

# Introducción a la Computación Evolutiva

Carlos A. Coello Coello

*carlos.coellocoello@ccinvestav.mx*

CINVESTAV-IPN

Evolutionary Computation Group (EVOCINV)

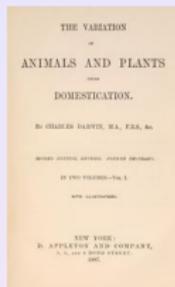
Departamento de Computación

Av. IPN No. 2508, Col. San Pedro Zacatenco

México, D.F. 07360, MEXICO

## Clase 2

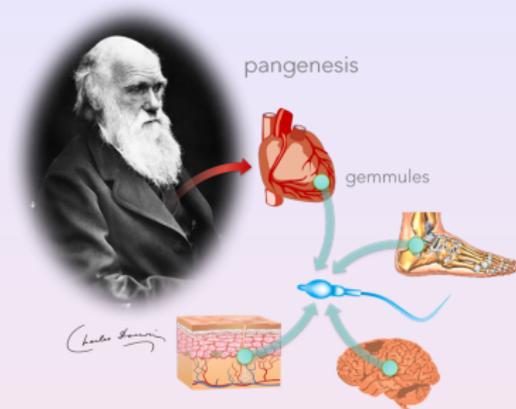
# Teorías sobre el origen de las especies



Para poder sustentar su teoría evolutiva, Darwin tenía que explicar el mecanismo de la herencia: ¿qué se hereda? ¿cómo se hereda? y ¿por qué se hereda?

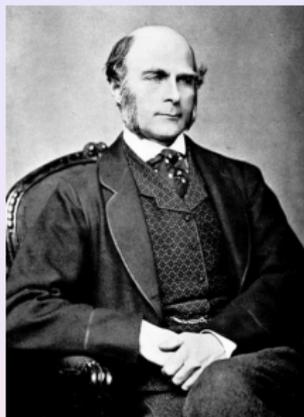
Para responder estas preguntas, Darwin inventa una teoría que se volvería tristemente célebre, a la que denominó **pangénesis** y que publicó en el libro **The Variation of Animals and Plants under Domestication** (1868).

# Teorías sobre el origen de las especies



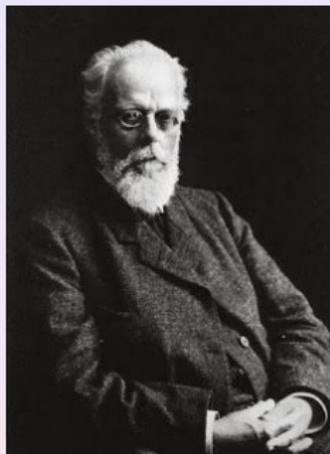
La teoría de la pangénesis nos dice que cada órgano y tejido del cuerpo de un ser vivo genera un tipo de células llamadas *gémulas*. De acuerdo a esta teoría, estos sedimentos van a parar al torrente sanguíneo y de ahí a los gametos sexuales, por lo que son lo que realmente se hereda. Cada vez que un órgano se desarrolla con su uso, éste genera más gémulas, lo que provoca que todos los desarrollos orgánicos que un organismo tenga en vida, vayan a parar a sus células sexuales.

# Teorías sobre el origen de las especies



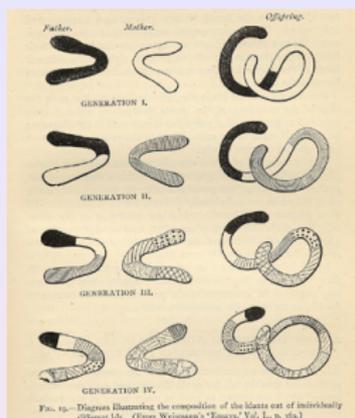
Claramente, la teoría de la pangénesis está alineada con el Lamarckismo y fue refutada por Francis Dalton (quien era primo de Charles Darwin). En una larga serie de experimentos realizados entre 1869 y 1871, Galton hizo transfusiones de sangre entre razas diferentes de conejos y examinó las características de sus hijos. No encontró ninguna evidencia de caracteres transmitidos en la sangre transfundida, lo cual refutaba la teoría de la pangénesis. Darwin murió sin conocer el trabajo pionero realizado por Gregor Mendel en torno a las leyes de la herencia.

# Teorías sobre el origen de las especies



El científico alemán August Weismann formuló la denominada **teoría del plasma germinal** hacia finales del siglo XIX. De acuerdo a esta teoría, la herencia, en un organismo multi-celular, se efectúa únicamente por medio de células *germinales* (la unión de los espermatozoides con el óvulo).

# Teorías sobre el origen de las especies



Según Weismann, las otras células del cuerpo, son las *somáticas* y NO funcionan como agentes hereditarios. Afirmó, además, que este efecto es unidireccional: las células germinales producen células somáticas, pero no puede transmitirse información genética de células somáticas a células germinales. A esto se le conoce como la **barrera de Weismann**.

# Teorías sobre el origen de las especies



La barrera de Weismann tiene implicaciones para la terapia genética en humanos. Si la barrera de Weismann es permeable, entonces los tratamientos genéticos de células somáticas en realidad pueden dar lugar a un cambio heredable en el genoma, los cuales podrían afectar a la especie, en lugar de afectar solo a un individuo. De hecho, esto permitiría redimir el Lamarckismo. Pese a eso, la barrera de Weismann ha sido cuestionada por algunos científicos.

# Teorías sobre el origen de las especies



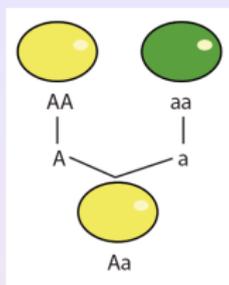
Weismann realizó un experimento en el que cortó las colas de un grupo de ratas durante 22 generaciones (1,592 ratas en total). Weismann reportaría: “durante cinco generaciones, se produjeron 901 ratas jóvenes a partir de padres mutilados artificialmente, y no se obtuvo ni un solo ejemplo de una cola rudimentaria, ni hubo ninguna otra anomalía en esta extremidad”. Esto demostraba claramente que no era posible heredar mutilaciones ocurridas durante el tiempo de vida, y corroboraba su teoría del plasma germinal.

# Teorías sobre el origen de las especies



El monje austriaco Johann Gregor Mendel realizó una serie de experimentos con chícharos durante una buena parte de su vida, enunciando a partir de ellos las tres leyes básicas que gobiernan la herencia. Los resultados de su trabajo los publicó en 1866 en un artículo titulado "*Experiments on Plant Hybridization*", pero tuvo poco impacto (sólo obtuvo 3 citas en sus primeros 35 años), hasta que fue re-descubierto a principios del siglo XX.

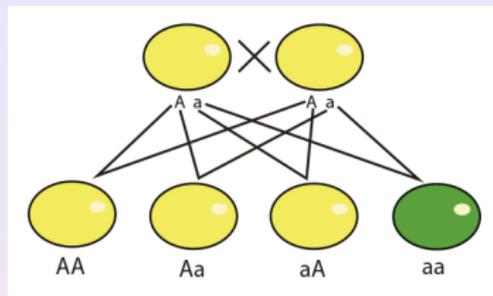
# Teorías sobre el origen de las especies



Las leyes de Mendel son las siguientes:

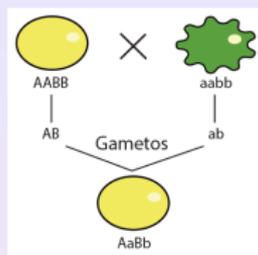
1. **Ley de la Uniformidad:** Al cruzar dos variedades de una especie de raza pura, cada uno de los híbridos de la primera generación tendrá caracteres determinados similares en su fenotipo. Esto se debe a que las razas puras tienen un gen dominante ('A' en la figura de arriba) o un gen recesivo ('a' en la figura de arriba). El genotipo dominante será entonces el que determine la característica o características principales de la primera generación del cruce, pero al mismo tiempo, también serán similares fenotípicamente entre sí, es decir, entre cada individuo de la primera generación. En la figura de arriba, al cruzar un chícharo amarillo (dominante) con uno verde (recesivo), se obtiene un chícharo amarillo.

# Teorías sobre el origen de las especies



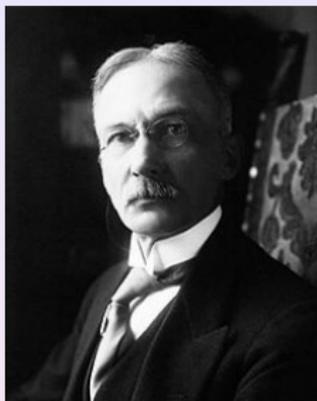
- 2. Ley de la Segregación:** Dice que para que exista la reproducción de dos individuos de una especie, primero debe existir la separación del alelo de cada uno de los pares para que de esta manera se transfiera la información genética al hijo. Un alelo es, la variante genética que permite determinar un rasgo o carácter. Existen entonces, alelos dominantes y alelos recesivos. Mendel, en su experimento, obtuvo solo semillas amarillas en la primera generación, pero en la segunda generación, los alelos se separaron para formar nuevas semillas verdes en menor proporción que las amarillas, pero aun así existentes (ver figura de arriba).

# Teorías sobre el origen de las especies



3. **Ley de Uniformidad:** Hay rasgos heredados que se obtienen de forma independiente, sin relación con el fenotipo. Esta ley se cumple con los genes que no están ligados, es decir, que se encuentran en diferentes cromosomas o que están en zonas muy separadas del mismo cromosoma. Para derivarla, Mendel realizó un cruce de chícharos que producían semillas amarillas y llanas, con chícharos que producían semillas verdes y con textura irregular. En la primera generación se pudieron obtener semillas amarillas y llanas (ver figura de arriba). Sin embargo, al cruzar esta primera generación para obtener una segunda generación, se obtuvieron nuevos tipos de semillas con caracteres diversos pero relacionados con la generación parental (v.g., semillas amarillas y lisas, amarillas y rugosas, verdes y lisas, y verdes y rugosas).

# Teorías sobre el origen de las especies



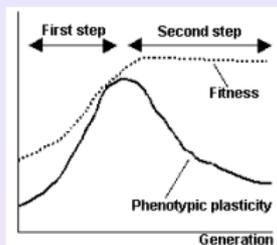
Una pieza interesante de las teorías evolutivas es el denominado **Efecto Baldwin**, conocido también como **evolución baldwiniana** o **evolución ontogénica**. Esta teoría se propuso en un artículo de 1896, titulado **A New Factor in Evolution**, el cual fue escrito por el psicólogo norteamericano **James Mark Baldwin**.

# Teorías sobre el origen de las especies



Baldwin propuso la noción de **plasticidad fenotípica**, que es la capacidad de un organismo para adaptarse a su ambiente durante su tiempo de vida. La capacidad de aprendizaje es el ejemplo más obvio de plasticidad fenotípica, aunque no es el único. Debe aclararse, sin embargo, que la plasticidad fenotípica es típicamente costosa para un individuo. Por ejemplo, aprender requiere energía y tiempo.

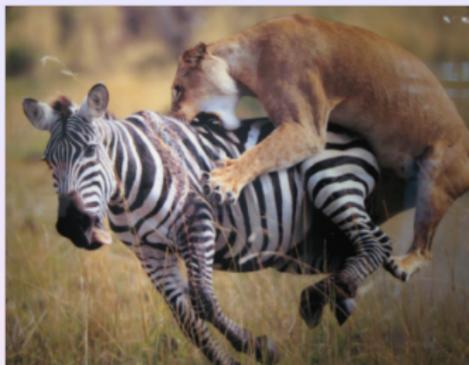
# Teorías sobre el origen de las especies



## Efecto Baldwin

- Supongamos que el aprendizaje no tiene un efecto directo en el ADN de un individuo.
- Pero supongamos también que la capacidad de aprendizaje reduce la necesidad de codificar ciertas características en el ADN de un individuo.
- En consecuencia, la capacidad de aprendizaje de los individuos, mejorará la diversidad.
- Una mayor diversidad contribuirá a una evolución más acelerada.
- **Conclusión:** El aprendizaje individual incrementa (indirectamente) la velocidad de la evolución.

# Teorías sobre el origen de las especies



## Efecto Baldwin

Un ejemplo:

- Aparece un nuevo depredador en el ambiente.
- Los individuos que aprendan a evitarlo serán seleccionados.
- Un incremento en el número de individuos con esta capacidad de aprendizaje, incrementará la diversidad.
- Esto resultará en una evolución más acelerada.

# Teorías sobre el origen de las especies



## Neo-Darwinismo

Hoy se usa el término **Neo-Darwinismo** para describir a la síntesis moderna de la teoría de la evolución de Darwin con la genética de Mendel y la teoría del plasma germinal de Weismann.

# Teorías sobre el origen de las especies



## Neo-Darwinismo

El pensamiento evolutivo actual gira en torno al Neo-Darwinismo, el cual establece que toda la vida en el planeta puede ser explicada a través de sólo 4 procesos:

- 1 Reproducción
- 2 Mutación
- 3 Competencia
- 4 Selección



## ¿Y la computación evolutiva?

Como veremos a lo largo de este curso, la computación evolutiva se basa en el Neo-Darwinismo para simular el proceso evolutivo de una población de individuos que están sujetos a un proceso de selección basado en su aptitud y que se recombinan (sexualmente) entre ellos y se someten a mutaciones aleatorias, de tal forma que los individuos más aptos produzcan descendientes que sean “mejores” que ellos. Este proceso lo usamos para resolver problemas (típicamente de optimización o de clasificación) de alta complejidad.

# Historia de la Computación Evolutiva



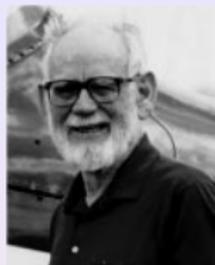
La evolución natural ha sido vista por los científicos como un proceso de aprendizaje desde los 1930s, como se manifiesta en el libro del destacado fisiólogo norteamericano Walter Bradford Cannon titulado **The Wisdom of the Body**, publicado en 1932.



El célebre matemático Alan Mathison Turing reconoció también una conexión “obvia” entre la evolución y el aprendizaje de máquina en su célebre artículo de 1950.

Alan Turing, “**Computing Machinery and Intelligence**, *Mind*, Vol. LIX, No. 236, pp. 433–460, October 1950.

# Historia de la Computación Evolutiva



A fines de los 1950s y principios de los 1960s, el biólogo inglés Alex S. Fraser publicó una serie de trabajos sobre la evolución de sistemas biológicos en una computadora digital, dando la inspiración para lo que después se convertiría en el algoritmo genético. Fraser murió en 2002.

A.S. Fraser, “**Simulation of genetic systems by automatic digital computers. I. Introduction**”, *Australian Journal of Biological Sciences*, Vol. 10, pp. 484–491, 1957.

David Fogel, “**In Memoriam Alex S. Fraser (1923–2002)**”, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 5, pp. 429–430, October 2002.

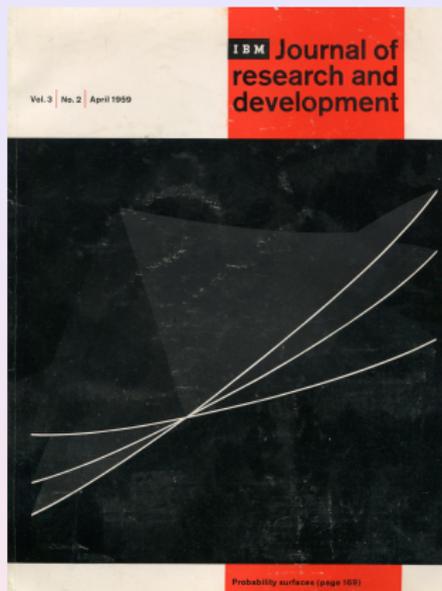


# Historia de la Computación Evolutiva



Aproximadamente en la misma época de Fraser, el estadístico inglés George E. P. Box propuso un enfoque evolutivo para la optimización de la producción industrial. Su técnica, denominada EVOP (*Evolutionary Operation*) sigue en uso hoy en día en la industria química.

# Historia de la Computación Evolutiva



R. M. Friedberg fue uno de los primeros científicos en intentar evolucionar programas de computadora (a fines de los 1950s). Sus experimentos no fueron muy exitosos, y originaron una avalancha de críticas de parte de los investigadores de la IA clásica.



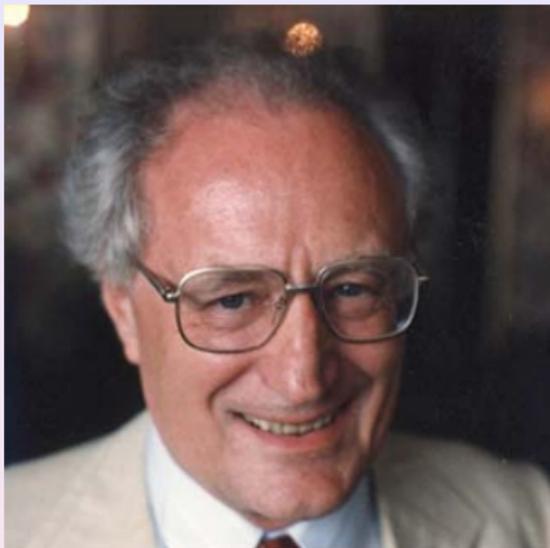
George J. Friedman fue tal vez el primero en proponer una aplicación de las técnicas evolutivas a la robótica: en su tesis de maestría que data de los 1950s, propuso evolucionar una serie de circuitos de control similares a las redes neuronales de hoy en día.

# Historia de la Computación Evolutiva



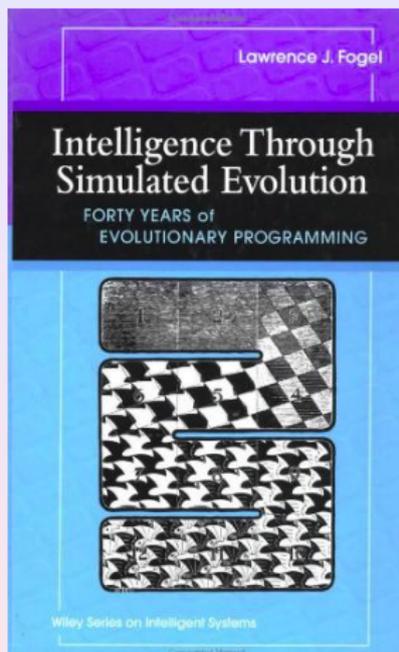
Nils Aall Barricelli fue un matemático noruego-italiano que desarrolló las que probablemente fueron las primeras simulaciones de un sistema evolutivo en una computadora digital, entre 1953 y 1956. Sus experimentos siguieron los lineamientos de una disciplina bautizada a principios de los 1980s como **vida artificial**.

# Historia de la Computación Evolutiva



Hans J. Bremermann fue tal vez 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 (sexual o asexual), selección y mutación, en lo que sería otro claro predecesor del algoritmo genético.

# Historia de la Computación Evolutiva



Lawrence J. Fogel concibió el uso de la evolución simulada en la solución de problemas (sobre todo de predicción) hacia mediados de los 1960s. A su técnica la denominó **Programación Evolutiva**. Fogel falleció en 2007.

# Historia de la Computación Evolutiva



Ingo Rechenberg, Hans-Paul Schwefel y, más tarde, Peter Bienert, desarrollaron hacia mediados de los 1960s un método de ajustes discretos aleatorios inspirado en el mecanismo de mutación que ocurre en la naturaleza. Esta técnica fue llamada la **estrategia evolutiva**.

# Historia de la Computación Evolutiva

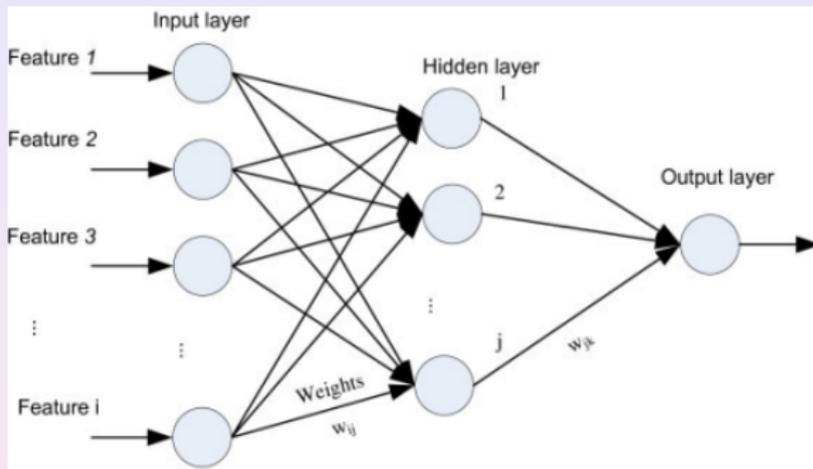


John H. Holland desarrolló a principios de los 1960s los “planes reproductivos” y “adaptativos” en un intento por hacer que las computadoras aprendieran imitando el proceso de la evolución. Esta técnica sería después conocida mundialmente como el **algoritmo genético**.



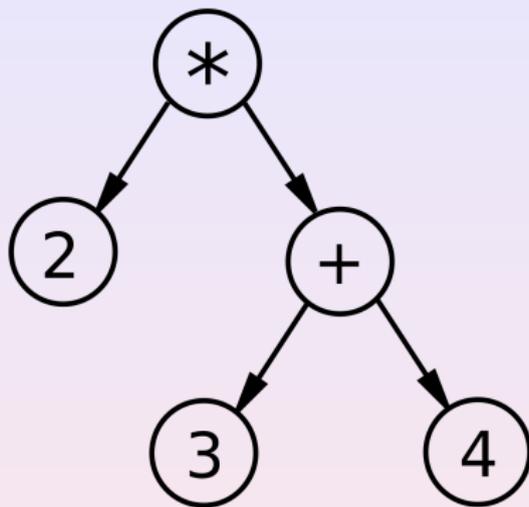
Michael Conrad y H. H. Pattee se cuentan entre los primeros en simular un ecosistema artificial jerárquico en el que un conjunto de organismos unicelulares estaban sujetos a una estricta ley de conservación de la materia que les inducía a competir por sobrevivir. Conrad falleció en 2000.

# Historia de la Computación Evolutiva



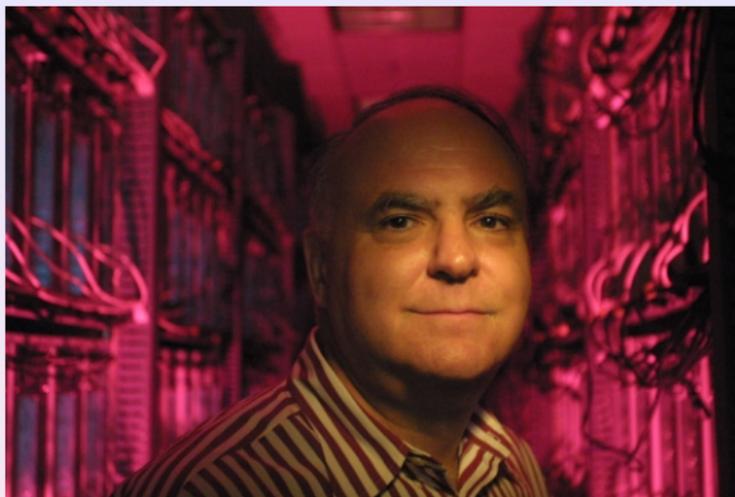
Conrad propuso también en los 1970s un “modelo de circuitos de aprendizaje evolutivo” en el cual especuló sobre la posibilidad de que el cerebro use el mismo tipo de mecanismos que usa la evolución para aprender. Su técnica fue uno de los primeros intentos por utilizar algoritmos evolutivos para entrenar redes neuronales.

# Historia de la Computación Evolutiva



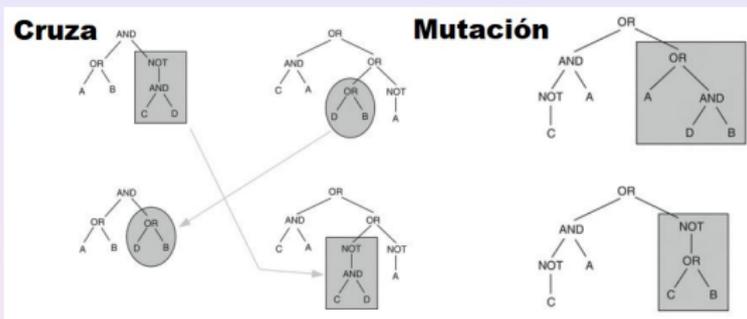
Aunque los primeros intentos por evolucionar programas se remontan a los 1950s y 1960s, fue hasta los 1980s en que se obtuvieron resultados satisfactorios. J.F. Hicklin (1986) y C. Fujiki (1986) usaron expresiones-S en LISP para representar programas cuyo objetivo era resolver problemas de teoría de juegos.

# Historia de la Computación Evolutiva



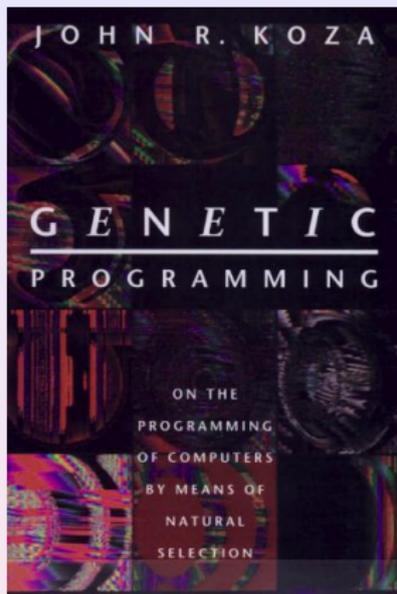
Nichael Lynn Cramer (1985) y posteriormente, John R. Koza (1989) propusieron (de forma independiente) el uso de una representación de árbol en la que se implementó un operador de cruza para intercambiar sub-árboles entre los diferentes programas de una población generada al azar (con ciertas restricciones impuestas por la sintaxis del lenguaje de programación utilizado).

# Historia de la Computación Evolutiva



La diferencia fundamental entre el trabajo de Cramer y el de Koza es que el primero usó una función de aptitud interactiva (es decir, el usuario debía asignar a mano el valor de aptitud de cada árbol de la población), mientras el segundo logró automatizarla.

# Historia de la Computación Evolutiva



La propuesta de Koza fue la que se impuso a la larga, y más tarde se denominó **Programación Genética**. Hoy en día es muy popular y cuenta con una amplia gama de aplicaciones, así como con un evento especializado (EuroGP).

# Historia de la Computación Evolutiva



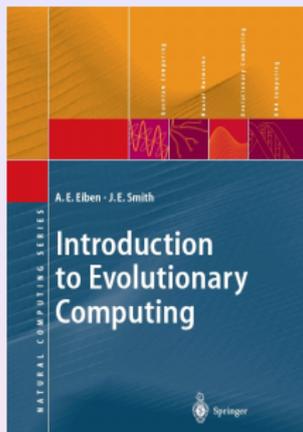
El biólogo Thomas S. Ray desarrolló a principios de los 1990s un simulador muy original en el que evolucionaba programas en lenguaje ensamblador, los cuales competían por ciclos de CPU de una computadora, a la vez que intentaban reproducirse (o sea, copiarse a sí mismos) en la memoria de dicha computadora.

# Historia de la Computación Evolutiva



```
E:\Program\Tierra\Win\Tierra\Tierra.exe
InstExec = 57.695631 Cells = 594 Genotypes = 297 Sizes = 78
Extracted = 0071aaz @ 10 normal
=====
36abh 70 | *****
69abp 31 | *****
36abz 17 | *****
40acd 17 | *****
37abz 14 | *****
  acn  9 | *****
  ach  11 | *****
30aen  7 | *****
45adc  9 | *****
74aax  9 | *****
36acu  8 | *****
71aaz  8 | *****
36aci  5 | *****
44abu  6 | *****
39adu  5 | *****
36acs  7 | *****
70ahn  5 | *****
36acp  4 | *****
40acn  4 | *****
44abo  4 | *****
46acl  2 | *****
68abg  4 | *****
=====
Press Any Key For Menu ...
```

En este simulador, denominado **Tierra**, se partía de un programa único con la capacidad de auto-replicarse, al que se denominaba “ancestro”. Con base en este programa se generaban “criaturas” nuevas (segmentos de código), las cuales a su vez se podían sub-dividir para dar origen a nuevas criaturas. Tierra es uno de los pocos intentos por simular un ecosistema con el propósito expreso de observar los comportamientos que emergen de la dinámica evolutiva del mismo.



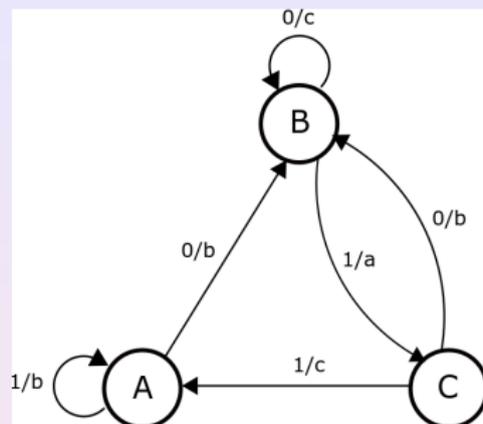
Tres son los paradigmas principales que conforman la computación evolutiva:

- Programación evolutiva
- Estrategias Evolutivas
- Algoritmos Genéticos



Lawrence J. Fogel propuso en 1960 una técnica denominada **programación evolutiva**, en la cual la inteligencia se ve como un comportamiento adaptativo.

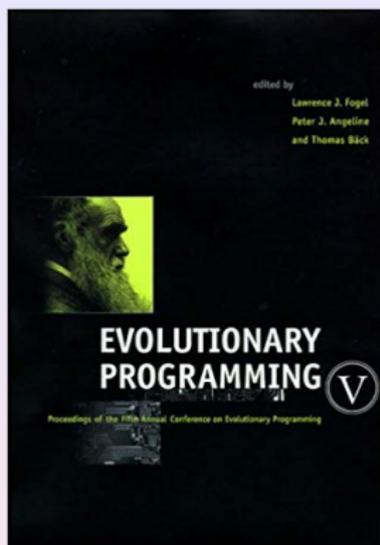
# Ejemplo de Programación Evolutiva



Autómata finito de 3 estados. Los símbolos a la izquierda del "/" son de entrada, y los de la derecha son de salida. El estado inicial es **C**.

Estado Actual	C	B	C	A	A	B
Símbolo de Entrada	0	1	1	1	0	0
Estado siguiente	B	C	A	A	B	B
Símbolo de salida	b	a	c	b	b	c

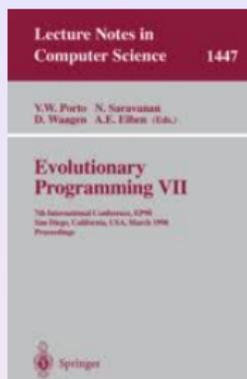
# Ejemplo de Programación Evolutiva



Para este ejemplo, existen 5 tipos de mutadores: (1) cambiar un símbolo de salida, (2) cambiar una transición, (3) agregar un estado, (4) borrar un estado y (5) cambiar el estado inicial. El borrado de un estado y el cambio del estado inicial sólo se permiten en máquinas de más de un estado. Las mutaciones se eligen con respecto a una distribución probabilística, que es normalmente uniforme.



La programación evolutiva enfatiza los nexos de comportamiento entre padres e hijos, en vez de buscar emular operadores genéticos específicos (como en el caso de los algoritmos genéticos).

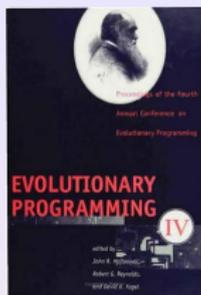


## Algoritmo Básico

- 1 Generar aleatoriamente una población inicial.
- 2 Aplicar mutación (normalmente hay varios operadores disponibles).
- 3 Calcular la aptitud de cada hijo y usar un proceso de selección mediante torneo (normalmente estocástico) para determinar cuáles serán las soluciones que se retendrán.
- 4 Regresar al paso 2, a menos que se haya alcanzado un cierto criterio de paro.



La programación evolutiva es una abstracción de la evolución al nivel de las especies, por lo que no se requiere el uso de un operador de recombinación ya que especies diferentes no se pueden cruzar entre sí. Asimismo, usa selección probabilística.



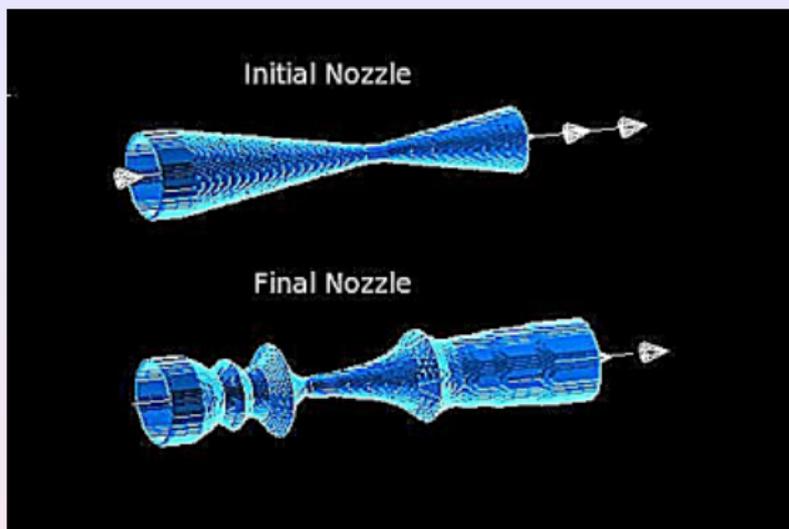
## Aplicaciones de la Programación Evolutiva

- Predicción
- Generalización
- Juegos
- Control automático
- Problema del viajero
- Planeación de rutas
- Diseño y entrenamiento de redes neuronales
- Reconocimiento de patrones



Las **estrategias evolutivas** fueron desarrolladas en 1964 por un grupo de estudiantes alemanes de ingeniería encabezado por Ingo Rechenberg. Su objetivo era resolver problemas hidrodinámicos de alto grado de complejidad.

# Estrategias Evolutivas



La versión original (1+1)-EE usaba un solo padre y con él se generaba un solo hijo. Este hijo se mantenía si era mejor que el padre, o de lo contrario se eliminaba (a este tipo de selección se le llama **extintiva**, porque los peores individuos tienen una probabilidad de cero de ser seleccionados).



En la (1+1)-EE, un individuo nuevo es generado usando:

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

donde  $t$  se refiere a la generación (o iteración) en la que nos encontramos, y  $N(0, \vec{\sigma})$  es un vector de números Gaussianos independientes con una media de cero y desviación estándar  $\vec{\sigma}$

## Ejemplo de una (1+1)-EE

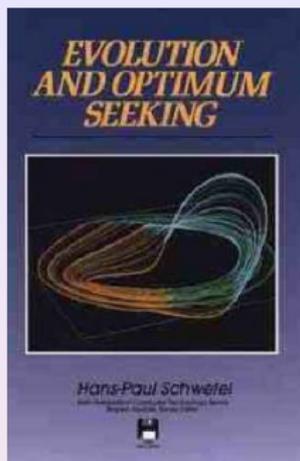
Supongamos que queremos **maximizar**:

$$f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$

$$\text{donde } -2.048 \leq x_1, x_2 \leq 2.048$$

Ahora, supongamos que nuestra población consiste del siguiente individuo (generado de forma aleatoria):

$$(\vec{x}^t, \vec{\sigma}) = (-1.0, 1.0), (1.0, 1.0)$$



## Ejemplo de una (1+1)-EE

Supongamos también que las mutaciones producidas son las siguientes:

$$x_1^{t+1} = x_1^t + N(0,1.0) = -1.0 + 0.61 = -0.39$$

$$x_2^{t+1} = x_2^t + N(0,1.0) = 1 + 0.57 = 1.57$$

## Ejemplo de una (1+1)-EE

Ahora, comparamos al padre con el hijo:

Padre:

$$f(x^t) = f(-1.0, 1.0) = 4.0$$

Hijo:

$$f(x^{t+1}) = f(-0.39, 1.57) = 201.416$$

Dado que:

$$201.416 > 4.0$$

el hijo reemplazará al padre en la siguiente generación.

# Estrategias Evolutivas



Ingo Rechenberg (1973) introdujo el concepto de población, al proponer una estrategia evolutiva llamada  $(\mu + 1) - EE$ , en la cual hay  $\mu$  padres y se genera un solo hijo, el cual puede reemplazar al peor padre de la población (selección extintiva).



Schwefel (1975) introdujo el uso de múltiples hijos en las denominadas  $(\mu + \lambda) - EEs$  y  $(\mu, \lambda) - EEs$ . La notación se refiere al mecanismo de selección utilizado:

- (a) En el primer caso, los  $\mu$  mejores individuos obtenidos de la unión de padres e hijos sobreviven.
- (b) En el segundo caso, sólo los  $\mu$  mejores hijos de la siguiente generación sobreviven.

## Convergencia

Rechenberg formuló una regla para ajustar la desviación estándar de forma determinística durante el proceso evolutivo de tal manera que el procedimiento convergiera hacia el óptimo.

La denominada “**regla de éxito 1/5**”, dice lo siguiente:

*“La razón entre mutaciones exitosas y el total de mutaciones debe ser exactamente  $\frac{1}{5}$ . Si es mayor, entonces debe incrementarse la desviación estándar. Si es menor, entonces debe decrementarse.”*

## Convergencia

Formalmente:

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

donde  $n$  es el número de dimensiones,  $t$  es la generación,  $p_s$  es la frecuencia relativa de mutaciones exitosas medida sobre intervalos de (por ejemplo)  $10 \cdot n$  individuos, y  $c=0.817$  (este valor fue derivado teóricamente por Schwefel).  $\sigma(t)$  se ajusta cada  $n$  mutaciones.

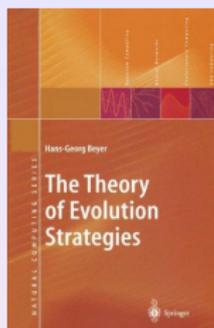
## Auto-Adaptación

En las estrategias evolutivas se evoluciona no sólo a las variables del problema, sino también a los parámetros mismos de la técnica (es decir, las desviaciones estándar). A esto se le llama “**auto-adaptación**”.

Los padres se mutan usando las siguientes fórmulas:

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

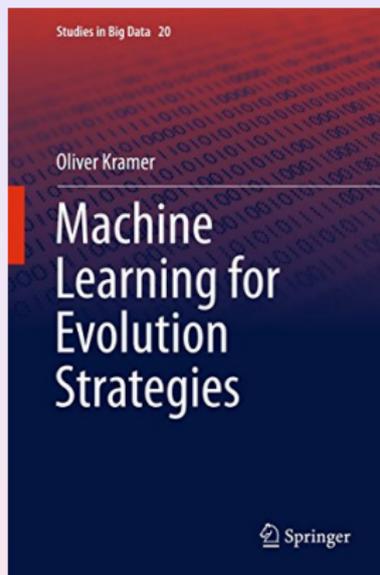
donde  $\tau$  y  $\tau'$  son constantes de proporcionalidad que están en función de  $n$ .



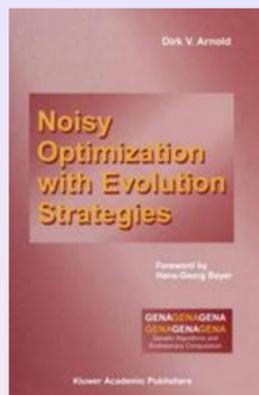
## Recombinación

Los operadores de recombinación de las estrategias evolutivas pueden ser:

- **Sexuales:** el operador actúa sobre 2 individuos elegidos aleatoriamente de la población de padres.
- **Panmíticos:** se elige un solo padre al azar, y se mantiene fijo mientras se elige al azar un segundo padre (de entre toda la población) para cada componente de sus vectores.



Las estrategias evolutivas simulan el proceso evolutivo al nivel de los individuos, por lo que la recombinación es posible. Asimismo, usan normalmente selección determinística.



## Aplicaciones

- Problemas de ruteo y redes
- Bioquímica
- Óptica
- Diseño en ingeniería
- Magnetismo

# Estrategias Evolutivas vs. Programación Evolutiva

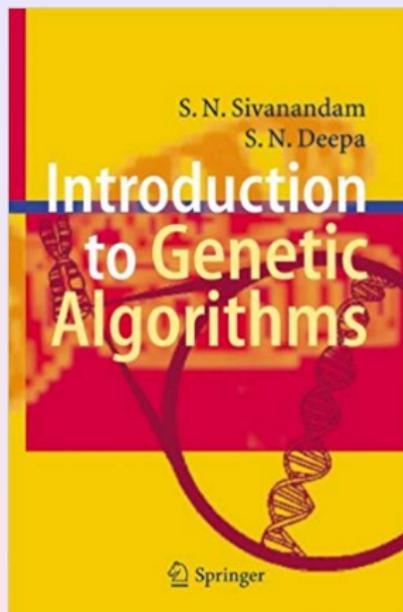
La Programación Evolutiva usa normalmente selección estocástica, mientras que las estrategias evolutivas usan selección determinística.

Ambas técnicas operan a nivel fenotípico.

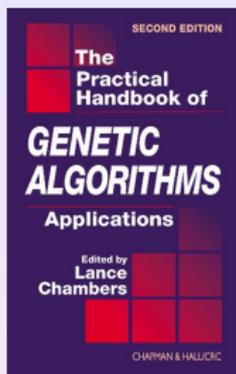
La programación evolutiva es una abstracción de la evolución al nivel de las especies, por lo que no se requiere el uso de un operador de recombinación (diferentes especies no se pueden cruzar entre sí). En contraste, las estrategias evolutivas son una abstracción de la evolución al nivel de un individuo, por lo que la recombinación es posible.



Los algoritmos genéticos (denominados originalmente “planes reproductivos”) fueron desarrollados por John H. Holland a principios de los 1960s. Su motivación principal fue el aprendizaje de máquina.

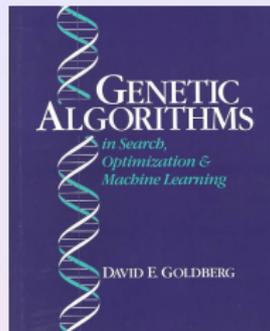


El algoritmo genético enfatiza la importancia de la cruce sexual (operador principal) sobre el de la mutación (operador secundario), y usa selección probabilística.



## Algoritmo Básico

- Generar (aleatoriamente) una población inicial
- Calcular aptitud de cada individuo
- Seleccionar (probabilísticamente) con base en aptitud
- Aplicar operadores genéticos (cruza y mutación) para generar la siguiente población
- Ciclar hasta que cierta condición se satisfaga



La representación tradicional es la cadena binaria del tipo:

$\underbrace{0110}_{\text{Cadena1}} \quad \underbrace{1101}_{\text{Cadena2}} \quad \underbrace{0011}_{\text{Cadena3}} \quad \underbrace{1001}_{\text{Cadena4}}$

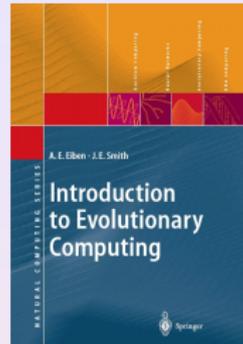
A la cadena se le llama **cromosoma**. A cada posición de la cadena se le denomina **gene** y al valor dentro de esta posición se le llama **alelo**.

# Algoritmos Genéticos

Para poder aplicar el Algoritmo Genético se requiere de los 5 componentes básicos siguientes:

- Una representación de las soluciones potenciales del problema.
- Una forma de crear una población inicial de posibles soluciones (normalmente, un proceso aleatorio).
- Una función de evaluación que juegue el papel del ambiente, clasificando las soluciones en términos de su **“aptitud”**.
- Operadores genéticos que alteren la composición de los hijos que se producirán para las siguientes generaciones (normalmente, cruza y mutación).
- Valores para los diferentes parámetros que utiliza el Algoritmo Genético (tamaño de la población, probabilidad de cruza, probabilidad de mutación, número máximo de generaciones, etc.)

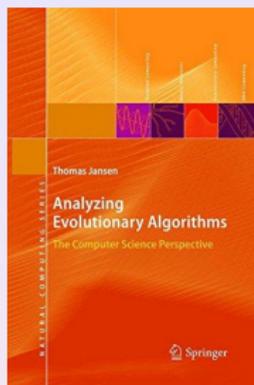
# Algoritmos Genéticos vs. otros Algoritmos Evolutivos



El Algoritmo Genético usa selección probabilística al igual que la Programación Evolutiva, y en contraposición a la selección determinística de las Estrategias Evolutivas.

El Algoritmo Genético usa representación binaria para codificar las soluciones a un problema, por lo cual se evoluciona el genotipo y no el fenotipo como en la Programación Evolutiva o las Estrategias Evolutivas.

# Algoritmos Genéticos vs. otros Algoritmos Evolutivos



El operador principal en el Algoritmo Genético es la cruce, y la mutación es un operador secundario. En la Programación Evolutiva, no hay cruce y en las Estrategias Evolutivas, la cruce es un operador secundario.

Los Algoritmos Genéticos no son, en su versión estándar, auto-adaptativos, a diferencia de las Estrategias Evolutivas. Existe también una versión auto-adaptativa de la Programación Evolutiva.

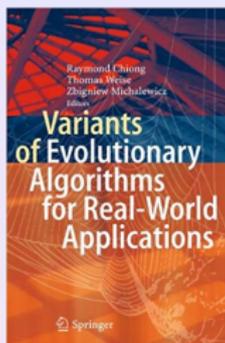
# Algoritmos Genéticos vs. otros Algoritmos Evolutivos



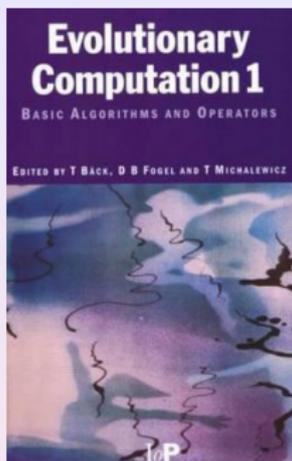
Ha sido demostrado matemáticamente [Rudolph, 1994] que el Algoritmo Genético requiere de elitismo para poder converger al óptimo.

Günter Rudolph, “**Convergence Analysis of Canonical Genetic Algorithms**”, *IEEE Transactions on Neural Networks*, Vol. 5, pp. 96–101, January 1994.

# Aplicaciones de los Algoritmos Genéticos



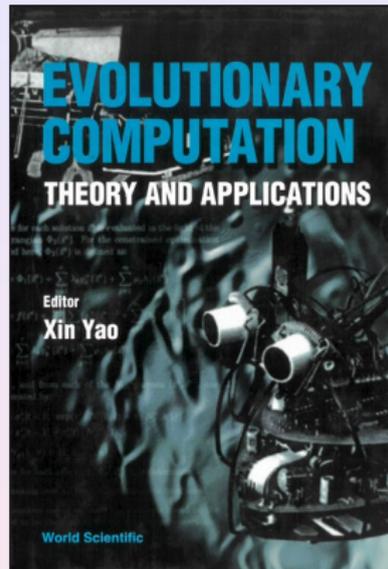
- Optimización (estructural, de topologías, numérica, combinatoria, etc.)
- Aprendizaje de máquina (sistemas de clasificadores).
- Bases de datos (optimización de consultas).
- Reconocimiento de patrones (por ejemplo, procesamiento de imágenes).
- Generación de gramáticas (regulares, libres de contexto, etc.)
- Planeación de movimientos de robots.
- Predicción.



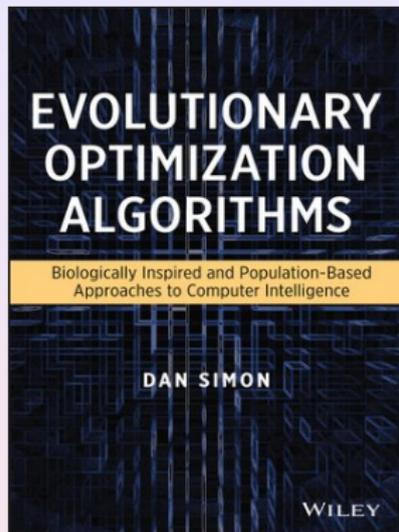
Hoy en día, suele resultar difícil distinguir un algoritmo perteneciente a un paradigma, de otro, por lo cual es más común utilizar el término **Algoritmo Evolutivo** en un sentido genérico para referirse a cualquier algoritmo estocástico que selecciona soluciones con respecto a una función de aptitud.

En general, los algoritmos evolutivos requieren de 4 elementos básicos:

- Una codificación de las estructuras (o sea, los individuos) que serán evolucionadas.
- Operadores que actúen sobre los “individuos” (típicamente, cruza y mutación).
- Una función de aptitud que nos indique qué tan bueno es un individuo con respecto a los demás.
- Un mecanismo de selección para elegir a los padres que producirán nuevos individuos.

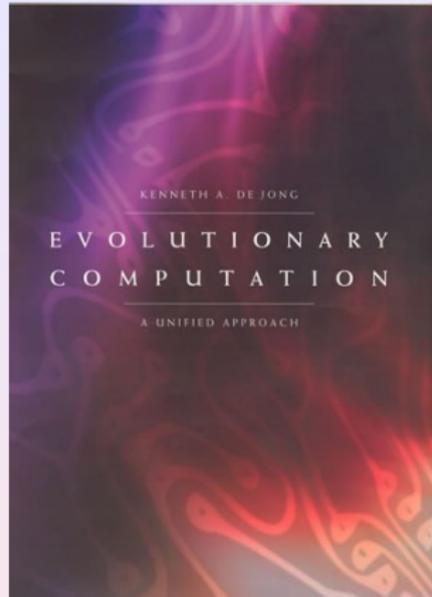


Los algoritmos evolutivos usan una población de soluciones potenciales en vez de buscar con un solo individuo a la vez, lo cual los hace menos sensibles a quedar atrapadas en mínimos/máximos locales.



Los algoritmos evolutivos no necesitan conocimientos específicos sobre el problema que intentan resolver, pero si tal conocimiento está disponible, pueden aprovecharlo.

# Algoritmos Evolutivos vs. Técnicas Tradicionales



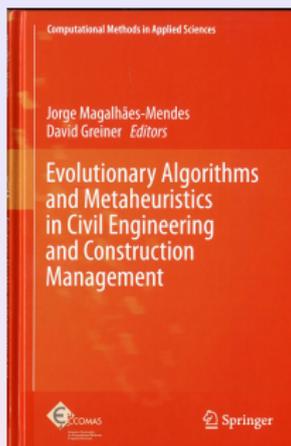
Los algoritmos evolutivos usan operadores probabilísticos, mientras que las técnicas tradicionales utilizan operadores determinísticos.

# Ventajas de los Algoritmos Evolutivos



- Simplicidad Conceptual.
- Amplia aplicabilidad.
- Superiores a las técnicas tradicionales en muchos problemas del mundo real.
- Tienen el potencial para incorporar conocimiento sobre el dominio y para hibridizarse con otras técnicas de búsqueda/optimización.
- Pueden explotar fácilmente las arquitecturas en paralelo.
- Son robustos a los cambios dinámicos.

# Desventajas de los Algoritmos Evolutivos



- No pueden garantizar convergencia al óptimo global, excepto para casos muy particulares.
- Debido a su naturaleza estocástica, requieren muestrear el espacio de búsqueda, lo que implica un mayor costo computacional, el cual hace que no puedan ser utilizadas en ciertas aplicaciones del mundo real que requieren de cómputo intensivo (p.ej., problemas de ingeniería aeronáutica).

# Comparaciones entre Algoritmos Evolutivos

	<b>Estrategias Evolutivas</b>	<b>Programación Evolutiva</b>	<b>Algoritmo Genético</b>
Representación	Real	Real	Binaria
Función de Aptitud	Valor de la Función Objetivo	Valor de la Función Objetivo ajustada	Valor de la Función Objetivo ajustada
Auto-Adaptación	Desviaciones Estándar y ángulos de rotación	Ninguna Varianzas (PE-estándar), Coeficientes de correlación (meta-PE)	Ninguna
Mutación	Gaussiana, operador principal	Gaussiana, operador único	Inversión de bits, operador secundario
Recombinación	Discreta e intermedia, sexual y panmítica, importante para la auto-adaptación	Ninguna	Cruza de z-puntos, cruza uniforme, únicamente sexual, operador principal

# Comparaciones entre Técnicas

	<b>Estrategias Evolutivas</b>	<b>Programación Evolutiva</b>	<b>Algoritmo Genético</b>
Selección	Determinística, extintiva o basada en la preservación	Probabilística, extintiva	Probabilística, basada en la preservación
Restricciones	Restricciones arbitrarias de desigualdad	Ninguna	Límites simples mediante el mecanismo de codificación
Teoría	Velocidad de Convergencia para casos especiales, (1+1)-ES, (1+ $\lambda$ )-ES, Convergencia Global para $(\mu + \lambda)$ -ES	Velocidad de Convergencia para casos especiales, (1+1)-PE, Convergencia Global para meta-PE	Teoría de los Esquemas, Convergencia Global para la versión elitista