Reconocimiento de patrones

Clase 13: un poco de algoritmos evolutivos

Para el día de hoy…

- Algoritmos evolutivos

Agrupamiento como optimización no lineal

- El problema de agrupamiento se puede ver como realizar una partición de N observaciones $x_1, x_2, \dots, x_d$, donde x_i es un vector de características de dimensión d, en k conjuntos $(k < d)$ $C = \{C_1, C_2, \dots, C_k\}$ tal que la distancia de cada observación a su partición más cercada es minimizada.

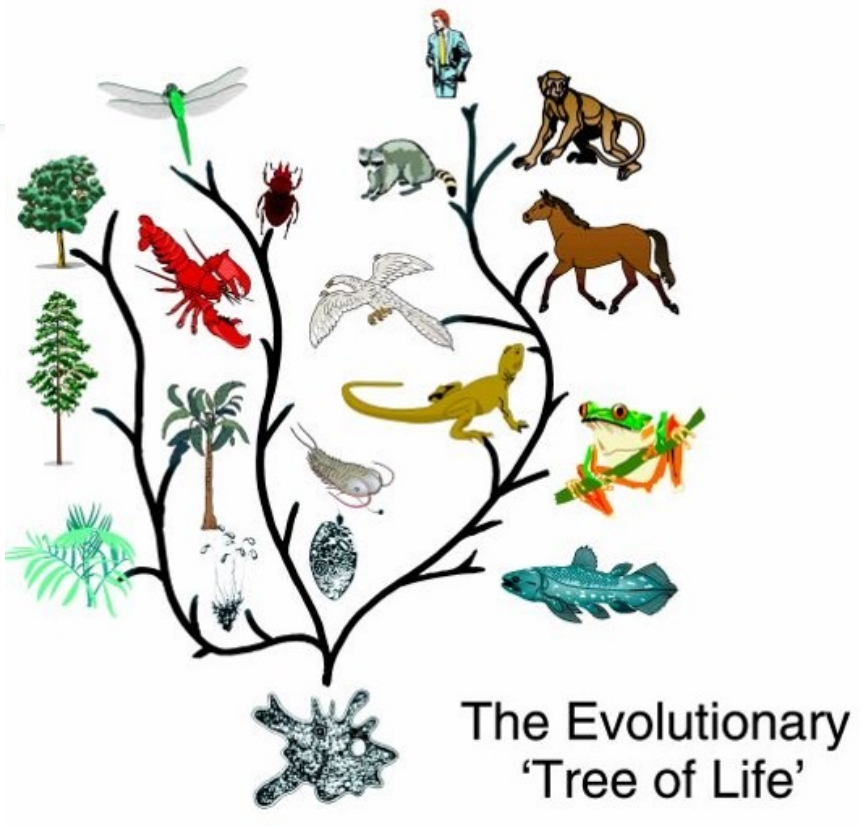
$$\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left|\left|x_i - \mu_i\right|\right|^2$$

Donde μ_i es la media de las observaciones en el conjunto i.

Algoritmos evolutivos

- La evolución nos ha traído hasta aquí
- Se trata de un proceso natural bajo el cual aquellos individuos más aptos tienen mayor oportunidad de pasar sus genes a otras generaciones
- Supervivencia del más apto

- ¿Funcionará para resolver nuestros problemas?
- La idea: diseñar agentes que simulen el proceso evolutivo para resolver problemas



Elementos para simular el proceso evolutivo

- Codificar las estructuras que se replicarán
- Operaciones que afecten a los individuos
- Una función de aptitud
- Un mecanismo de selección

Tipos de algoritmos evolutivos

- Estrategias evolutivas
- Programación evolutiva
- Algoritmos genéticos

Programación evolutiva

Lawrence J. Fogel la propone en los 60s

Enfatiza los nexos de comportamiento entre padres e hijos

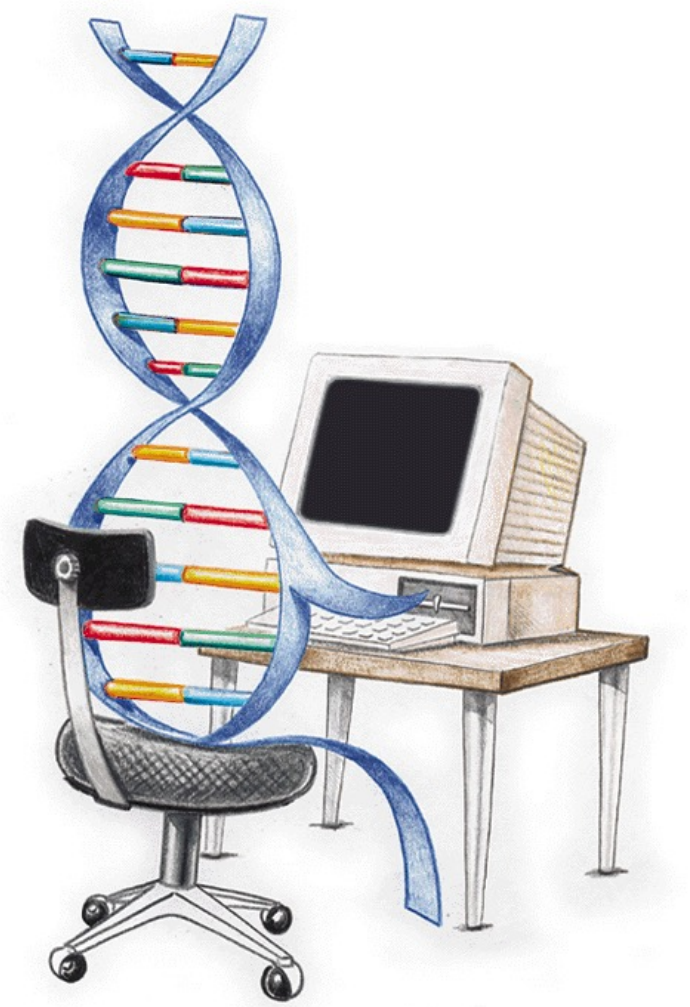
Abstracción a nivel de especies

Estrategias evolutivas

- Ingo Rechenber, lidera el grupo que propone el algoritmo en 1964 para problemas hidrodinámicos
- La versión original (1+1)-EE solo usaba un padre y generaba un hijo
- El hijo se mantenía si era mejor que el padre
- El nuevo individuo era generado con

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

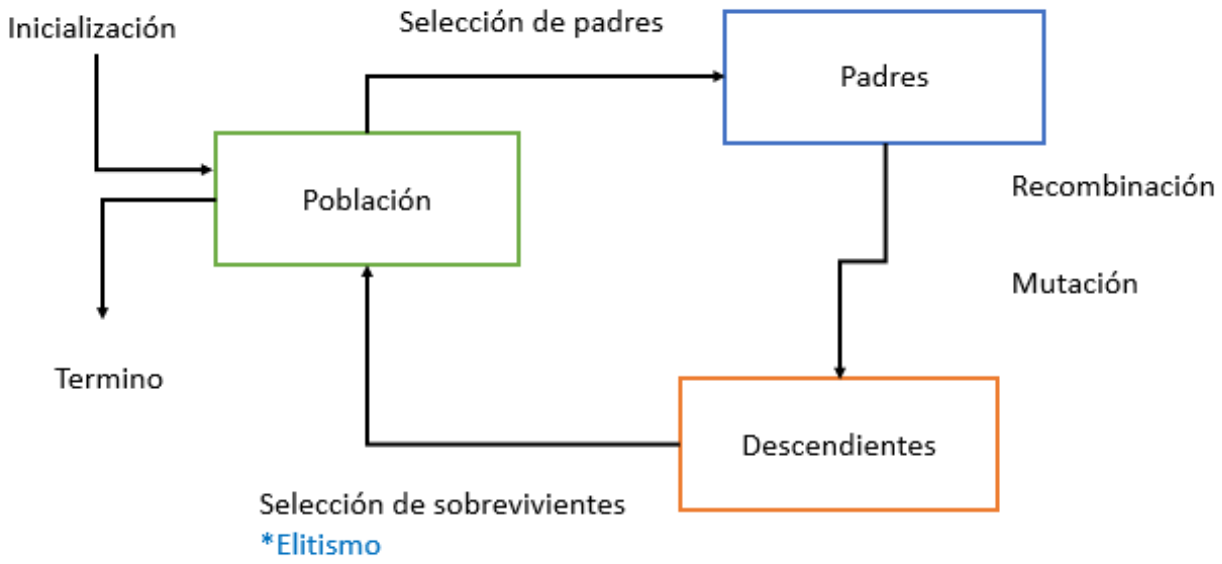
- Rechenberg propone $(\mu + 1)$-EE
- Schwefer propone el esquema $(\mu + \lambda)$-EE y (μ, λ)-EE
- CMA-ES es uno de los métodos más utilizados para optimización



Algoritmos genéticos

- Desarrollados por John H. Holland a principios de los 1960s para problemas de aprendizaje máquina
- Son metaheurísticas bio-inspiradas que tratan de emular la selección natural
- La codificación utilizada en los Algoritmos Genéticos es una abstracción del genotipo
- Son algoritmos muy útiles para resolver problemas de optimización no líneal
- Requieren poca información del dominio
- Son fáciles de usar y de implementar
- NO dan garantía de optimalidad y tampoco pueden terminar a que distancia se encuentran del optimo
- Requieren un gran número de evaluaciones de función

El 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

Selección

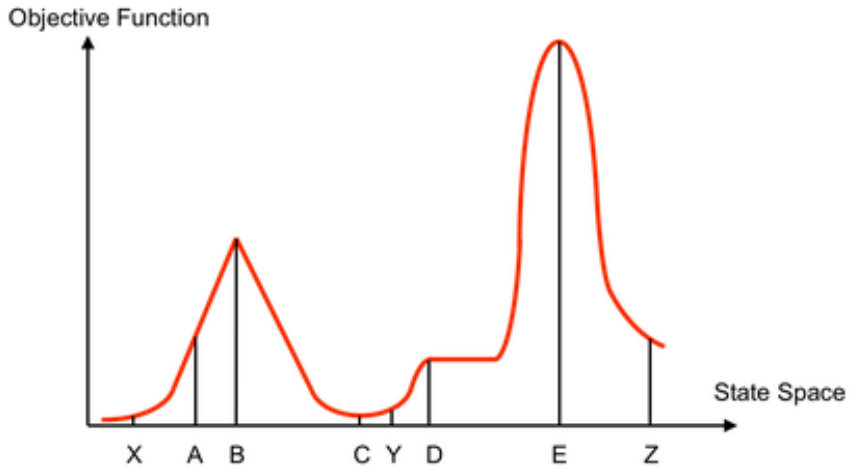
Proporcional

Ruleta

Torneo

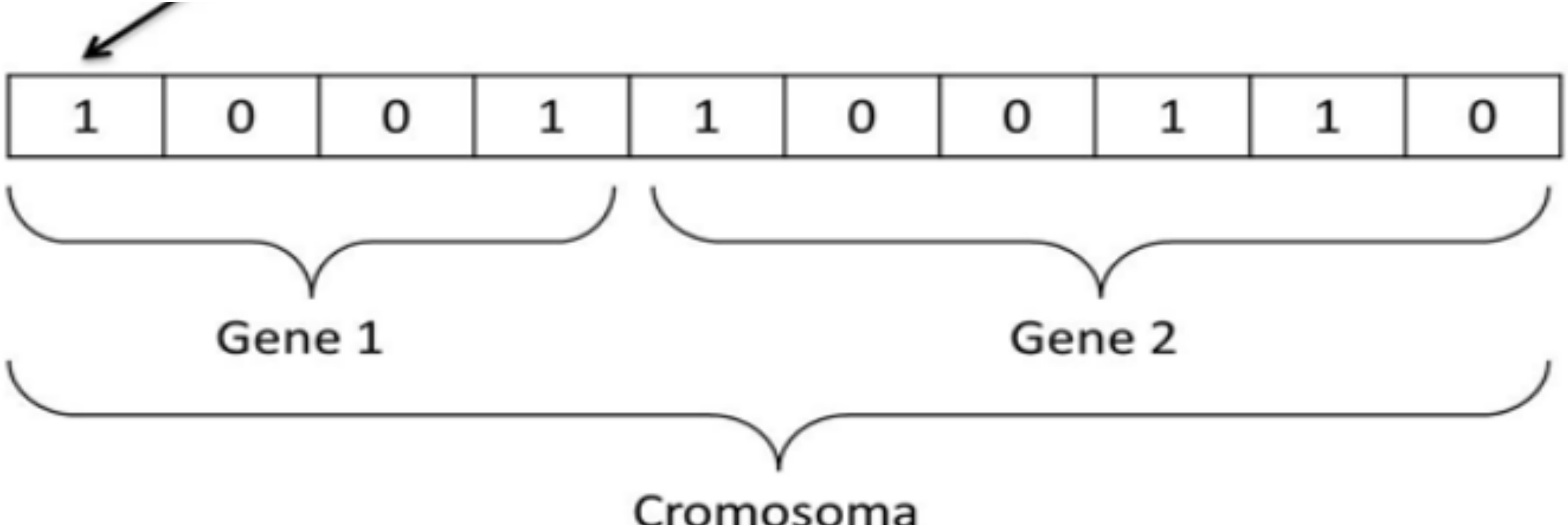
Selección por torneo

- Dado el tamaño del torneo s
- $padres = \emptyset$
- Hasta que se complete el número de padres μ
  - Elegir s individuos de forma aleatoria
  - Seleccionar al ganador i basado en su aptitud tal que $f(i) \leq f(j)\ \forall j \in \lambda$
  - $padres = padres \cup \{i\}$
- Regresar conjunto de padres

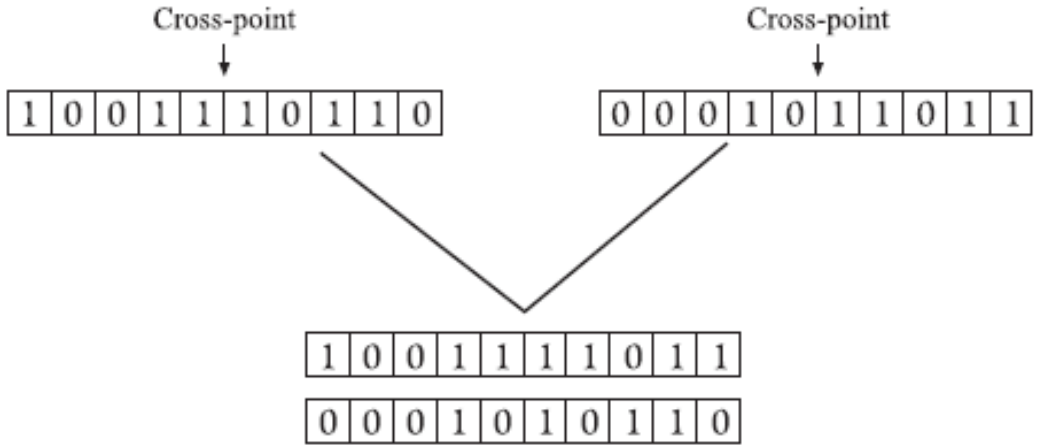


Representación

- La representación tradicional es la binaria
- A la cadena binaria se le llama “cromosoma”
- A cada cadena binaria que codifica una variable se le denomina “gene”
- Al valor dentro de esta posición se le llama “alelo”



Operadores: cruza de un punto



1. Se elige una posición en el cromosoma de forma aleatoria (cross-point)
2. El primer hijo tendrá los elementos hasta el cross-point del primer padre y del cross-point en adelante del segundo.
3. El segundo hijo tendrá los elementos hasta el cross-point del segundo padre y del cross-point en adelante del primero.

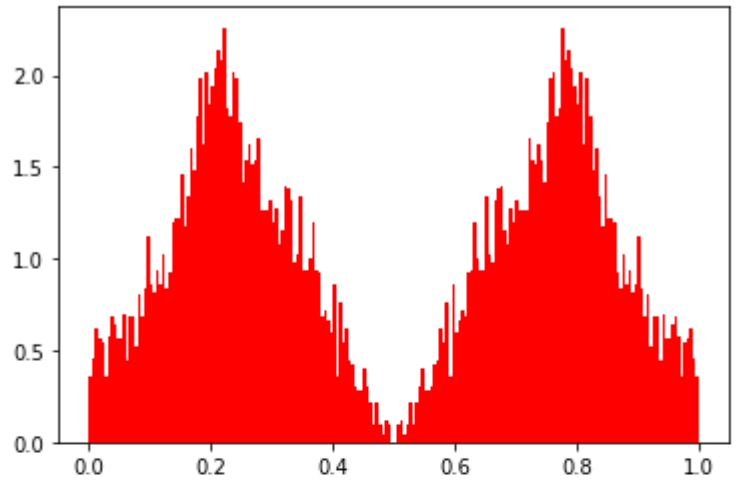
También existen cruzas de dos puntos, tres, hasta de n puntos

Operadores: mutación

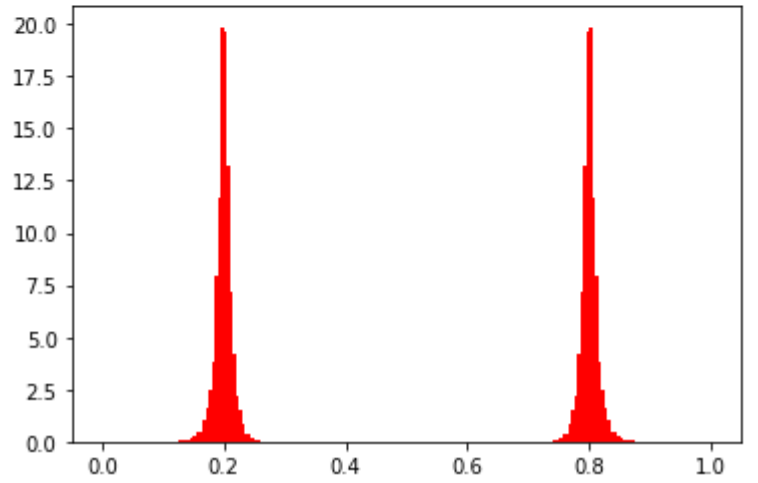
- Para cada alelo del cromosoma, generar un número aleatorio y si es menor a Pm cambiar el valor de 0 a 1 y de 1 a 0 según corresponda
- Normalmente el valor de Pm es muy bajo, se ha sugerido $P_m = \frac{1}{L}$ donde L es el tamaño del cromosoma

Simulated Binary Crossover (SBX)

- Propuesta por Deb y Agrawal en 1995
- Intenta emular el efecto de la cruza de un punto de
- Algoritmo
  - $u = X \sim U(0,1)$
  - $\beta = \begin{cases} (2u)^{\frac{1}{\eta_c+1}} & si\ u \leq 0.5 \\ \frac{1}{2(1-u)}^{\frac{1}{\eta_c+1}} & de\ lo\ contrario \end{cases}$
  - $h_1 = \frac{1}{2}(p_1 + p_2 - \beta|p_2 - p_1|)$
  - $h_2 = \frac{1}{2}(p_1 + p_2 + \beta|p_2 - p_1|)$




```
[3]: show(30)
```



Parameter-Based Mutation

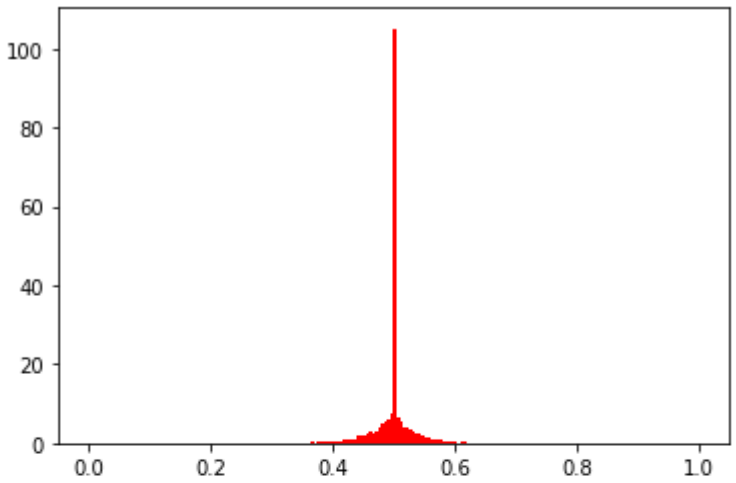
- Propuesto por Deb en 1995
- Algoritmo
  - Dado $p = \langle V_1, \dots, V_m \rangle$
  - Regresar $p' = \langle V_1, \dots, V_k' \dots, V_m \rangle$
  - Donde

$$V_k' = V_k + \delta_q (ub_k - lb_k)$$

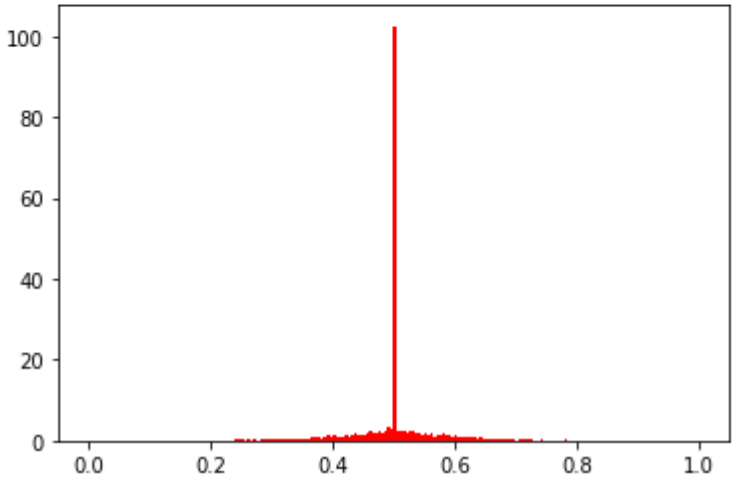
$$\delta_q = \begin{cases} [2u + (1-2u)(1-\delta)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} - 1 & si\ u \sim U(0,1) \leq 0.5 \\ 1 - [2(1-u) + 2(u-0.5)(1-\delta)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} & de\ lo\ contrario \end{cases}$$

$$\delta = \frac{\min [V_k - lb_k, ub - V_k]}{ub_k - lb_k}$$

$$\eta_m \in \mathbb{N}$$




```
[2]: show(10)
```



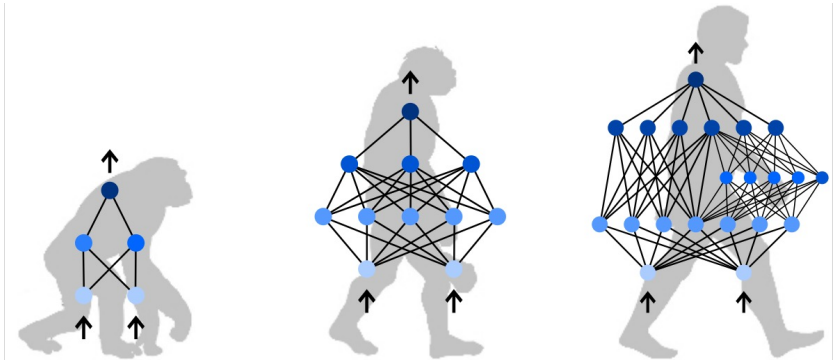
Comparativo de los algoritmos

	Estrategia evolutiva	Programación evolutiva	Algoritmo genético
Representación	Real	Real	Binaria
Función de aptitud	Valor de función objetivo	Valor de la función objetivo ajustada	Valor de la función objetivo ajustada
Auto-Adaptación	Desviación estándar	Ninguna	Ninguna
Mutación	Gausiana, operador principal	Gausiana	Inversión de bits (secundario)
Recombinación	Discreta e intermedia	Ninguna	Cruza de dos puntos, uniforma (principal)
Selección	Determinística, extintiva	Probabilística, extintiva	Probabilística
Restricciones	Desigualdad	Ninguna	Limites simples mediante codificación

Los requisitos para usar un algoritmo genético

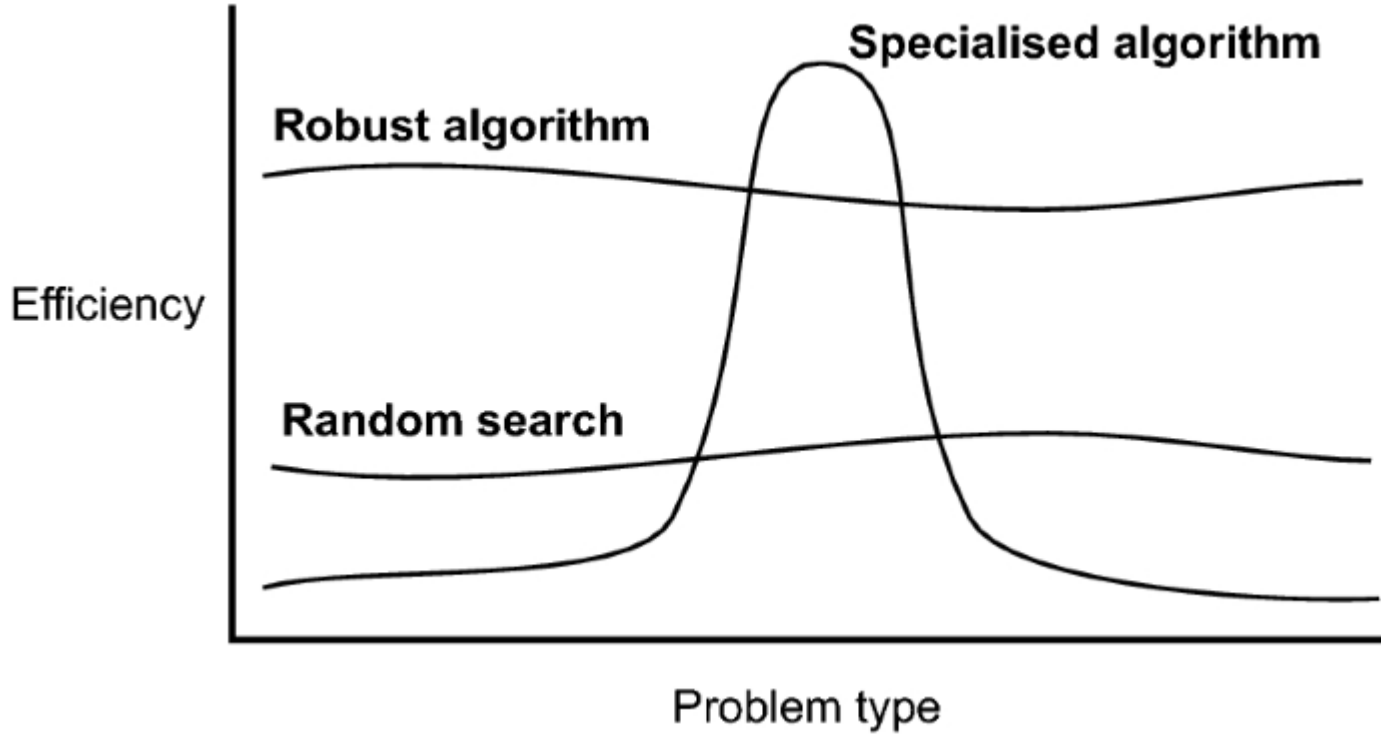
- Una representación de las soluciones potenciales del problema.
- Una forma de crear una población inicial de posibles soluciones (normalmente un proceso aleatorio).
- Una función de evaluación que juegue el papel del ambiente, clasificando las soluciones en términos de su "aptitud".
- Operadores genéticos 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.)

Algunas notas generales…



- Al tratar con problemas discretos o combinatorios se suele preferir búsqueda tabú y recocido simulado
- Para problemas continuos, los algoritmos evolutivos suelen dar buenos resultados así como para problemas mixtos.
- Además… los evolutivos se han utilizado para:
  - generar software (programación genética)
  - aprendizaje maquina (aprendizaje máquina evolutivo)
  - Redes neuronales (neuroevolución)

No free lunch theorem



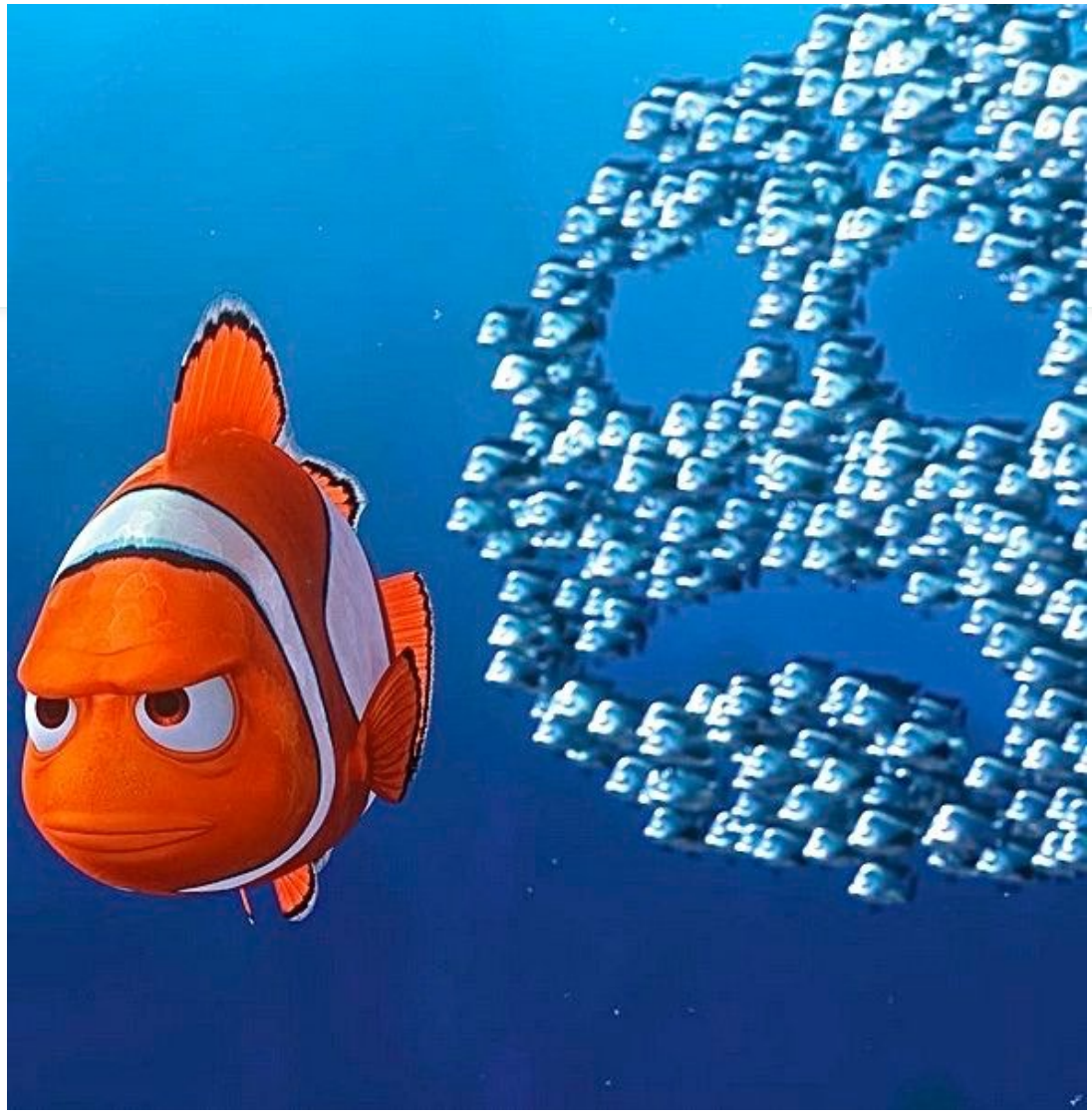
Otro comentario

In practice, genetic algorithms have had a widespread impact on optimization problems, such as circuit layout and job-shop scheduling. At present, it is not clear whether the appeal of genetic algorithms arises from their performance or from their æsthetically pleasing origins in the theory of evolution. Much work remains to be done to identify the conditions under which genetic algorithms perform well.

In practice, genetic algorithms have their place within the broad landscape of optimization methods (Marler and Arora, 2004), particularly for complex structured problems such as circuit layout or job-shop scheduling, and more recently for evolving the architecture of deep neural networks (Miikkulainen *et al.*, 2019). It is not clear how much of the appeal of genetic algorithms arises from their superiority on specific tasks, and how much from the appealing metaphor of evolution.

Otras metaheurísticas

- Bio inspirados
  - Cumulo de partículas
  - Colonia de hormigas
  - Colonia de abejas
  - Búsqueda Cuckoo
  - Algoritmos genéticos
- Programación matemática
  - Evolución diferencial
  - Algoritmos de estimación de distribución



Lo que sigue…

4. Clasificación de patrones por medio de funciones de similitud	
4.1	Revisión de probabilidad: densidad y momentos de una variable aleatoria. Densidad, correlación y covarianzas de vectores de variables aleatorias. Densidades condicionales
4.2	Clasificación de patrones como un problema de decisión estadístico
4.3	Clasificación tipo Bayes. Caso de Bayes aplicado a patrones normales.
4.4	Probabilidad de error

Para la otra vez…

- Introducción a reconocimiento de patrones por funciones de similitud