



# UAEM

Universidad Autónoma  
del Estado de México



## Algoritmos Genéticos

**Centro Universitario UAEM Valle de México**

**Unidad 1. Optimización: Introducción a la  
Optimización y los Algoritmos Evolutivos**



**Maestría en Ciencias de la Computación (MASCO)**

**Ph. D. Victor Manuel Landassuri Moreno**

**[vmlandassurim@uaemex.mx](mailto:vmlandassurim@uaemex.mx)**

**[landassuri@gmail.com](mailto:landassuri@gmail.com)**

# **Unidad 1. Optimización: Introducción a la optimización y los algoritmos evolutivos**

# Unidad 1

1.1 Introducción

1.2 Teoría de la Evolución

1.3 Búsqueda y optimización

1.4 Algoritmos Evolutivos

1.5 Aplicabilidad de los Algoritmos Evolutivos

# 1.1 Introducción

# Introducción

Los Algoritmos Evolutivos (**AEs**) son:

- Técnicas de resolución de problemas de búsqueda y optimización
- Inspirada en la teoría de la evolución

Tienen:

- Características de **búsqueda aleatoria** con características de **búsqueda rígida** para encontrar a los individuos mas adaptados

# Introducción

- Los AEs proporcionan un esquema general para la solución del problema
- Se pueden especificar ciertos componentes y puede ser aplicable a toda una clase de problemas
- La parte más delicada es el diseño de la función de adaptación
  - Usada en el proceso de selección del algoritmo

# Introducción

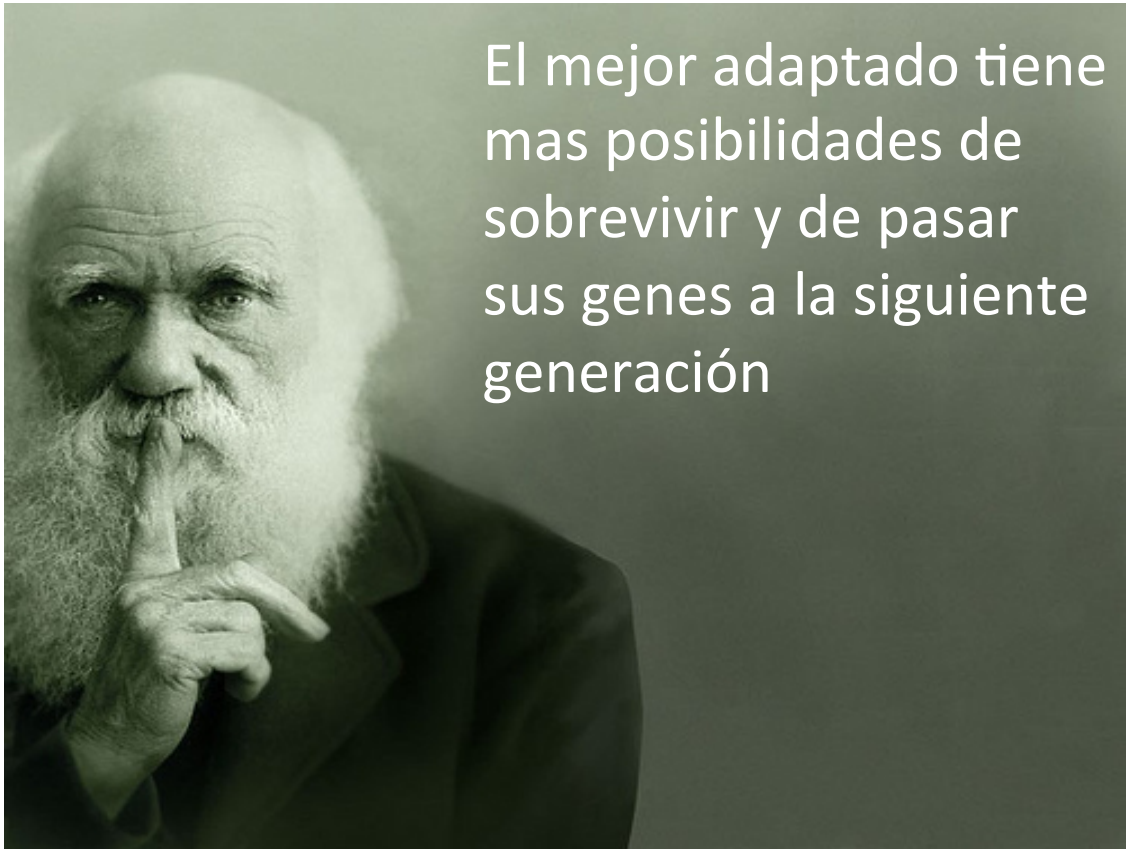
- Es fácil agregar conocimiento intrínseco del problema
  - Sí se conoce
  - Y puede ayudar a mejorar significativamente el rendimiento de él
- Se debe saber con exactitud que se desea:
  - Evolucionar un conjunto de posibles soluciones para entender los mecanismo evolutivos
    - e.g. entender como una especie llega a ser como es
  - Se desea resolver únicamente un problema o tarea
    - e.g. obtener una solución para aplicarla

# Introducción

- Los Algoritmos Evolutivos (AEs) son la rama principal de donde se desprenden los Algoritmos Genéticos (AGs)
- Todos estos no garantiza llegar al óptimo
- Pero si una aproximación, dependiendo de los recurso destinados a la búsqueda



## 1.2 Teoría de la Evolución



El mejor adaptado tiene  
mas posibilidades de  
sobrevivir y de pasar  
sus genes a la siguiente  
generación

## Qué sabemos de la Evolución?

Especies

Individuos

Reproducción

- Recombinación
- Mutaciones

Proceso de Selección

Elitismo

### Charles Darwin

Teoría de la Evolución

**Selección natural**

1. Nuevos individuos son generados
  - A pesar de que muchos mueran
2. Diferentes tasas de supervivencia y reproducción
3. Herencia

# Teoría de la Evolución

- Los AEs parten de la teoría de la evolución de Charles Darwin
  - La evolución de las especies se debe al principio de la selección natural
    - Favorecen la supervivencia y multiplicación de aquellas especies mejor adaptadas de su entorno
  - Otro elemento son las mutaciones
    - Pequeñas variaciones en los individuos
  - La reproducción produce nuevos individuos usando los genes de los padres

# Teoría de la Evolución

- La herencia de las características físicas de padres a hijos viene de la teoría del neodarwinismo
- Las características físicas de un individuo, su fenotipo, son consecuencias de su información genética, o genotipo
  - Cadenas de genes con relaciones complejas
  - Constituyen las unidades de transferencia de la herencia

# Teoría de la Evolución

- Los individuos están agrupados en poblaciones de ellos
- La adaptación de un individuo, es la tendencia relativa al resto de los individuos para sobrevivir y dejar descendencia

## 1.3 Búsqueda y optimización

# Búsqueda - optimización

- La mayoría de los problemas de la industria se pueden ver como un problema de búsqueda u optimización:
  - Dado un sistema
    - Busca un conjunto de valores que lo configuren
    - Valores que permiten optimizar su comportamiento
      - Rendimiento, calidad, costo, etc.

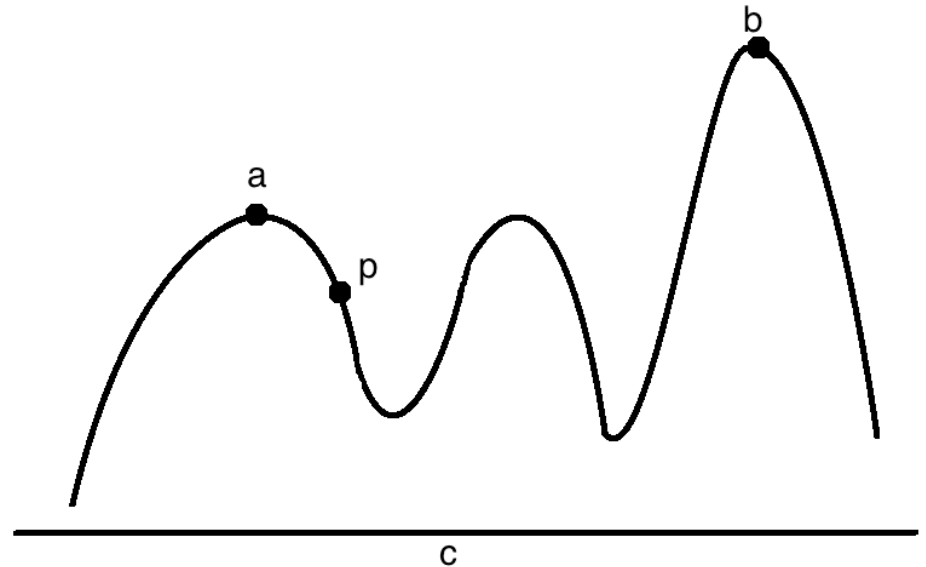
# Espacio de búsqueda

- Se define como:
  - La caracterización de cada una de las soluciones para un problema en particular.
  - i.e., tener en cuenta todas las posibles combinaciones del problema, dadas las variables del mismo.



# Técnica de escalada

- Esta técnica consiste en determinar la pendiente de la vecindad y seleccionar el punto de mayor pendiente
- Si el valor de la función a optimizar en el nuevo punto es mejor que el anterior, este se convierte en el actual



# Espacio de búsqueda

- Sin embargo los problemas del mundo real, suelen ser de los mas complejos, computacionalmente hablando
  - No existe un método que los resuelva en un número de pasos que sea una función polinómica del tamaño del problema
- Por lo que métodos deterministas como el de la **escalada** no son capaces de resolver problemas a partir de cierto tamaño

# Espacio de búsqueda

- Una alternativa es usar búsqueda aleatoria
  - Tomando al azar puntos dentro de la zona de búsqueda
  - Sin embargo es ineficiente
  - De ahí, que una buena practica al hacer un AE, es crear un método de búsqueda aleatoria, y comparar los resultados, siendo el esperado, que el AE funcione más eficientemente.
- Los AEs están en un punto intermedio de la búsqueda aleatoria y la búsqueda dirigida por la selección

# **1.4 Algoritmos Evolutivos / Cómputo Evolutivo**

# 1.4 Cómputo Evolutivo

## Contenido de la Sección 1.4

1.4.1 Cómputo Evolutivo

1.4.2 AE simple

1.4.3 Funcionamiento de un AE

1.4.4 Tipos de Algoritmos Evolutivos

1.4.4.1 Algoritmos Genéticos

1.4.4.2 Programación evolutivo

1.4.4.3 Estrategias evolutivas

1.4.4.4 Programación Genética

## 1.4.1 Cómputo Evolutivo

# ¿Porqué cómputo evolutivo?

¿Cómo son los algoritmos actuales?:

- Rígido e inflexible
- Frágil
- No generaliza ni aprende
- Es necesario dejar todo hecho
- Nunca crece
- No es autónomo
- Lento
- ...

Se inspira, no es copia

# Computación inspirada en la Naturaleza



# ¿Qué es el Cómputo Evolutivo?

- Es el estudio de sistemas de cómputo, el cual se inspira en la evolución natural
- Toma un principio básico: sobrevive el mas apto
- El cómputo evolutivo se emplea en aprendizaje, diseño y optimización
- No requiere un dominio completo de conocimiento de la tarea en cuestión
  - Sin embargo, este se puede incorporar en la solución de los problemas (previous domain knowlegde)

## 1.4.2 Un Algoritmo Evolutivo (AE) simple

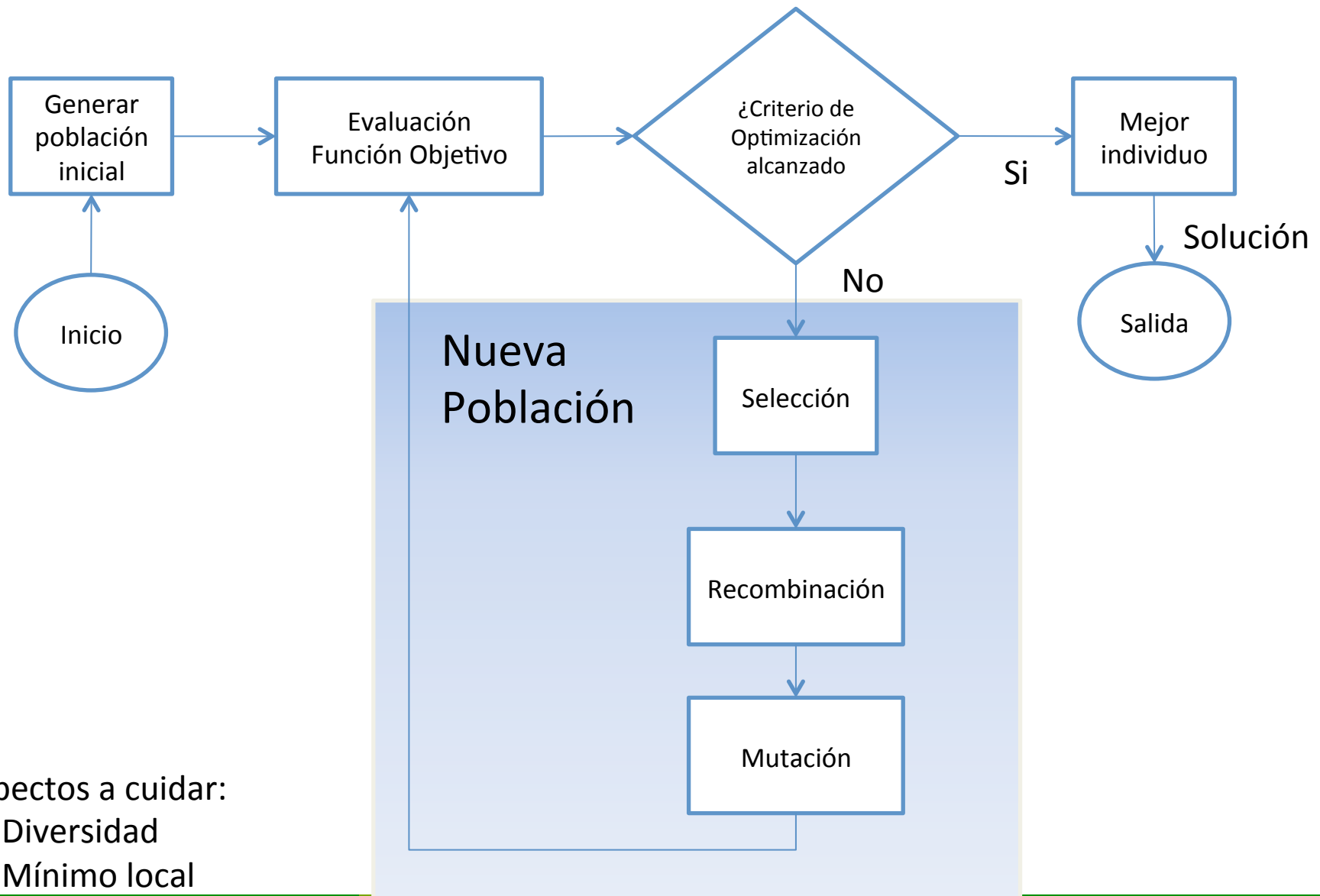
# Esquema general de un Algoritmo Evolutivo (AE)

- Procesan un conjunto simultaneo de soluciones al mismo tiempo (individuos en la población).
- La composición de la población se va modificando en cada iteración del algoritmo
  - Generaciones
  - Su generación dependerá de los operadores genéticos
- La selección es clave para que individuos altamente adaptados tengan más probabilidades de pasar sus genes, contrario a los menos adaptados
  - Escapar de óptimos locales

# AE simple

1. Generar la población inicial  $P(0)$  aleatoriamente y hacer  $i \leftarrow 0$ ;
2. Repetir
  - a. Evaluar la adaptabilidad de cada individuo  $p(i)$ ;
  - b. Seleccionar padres de  $P(i)$  basados en la adaptabilidad de  $P(i)$ ;
  - c. Generar descendencia de los padres seleccionados usando cruzamiento y mutación para formar  $P(i+1)$
  - d.  $i \leftarrow i + 1$ ;
3. Parar hasta que criterio de para sea satisfecho

# Diagrama de flujo



Aspectos a cuidar:

- Diversidad
- Mínimo local

# Basados en población generar-probar

- Generación: Mutar o recombinar individuos en la población
- Probar: Selecciona a la siguiente generación de los padres e hijos
- Mientras que hacer una analogía hacia el aspecto biológico puede parecer novedoso, no hay que olvidar las raíces de la vieja IA y la nueva IA (GOFAI)

# 1.4.3

## Funcionamiento de un AE simple

## Maximizar $f(x) = x^2$

- X pertenece a los enteros, en el intervalo  $[0, 31]$ 
  - Recordatorio de intervalos

$$(a, b) = ]a, b[ = \{x \in \mathbb{R} \mid a < x < b\},$$

$$[a, b) = [a, b[ = \{x \in \mathbb{R} \mid a \leq x < b\},$$

$$(a, b] = ]a, b] = \{x \in \mathbb{R} \mid a < x \leq b\},$$

$$[a, b] = [a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}.$$



## Maximizar $f(x) = x^2$

- X pertenece a los enteros, en el intervalo  $[0, 31]$ ,  
 $x = 0 \dots 31$
- Posteriormente es codificar: representación de los cromosomas
  - Representación binaria
  - 5 bit para representar hasta el 31
- Tamaño de la población es 4

# Maximizar $f(x) = x^2$

## 1. Generar población aleatoria

- 01101
- 11000
- 01000
- 10011

Estos son los cromosomas o genotipos

## Maximizar $f(x) = x^2$

2. Calcular el valor de adaptabilidad de cada individuo
  - a. Decodificar los individuos en enteros (fenotipos)
    - $01101 \rightarrow 13$
    - $11000 \rightarrow 24$
    - $01000 \rightarrow 8$
    - $10011 \rightarrow 19$

## Maximizar $f(x) = x^2$

b. Evaluar la adaptabilidad de acuerdo a  $f(x) = x^2$

- $13 \rightarrow 169$
- $24 \rightarrow 576$
- $8 \rightarrow 64$
- $19 \rightarrow 361$

## Maximizar $f(x) = x^2$

3. Seleccionar a dos individuos basados en su adaptabilidad, si se usa la selección de ruleta:

$$p_i = \frac{f_i}{\sum_j f_j}.$$

- $p_1(13) = 169/1170 = 0.14$
- $p_2(24) = 576/1170 = 0.49$
- $p_3(8) = 64/1170 = 0.06$
- $p_4(19) = 361/1170 = 0.31$

# Maximizar $f(x) = x^2$

Cruzamiento, punto aleatorio en 4

- 01101
- 11000

Descendencia 0110<sup>0</sup> y 1100<sup>1</sup>

Cruzamiento con punto aleatorio en 2

- 10011
- 11000

Descendencia <sup>1</sup>0000 y <sup>1</sup>1011

Ahora la población intermedia es: 01100, 11001, 10000 11011

# Maximizar $f(x) = x^2$

4. Aplicar mutación a la población intermedia con una pequeña probabilidad.
  - Un método simple es el cambio de bit
  - Aleatoriamente se escoge y se tiene la nueva población  $P(1)$

0110**1**, 11001, **0**0000 11011

5. ir al paso 2 si no hay paro

## **1.4.4 Tipos de Algoritmos Evolutivos**

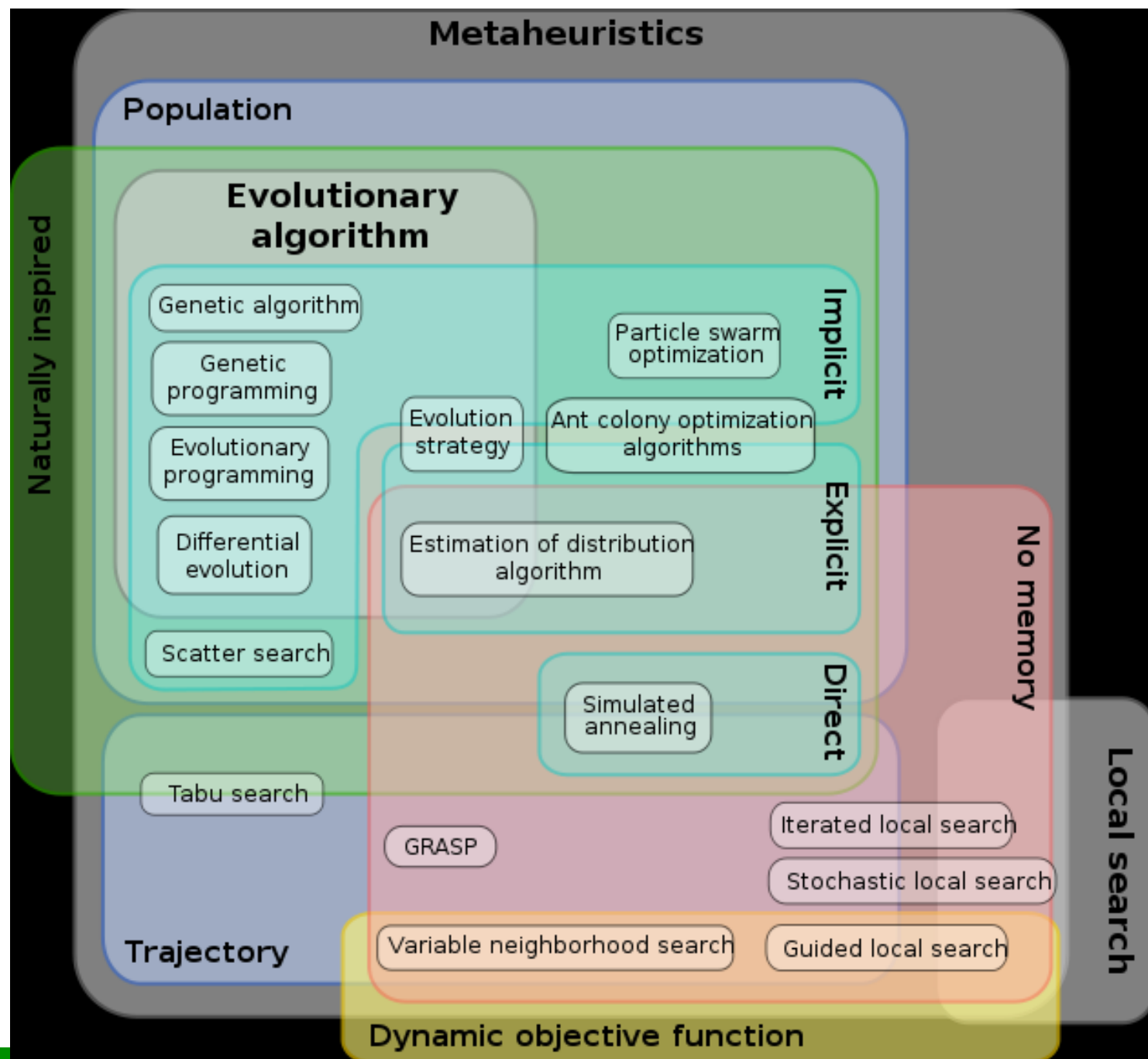


# Variantes de AEs

- Las variantes principalmente están distinguidas por la forma de representación de los individuos:
  - AGs utilizan una representación binaria o de números reales.
  - Programación Evolutiva: Evolucionan una población de programas para resolver un problema en general.
  - Entre otros que más adelante se explican.

## Diversos tipos:

- Hay muchos AEs con diferentes
  - Bases históricas
  - Representaciones
  - Variaciones de operadores
  - Técnicas de selección
- En si, los AEs se refieren a toda una familia de algoritmos:
  - Algoritmos Genéticos
  - Programación evolutivo
  - Estrategias evolutivas
  - Programación Genética



## 1.4.4.1 Algoritmos Genéticos

- Formulados por Holland y sus alumnos a entre los 1960s y 1970s
  - Búsqueda adaptativa
  - Individuos de cadenas binarias (cromosomas)
  - Simula evolución Darwiniana
  - Los operadores de búsqueda se aplican solo a los genotipos
  - Enfatizan el cruzamiento, la mutación es un operador secundario
  - Usualmente se usa el método de selección de ruleta

## 1.4.4.2 Programación Evolutiva

- Propuesto por Fogel, et. al. a mediados de los 1960s
  - Para simular inteligencia
- Maquinas de estado finito (FSMs)
  - Usadas para representar individuos
  - Así como, vectores de valores reales para optimización numérica
- Cercanos a la evolución Lamarckiana
- Operadores de búsqueda (solo mutaciones) son aplicados al **fenotipo**
- No usan recombinación (cruzamiento)
- Usualmente se usa selección por torneo.

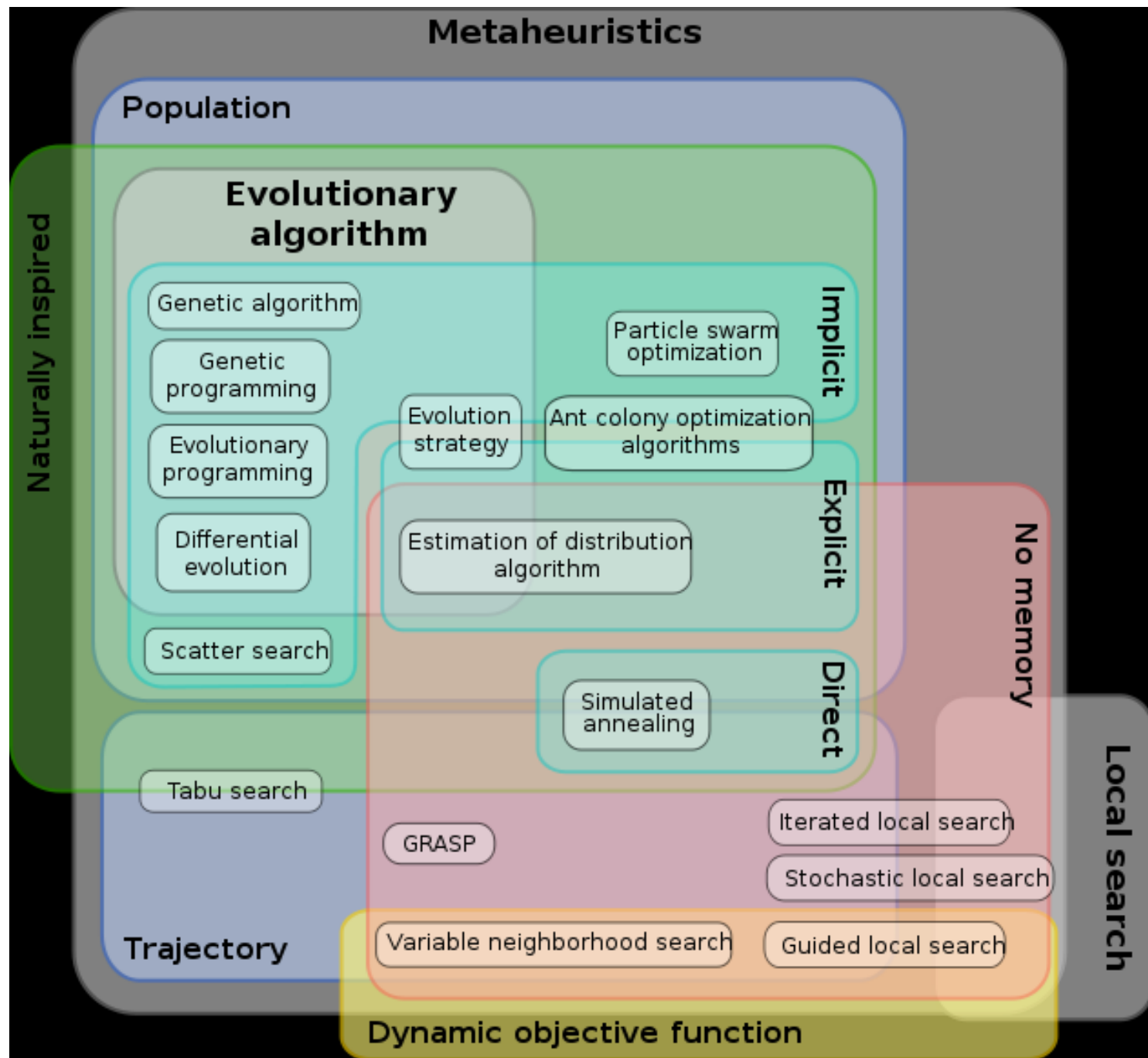
### 1.4.4.3 Estrategias evolutivas

- Propuesto por Rechenberg and Schwefel a mediados de los 1960s
  - Optimización numérica
  - Vectores de valores reales son usados
  - Cercanos a la evolución Lamarckiana
  - No tienen recombinación
  - Usan mutaciones auto adaptativas

## 1.4.4.4 Programación Genética

- Propuesto por Garis para la evolución de redes neuronales artificiales (RNAs)
- Usados por Koza para indicar como AG se aplican a evolucionar programas de computadora
- Representación de árboles
  - Expresiones LISP, son usualmente usados para representar individuos
- Cruzamiento y selección usados

Volvamos a revisarlo





# Termino preferido

- Los AEs tiene los mismo problemas que la IA
  - Representación
  - Búsqueda
- Todas las variantes se basan en población – generar y probar, y tienen mas similitudes que diferencias
- El mejor termino para referirse a ellos es Algoritmos Evolutivos (AEs)

# 1.5 Aplicabilidad

# Aplicabilidad de los AEs

- Las áreas principales donde se suelen usar son:
  - Optimización
  - Aprendizaje
  - Diseño
  - Teoría

# Optimización Evolutiva

- Optimización numérica (global)
- Optimización combinatoria (problemas NP duros)
- Optimización Mixta
- Optimización con restricciones
- Optimización Multiobjetivo
- Optimización en tareas dinámicas y/o en escenarios inciertos

# Aprendizaje Evolutivo

- Se puede dividir en supervisado, no supervisado y en aprendizaje reforzado
  1. Learning classifier systems (sistemas basados en reglas)
  2. Evolución de Redes Neuronales Artificiales
  3. Evolutivos de Sistemas de lógica difusa
  4. Aprendizaje co-Evolutivo
  5. Modularización automática de sistemas de aprendizaje (Machine Learning) por nichos y creación de especies

# Diseño Evolutivo

- Computación evolutiva es particularmente buena en la exploración de entornos no convencionales
  - Difíciles de obtener a mano
- Diseño evolutivo de:
  - RNAs
  - Circuitos electrónicos
  - Hardware Evolutivo
  - Diseño creativo interactivo usando AEs

# Teórica de cómputo evolutivo

- Explica
  - Cómo
  - Cuándo y
  - Porqué

los AEs funcionan

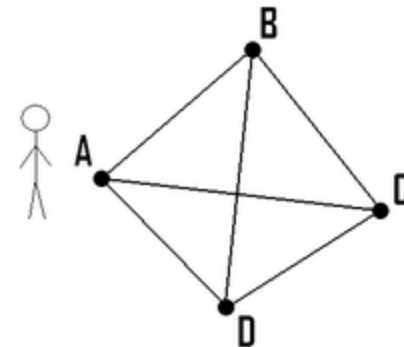
# Otros

General:

Optimización - Problemas NP completos

- Diseño automatizado
- Diseño de sistemas de distribución de aguas
- En Teoría de juegos, resolución de equilibrios.
- Análisis de expresión de genes.
- Optimización de estructuras moleculares.
- Predicción.
- Aplicaciones en planificación de procesos industriales.
- Selección óptima de modelos matemáticos para la descripción de sistemas biológicos.

- Construcción de horarios en grandes universidades, evitando conflictos de clases.
- Problema del viajante.



- Optimización de Redes Neuronales





# Resumen

- Existen problemas realmente difíciles de resolver, los cuales no se pueden aproximar con una solución de forma polinomial
- Se puede discretizar el problema, en soluciones mas pequeñas (individuos de un AE), las cuales se puedan evaluar en tiempo polinomial
- Usando ideas de la Evolución Natural, se tienen a los Algoritmos Evolutivos (AE) los cuales son toda una familia de técnicas enfocadas a diversas áreas.

# Resumen

- Los algoritmos Evolutivos son técnicas de búsqueda y optimización basadas en población, generación y prueba
- El cómputo evolutivo puede usarse en el diseño, optimización y aprendizaje
- El cómputo evolutivo es flexible y robusto
- Son técnicas útiles, pero puede haber otras mas apropiadas al problema

## Guión explicativo

- Este juego de diapositivas proporciona la introducción a los AE así como a los conceptos de optimización
- Se deben de leer en el orden en el que aparecen
- El siguiente bloque de diapositivas (no mostradas aquí), ahondaremos en los pasos fundamentales de los AGs.

# Referencias sugeridas

- Au, W.-H.; Chan, K.C.C.; Xin Yao, "A novel evolutionary data mining algorithm with applications to churn prediction," *IEEE Transactions on Evolutionary Computation*, vol.7, no.6, pp.532,545, Dec. 2003.
- 1997. *Handbook of Evolutionary Computation* (1st ed.). Thomas Back, David B. Fogel, and Zbigniew Michalewicz (Eds.). IOP Publ. Ltd., Bristol, UK.
- Mitchell Melanie. An introduction to Genetic Algorithms, A bradford book The MIT Press, 1999.
- Marcos Gestal Pose. Introducción a los algoritmos Genéticos, Universidad de Coruña, Depto. Tecnologías de la Información y las Comunicaciones.
- X. Yao. Evolutionary computation: A gentle introduction. In *Evolutionary Optimization*, R. Sarker, M. Mohammadian and X. Yao (eds.), Chapter 2, pp. 27-53, Kluwer Academic Publishers, Boston, 2002. (ISBN 0-7923-7654-4)