



Computación Evolutiva

Fernando Berzal, berzal@acm.org

Computación Evolutiva



Orígenes

- Evolución
- Genética

Historia de la computación evolutiva

Algoritmos evolutivos

- Estrategias de evolución
- Algoritmos genéticos
- Programación genética
- Programación evolutiva

Aplicaciones



Computación Evolutiva



La naturaleza siempre ha servido como fuente de inspiración para ingenieros y científicos.

El mejor mecanismo de resolución de problemas conocido es...

- El cerebro (humano), que creó “la rueda, Nueva York, las guerras y demás” [Douglas Adams: Hitch-Hiker’s Guide to the Galaxy] ➔ **Neurocomputación**
- El mecanismo de evolución que creó el cerebro humano. ➔ **Computación Evolutiva**



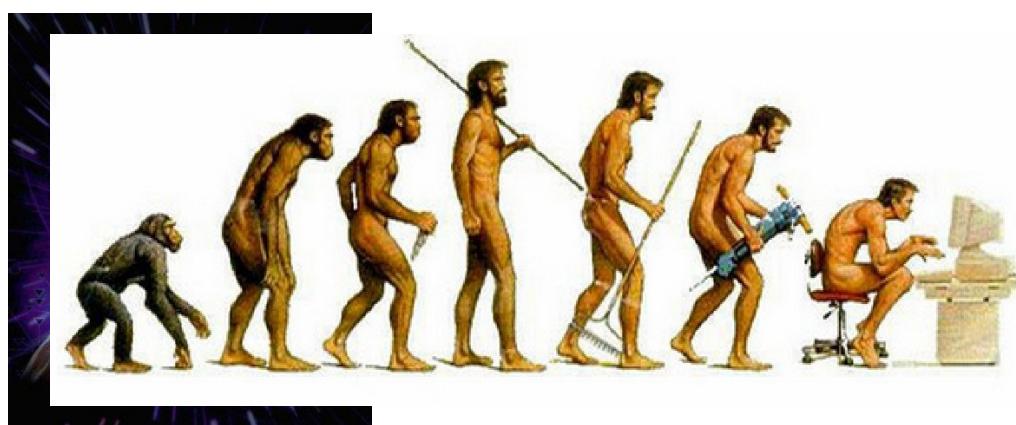
Orígenes



Creacionismo



Evolución



Orígenes



Georges-Louis Leclerc, Conde de Buffon (1707-1788)

[naturalista, matemático, cosmólogo y autor enciclopédico francés]

Histoire naturelle, générale et particulière

(36 volúmenes publicados de 1749 a 1788)



Padre del evolucionismo:

- El primero en discutir problemas evolutivos con un espíritu científico (100 años antes que Darwin), p.ej. consideró las similitudes entre el hombre y los simios y especuló sobre la posible existencia de un ancestro común (aunque él mismo negó esa posibilidad).

http://en.wikipedia.org/wiki/Georges-Louis_Leclerc,_Comte_de_Buffon



Orígenes



James Burnett, Lord Monboddo (1714-1799)

[juez, lingüista, filósofo y deísta escocés]

Of the Origin and Progress of Language

(6 volúmenes publicados de 1773 a 1792)



Precursor de la teoría de la evolución:

- Primero en formular la hipótesis del origen común de toda la humanidad (i.e. todos los hombres proceden de la misma región de la tierra).
- Uno de los primeros en postular que todos los simios y antropoides tienen un origen común.

http://en.wikipedia.org/wiki/James_Burnett,_Lord_Monboddo



Orígenes



Jean-Baptiste Pierre Antoine de Monet, Chevalier de Lamarck
a.k.a. Jean Baptiste Lamarck (1744-1829)
[naturalista francés]

Philosophie zoologique (1809)

- Primera teoría evolutiva coherente:
Los organismos pueden adquirir nuevas características por la influencia de su entorno.
- Herencia de caracteres adquiridos, i.e. capacidad de los organismos de trasladar a sus descendientes los caracteres adquiridos en vida («Lamarckismo»).



http://en.wikipedia.org/wiki/Jean-Baptiste_Lamarck



Orígenes



Lamarckismo

- **Primera ley:** En todo animal que no ha traspasado el término de sus desarrollos, el uso frecuente y sostenido de un órgano cualquiera lo fortifica poco a poco, dándole una potencia proporcionada a la duración de este uso, mientras que el desuso constante de tal órgano le debilita y hasta lo hace desaparecer.
- **Segunda ley:** Todo lo que la Naturaleza hizo adquirir o perder a los individuos por la influencia de las circunstancias en que su raza se ha encontrado colocada durante largo tiempo, y consecuentemente por la influencia del empleo predominante de tal órgano, o por la de su desuso, la Naturaleza lo conserva por la generación en los nuevos individuos, con tal de que los cambios adquiridos sean comunes a los dos sexos, o a los que han producido estos nuevos individuos.

Jean-Baptiste Lamarck: **Filosofía zoológica**

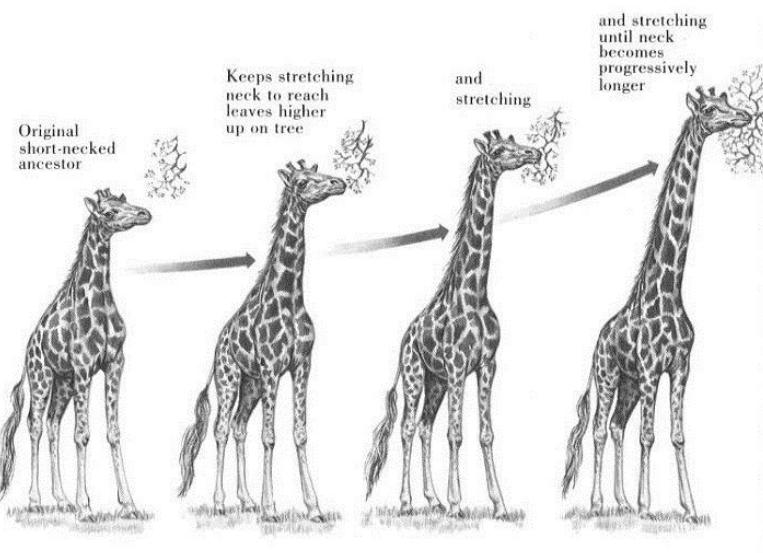


Orígenes



Lamarckismo

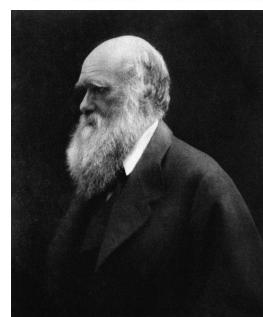
“When an organism is ‘in need’ of an organ, sooner or afterward it will occur.”



Orígenes



Charles Robert Darwin, FRS (1809-1882)



**On the Origin of Species
by Means of Natural Selection, or the Preservation
of Favoured Races in the Struggle for Life (1859)**

- Los individuos son ligeramente distintos entre sí y estas pequeñas variaciones hacen que cada uno tenga distintas capacidades para adaptarse a su medio ambiente, así como para reproducirse y para transmitir sus rasgos a sus descendientes.

http://en.wikipedia.org/wiki/Charles_Darwin



Orígenes



Darwinismo

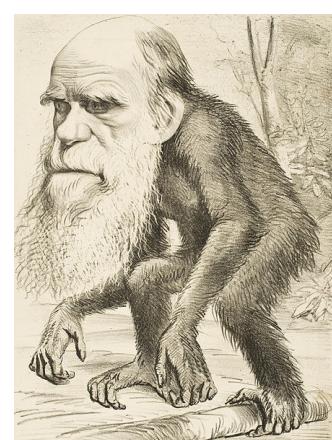
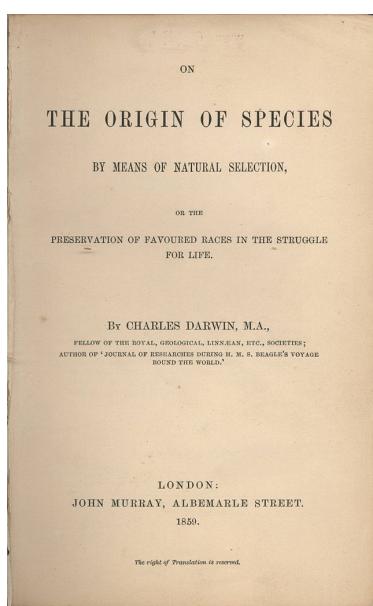
- Con el paso del tiempo (generaciones), los rasgos de los individuos que mejor se adaptaron a las condiciones del medio ambiente se vuelven más comunes, haciendo que la población, en su conjunto, evolucione ("descendencia con modificación").
- Del mismo modo, la naturaleza selecciona las especies mejor adaptadas para sobrevivir y reproducirse ("selección natural").



Orígenes



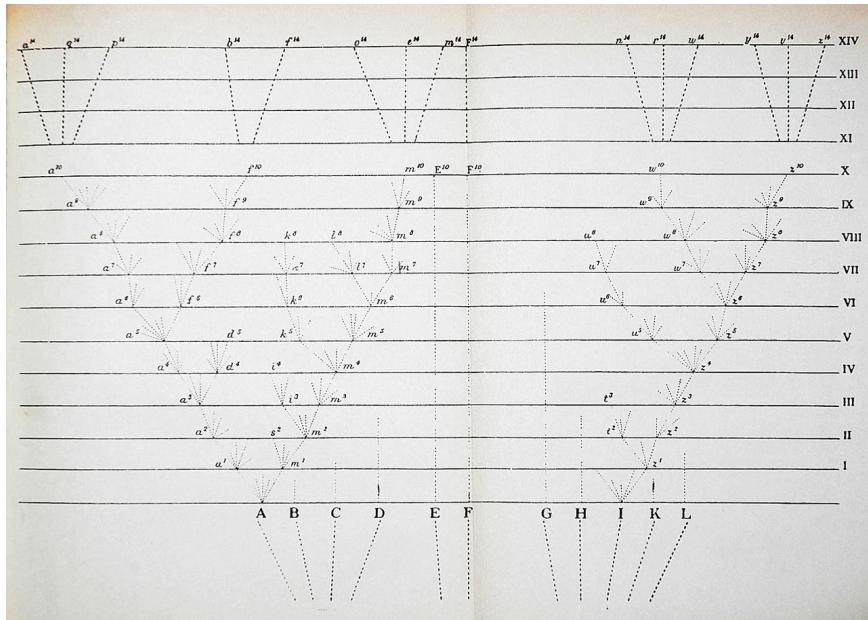
Darwinismo



Orígenes



Darwinismo: The divergence of species (1859)



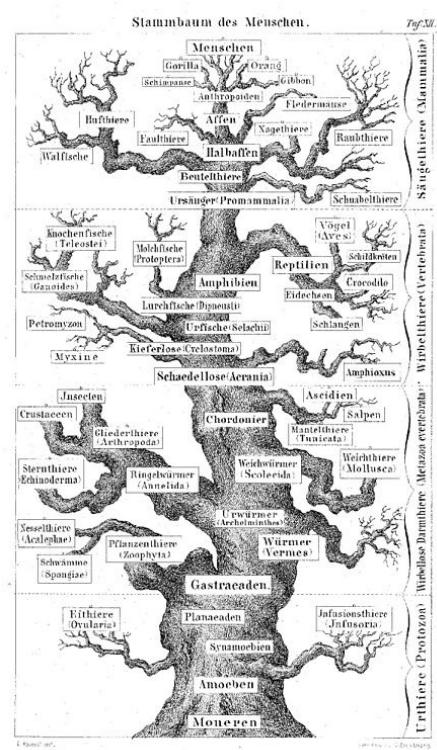
Orígenes



Pedigree of Man

(Ersnt Haeckel, 1874)

Modelo lineal de la evolución

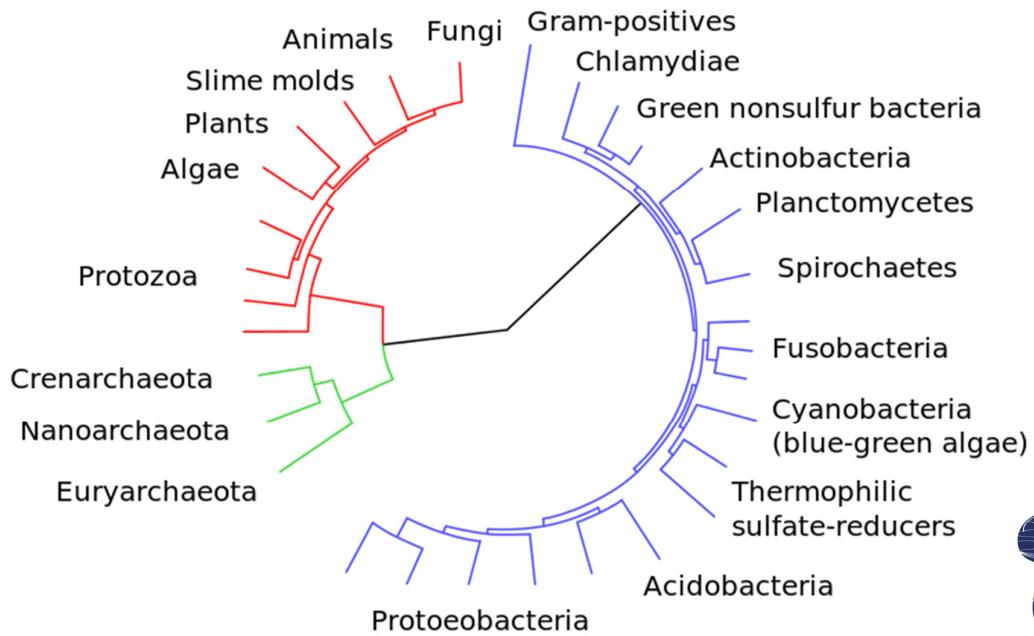


Orígenes



Árbol filogenético

Three-domain system (Carl Woese, 1977)

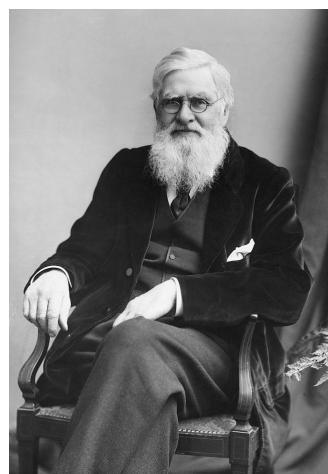


Orígenes



Alfred Russel Wallace (1823-1913)

**On the Tendency of Varieties
to Depart Indefinitely
from the Original Type** (1858)



- Concibió la teoría de la evolución por medio de la selección natural de forma independiente a Darwin.

http://en.wikipedia.org/wiki/Alfred_Russel_Wallace

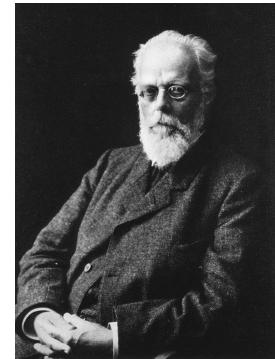


Orígenes



Friedrich Leopold August Weismann (1834-1914)

Germ-Plasm, a theory of Heredity (1893)



Teoría del plasma germinal:

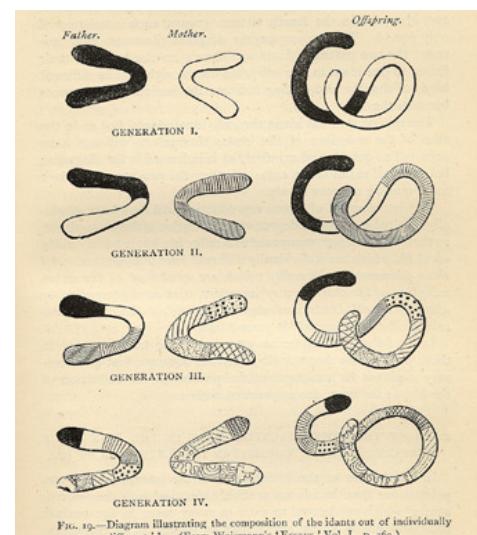
- La herencia, en un organismo multicelular, se efectúa únicamente por medio de células germinales (gametos).
- Las demás células del cuerpo (las células somáticas) NO funcionan como agentes hereditarios.
- Barrera de Weismann: Las células germinales producen células somáticas, pero no al revés, por lo que no puede transmitirse información genética de células somáticas.

http://en.wikipedia.org/wiki/August_Weismann



Weismannismo

- Cualquier capacidad adquirida por un organismo durante su vida no puede transmitirse a la siguiente generación (refutación del Lamarckismo).



Orígenes



Gregor Johann Mendel (1822-1884)

Experiments on Plant Hybridization (1866)



Leyes de Mendel:

Reglas básicas que gobiernan la transmisión por herencia de las características de los padres a los hijos.

http://es.wikipedia.org/wiki/Leyes_de_Mendel

Semilla		Flor		Vaina		Tallo	
Forma	Cotiledones	Color		Forma	Color	Lugar	Tamaño
Gris y Redondo	2 Amarillo	Blanco		Lleno	Amarillo	Vainas axilares. Las flores crecen a los lados	Largo (~3m)
Blanco y Arrugado	2 Verde	Violeta		Constrefido	Verde	Vainas terminales. Las flores crecen en la cúspide	Corto (~30cm)
	1	3		4	5		6
							7

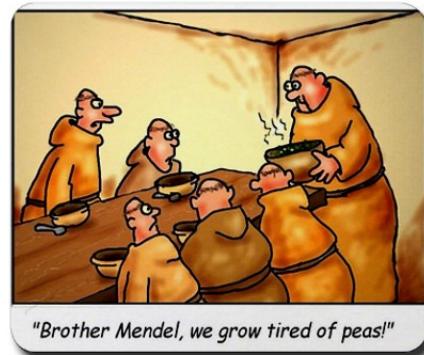
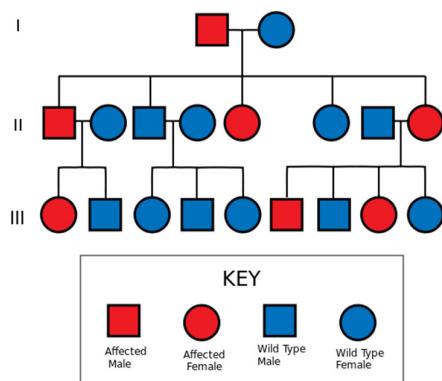
http://es.wikipedia.org/wiki/Gregor_Mendel



Orígenes



Árboles genealógicos



Árbol autosómico dominante



Orígenes



James Mark Baldwin (1861-1934)

A New Factor in Evolution (1896)



- Las condiciones genéticas transmisibles por herencia pueden hacer más fácil el aprendizaje de técnicas y trucos que sólo poseen aquellos que tengan una variante evolutiva determinada.
- Evolución baldwiniana u ontogenética (efecto Baldwin): Las condiciones epigenéticas pueden ser igual o más importantes que la selección natural.

http://en.wikipedia.org/wiki/James_Mark_Baldwin



Orígenes



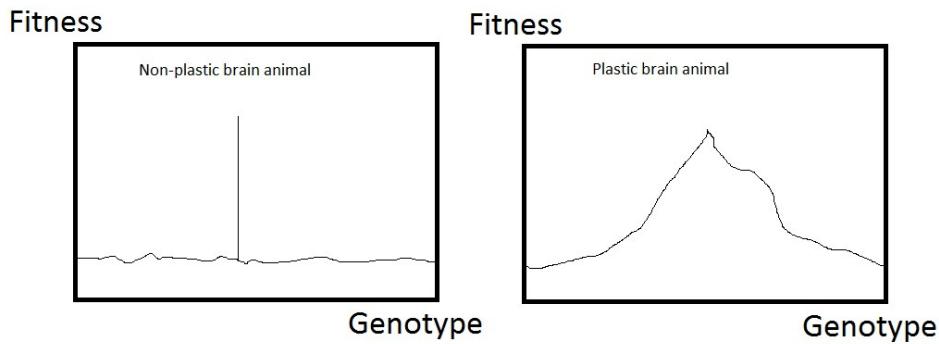
El efecto Baldwin: Evolución & aprendizaje

- La descendencia seleccionada tenderá hacia una mayor capacidad para aprender nuevas habilidades y no estar constreñida a habilidades genéticamente codificadas y relativamente fijas.
- Plasticidad fenotípica: Capacidad de un organismo para adaptarse a su ambiente durante su tiempo de vida (p.ej. capacidad de aprendizaje)





El efecto Baldwin: Evolución & aprendizaje



El efecto Baldwin: Evolución & aprendizaje

- El comportamiento sostenido de una especie o grupo puede guiar la evolución de esa especie:
Las habilidades que inicialmente requieren el aprendizaje son finalmente reemplazadas por la evolución de sistemas genéticamente determinados que no requieren aprendizaje.
- El aprendizaje individual puede explicar fenómenos evolutivos que parecen apoyar la herencia Lamarckiana (sin recurrir a ella).



Orígenes



Sir Julian Sorell Huxley, FRS (1887-1975)

Evolution: The Modern Synthesis (1942)

“Neodarwinismo”

Síntesis moderna comúnmente aceptada de la evolución:

- Selección natural (Darwin).
- Genética clásica
 - Leyes de Mendel (herencia)
 - Teoría de Sutton-Bovery (cromosomas)
- Reproducción
 - Teoría del plasma germinal (Weismann)

http://en.wikipedia.org/wiki/Modern_evolutionary_synthesis



Orígenes



La teoría sintética defiende que los cambios graduales y la selección natural sobre ellos son el mecanismo principal del cambio evolutivo, rechazando otros mecanismos defendidos por otras teorías:

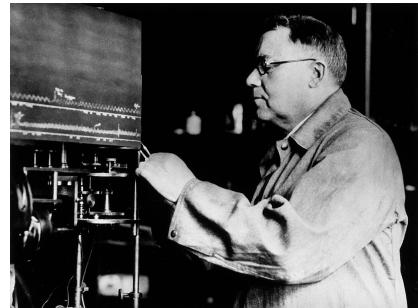
- **Saltacionismo:** Origen repentino de nuevas especies.
- **Lamarckismo:** Herencia de caracteres adquiridos.
- **Ortogénesis:** Fuerza intrínseca a la materia orgánica que conduciría a un progreso evolutivo.
- **Equilibrio puntuado:** Los cambios graduales sólo explican la microevolución, mientras que la macroevolución se produce por cambios bruscos.



Historia



La evolución natural ha sido vista como un proceso de aprendizaje desde los años 30 del siglo XX, con el trabajo del fisiólogo estadounidense Walter Bradford Cannon (1871-1945):



The Wisdom of the Body (1932).

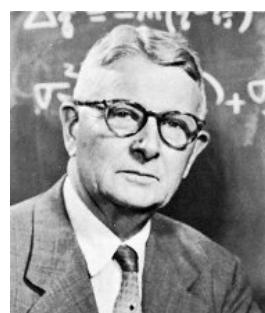
http://en.wikipedia.org/wiki/Walter_Bradford_Cannon



Historia

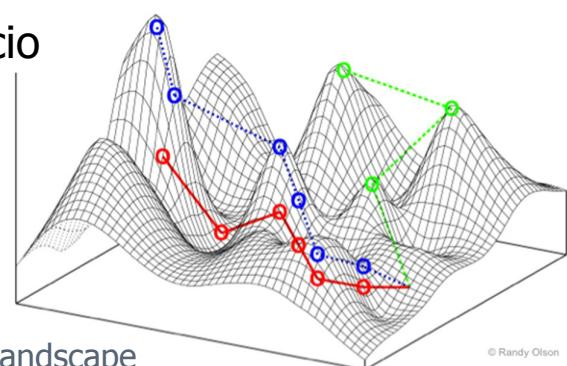


La idea de estudiar la evolución visualizando la distribución de los grados de adaptación [fitness] fue propuesta por el genetista Sewall Wright (1889-1988) en 1932.



Paisaje adaptativo [fitness landscape]

Individuos con n rasgos como puntos en un espacio $(n+1)$ -dimensional, con altura proporcional a su grado de adaptación o fitness.



http://en.wikipedia.org/wiki/Fitness_landscape

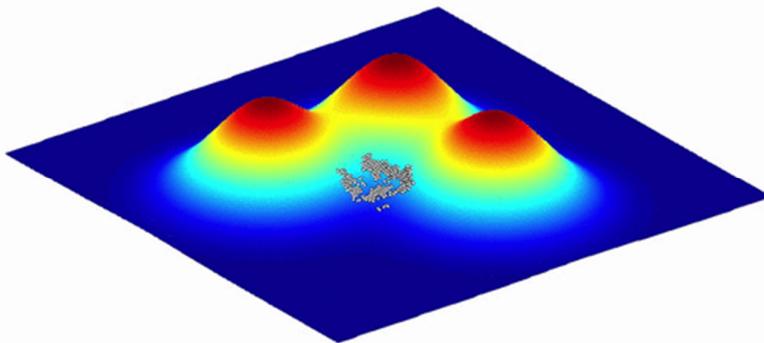


Historia



Una población es una nube de puntos que se mueve en el paisaje conforme se adapta a su entorno (evoluciona).

Static fitness landscape



Population size, $N = 2,304$
Mutation rate, $\mu = 0.05$ per trait

© Randy Olson and Bjørn Østman



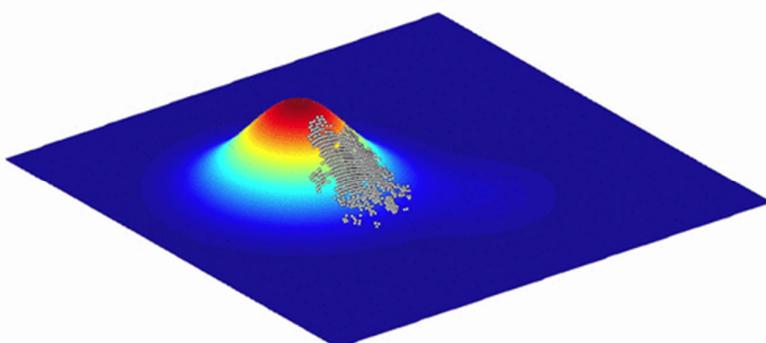
28

Historia



El entorno también puede cambiar a lo largo del tiempo...

Dynamic fitness landscape



Population size, $N = 2,304$
Mutation rate, $\mu = 0.5$ per trait

© Randy Olson and Bjørn Østman

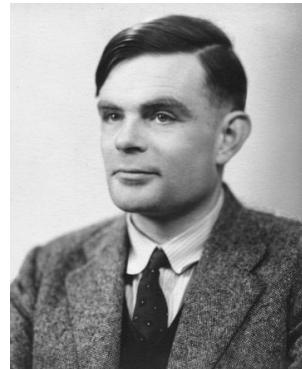


29

Historia



Alan Mathison Turing (1912-1954) reconoció también una conexión entre la evolución y el aprendizaje automático en un artículo de 1950:



Turing, Alan M.:

Computing Machinery and Intelligence,

Mind LIX (236): 433–460

DOI [10.1093/mind/LIX.236.433](https://doi.org/10.1093/mind/LIX.236.433)



Historia



En 1957, el estadístico inglés George E. P. Box, FRS (1919-2013) propuso un enfoque evolutivo para la optimización de la producción industrial.



Su técnica, denominada EVOP [Evolutionary Operation] se sigue utilizando en la industria.

Box, George E.P. (1957). "**Evolutionary Operation: A Method for Increasing Industrial Productivity**". Journal of the Royal Statistical Society. Series C (Applied Statistics) 6 (2): 81–101.
DOI 10.2307/2985505. JSTOR 2985505.



Historia



A finales de los 50, R.M. Friedberg fue de los primeros en intentar evolucionar programas de ordenador usando mutación y selección, aunque sus experimentos no fueron muy exitosos y fueron muy criticados en su época.



R.M. Friedberg (1958): "**A Learning Machine: Part I**".
IBM Journal of Research and Development 2(1):2-13.
DOI 10.1147/rd.21.0002

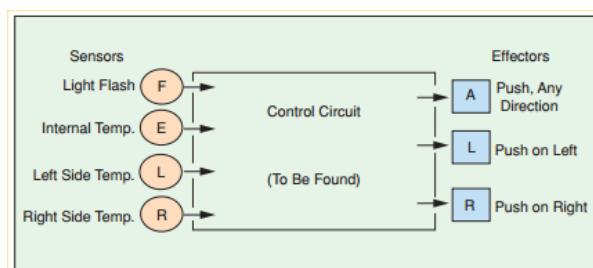
R.M. Friedberg, B. Dunham, J. H. North (1959): "**A Learning Machine: Part II**". IBM Journal of Research and Development 3(3):282-287. DOI 10.1147/rd.33.0282



Historia



George J. Friedman fue el primero en proponer una aplicación de las técnicas evolutivas a la robótica: en su tesis de maestría (UCLA, 1956) propuso evolucionar una serie de circuitos de control similares a las redes neuronales actuales.



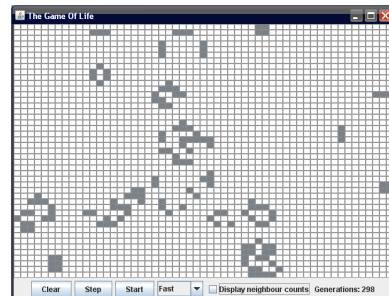
David Fogel (2007): "**George Friedman - Evolving circuits for robots**" [Historic Perspective]
IEEE Computational Intelligence Magazine 1(4):52-54.
DOI 10.1109/MCI.2006.329706



Historia



Nils Aall Barricelli (1912-1993) realizó las primeras simulaciones de un sistema evolutivo en una computadora digital, entre 1953 y 1956, en el Instituto de Estudios Avanzados de Princeton.



Barricelli, Nils Aall (1962). "**Numerical testing of evolution theories: Part I - Theoretical introduction and basic tests**". *Acta Biotheoretica* 16 (1-2): 69–98.
DOI 10.1007/BF01556771

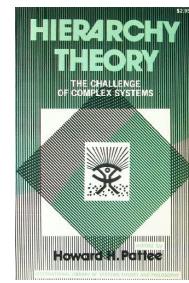
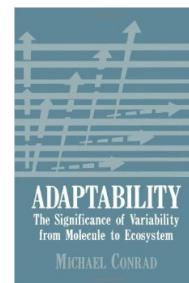


Historia



Michael Earl Conrad (1941-2000) y Howard Hunt Pattee (1926-) fueron de los primeros en simular un ecosistema artificial jerárquico en el que un conjunto de organismos unicelulares estaban sujetos a una estricta ley de conservación de la materia que les inducía a competir por sobrevivir.

- Michael Conrad (1983): **Adaptability**. Plenum Press, ISBN 0306412233.
- Howard H. Pattee (1973): **Hierarchy Theory: The Challenge of Complex Systems**. Georges Braziller, ISBN 0807606731.

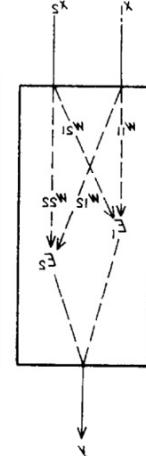


Historia



Conrad propuso también un “modelo de circuitos de aprendizaje evolutivo” en el cual especuló sobre la posibilidad de que el cerebro use el mismo tipo de mecanismos que usa la evolución para aprender (uno de los primeros intentos de utilizar algoritmos evolutivos para entrenar redes neuronales).

- M. Conrad (1969): “Computer experiments on the evolution of coadaptation in a primitive ecosystem,” Ph.D. dissertation, Stanford University.
- M. Conrad & H. H. Pattee (1970): “Evolution experiments with an artificial ecosystem,” Journal of Theoretical Biology 28:393–409.
DOI: 10.1016/0022-5193(70)90077-9
- M. Conrad (1974): “Evolutionary learning circuits,” Journal of Theoretical Biology 46:167–188.
DOI: 10.1016/0022-5193(74)90146-5



Historia



Tierra

El biólogo Thomas S. Ray desarrolló, a principios de los 90, un simulador en el que evolucionaban programas en lenguaje ensamblador, los cuales competían por los ciclos de CPU de un ordenador a la vez que intentaban reproducirse (o sea, copiarse a sí mismos) en la memoria del ordenador.



- Simulación de un ecosistema
[http://en.wikipedia.org/wiki/Tierra_\(computer_simulation\)](http://en.wikipedia.org/wiki/Tierra_(computer_simulation))



Algoritmos evolutivos



EVOLUCIÓN

RESOLUCIÓN DE PROBLEMAS

Entorno

↔ Problema

Individuo

↔ Solución candidata

Adaptación (fitness) ↔ Calidad de la solución



Algoritmos evolutivos



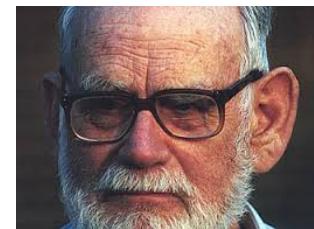
- Algoritmos genéticos
- Programación evolutiva
- Estrategias de evolución
- Programación genética



Algoritmos genéticos



A finales de los 1950s y principios de los 1960s, el biólogo inglés Alex S. Fraser (1923-2002) publicó una serie de trabajos sobre la evolución de sistemas biológicos en una computadora digital, sirviendo de inspiración para los algoritmos genéticos:



Fraser, A. S., "**Simulation of genetic systems by automatic digital computers. I. Introduction,**" Aust. J. Biol. Sci., vol. 10, pp. 484–491, 1957.



Algoritmos genéticos



Hans-Joachin Bremermann (1926-1996) fue el primero en ver la evolución como un proceso de optimización, además de realizar una de las primeras simulaciones con cadenas binarias que se procesaban por medio de reproducción, selección y mutación (predecesor de los algoritmos genéticos).



Chromosome 1	1 1 0 1 0 0 0 1 1 0 1 0
Chromosome 2	0 1 1 1 1 1 1 1 1 0 0

Bremermann, H.J. (1962): "**Optimization through evolution and recombination**". In: Self-Organizing systems 1962, edited M.C. Yovitts et al., Spartan Books, Washington, D.C. pp. 93–106.



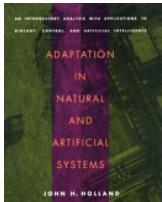
Algoritmos genéticos



John Henry Holland (1929-) desarrolla a principios de los 1960s los “planes reproductivos” y “adaptativos” en un intento por hacer que las computadoras aprendan imitando el proceso de la evolución.



John H. Holland (1962):
“Outline for a logical theory of adaptive systems”,
JACM 9(3):297–314. DOI 10.1145/321127.321128



John H. Holland (1975):
“Adaptation in Natural and Artificial Systems”
The University of Michigan Press, 1975
ISBN 0472084607



Algoritmos genéticos



- Se hace evolucionar una población de individuos (cada uno de los cuales representa una posible solución).
- La población se somete a acciones aleatorias semejantes a las de la evolución biológica (mutaciones y recombinaciones genéticas).
- Los individuos se seleccionan de acuerdo con una función de adaptación en función del cual se decide qué individuos sobreviven (los más adaptados) y cuáles son descartados (los menos aptos).



Algoritmos genéticos

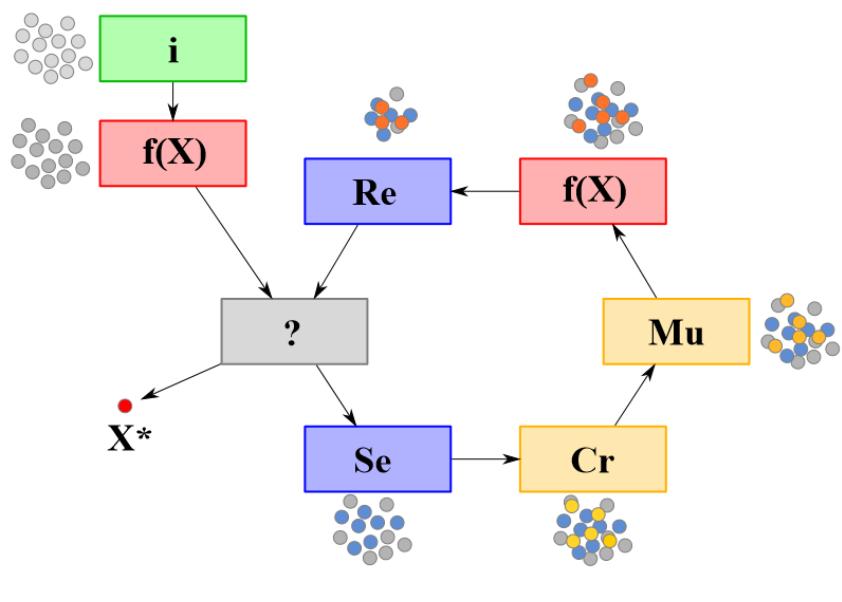


Fases

- Inicialización
- Evaluación

Repetición...

- Selección
- Cruce
- Mutación
- Evaluación
- Reemplazo



Algoritmos genéticos



Algoritmo

```
t ← 0
población(t) ← poblaciónInicial
EVALUAR(población(t))

while not (condición de terminación)
    t ← t + 1
    población(t) ← SELECCIONAR(población(t-1))
    población(t) ← CRUZAR(población(t))
    población(t) ← MUTAR(población(t))
    EVALUAR(población(t))

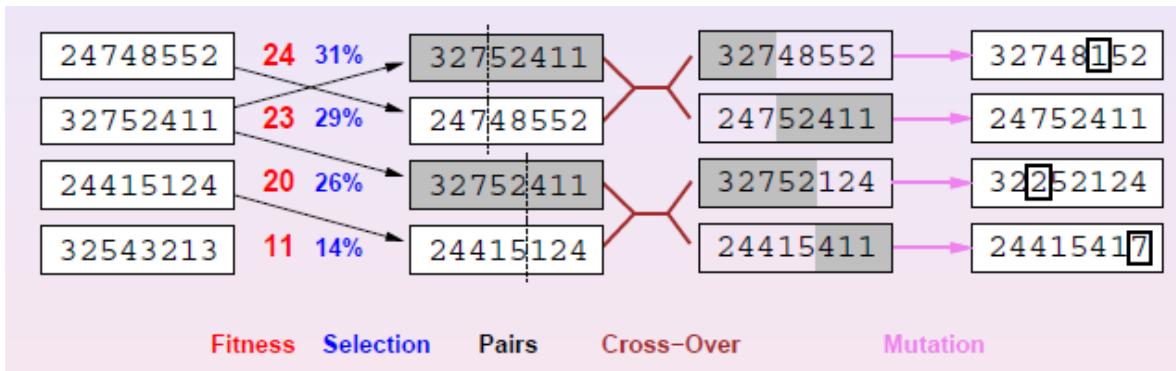
return población(t)
```



Algoritmos genéticos



Selección, cruce & mutación

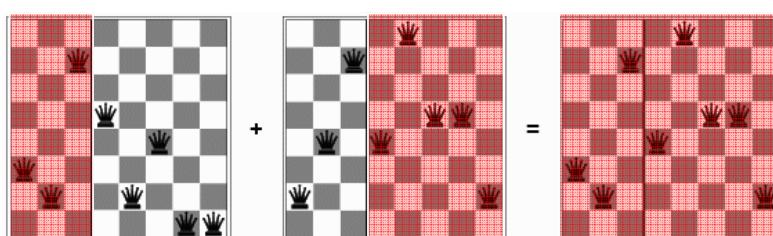


Algoritmos genéticos



Ejemplo: El problema de las N reinas

- Función de evaluación:
Número de parejas de reinas que no se atacan.
- Operador de cruce:



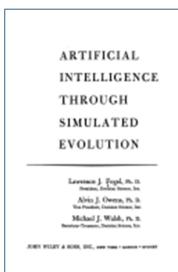
Programación evolutiva



Lawrence J. Fogel (1928-2007) fue el padre de la programación evolutiva: el uso de la evolución simulada en la solución de problemas reales.



Primera tesis doctoral en computación evolutiva:
“On the Organization of Intellect” (UCLA, 1964).



Primer libro de computación evolutiva (con Alvin Owens & Michael Walsh): **“Artificial Intelligence through Simulated Evolution”** (Wiley, 1966)



Programación evolutiva



- Evolución simulada como proceso de aprendizaje con el objetivo de generar “inteligencia artificial” (entendida como comportamiento adaptativo).
- La realización de predicciones acerca del entorno se consideró un requisito previo al comportamiento adaptativo.
- Por tanto, la capacidad de realizar predicciones se considera clave para la inteligencia.

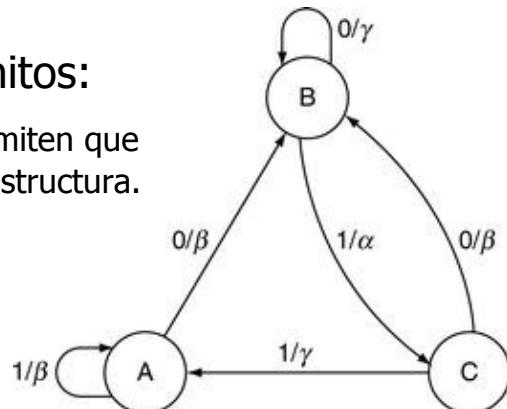


Programación evolutiva



- La estructura del “programa” que se pretende optimizar se mantiene fija, mientras que se permite que sus parámetros evolucionen.
- Fogel utilizó autómatas finitos:

NOTA: Propuestas posteriores permiten que no se tenga que mantener fija la estructura.



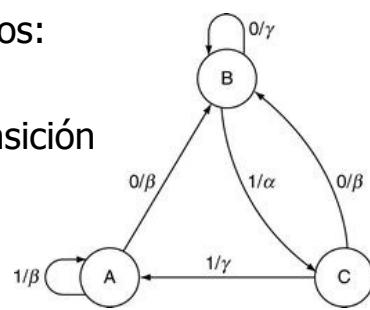
Programación evolutiva



- La evolución se consigue utilizando operadores de mutación.

En el caso de los autómatas finitos:

- Cambiar un símbolo de salida
- Cambiar el destino de una transición
- Agregar un estado
- Borrar un estado
- Cambiar el estado inicial



- Las mutaciones se realizan de acuerdo con una distribución de probabilidad (p.ej. uniforme).



Programación evolutiva



- La programación evolutiva modela la evolución a nivel de especies, por lo que no requiere el uso de un operador de recombinación (diferentes especies no se cruzan entre sí).

"Fogel considered the FSM representations to be equivalent to individual phenotypes being evaluated in an environmental context by selection. As such, the actual methods of modification were less important than simply offering phenotypic change over time in the population of solutions." [Fogel, 1999]



Programación evolutiva



Algoritmo básico

- Inicialización
(generación aleatoria de una población inicial)
- Variación
(operadores de mutación)
- Evaluación
(aptitud [fitness] de cada hijo)
- Selección
(torneo, normalmente estocástico, para determinar qué soluciones se mantendrán en la población)

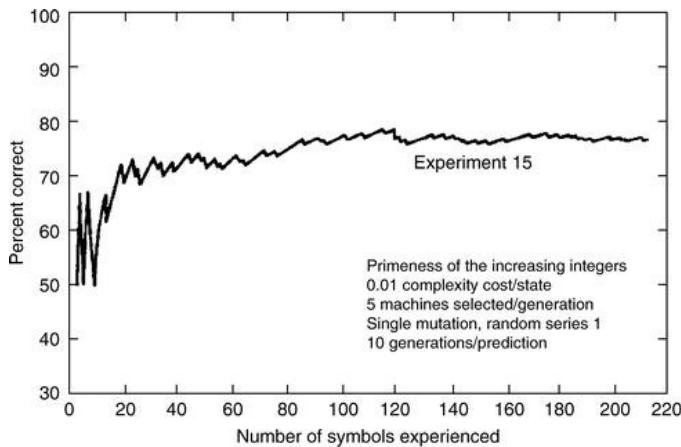


Programación evolutiva



Ejemplo

Autómata finito para predecir si un número es primo



Resultado: 0000...0000 :-(



Programación evolutiva



- Iterativamente se generan soluciones cada vez más aptas para su entorno dada una función de fitness.
- El tamaño de la población (μ) varía dependiendo del problema y de la capacidad de cálculo disponible.
- Cada padre da lugar a λ nuevos hijos (ajustando la probabilidad de cada tipo de mutación y el número de mutaciones por hijo se mantiene la diversidad de la población).



Gary B. Fogel: "**Evolutionary Programming**",
en G. Rozenberg et al. (eds.),
Handbook of Natural Computing, 2012
DOI 10.1007/978-3-540-92910-9_23



Programación evolutiva



Blondie24

Juego de damas (checkers/draughts)

David B. Fogel

<http://en.wikipedia.org/wiki/Blondie24>



- Un Algoritmo evolutivo ajusta 5046 pesos de la red neuronal que evalúa el valor de un movimiento.
- Distintos programas (con sus redes neuronales) juegan entre sí sin intervención humana ni conocimiento específico del juego.
- Después de 840 generaciones (6 meses), la mejor estrategia se probó con jugadores humanos por Internet: mejor que el 99.61% de jugadores



Estrategias de evolución



Ingo Rechenberg, Hans-Paul Schwefel y, más tarde, Peter Bienert, desarrollaron las **estrategias de evolución** como un método de ajustes discretos aleatorios inspirado en el mecanismo de mutación que ocurre en la naturaleza.



- Ingo Rechenberg (1971): Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution (PhD thesis).
- Hans-Paul Schwefel (1974): Numerische Optimierung von Computer-Modellen (PhD thesis). Reprinted by Birkhäuser Verlag, Basel, 1977

English edition: Numerical Optimization of Computer Models, John Wiley and Sons Ltd., 1980. ISBN 0471099880.



Estrategias de evolución



Ingo Rechenberg



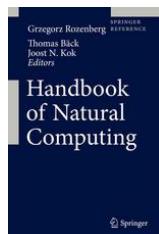
Wie klein kann man Hubschrauber bauen?
Zur Langen Nacht der Wissenschaften 2008 lässt Professor Ingo Rechenberg, Fachgebiet Bionik und Evolutionstechnik an der
Technischen Universität Berlin, eine von ihm entwickelte nur wenige Zentimeter kleine Rotorlibelle fliegen.
Veröffentlichung zur Ankündigung der Langen Nacht der Wissenschaften 2008 frei gegen Urheberennnung: Pressestelle der TUB/Dahl



Estrategias de evolución



- La versión original (1+1)-EE usaba un solo parente y de él se generaba un solo hijo. Este hijo se mantenía si era mejor que el parente, de lo contrario se eliminaba.
- Selección extintiva: los peores individuos tienen una probabilidad de cero de ser seleccionados.



Günter Rudolph: "**Evolutionary Strategies**",
en G. Rozenberg et al. (eds.),
Handbook of Natural Computing, 2012
DOI 10.1007/978-3-540-92910-9_22



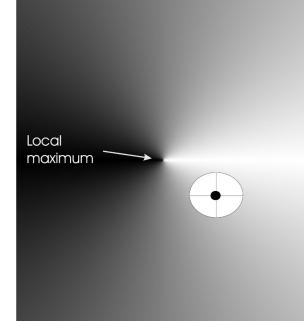
Estrategias de evolución



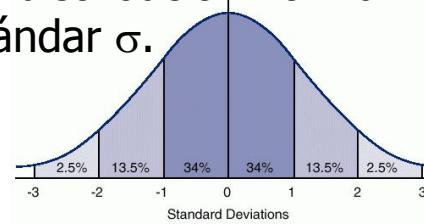
(1+1)-EE

Un individuo nuevo se genera de acuerdo con la expresión:

$$\vec{x}^{t+1} = \vec{x}^t + N(0, \vec{\sigma})$$



donde t se refiere a la generación (iteración) y $N(0, \sigma)$ es un vector de números generados aleatoriamente utilizando una distribución normal con media 0 y desviación estándar σ .



Estrategias de evolución



(μ+1)-EE

Ingo Rechenberg, 1973

- Se introduce el concepto de población (de tamaño μ).
- Se genera un solo hijo a partir de uno de los μ padres y este nuevo individuo reemplaza al peor parente de la población (selección extintiva).



Estrategias de evolución



($\mu+\lambda$)-EE & (μ,λ)-EE

Hans-Paul Schwefel, 1975

Se crean múltiples hijos por generación (λ).

- **($\mu+\lambda$)-EE:** Los μ mejores individuos del conjunto de padres e hijos sobreviven.
- **(μ,λ)-EE:** Sólo los μ mejores hijos sobreviven hasta la siguiente generación.

Presión selectiva elevada: $\lambda \approx 7 \mu$



Estrategias de evolución



Regla del éxito 1/5

- Regla de ajuste de la desviación estándar durante el proceso evolutivo para converger hacia el óptimo.
- La razón entre mutaciones exitosas y el total de mutaciones debe ser 1/5. Si es mayor, debe incrementarse la desviación estándar. Si es menor, debe decrementarse.



Estrategias de evolución



Regla del éxito 1/5

$$\sigma(t) = \begin{cases} \sigma(t-n)/c, & \text{si } p_s > 1/5 \\ \sigma(t-n) \cdot c, & \text{si } p_s < 1/5 \\ \sigma(t-n), & \text{si } p_s = 1/5 \end{cases}$$

donde n es el número de dimensiones,
t es la generación,
 p_s es la frecuencia relativa de mutaciones
exitosas (sobre intervalos de $10n$ individuos)
 $c = 0.817$

- $\sigma(t)$ se ajusta cada n mutaciones.

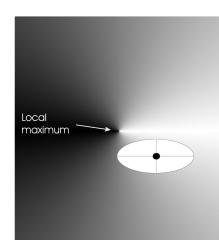


Estrategias de evolución



- No se ajustan sólo los parámetros del problema, sino también los de la propia técnica (**auto-adaptación**):

$$\begin{aligned}\sigma'(i) &= \sigma(i) \times \exp(\tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1)); \\ x'(i) &= x(i) + N(0, \sigma'(i))\end{aligned}$$



- Las estrategias evolutivas simulan la evolución a nivel de individuos, por lo que la **recombinación** es posible. Los operadores de recombinación pueden ser:
 - Sexuales (sobre 2 padres elegidos aleatoriamente).
 - Panmíticos (se elige un parente al azar, que se mantiene fijo mientras se elige un segundo parente para cada componente de su vector).



Estrategias de evolución



Características esenciales

- Inicialización
(generación aleatoria de una población inicial)
- Variación
(operadores de mutación y recombinación)
- Evaluación
(aptitud [fitness] de cada individuo)
- Selección
(selección determinística)
vs. selección estocástica
(programación evolutiva
& algoritmos genéticos)



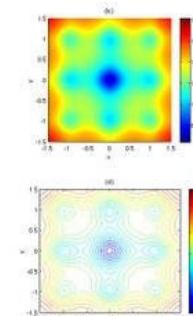
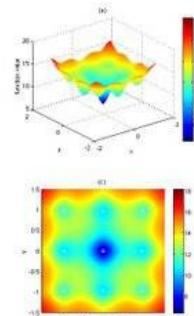
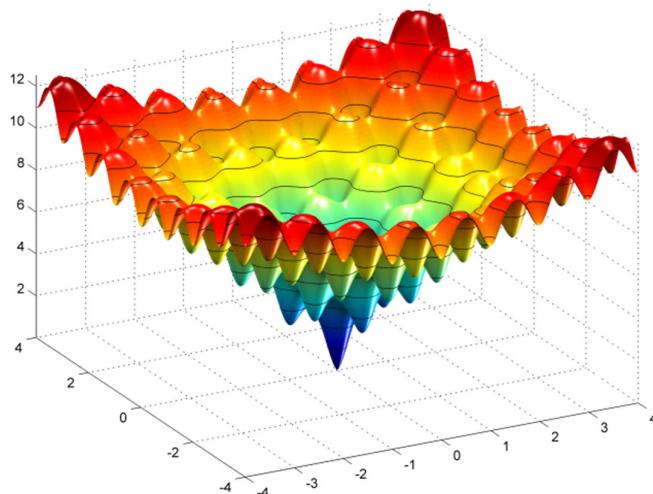
Estrategias de evolución



Ejemplo

Ackley function [Bäck et al '93]

$$f(x) = -20 \cdot \exp\left(-0.2 \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$$



Estrategias de evolución



Demo: Cuadrado mágico

© M. Herdy, TU Berlin



Application

Algoritmo evolutivo:

- Asignación inicial aleatoria.
- Creación de N “mutantes” a partir del individuo actual.
- Selección del mutante con menor error.

Parámetros:

- Step1 (mutación pequeña, lento, encuentra el óptimo)
- Step10 (mutación grande, rápido, falla al saltar por encima del óptimo)
- MStep (autoadaptación)
(mutación ajustada online, rápido, encuentra el óptimo).

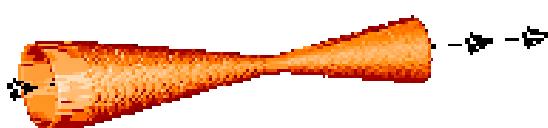


Estrategias de evolución

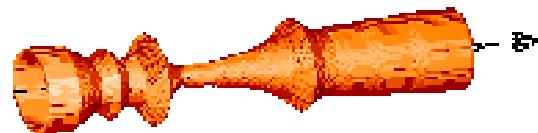


Ejemplo

Optimización de un “jet nozzle”



Initial shape



Final shape



Programación genética



Aunque los primeros intentos por hacer evolucionar programas se remontan a los 1950s y 1960s, hasta los 1980 no se obtuvieron resultados satisfactorios.

- J.F. Hicklin (1986) y C. Fujiki (1986) usaron expresiones-S en LISP para representar programas cuyo objetivo era resolver problemas de teoría de juegos.
- Michael Lynn Cramer (1985) y, posteriormente, John R. Koza (1989) propusieron (de forma independiente) el uso de una representación en árbol sobre la que se implementó un operador de cruce que permitía intercambiar subárboles entre los diferentes programas de una población generada al azar.



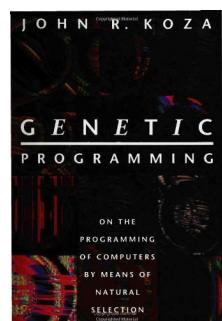
Programación genética



La propuesta de Koza fue la que se acabó imponiendo y, más tarde, se denominó Programación Genética.



- J.R. Koza (1972): **On Inducing a Non-Trivial, Parsimonious Grammar for a Given Sample of Sentences.** PhD Thesis. University of Michigan.



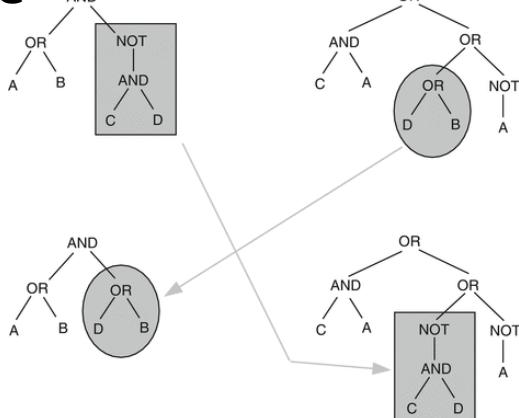
- J.R. Koza (1992): **Genetic Programming: On the Programming of Computers by Means of Natural Selection,** MIT Press. ISBN 0-262-11170-5



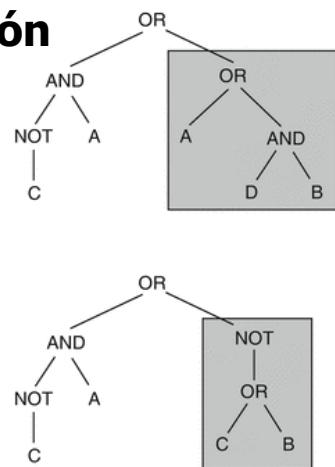
Programación genética



Cruce



Mutación



Leonardo Vanneschi, Riccardo Poli: "**Genetic Programming — Introduction, Applications, Theory and Open Issues**", en G. Rozenberg et al. (eds.), *Handbook of Natural Computing*, 2012
DOI 10.1007/978-3-540-92910-9_24



Algoritmos evolutivos



Comparación de las técnicas evolutivas

	Programación evolutiva	Estrategias de evolución	Algoritmos genéticos	Programación genética
Representación	Real	Real	Binaria++	Árbol
Selección	Estocástica	Determinística	Estocástica	
Variación	Mutación	Mutación & recombinación	Recombinación (cruce) & mutación	
Auto-adaptación	-	Desviaciones	No	



Aplicaciones



Como técnicas heurísticas, los algoritmos evolutivos se utilizan para solucionar problemas complejos con un coste computacional razonable, si bien no garantizan la optimalidad de las soluciones encontradas.

En muchas ocasiones, ni siquiera se puede determinar cómo de cerca del óptimo se encuentra la solución particular obtenida por medio de técnicas heurísticas.



Aplicaciones



- Las técnicas evolutivas usan poblaciones de soluciones potenciales en vez de un solo individuo, lo cual las hace menos sensibles a quedar atrapadas en óptimos locales.
- Las técnicas evolutivas no necesitan disponer de conocimiento específico sobre el problema que intentan resolver (aunque puede ayudar).



Aplicaciones



- Las técnicas evolutivas usan operadores probabilísticos, mientras las técnicas tradicionales utilizan operadores determinísticos.
- Aunque las técnicas evolutivas sean estocásticas, el hecho de que usen operadores probabilísticos no significa que operen de manera análoga a una simple búsqueda aleatoria: la función de aptitud o fitness ayuda a guiar el proceso de búsqueda.



Aplicaciones



Dónde utilizar técnicas evolutivas

- Problemas que no pueden resolverse de forma exacta en tiempo polinómico (clase NP).
- Aplicaciones en las cuales ni siquiera podemos establecer si existe una solución eficiente.

Dónde no utilizar técnicas evolutivas

- Problemas que se pueden resolver utilizando técnicas clásicas (p.ej. simplex para problemas de optimización lineal, programación dinámica...)



Aplicaciones



Las técnicas específicas para un problema

- Suelen funcionar mejor que los algoritmos genéricos de búsqueda en la mayoría de los casos.
- Pueden tener una utilidad limitada (p.ej. eficiencia).
- No siempre funcionan bien para todos los casos..

Los algoritmos evolutivos suelen ser robustos

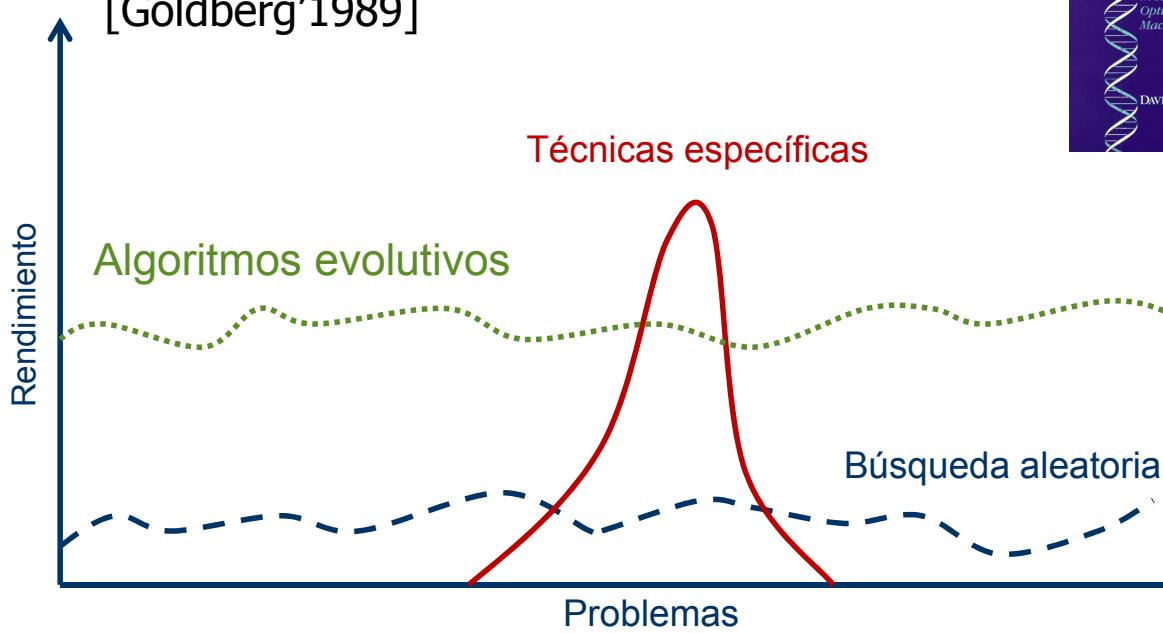
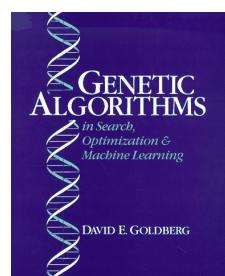
- Rendimiento aceptablemente bueno.
- Para un amplio abanico de problemas y casos.



Aplicaciones



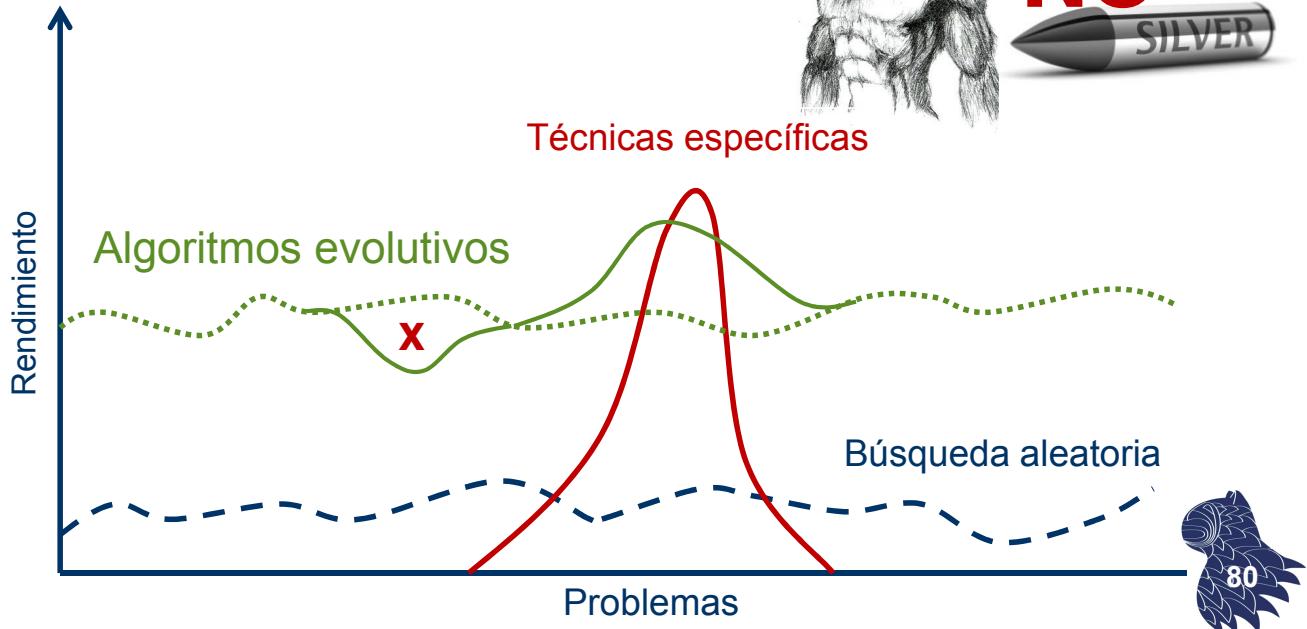
Los algoritmos evolutivos como técnica de resolución de problemas [Goldberg'1989]



Aplicaciones



Incorporación de conocimiento
sobre el problema concreto...

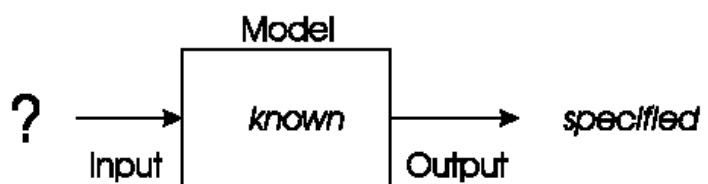


Aplicaciones



Problemas tipo

Optimización



Tenemos un modelo de nuestro sistema y buscamos entradas que nos permitan satisfacer un objetivo previamente establecido.

Ejemplos:

- Asignación de turnos y horarios en universidades, hospitales, call centers...
- Diseño sujeto a una serie de especificaciones y restricciones.

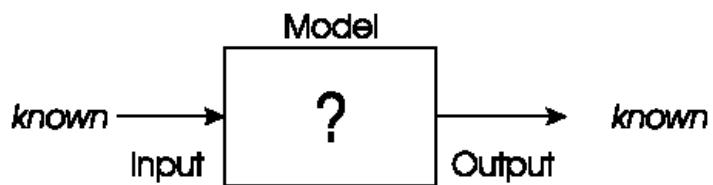


Aplicaciones



Problemas tipo

Modelado



Tenemos un conjunto de entradas junto con sus correspondientes salidas, a partir del cual queremos construir un modelo que proporcione la salida correcta para cada entrada.

"Evolutionary Machine Learning"

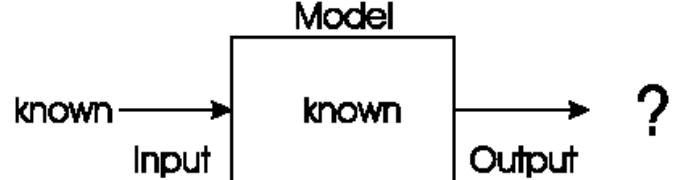


Aplicaciones



Problemas tipo

Simulación



Tenemos un modelo conocido y deseamos conocer cuáles serán las salidas que se obtendrán bajo distintas condiciones (diferentes entradas).

"Artificial Life"



Aplicaciones



Ventajas de las técnicas evolutivas

- Simplicidad (fáciles de implementar).
- Generalidad (aplicables en muchos ámbitos).
- Potencial para incorporar conocimiento sobre el dominio del problema que se pretende resolver y para hibridarse con otras técnicas de optimización.
- Paralelizables.



Aplicaciones



Críticas a las técnicas evolutivas

- Criticadas en sus orígenes (1960s) por los investigadores de la IA simbólica: Se creía que una simple búsqueda aleatoria podía superarlas.
- La programación automática fue considerada una “moda pasajera” en IA: el enfoque evolutivo fue visto como “un intento más” por lograr lo imposible.
- Muchos creen que un AG funciona igual que una técnica de ascensión de colinas que comienza desde varios puntos (se ha demostrado que no es cierto).
- Algunos especialistas de IA clásica las consideran “mal fundamentadas” e “inestables”.

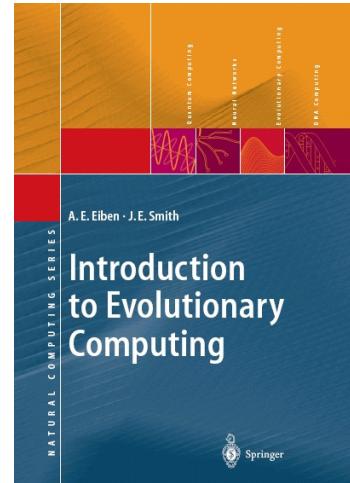


Bibliografía



Lecturas recomendadas

- A.E. Eiben & J.E. Smith:
Introduction to Evolutionary Computing
Springer, 2nd printing, 2007
ISBN 3540401849
<http://www.cs.vu.nl/~gusz/ecbook/ecbook.html>



[Capítulos 1 & 2: Introduction & Evolutionary Algorithms]

Lecturasopcionales

- [Capítulo 4: Evolution strategies]
- [Capítulo 5: Evolutionary programming]



Bibliografía



Bibliografía en castellano

- Carlos Artemio Coello Coello:
Introducción a la Computación Evolutiva.
CINVESTAV-IPN, 2014.
<http://delta.cs.cinvestav.mx/~ccarlo/compevol/apuntes.pdf>

[Capítulo 1: Conceptos básicos]

[Capítulo 2: Un vistazo histórico a la computación evolutiva]

[Capítulo 3: Principales paradigmas]

[Capítulo 16: Técnicas evolutivas alternativas]



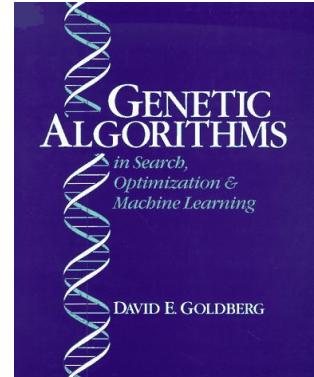
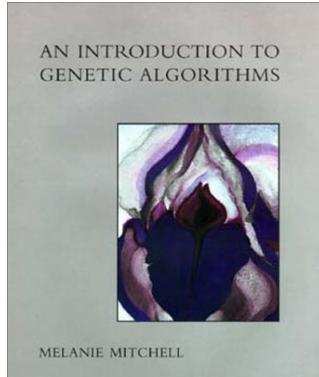
Bibliografía



Bibliografía complementaria

- Melanie Mitchell:
An Introduction to Genetic Algorithms
MIT Press, 1996.
ISBN 0262133164

- David E. Goldberg:
Genetic Algorithms in Search, Optimization & Machine Learning.
Addison-Wesley, 1989.
ISBN 0201157675



88