

# Introducción a la Computación Evolutiva

Dr. Carlos A. Coello Coello

Departamento de Computación

CINVESTAV-IPN

Av. IPN No. 2508

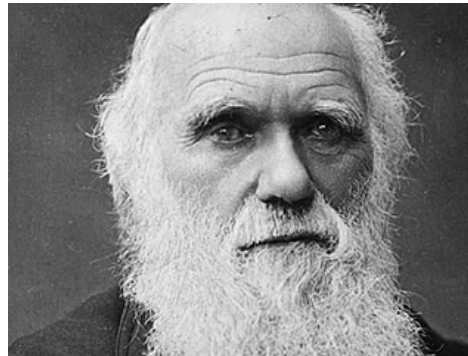
Col. San Pedro Zacatenco

México, D.F. 07300

email: [ccoello@cs.cinvestav.mx](mailto:ccoello@cs.cinvestav.mx)

http: [//delta.cs.cinvestav.mx/~ccoello](http://delta.cs.cinvestav.mx/~ccoello)

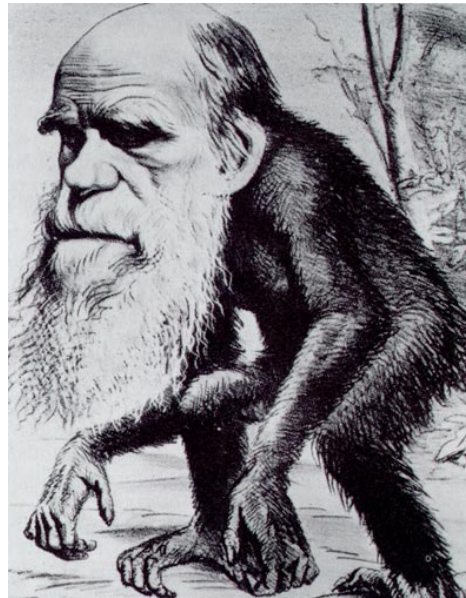
El naturalista inglés Charles Darwin presentó en 1858 los primeros bosquejos de su (ahora famosa) teoría sobre el origen de las especies. Su libro, titulado **On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life**, se publicó el 24 de noviembre de 1859 y se considera como una de las obras científicas más importantes de todos los tiempos.



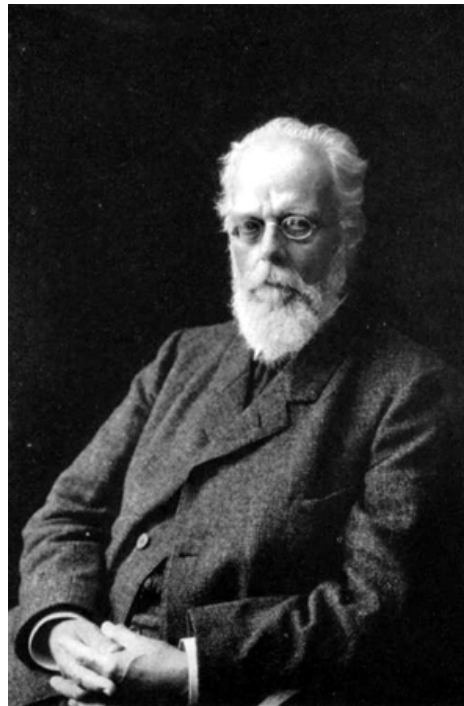
Darwin entendió que toda población consiste de individuos ligeramente distintos entre sí y que estas pequeñas variaciones hacen que cada uno tenga distintas capacidades para adaptarse a su medio ambiente, así como para reproducirse y para transmitir sus rasgos a sus descendientes.



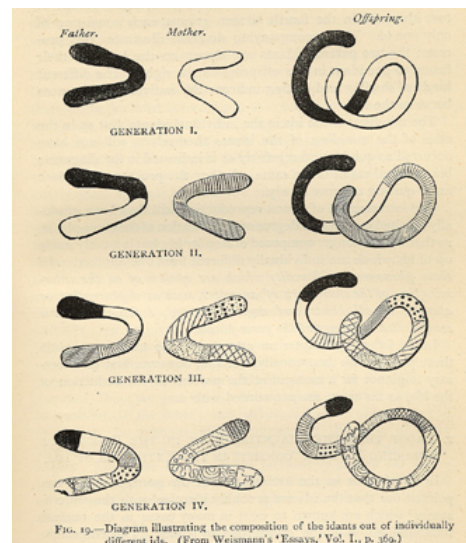
Con el paso del tiempo (o generaciones), los rasgos de los individuos que mejor se adaptaron a las condiciones del medio ambiente, se vuelven más comunes, haciendo que la población, en su conjunto, evoluciones. Darwin llamó a este proceso “descendencia con modificación”. Del mismo modo, la naturaleza selecciona las especies mejor adaptadas para sobrevivir y reproducirse. A este proceso, Darwin lo denominó “selección natural”.



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).



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**.



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.



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 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.





Una pieza interesante de las teorías evolutiva 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.



Este artículo 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.



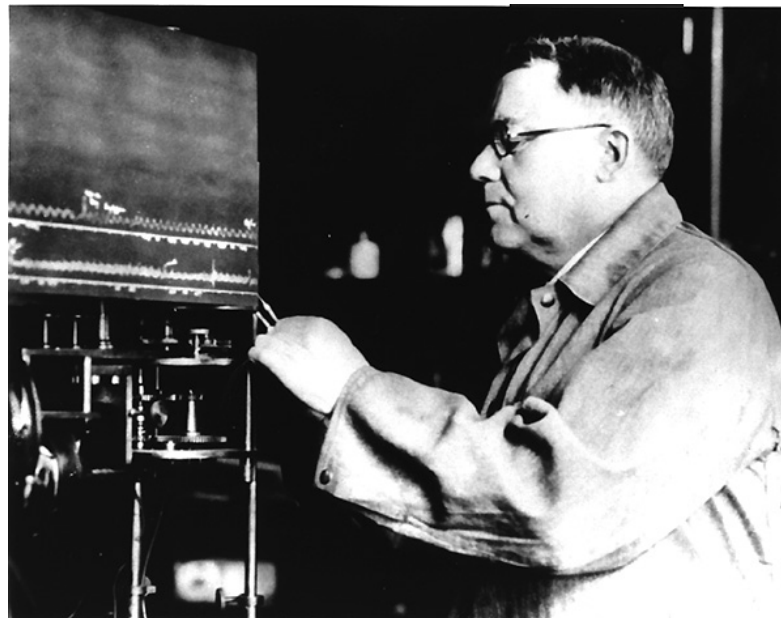
Hoy se usa el término **Neo-Darwinismo** para describir a la síntesis moderna de la teoría de la evolución de Darwin con la genética de Mendel y la teoría del plasma germinal de Weismann.



- 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:
  - Reproducción
  - Mutación
  - Competencia
  - Selección

## Historia de la Computación Evolutiva

La evolución natural ha sido vista como un proceso de aprendizaje desde los 1930s, con el trabajo de Walter Bradford Cannon (*The Wisdom of the Body*).



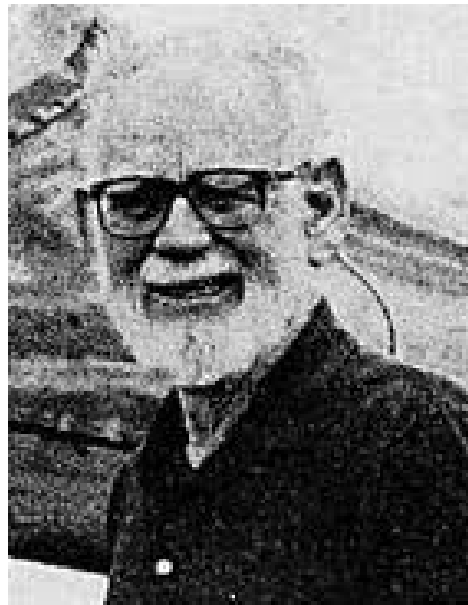
## Historia de la Computación Evolutiva

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



## Historia de la Computación Evolutiva

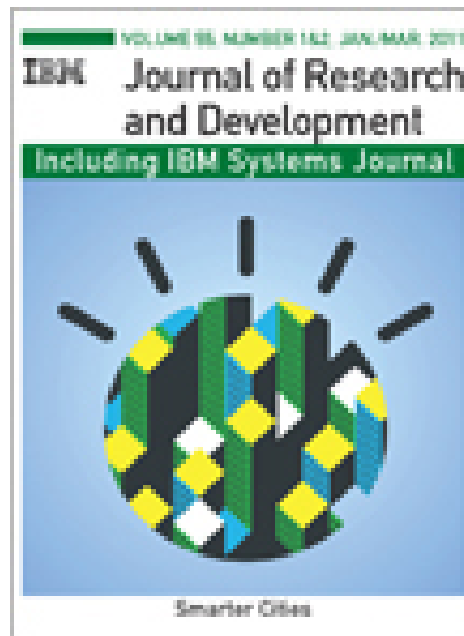
Aproximadamente en la misma época de Fraser, el estadístico inglés George E. P. Box propuso un enfoque evolutivo para la optimización de la producción industrial. Su técnica, denominada EVOP (*Evolutionary Operation*) sigue en uso hoy en día en la industria química.





## 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.



## Historia de la Computación Evolutiva

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 Bielecki, 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 se denominó **estrategias evolutivas**.



## 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 aprendan imitando el proceso de la evolución. Esta técnica sería después conocida mundialmente como el **algoritmo genético**.



## Historia de la Computación Evolutiva

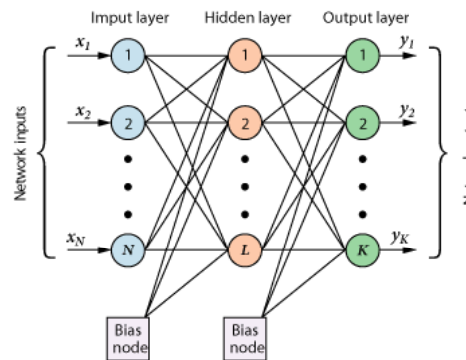
Michael Conrad y H. H. Pattee se cuentan entre los primeros en simular un ecosistema artificial jerárquico en el que un conjunto de organismos unicelulares estaban sujetos a una estricta ley de conservación de la materia que les inducía a competir por sobrevivir. Conrad falleció en 2000.





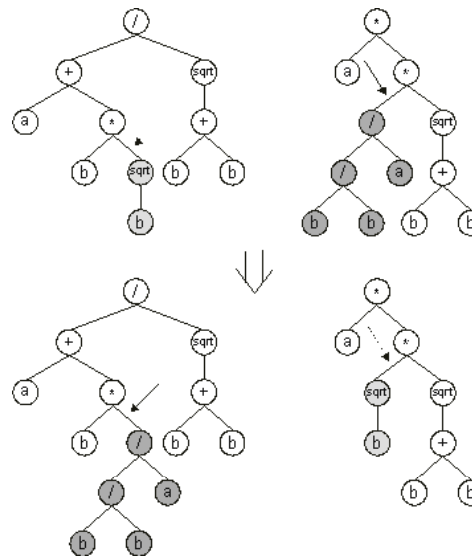
## Historia de la Computación Evolutiva

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).



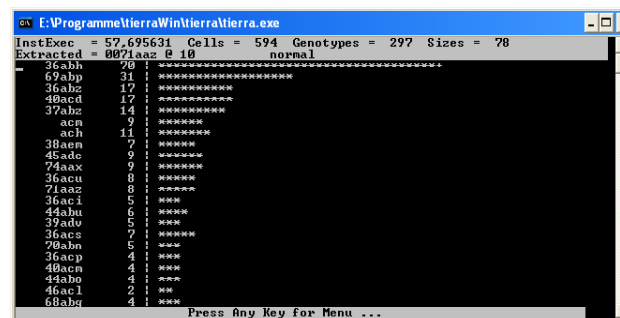
## Historia de la Computación Evolutiva

El biólogo Thomas S. Ray desarrolló a principios de los 1990s un simulador muy original en el que evolucionaban 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

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.



```

E:\Programme\TierraWin\Tierra\Tierra.exe
InstExec = 57,695631 Cells = 594 Genotypes = 297 Sizes = 78
Extracted = 8071aaz @ 10 normal
-----
36abh 70 : *****
69abp 31 : *****
36abz 12 : *****
40acd 17 : *****
37abz 14 : *****
acp 9 : *****
ach 11 : *****
38aen 7 : *****
45adc 9 : *****
74aax 9 : *****
36acu 8 : *****
71aaz 8 : *****
36aci 5 : *****
44abu 6 : *****
39adv 5 : *****
36acs 7 : *****
70abn 5 : *****
36acp 4 : *****
40acn 4 : *****
44abo 4 : *****
46acl 2 : *****
68abq 4 : *****
Press Any Key For Menu ...

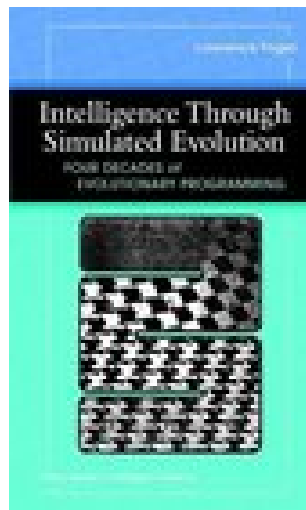
```

Tres son los paradigmas principales que conforman la computación evolutiva:

- Programación evolutiva
- Estrategias Evolutivas
- Algoritmos Genéticos



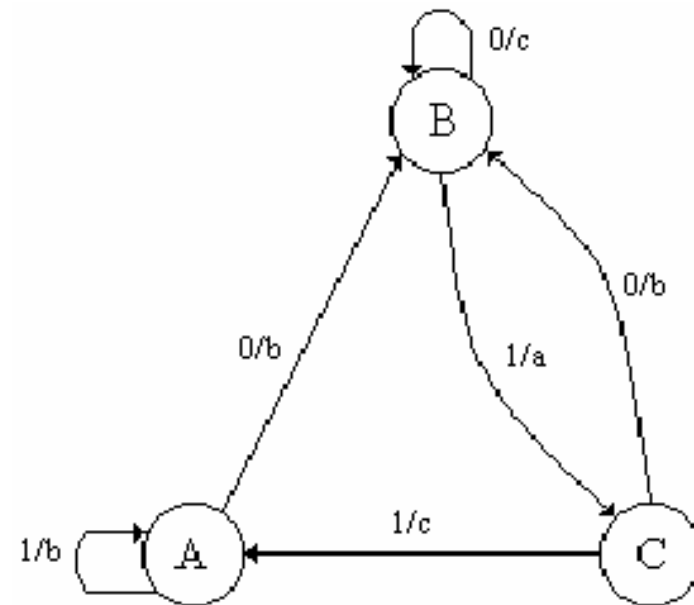
Lawrence J. Fogel propuso en 1960 una técnica denominada “programación evolutiva”, en la cual la inteligencia se ve como un comportamiento adaptativo.



## 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.*



## Ejemplo de Programación Evolutiva

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

## Ejemplo de Programación Evolutiva

Para este ejemplo, existen 5 tipos de mutadores: cambiar un símbolo de salida, cambiar una transición, agregar un estado, borrar un estado y 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).



- El algoritmo básico de la programación evolutiva es el siguiente:
  - Generar aleatoriamente una población inicial.
  - Se aplica mutación.
  - Se calcula la aptitud de cada hijo y se usa un proceso de selección mediante torneo (normalmente estocástico) para determinar cuáles serán las soluciones que se retendrán.

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). Asimismo, usa selección probabilística.



- Algunas aplicaciones de la programación evolutiva son:
  - 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.



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**:

$$f(\vec{x}_1, \vec{x}_2) = 100(\vec{x}_1^2 - \vec{x}_2)^2 + (1 - \vec{x}_1)^2$$

$$\text{donde } -2.048 \leq \vec{x}_1, \vec{x}_2 \leq 2.048$$

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:

$$\begin{aligned}\vec{x}_1^{t+1} &= \vec{x}_1^t + N(0,1.0) = -1.0 + 0.61 = -0.39 \\ \vec{x}_2^{t+1} &= \vec{x}_2^t + N(0,1.0) = 1 + 0.57 = 1.57\end{aligned}$$

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

Ahora, comparamos al padre con el hijo:

Padre:

$$f(\vec{x}^t) = f(-1.0, 1.0) = 4.0$$

Hijo:

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

Dado que:

$$201.416 > 4.0$$

el hijo reemplazará al padre en la siguiente generación.

- 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 de las Estrategias Evolutivas

- 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.

## Convergencia de las Estrategias Evolutivas

Esta regla se conoce como la “regla del éxito  $1/5$ ”, y en palabras dice:

*“ La razón entre mutaciones exitosas y el total de mutaciones debe ser  $1/5$ . Si es más, entonces debe incrementarse la desviación estándar. Si es menos, entonces debe decrementarse”.*

## Convergencia de las Estrategias Evolutivas

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”.

## 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  $n$ .

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.

## 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.



## Estrategias Evolutivas vs Programación Evolutiva

- 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.

- Algunas aplicaciones de las estrategias evolutivas son:
  - Problemas de ruteo y redes
  - Bioquímica
  - Optica
  - Diseño en ingeniería
  - Magnetismo

- 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 cruza sexual (operador principal) sobre el de la mutación (operador secundario), y usa selección probabilística.

- El algoritmo básico es el siguiente:
  - 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:

$$\begin{array}{cccc} \underbrace{0110} & \underbrace{1101} & \underbrace{0011} & \underbrace{1001} \\ \textit{Cadena1} & \textit{Cadena2} & \textit{Cadena3} & \textit{Cadena4} \end{array}$$

- 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”.

- 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
- 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.)