{aligned}$$



$[1,2(1,5)^4]$-ES

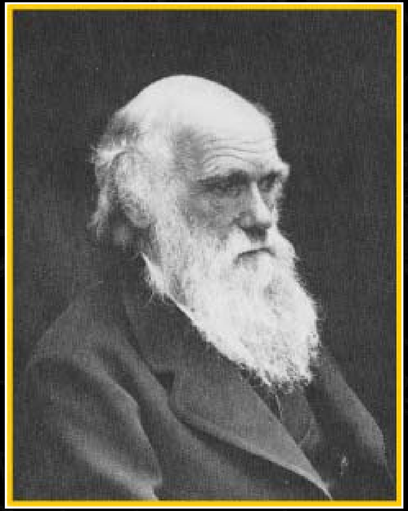
The Nested ES

Example of ES: Evolution of an Eye Lens



„To suppose that the eye, with all its inimitable contrivances for adjusting the focus to different distances, for admitting different amounts of light, and for the correction of spherical and chromatic abberation, could have been formed by natural selection, seems, I freely confess, absurd in the highest possible degree.“

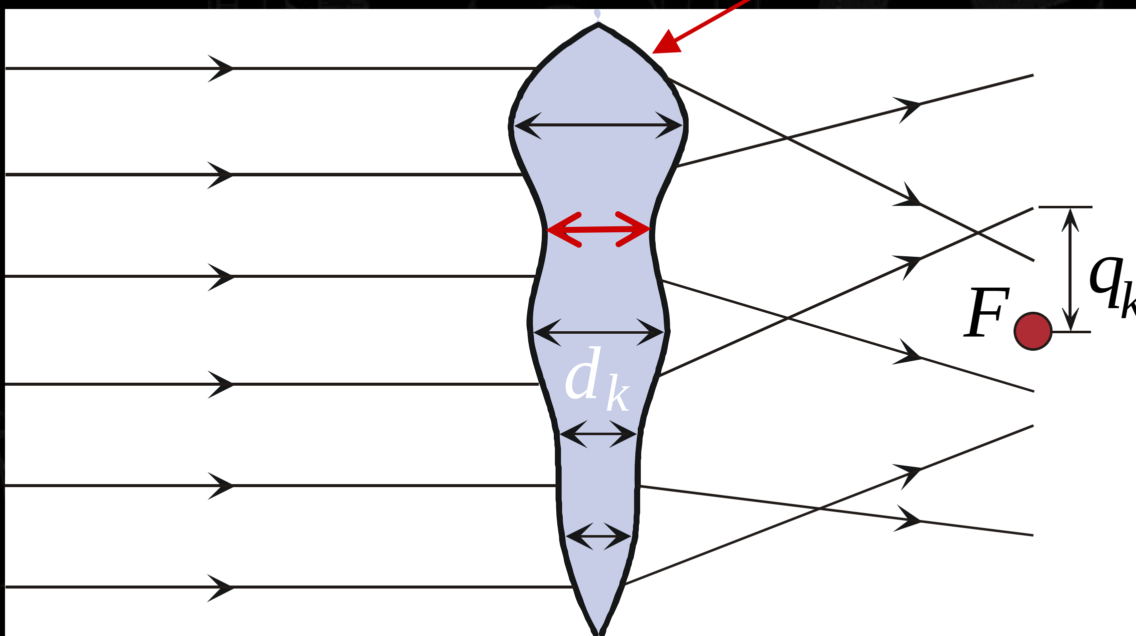
From Charles Darwin: *„The Origin of Species“*



Evolution of an Eye Lens

Computer-simulated evolution of a convergent lens

Flexible glass body



$$\sum q_k^2 \rightarrow \text{Minimum}$$

Shape of lens is appropriately parameterized (encoded in the genome)

http://www.bionik.tu-berlin.de/institut/xstart.htm

Genetic Programming

"How can computers learn to solve problems without being explicitly programmed? In other words, how can computers be made to do what is needed to be done, without being told exactly how to do it?" *(Attributed to A. Samuel, 1959)*

Genetic Programming

Goal: Evolve entire computer programs (Koza, 1992; 1994)

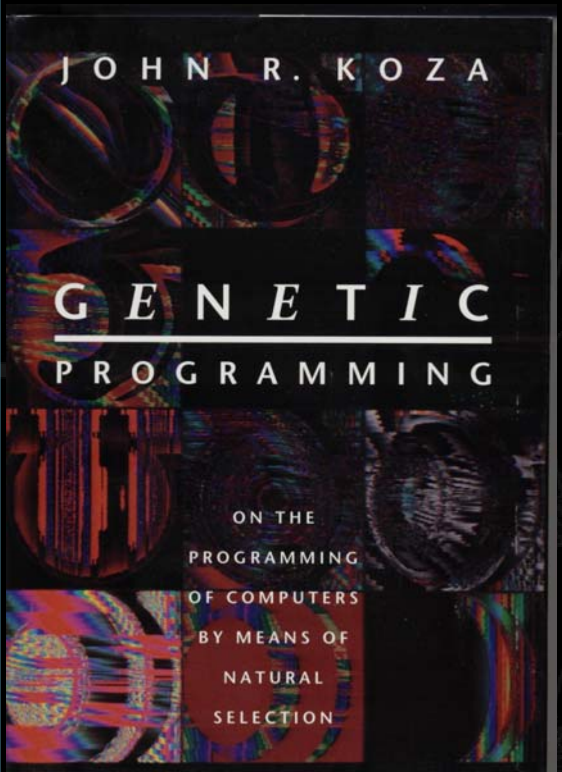
Simple solution would be to directly encode a program into a fixed-length bit-string

Problems:

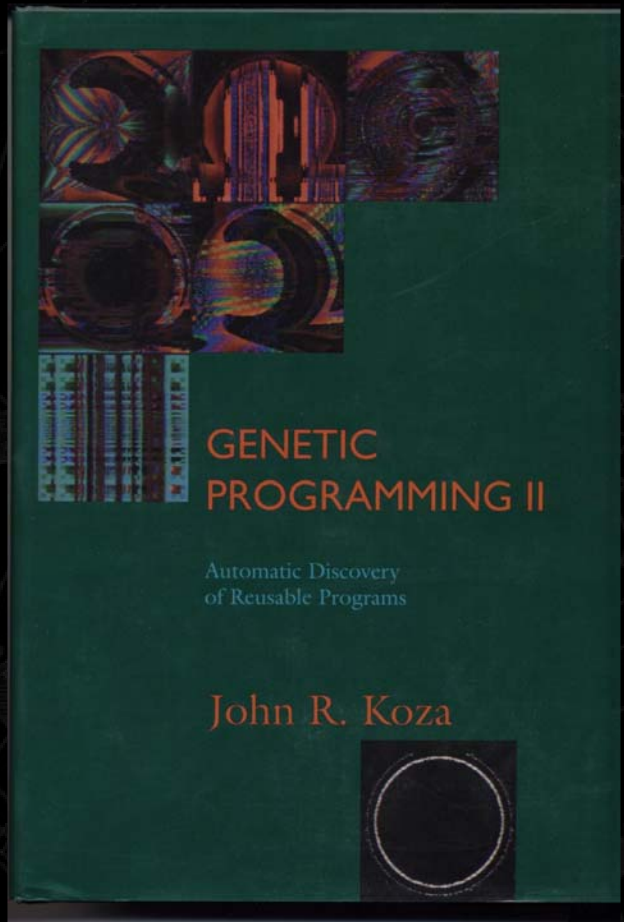
a) Bit-strings do not represent valid computer programs (mutation and xover would lead to non-functioning programs → fitness =0 → no feedback to evolution → no progress)

b) Genomes have typically a fixed length → would limit maximum potential complexity of the program

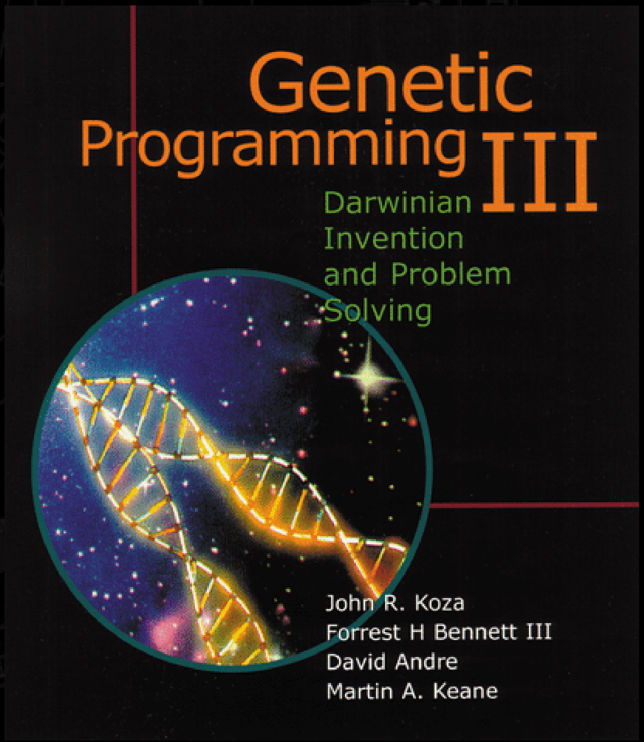
Genetic Programming



1992

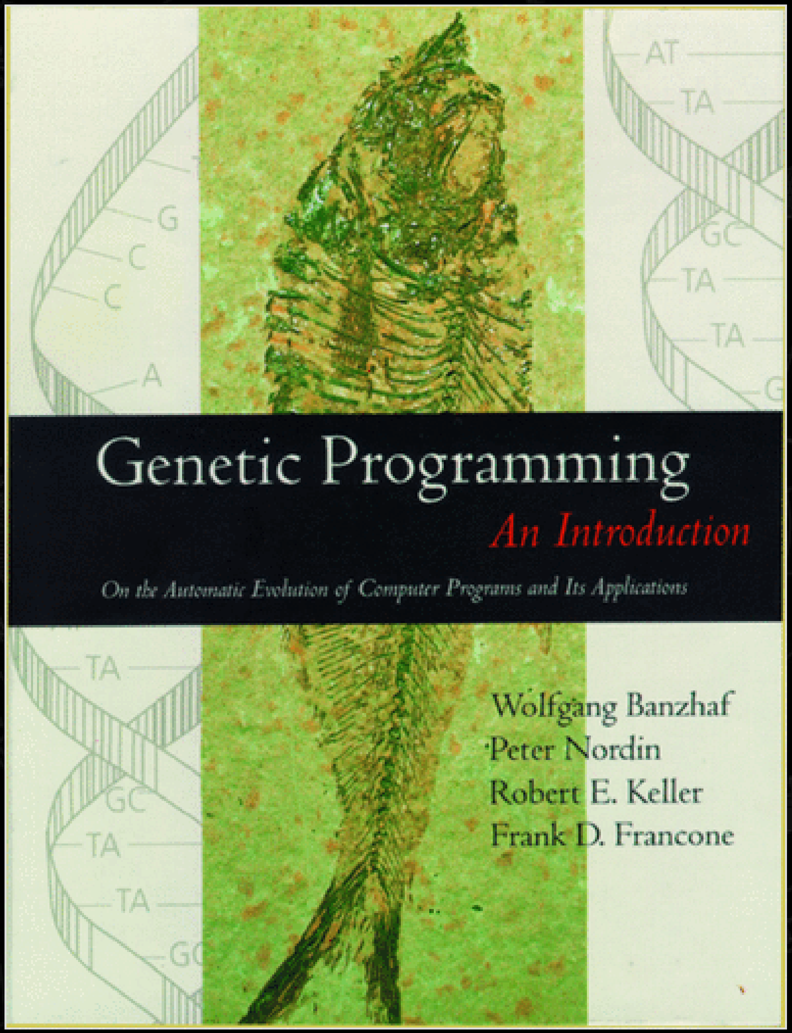


1994



1999

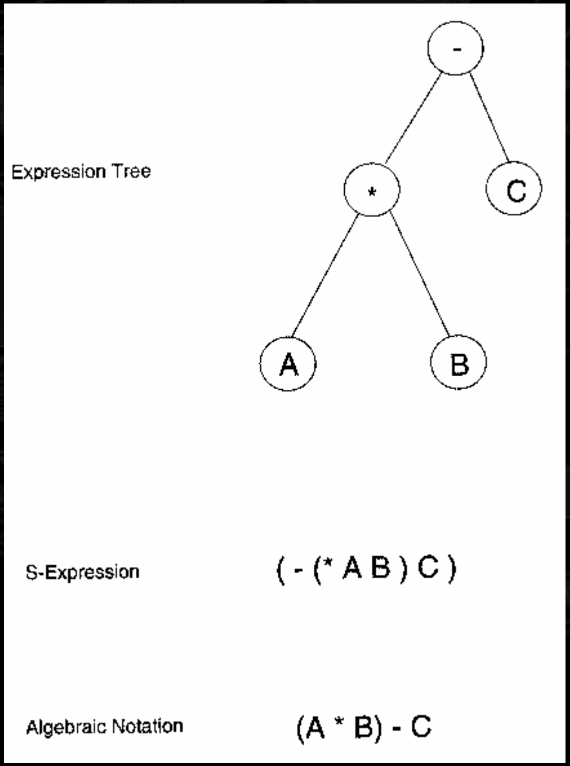
Genetic Programming



1998

Genetic Programming

Solution: represent programs using tree-shaped, linear, or graph-like chromosomes in which *nodes* represent processor operations, instructions (e.g. add, multiply) and *leaves* represent the parameters (e.g. program data)



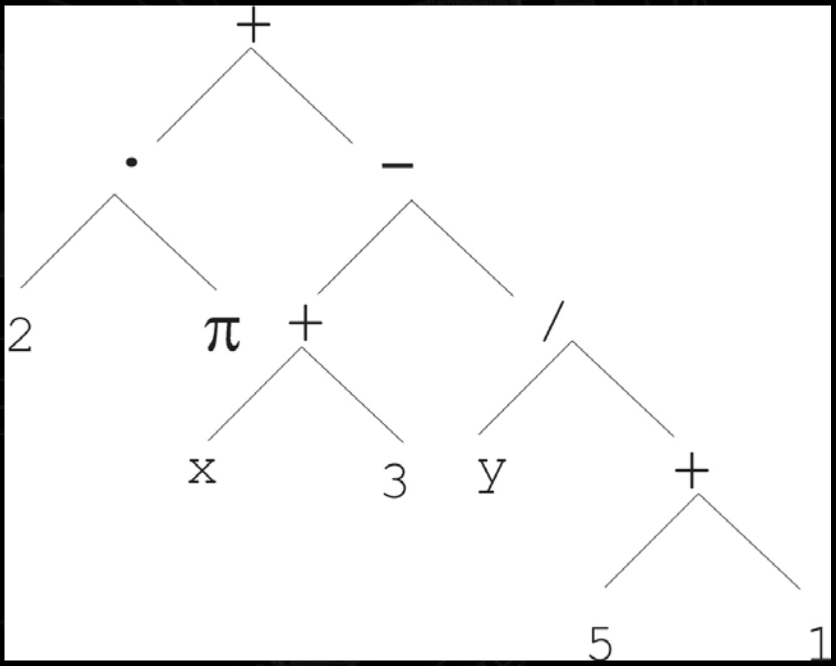
Genetic Programming

$$2\cdot\pi+\left((x+3)-\frac{y}{5+1}\right)$$

$$(+,(*,2,\pi),(-,(+,x,3),(/,y,(+,5,1))))$$

Linear representations (prefix/postfix)

$$((2,\pi,*),((x,3,+),((5,1,+),y,/),-),+)$$



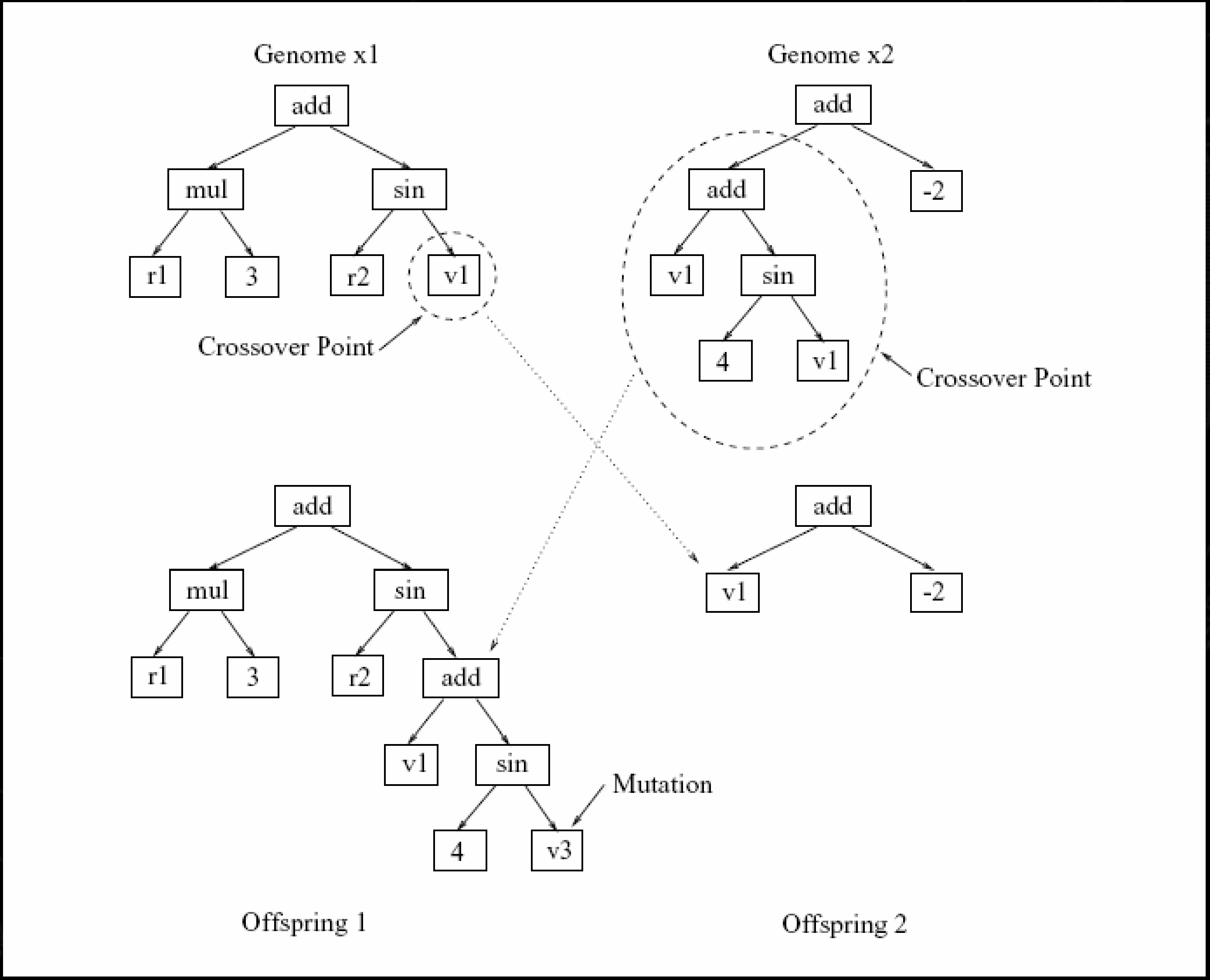
Tree structure

Genetic Programming

Steps:

1. Choose the initial set of functions and terminals, e.g. F={+,-,*,%,IFLTE} and T={X,Y,Z,R}
2. Create initial population of random programs
3. Start big loop
    1. ...
    2. Apply genetic operators:
        1. Crossover exchanges sub-trees between parents or within the same individual by selecting two random nodes and swapping all node dependencies
        2. Mutation works by deleting or adding nodes or sub-trees (mutation operator is rarely used, or mutation rate is small)

Genetic Programming



GP: Mutation

Mutation has two parameters:

- Probability p_m to choose mutation vs. recombination
- Probability to chose an internal point as the root of the subtree to be replaced

Remarkably p_m is advised to be 0 (Koza'92) or very small, like 0.05 (Banzhaf et al. '98)

GP: Crossover

Most common recombination: exchange two randomly chosen subtrees among the parents

Recombination has two parameters:

- Probability p_c to choose recombination vs. mutation
- Probability to chose an internal point within each parent as crossover point

The size of offspring can exceed that of the parents

GP: Selection

Parent selection typically fitness-proportionate

Over-selection in very large populations

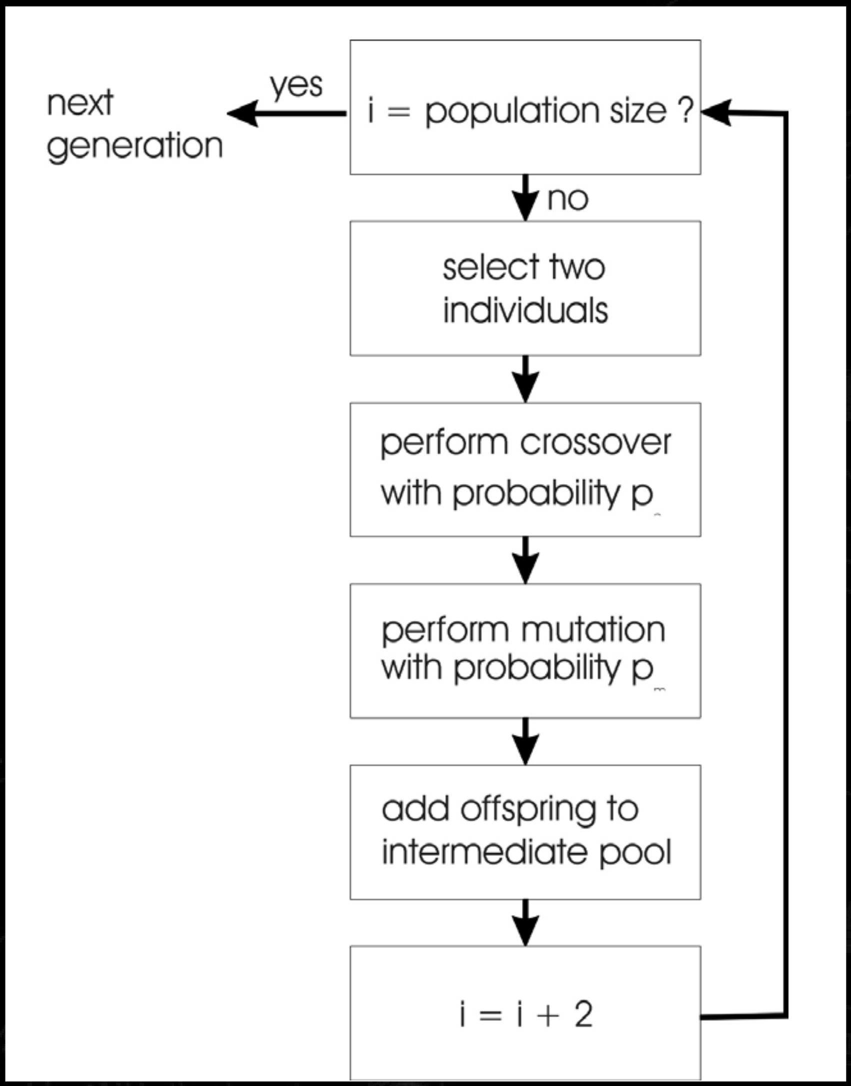
Rank population by fitness and divide it into two groups:

group 1: best x% of population

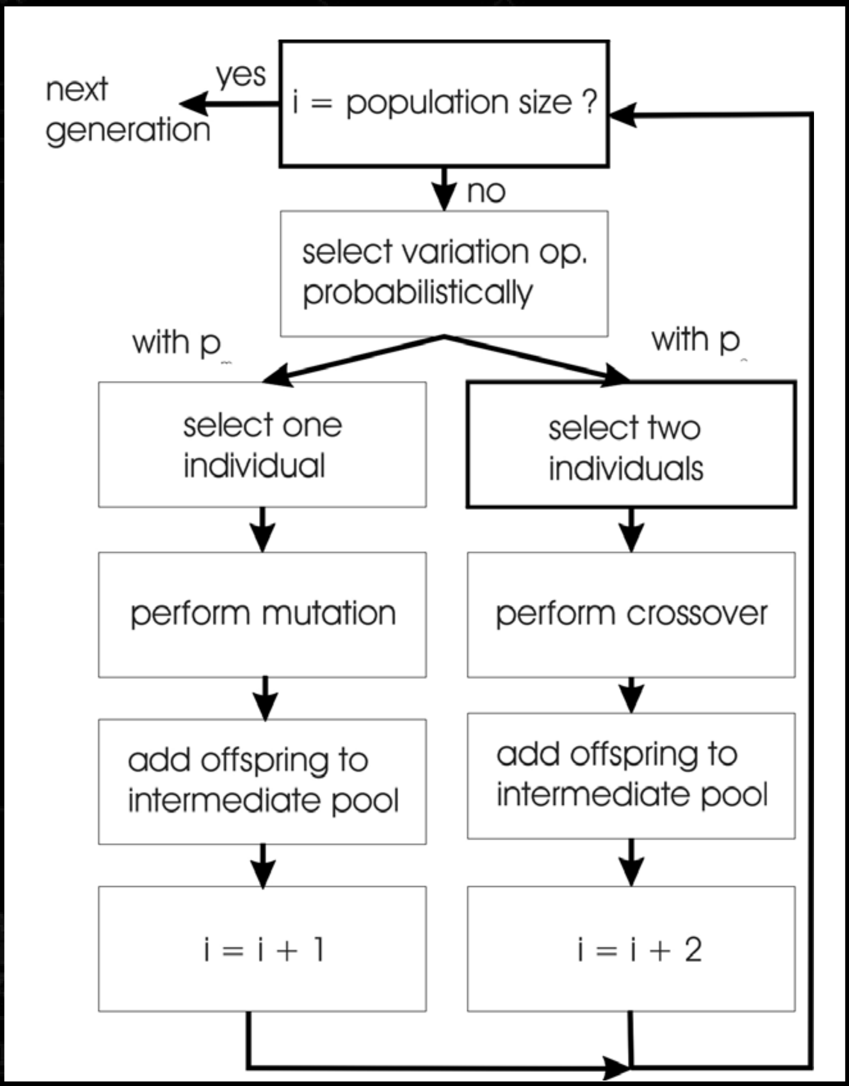
group 2: other (100-x)%

- 80% of selection operations chooses from group 1, 20% from group 2
- for pop. size = 1000, 2000, 4000, 8000 x = 32%, 16%, 8%, 4%
- motivation: to increase efficiency, %'s come from rule of thumb

Genetic Programming vs. GA, ES, EP



GA flowchart



GP flowchart

GP: Example

http://alphard.ethz.ch/gerber/approx/default.html

Observations??

GP: Bloat

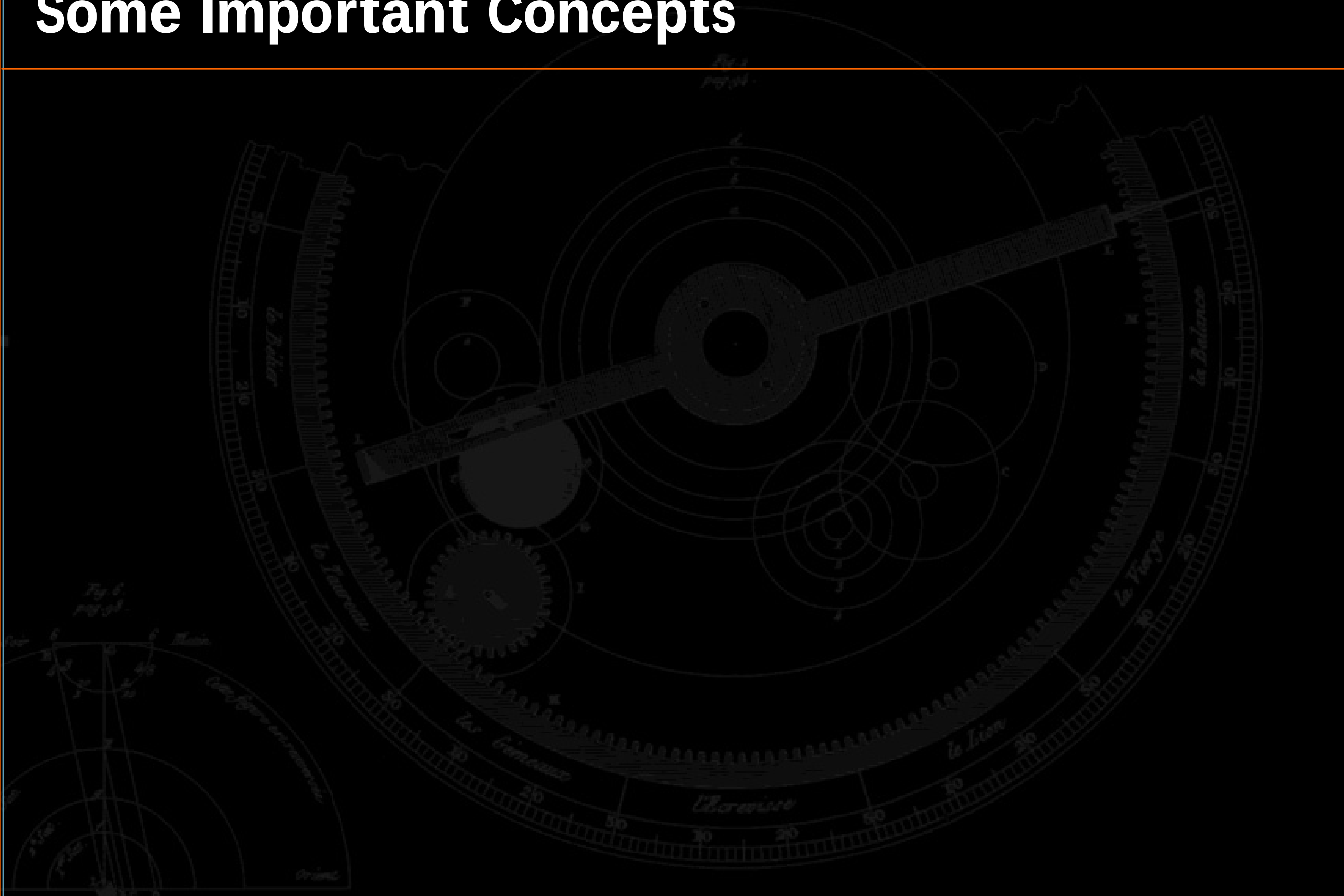
Bloat = "survival of the fattest", i.e., the tree sizes in the population are increasing over time

Ongoing research and debate about the reasons

Needs countermeasures, e.g.

- Prohibiting variation operators that would deliver "too big" children
- Parsimony pressure: penalty for being oversized; shorter genomes have an advantage ("survival of the shortest")

Some Important Concepts



(Competitive) Co-Evolution

From Wikipedia: "In biology, mutual evolutionary influence between two or more species."

Successive adaptations in one group put pressure on another group to catch up. Each group may become bigger, faster, more lethal, more intelligent, etc.

Evolutionary arms race (Dawkins and Krebs, 1979)

Artificial Co-Evolution

Competitive co-evolution has several features that may potentially enhance the adaptation power of artificial evolution:

1) Competing populations may reciprocally drive one another to increasing levels of behavioral complexity by producing an evolutionary "arms race" (populations and evolving challenges will become progressively more complex)
2) Larger variety of tasks faced by every single individual (opponents faced by organism are likely to change over generations); abilities for which individuals are selected are more general
3) Computational appeal: ever changing fitness landscape caused by changes in co-evolving species is potentially useful in preventing stagnation in local minima

Co-Evolution and Landscapes

Co-Evolution and Lanscapes

- The fitness of a genotype will vary over time according to the genotypes present in the co-evolving population
- The fitness landscape is no longer static but changes over time
- E.g. a highly fit cheetah of 6 mio. years ago might not be able to catch any modern gazelles

Building a Co-Evolutionary GA

- Two GAs running in tandem
- Fitness of an individual in population A is determined by how well it does against one or more members of population B, and vice versa
- In other respects the GAs are standard

Example

From D. Hillis (1990) "Co-evolving Parasites Improve Simulated Evolution as an Optimization Procedure"

Objective: Use a GA to solve an optimization problem in computer science – find the smallest size sorting network for n=16 elements.

Co-evolution as an engineering tool to develop sorting networks

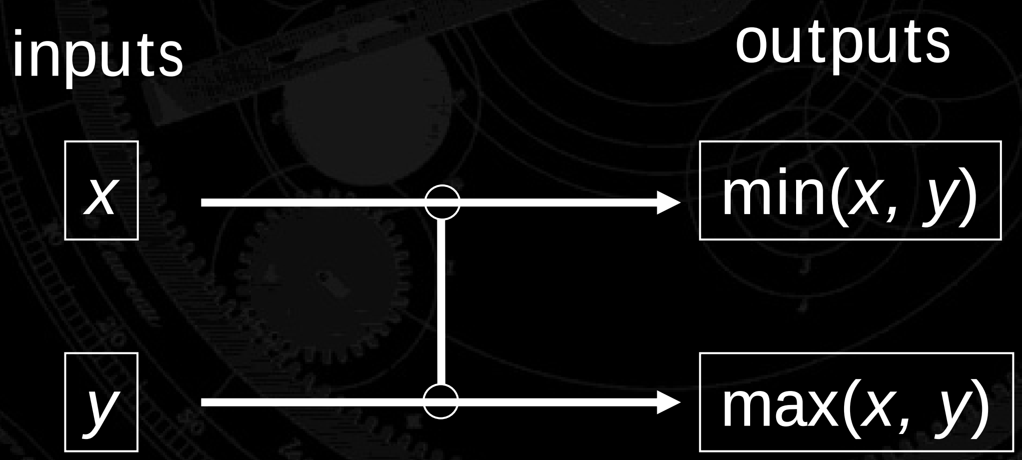
What is a Sorting Network?

Sorting networks describe algorithms for sorting a fixed-length list of numbers

What is a Sorting Network?

Suppose we want to sort two numbers according their order

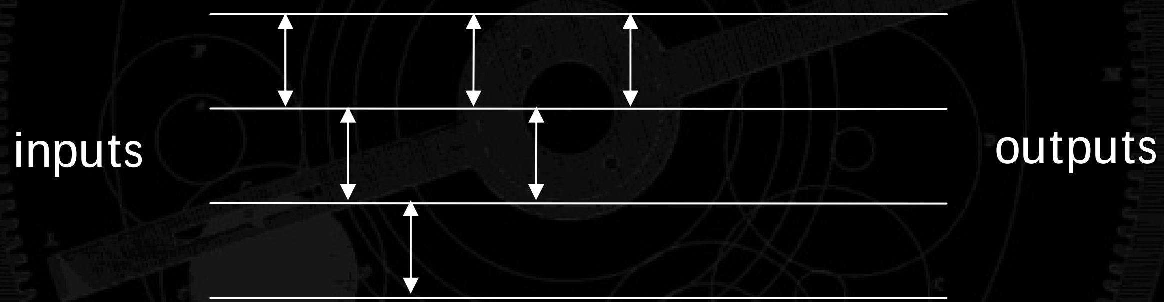
Here is one way to do it:



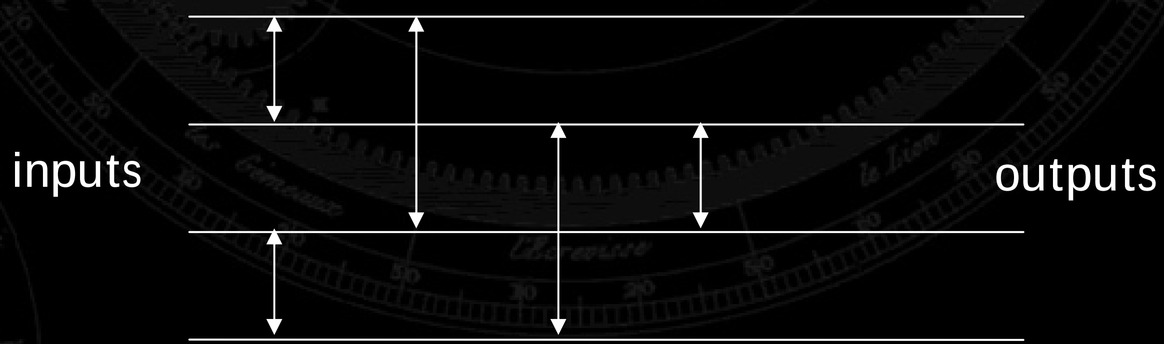
What is a Sorting Network?

Let’s sort 4 numbers

Network 1: (6 comparison-exchange operations)



Network 2: (5 comparison-exchange operations)



Co-Evolving Sorting Networks

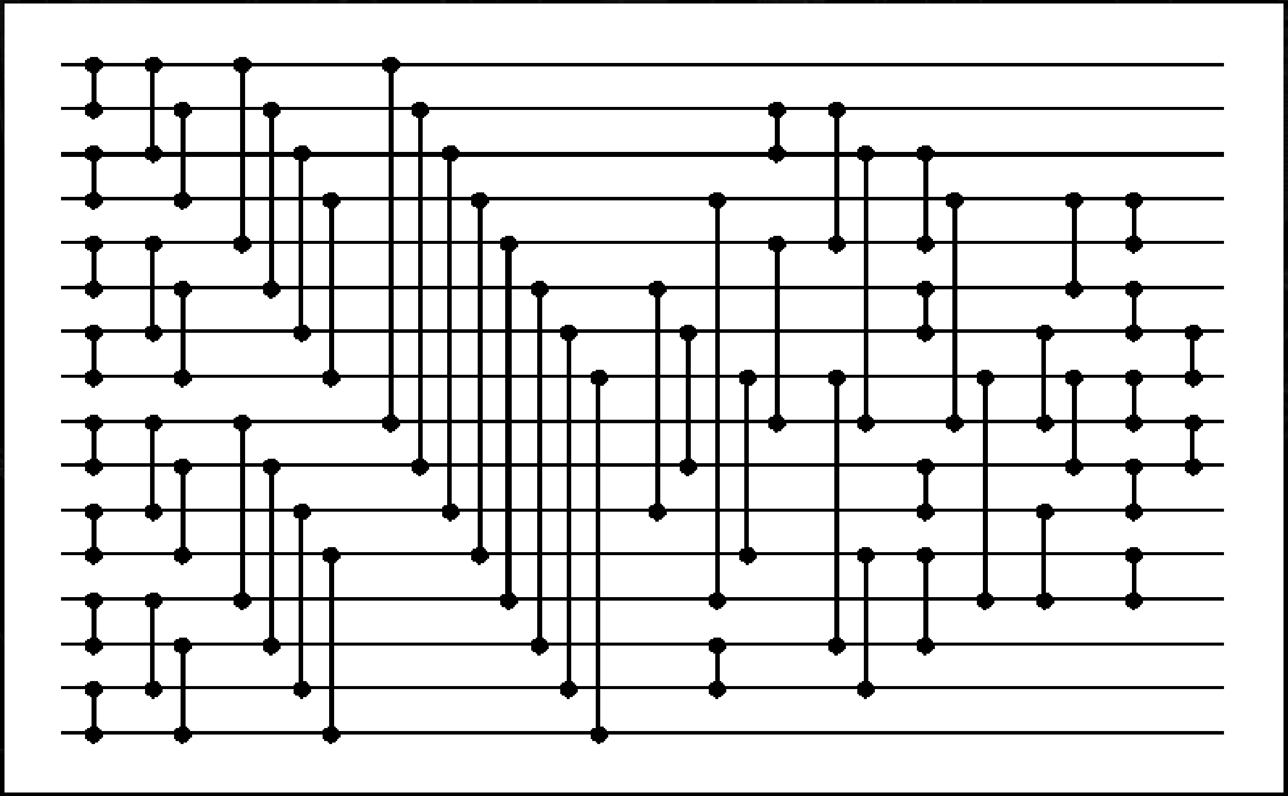
Much effort has gone into designing sorting networks that are not only correct but minimal (e.g. 15 vs. 12 comparisons)

Co-Evolving Sorting Networks

- In his first experiment, Hillis used a standard GA to evolve minimal correct networks for sorting lists of 16 numbers
  - 1-point crossover
  - Low mutation rate (1/1000)
  - Npop = 65536
  - Ngen = 5000
- The fitness function was the percentage of cases sorting correctly, given a series of random lists to sort. There was hence only an implicit pressure towards fewer comparisons

Co-Evolving Sorting Networks

Smallest network previously discovered (by hand) used 60 comparisons. <u>The best network discovered by GA used 65 comparisons</u>

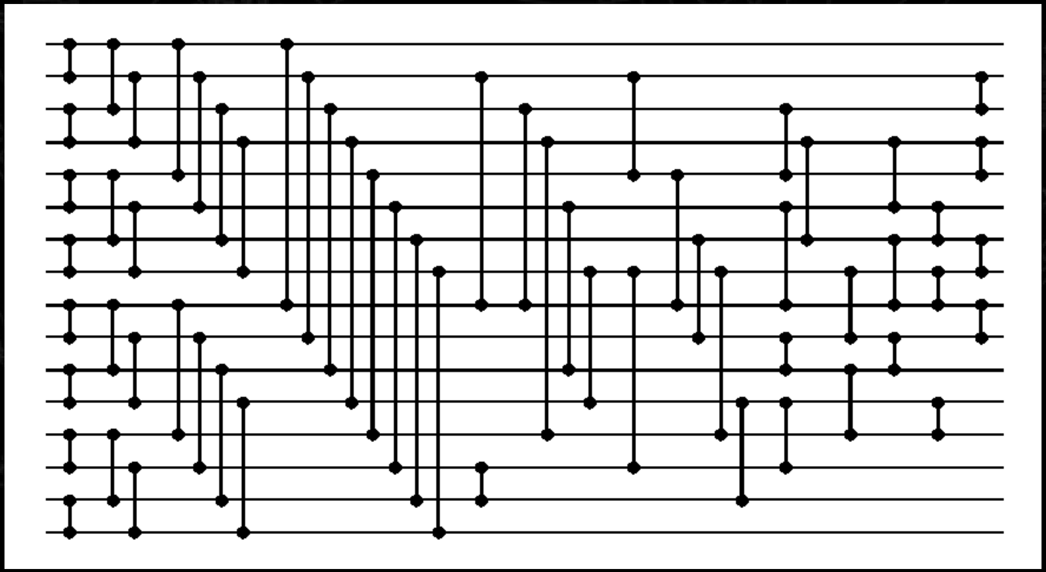


Co-Evolving Sorting Networks

- Hillis used co-evolution hoping to start and arms race between sorting networks and their parasite opponents
- A “parasite population” was implemented as a collection of 16 lists of numbers to be sorted
- The fitness function for the “parasite population” was the average percentage of cases that the networks failed to sort (for the parasite population: the higher the percentage the better)

Co-Evolving Sorting Networks

- Co-evolution thus pushed the parasites towards hard-to-sort lists, and pushed the networks towards robust sorting algorithms
- The smallest correct sorting network to evolve used only 61 comparisons – just short of equaling the best human-designed network



Lamarckian Evolution

Organisms learn useful *adaptations* during their lifetime

Adaptations embody <u>knowledge about the environment</u> gained through exploration and experimentation → it seems wasteful to lose knowledge between generations

Lamarckian evolution is the obvious solution: Transfer acquired knowledge back into the genome, that is, the phenotype influences the genotype!

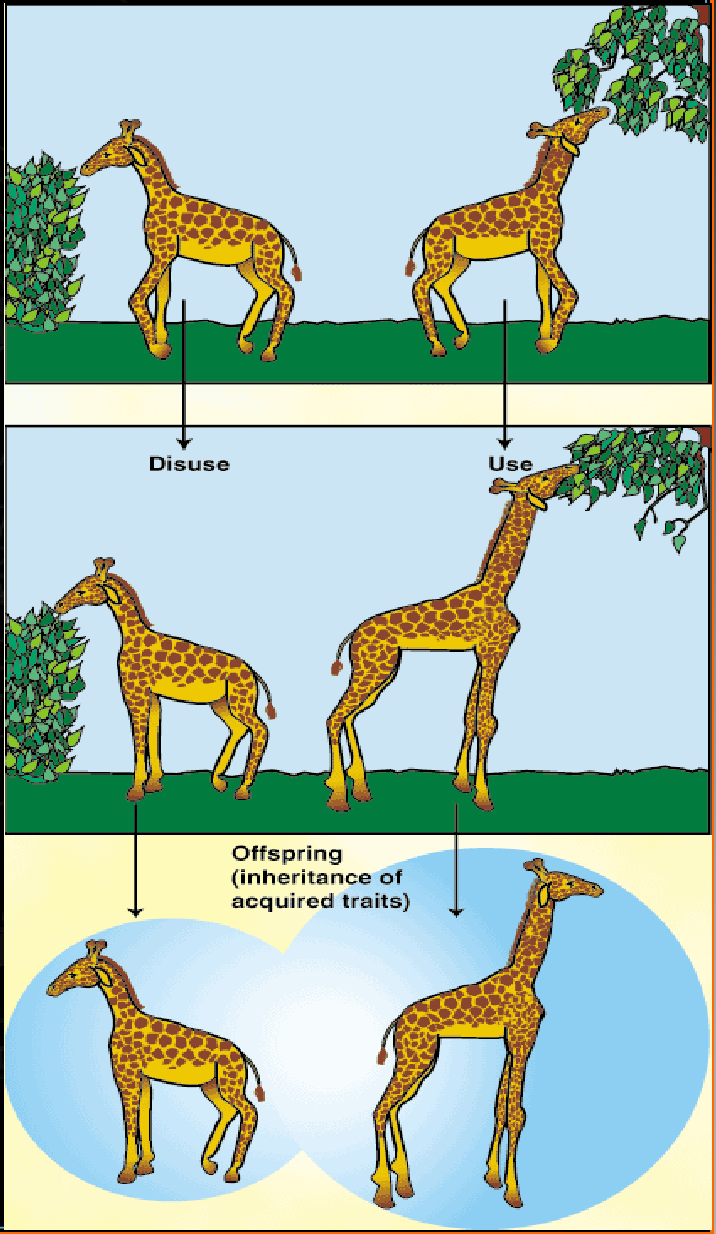
Lamarckism is a doctrine *admitting* the *possibility* of the (genotypic) inheritance of acquired (phenotypic) characters by individual organisms in evolutionary processes

So far, Lamarckian evolution has never been substantiated in any biological system

Note: There have been successful attempts in the digital verse!

Lamarckian Evolution: Examples

- Giraffes stretching their necks to reach leaves high in trees, strengthen and gradually lengthen their necks. These giraffes have offspring with slightly longer necks
- A blacksmith, through his work, strengthens the muscles in his arms. His sons will have similar muscular development when they mature



Evolution and Learning

In the absence of Lamarckian evolution, one might conclude that lifetime learning cannot impact evolution

One would be wrong

Learning *can* be very effective in guiding evolutionary search

- Even when that learning is *not* communicated to the genotype
- Even when that learning is unrelated to specific survival tasks or the fitness selection criteria

Examples!

Baldwin Effect

Useful adaptations allow an organism to survive and reproduce, increasing its evolutionary fitness

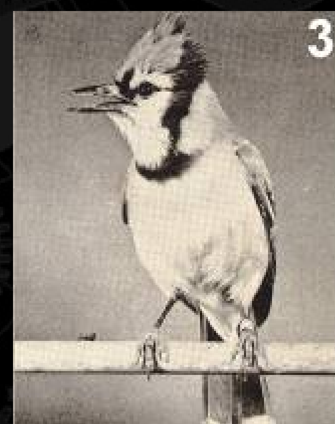
Evolution then selects for organisms that are ever more capable of learning these adaptations

Baldwin Effect

A blue jay
eats a
monarch

But it does‘t
taste

Because of
nausea the
feathers struggle



Out with the
poison

And the
teaching isn‘t
forgotten

Subjective selection in nature

Punctuated Equilibrium (“Punkeek”)

The Modern Synthesis = Neo-Darwinism, theory of evolution in full bloom

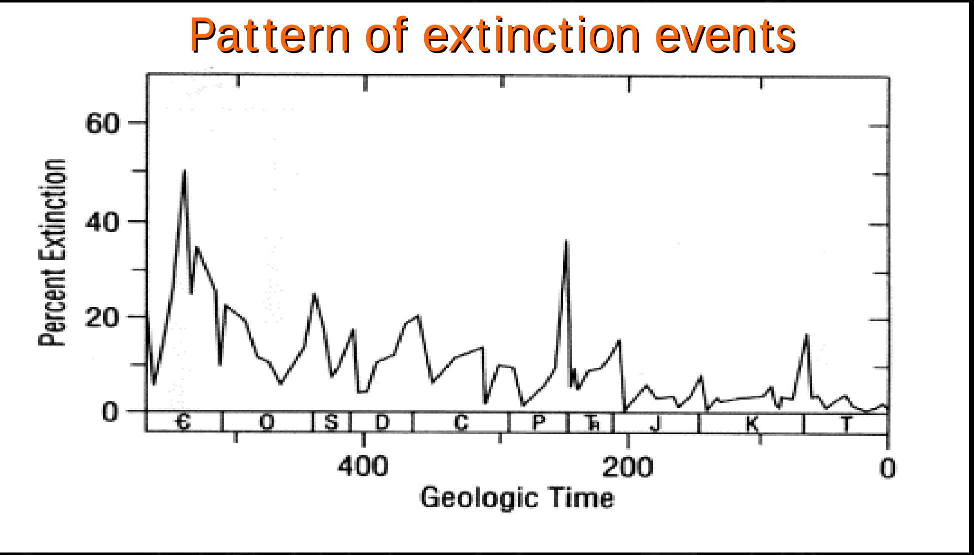
The four horsemen of the Modern Synthesis (~1940s):

1. Theodozius Dobzhansky – population genetics
2. Ernst Mayr – ecology, biogeography (zoology)
3. G. Ledyard Stebbins – botany
4. George Gaylord Simpson – paleontology (mammals)

- Viewed evolution as gradual process, taking many years
- “Natura non facit saltum” – nature does not make leaps

Punctuated Equilibrium ("Punkeek")

It is estimated that among four billion species which have existed on the Earth since life first appeared less than 50 million (about 1 percent) are still alive

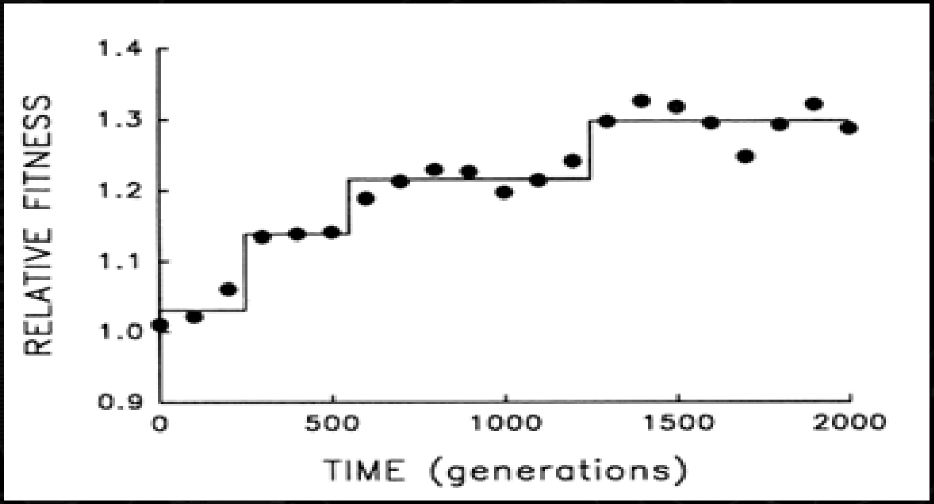


From the pattern of extinction events: there are periods of relatively little activity (stasis) interrupted by narrow intervals, or bursts, with large activity in the history of biological evolution

"Punctuated equilibrium" was first proposed by Gould and Eldredge

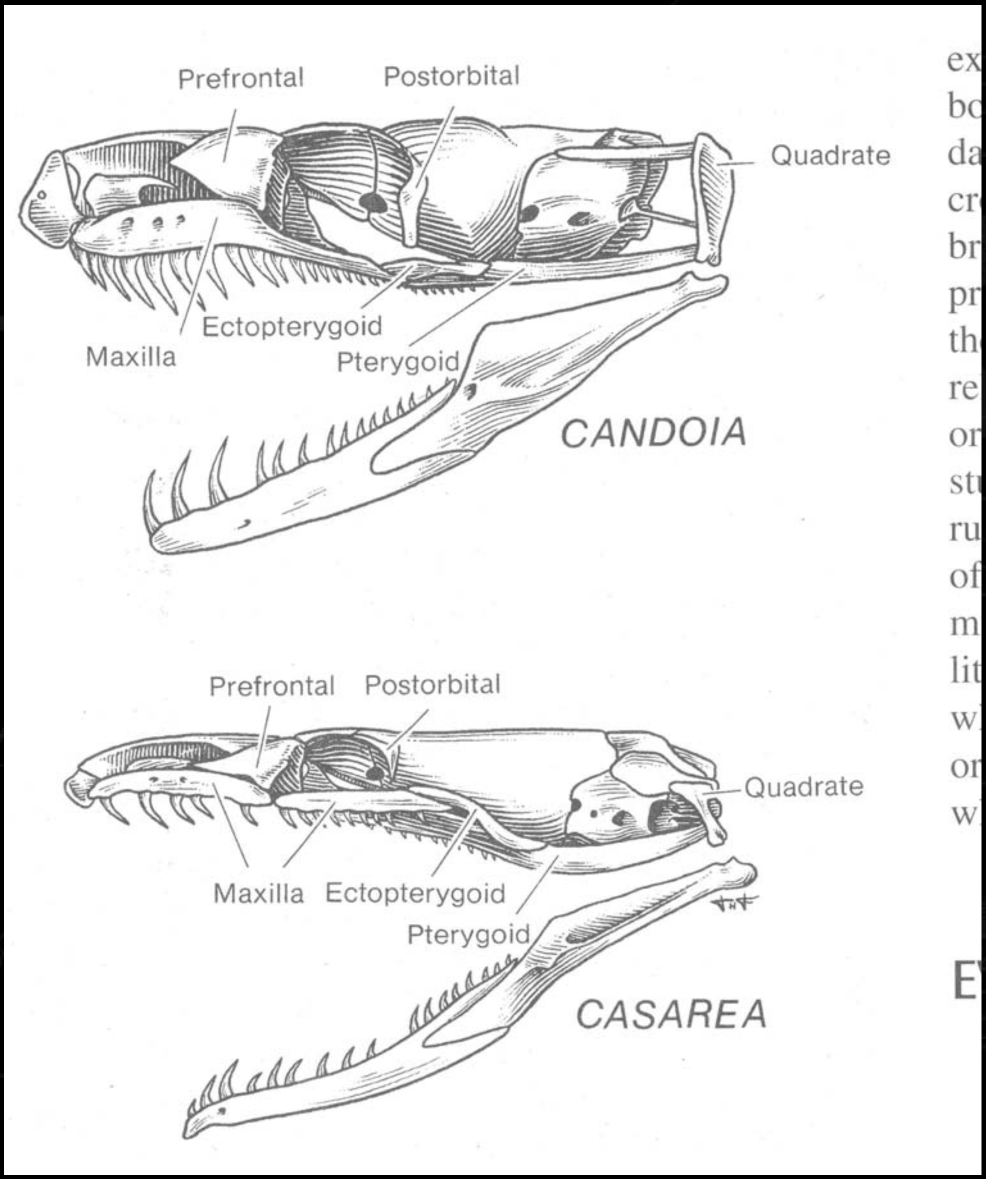
Punctuated Equilibrium ("Punkeek")

Lenski's experiment on E-coli



- Evolution in a test tube (Lenski's experiment on E-coli)
- Fossil evidence (Jackson and Cheetham) on punctuated Equilibrium
- Gradualism vs. Punctuated Equilibrium

Example of Punkeek?



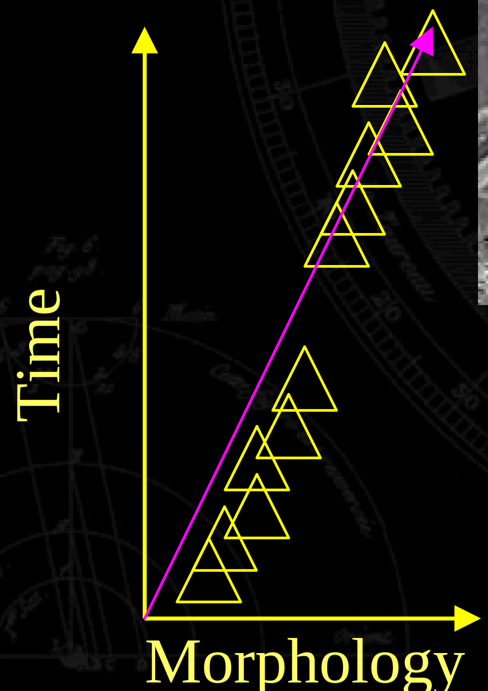
There were scientific dissenters at the time, e.g. Goldschmidt – developmental genetics, "hopeful monsters". Although rejected when he proposed it in 1940's turned out later he was right.

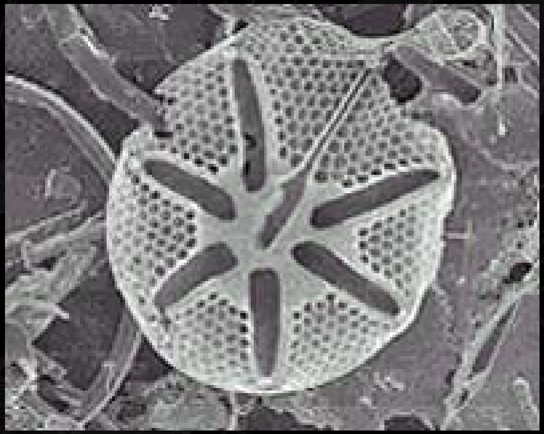
Example of a Goldschmidt break in the evolution of a modern snake.

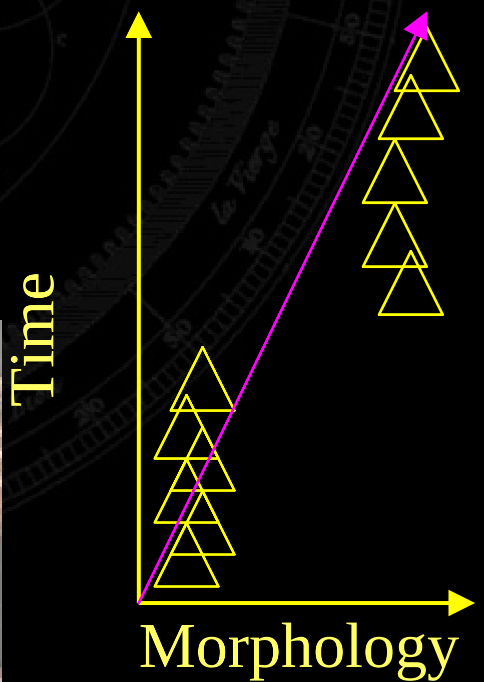
Casarea, found only on island of Mauritius, has a joint between the two bones of its maxilla, allowing it to gape its jaws wider. <u>No functional intermediates are possible</u> between the original morphology of *Candoia* and the final morphology of *Casarea*

Punkeek

Phyletic gradualism vs punctuated equilibrium.
Where we do not see gradual morphological change, is this because of gaps in the fossil record, or because the change was not gradual?







Punkeek

Self-organized criticality anyone?