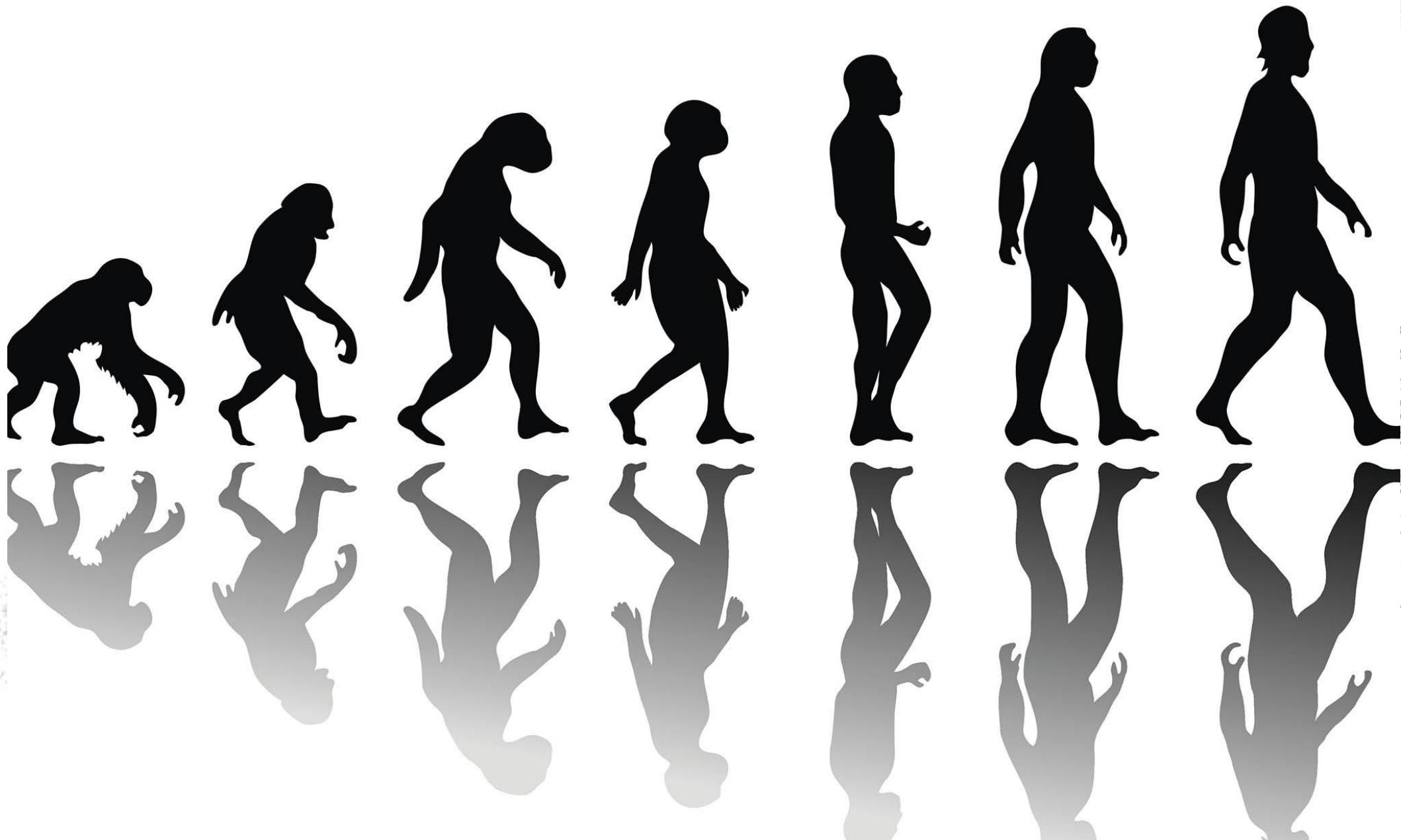


# Algoritmos de optimización bioinspirados (Parte 2)

Inteligencia Artificial  
Segundo semestre 2019

Profesor: Jorge Maturana  
[jorge.maturana@inf.uach.cl](mailto:jorge.maturana@inf.uach.cl)

# La evolución de las Especies



# La evolución de las Especies

- Charles Darwin (1809-1882)
- “El origen de las especies” (1859)
- HMS Beagle, 1831-1836

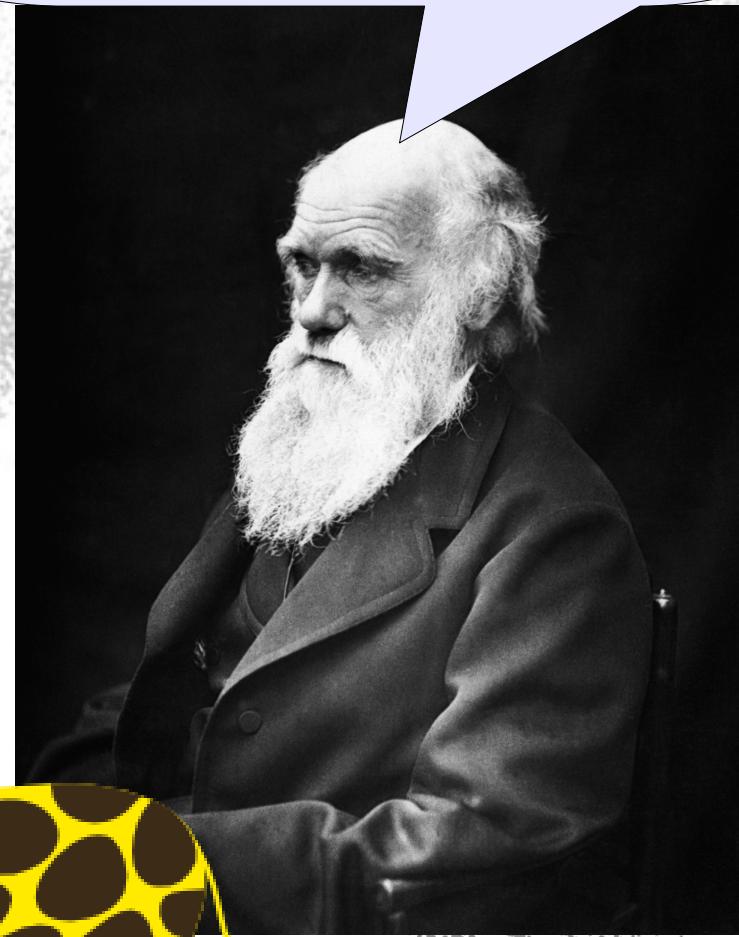
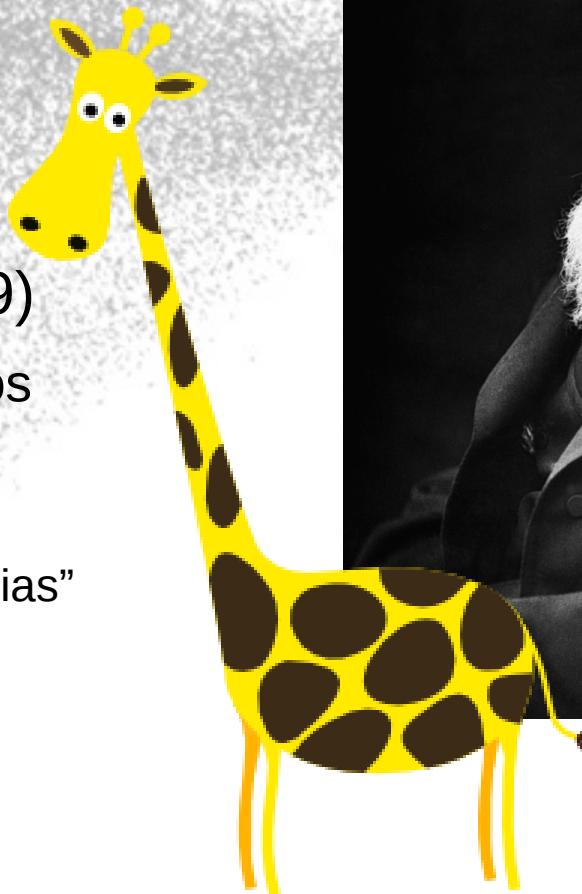
***“la supervivencia del más apto”***

- Idea de evolución: Jean-Baptiste Lamarck (1744-1829)
  - Teoría alquímica basada en dos fuerzas:
    - “El poder de la vida”
    - “La influencia de las circunstancias”



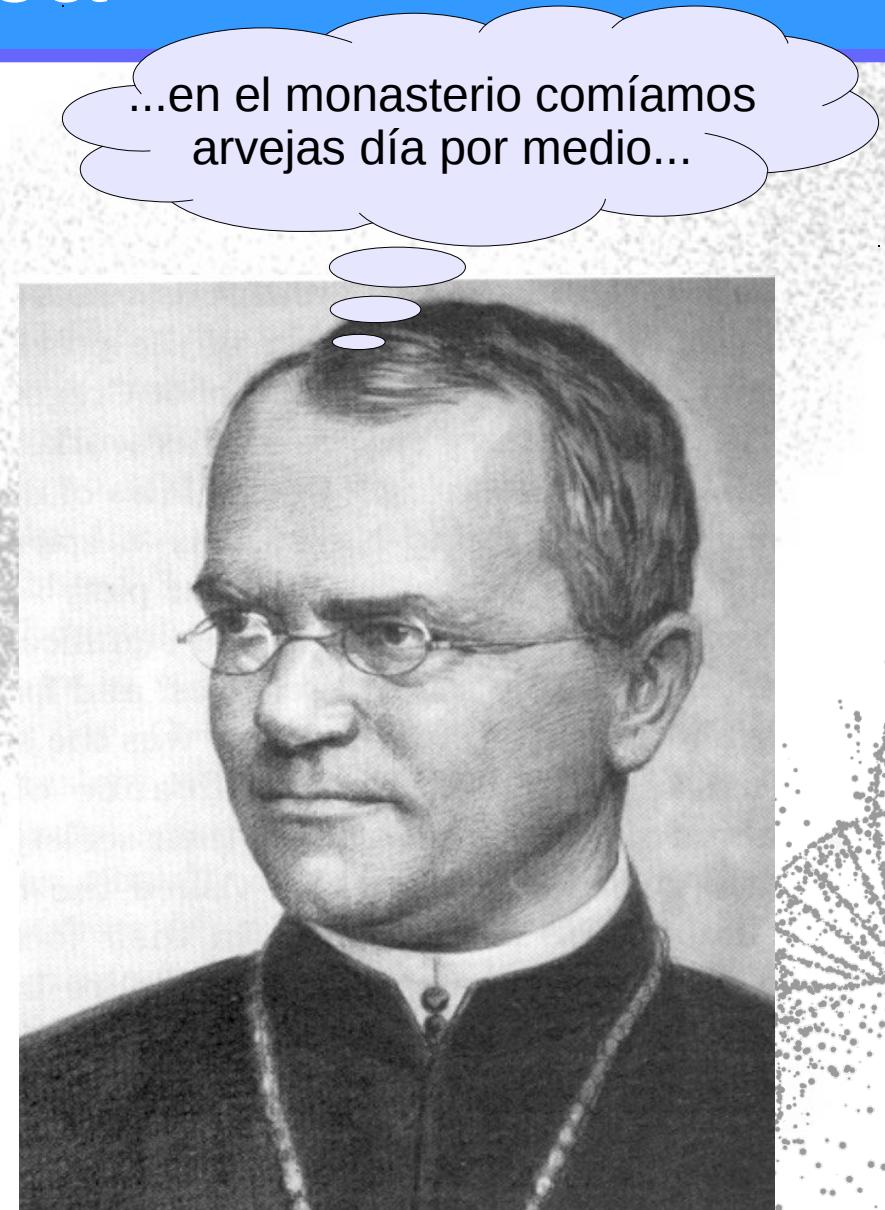
philipmartin.info

**Charly** Los más aptos sobreviven,  
el resto... cooperó!, jkjajkaja xd lol  
159 years ago via book [Like](#)



# Genética

- Gregor Mendel (1822-1884)
- Estudio de variación en plantas
- Modelo genético combinatorio
  - genes dominantes
  - genes recesivos
- El genotipo define las características
- Redescubierto por Hugo de Vries y Carl Correns, en 1900  
→ Evolución “Darwiniana”



# Modelo Evolutivo (simplificado)

- Genes determinan las características de los individuos
- Características determinan las opciones de los individuos
  - Lucha por los recursos
  - Lucha por las parejas
- “Supervivencia del más apto”

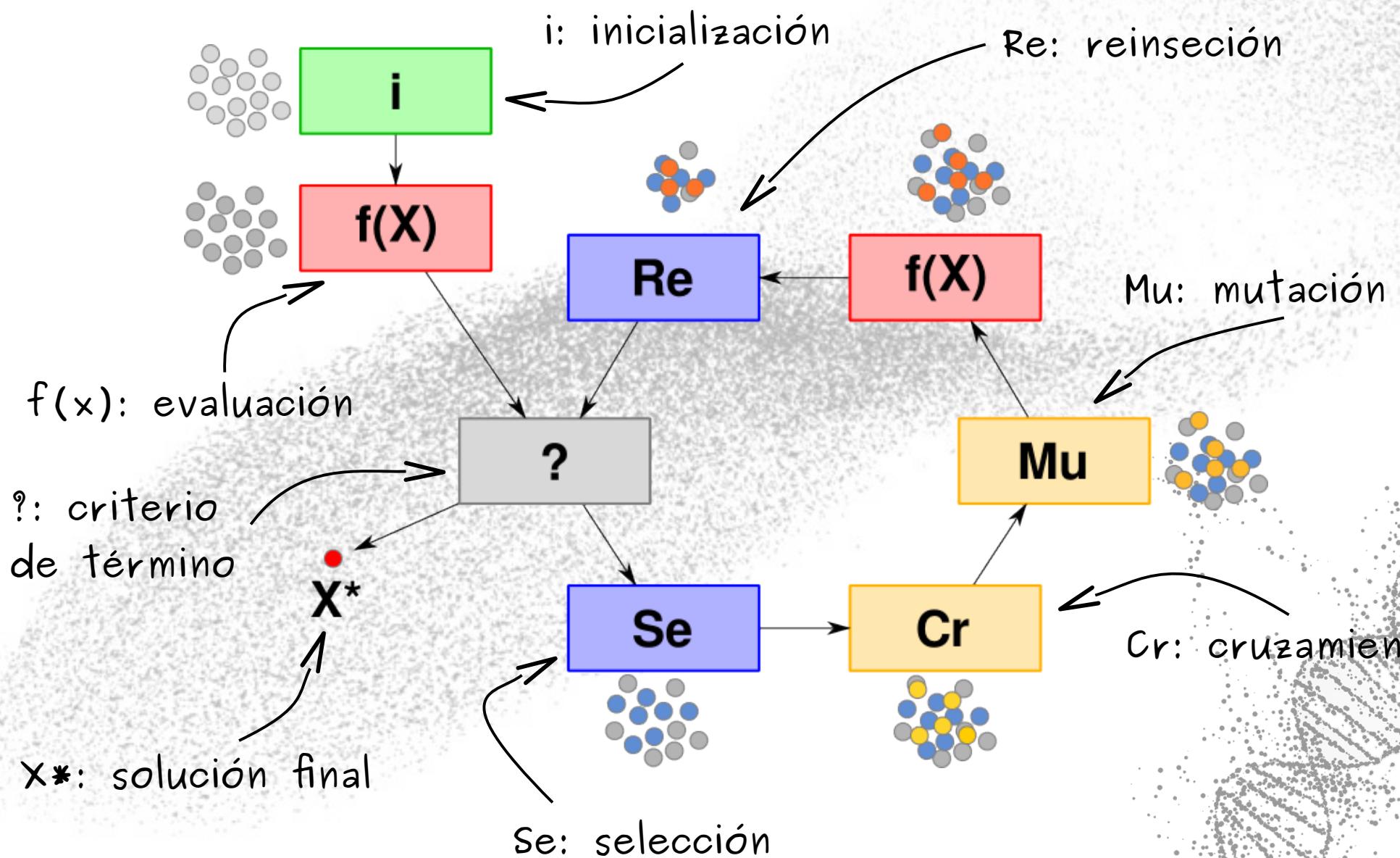
**Evolución: mejora continua = optimización**

# Evolución natural y artificial

- Paralelo entre naturaleza y problemas de optimización:

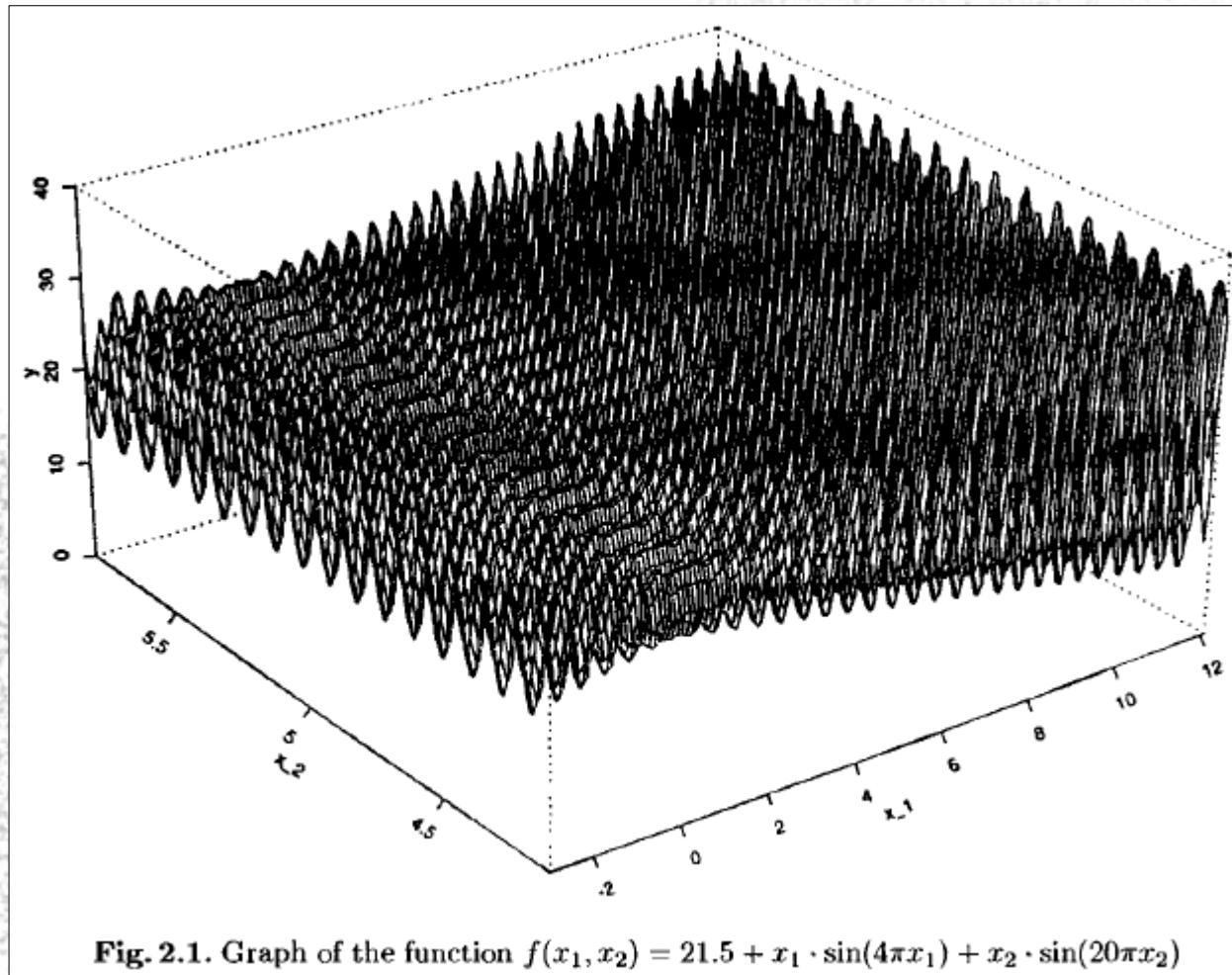
Natural	Artificial
Población de individuos	Conjunto de soluciones
ADN del Individuo	Conjunto de variables a determinar
Aptitud	Función objetivo
Mutación	Perturbación aleatoria
Cruzamiento	Combinación de variables

# Algoritmo Evolutivo



# Ejemplo

- Optimización de una función en 2D



# Representación (encoding)

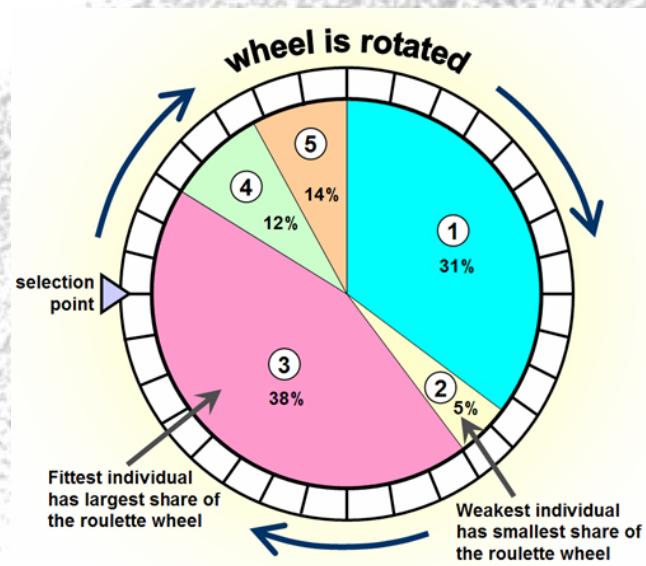
- Usando Algoritmos Genéticos (AG), propuestos por John Holland (1975)
- Rango de variables:

$$3.0 \leq x_1 \leq 12.1, 4.1 \leq x_2 \leq 5.8$$

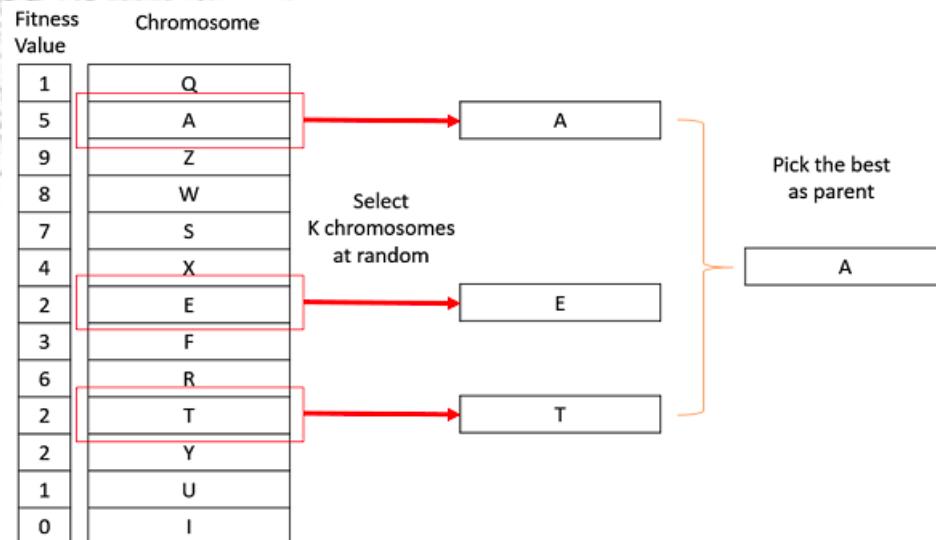
- Representación clásica de los Algoritmos Genéticos: strings binarios
  - Cada variable será representada por un string de bits
  - 000...0 corresponde al valor mínimo, 111...1 al valor máximo
- Los Algoritmos Evolutivos (Clase más general) admiten **otras representaciones**:
  - Reales
  - Permutaciones
  - Árboles
  - Cualquier otra estructura de datos, variable o no
- Se determina un tamaño de población (e.g., 20) y se genera una población de individuos inicializados al azar

# Evaluación y selección

- Cada individuo es decodificado y evaluado: función de **fitness**
- Se crea una nueva población de 20 individuos seleccionando con preferencia a los mejores
  - Idea: sólo los mejores tienen derecho a procrear
  - Método base: Roulette wheel
  - N.B. Algunos pueden quedar seleccionados más de una vez

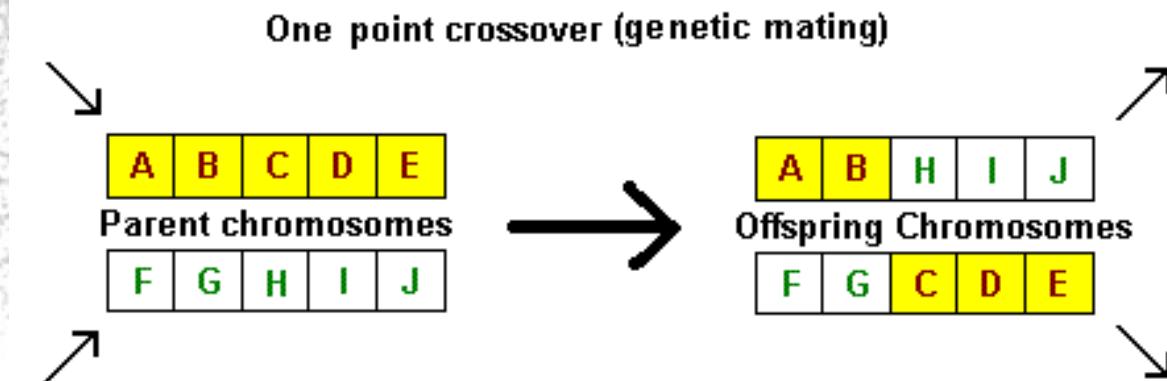


Mejor opción: **tournament selection**



# Cruzamiento

- Se elige con alguna probabilidad distintos individuos para mezclar sus cromosomas
  - Típicamente probabilidad entre 0.2 y 0.8
  - Una vez elegidos los dos **padres**, elegir al azar un punto de corte e intercambiar las dos mitades, para producir dos **hijos**
  - Este cruzamiento se conoce como “de un punto”
    - Existen otros
  - Los hijos reemplazan a los padres en la población (la secundaria)



# Mutación

- Una vez cruzados, comienza la mutación
  - Cambio con probabilidad baja ( $\sim 0.01$ )
  - Para la codificación de string de bits, se acostumbra a decidir bit a bit
  - Si un número al azar es inferior a la probabilidad de mutación, se hace un swap del bit ( $0 \rightarrow 1$  ó  $1 \rightarrow 0$ )

**Mutation**

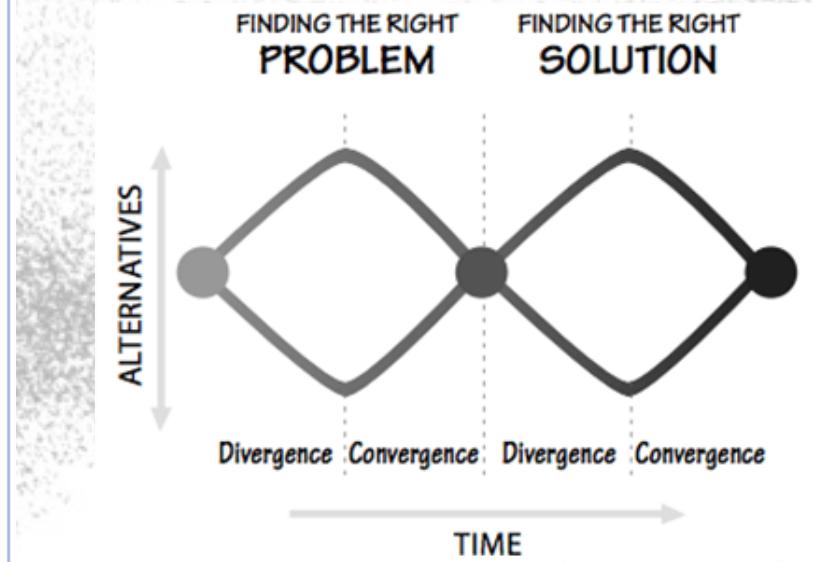
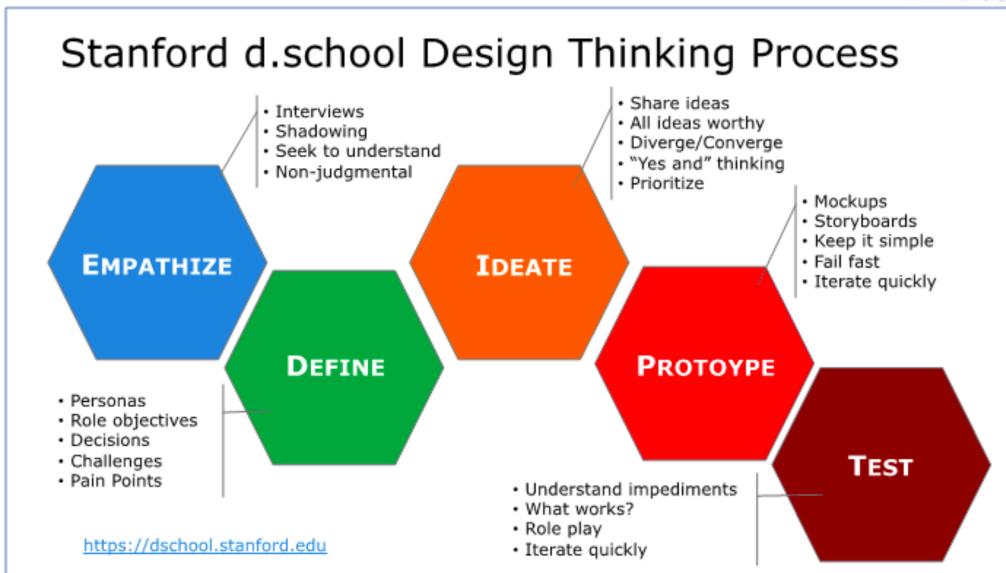
01101010110  $\Rightarrow$  01101000110

# Reinserción

- En general existen dos estrategias de reinserción:
  - **Generacional**
    - Los individuos nuevos “pisan” a los antiguos
    - La población se renueva constantemente
    - Exploración relativamente alta
  - **Steady State**
    - Los individuos deben competir con sus padres
    - A medida que avanza la búsqueda, aumenta la explotación
    - Eventual atasco en la búsqueda

# Exploración vs Explotación

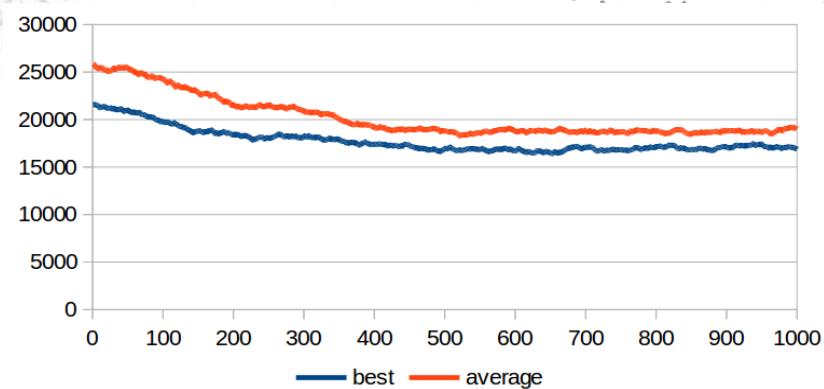
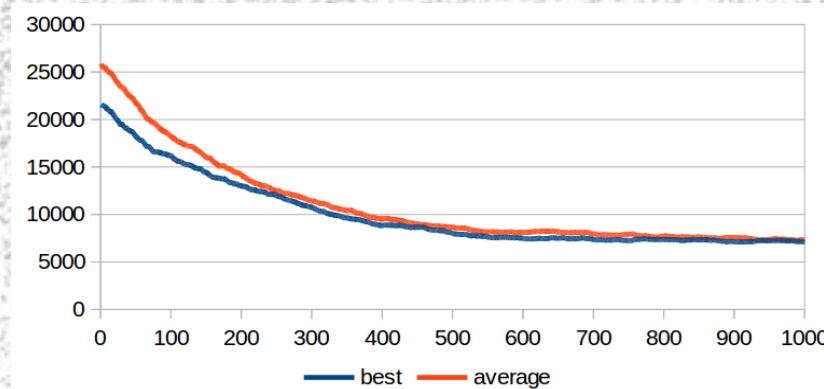
- Dos formas de pensar:



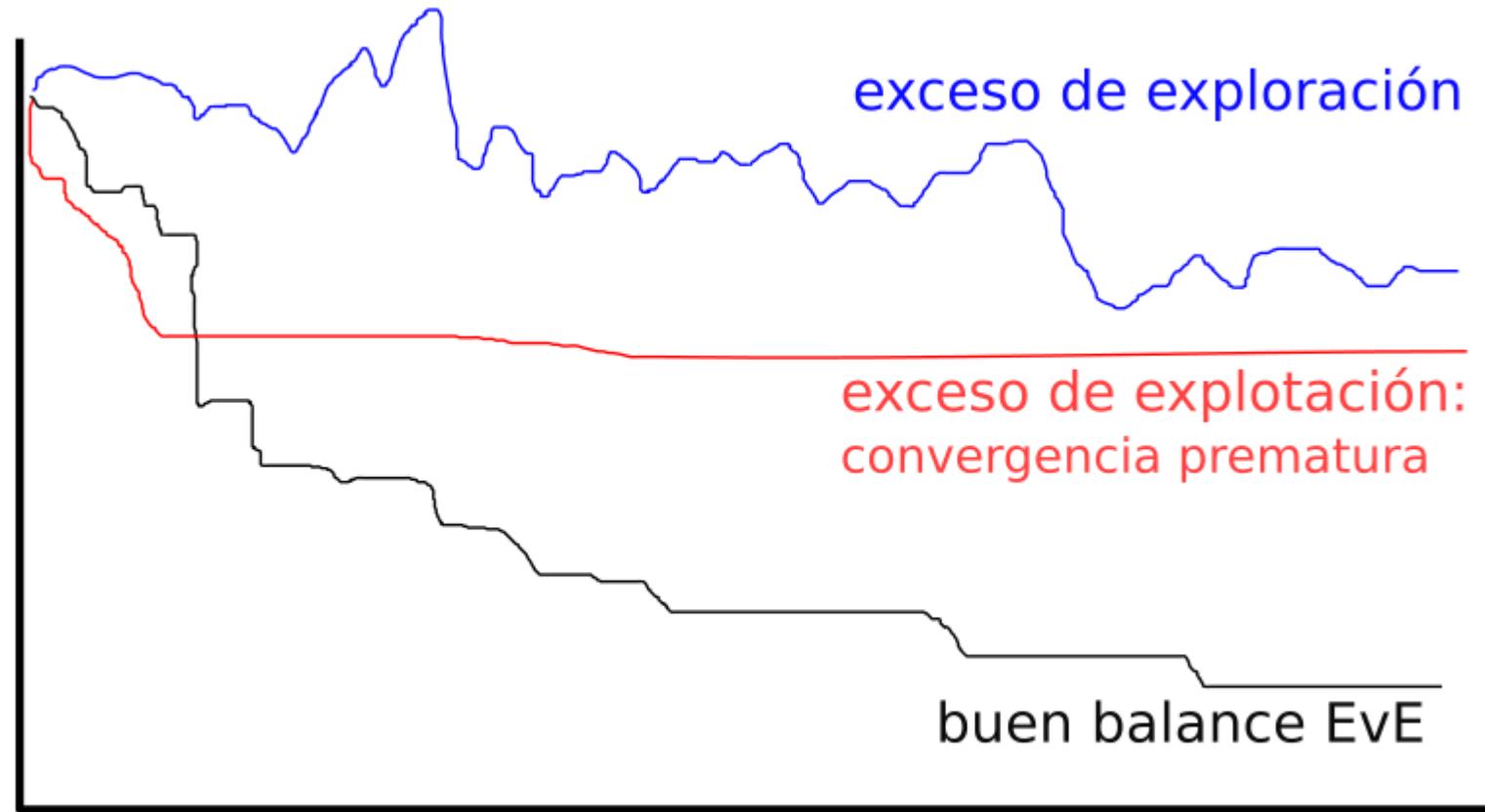
- Dos formas de enfrentar la búsqueda
  - Buscar ampliamente? → Explorar / diversificar
  - Refinar búsqueda? → Explotar / intensificar

# Lidiando con el azar

- La aleatoriedad dificulta el debug y la evaluación del algoritmo
- Tips:
  - Fijar o registrar semilla aleatoria para reproducir errores
  - Evitar usar una misma semilla siempre: caso puntual
  - Usar un gráfico de convergencia:



# Gráficas típicas de convergencia



# Resumen

- Ingeniería necesita optimizar problemas complejos
- Métodos clásicos no siempre funcionan
- Maldición de la dimensionalidad
- Metaheurísticas son buenas alternativas
- PSO imita el comportamiento social de aves y peces
  - Individuos se desplazan considerando información individual y compartida
- Algoritmos genéticos imitan teoría de la evolución
  - Individuos que se combinan, mutan y compiten
- Compromiso Exploración / Explotación
- Lidiando con el azar
  - Manejar semillas aleatorias
  - Gráfico de convergencia permite apreciarlo