



## Algoritmos Genéticos

UDG - CUCEI

- 1 Introducción
- 2 Terminología Básica
- 3 Algoritmo Genético
  - Inicialización
  - Evaluación de la función objetivo
  - Selección
  - Cruza
  - Mutación
- 4 Variantes
  - Algoritmo Genético Elitista
  - Selección por Ranking
  - Selección por Torneo
  - Cruza Plana y Aritmética
  - Mutación - Distribución normal

# Introducción

Los Algoritmos Genéticos (GA) son simulaciones de la selección natural y pueden resolver problemas de optimización.

Características básicas de la selección natural:

- Un sistema biológico incluye una población de individuos (algunos de ellos se pueden reproducir)
- Los individuos tienen tiempo de vida
- Existe variación en la población (mutación)
- La habilidad de la supervivencia esta correlacionada positivamente con la habilidad de reproducirse

# Introducción (continuación)

En general, los GA mejoran los individuos de la población resolviendo las siguientes operaciones durante un proceso iterativo:

- Selección
- Cruza
- Mutación



# Terminología Básica

- Población: Conjunto de individuos, donde cada individuo representa una posible solución.
- Individuo: Un solo miembro de la población. Cada individuo contiene un cromosoma.
- Cromosoma: Estructura de datos que contiene una cadena de parámetros de ajuste o diseño.
- Aptitud: El valor que se asigna a cada individuo y que indica que tan bueno es este con respecto a los demás.

# Terminología Básica (continuación)

- Generación: Es una iteración de la medida de aptitud y a la creación de una nueva población por medio de los operadores.
- Selección: es el proceso mediante el cual algunos individuos en una población son seleccionados para reproducirse, típicamente con base a su aptitud.
- Cruza o Cruce: Es un operador que forma un nuevo cromosoma combinando partes de cada uno de sus cromosomas padres.
- Mutación: Operador que forma un nuevo cromosoma a través de alteraciones de los valores de los genes del cromosoma.

# Algoritmo Genético Continuo

Existen diversas versiones del los GA. Entre los algoritmos clásicos se encuentran el GA discreto y el GA continuo.

Algunas de las diferencias entre estos algoritmos son las siguientes:

- GA discreto: Es el algoritmo pionero. El cromosoma es representado por una cadena binaria conformada por genes. Es necesario decodificar la información binaria. Se utiliza para resolver principalmente problemas de optimización combinatoria.
- GA continuo: En este algoritmo el cromosoma esta conformado por valores continuos. Es una version modificada del GA discreto para resolver problemas de optimización continua.

# Algoritmo Genético Continuo (continuación)

Las etapas del GA continuación son los siguientes:

- Generación aleatoria de la población (Inicialización)
- Cálculo de la aptitud de cada individuo (Evaluación de la Función Objetivo)
- Selección de padres (operación de Selección)
- Generación de hijos (operación de Cruza o Cruce)
- Mutación de hijos (operación de Mutación)



# Inicialización

La inicialización de los individuos de la población dentro del espacio de trabajo del espacio continuo, se realiza mediante la siguiente ecuación

$$\mathbf{x}_i = \mathbf{x}_l + (\mathbf{x}_u - \mathbf{x}_l) \mathbf{r}_i$$

donde  $i = 1, 2, 3, \dots, N$  con  $N$  como el tamaño total de la población y

- $\mathbf{x}_i$  individuo  $i$  en la población
- $\mathbf{x}_l$  límite inferior del espacio de búsqueda
- $\mathbf{x}_u$  límite superior del espacio de búsqueda
- $\mathbf{r}_i$  vector de números aleatorios

Además,  $\mathbf{x}_i, \mathbf{x}_l, \mathbf{x}_u, \mathbf{r}_i \in \mathbb{R}^D$  con  $D$  como el tamaño de la dimension del problema.

# Evaluación de la función objetivo

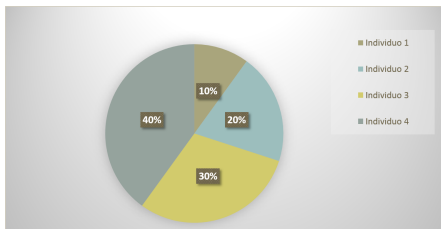
Para minimizar una Función Objetivo  $f(\mathbf{x})$  se propone evaluar la aptitud de cada individuo  $i$  como sigue

$$aptitud_i(\mathbf{x}_i) = \begin{cases} \frac{1}{1 + f(\mathbf{x}_i)} & \text{si } f(\mathbf{x}_i) \geq 0 \\ 1 + |f(\mathbf{x}_i)| & \text{si } f(\mathbf{x}_i) < 0 \end{cases}$$

esto debido a que los GA por naturaleza maximizan. Además, es necesario obtener valores positivos en la  $aptitud_i$  para poder realizar la operación de Selección.

# Selección

En la operación de selección, algunos individuos son seleccionados para reproducirse, típicamente con base a su aptitud. Existen diferentes tipos de selección. Como por ejemplo: selección por ruleta, selección por torneo, entre otros.



En la selección por ruleta, cada individuo es asignado dependiendo a la proporción de su aptitud. Un valor de aptitud grande tiene más probabilidades de ser seleccionado.

# Selección (continuación)

Selección por ruleta:

Por cada individuo  $i$  se asigna una probabilidad de selección como sigue

$$p_i = \frac{aptitud_i}{\sum_{k=1}^N aptitud_k}$$

donde

- $aptitud_i$  es la aptitud del individuo  $i$
- $N$  es el total de individuos en la población

Dependiendo del valor  $p_i$ , el individuo  $i$  puede ser seleccionado como un padre para crear descendencia.

## Selección (continuación)

---

**Algorithm 1** Selección por ruleta

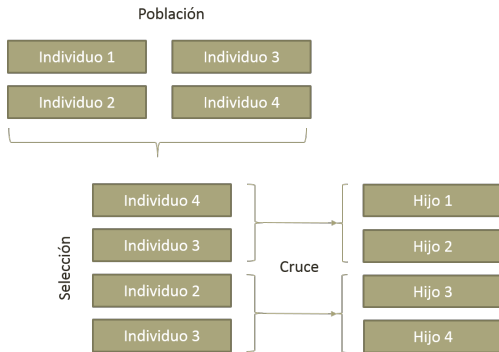
---

```
1:  $aptitud_{total} \leftarrow 0$ 
2: Desde  $i = 1$  Hasta  $N$ 
3:    $aptitud_{total} \leftarrow aptitud_{total} + aptitud_i$ 
4: Fin Desde
5: Desde  $i = 1$  Hasta  $N$ 
6:    $p_i \leftarrow \frac{aptitud_i}{aptitud_{total}}$ 
7: Fin Desde
8:  $r \leftarrow$  generar número aleatorio tal que  $r \in [0, 1]$ 
9:  $p_{sum} \leftarrow 0$ 
10: Desde  $i = 1$  Hasta  $N$ 
11:    $p_{sum} \leftarrow p_{sum} + p_i$ 
12:   Si  $p_{sum} \geq r$  Entonces
13:     seleccionar individuo  $i$  como padre
14:   Regresar
15: Fin Si
16: Fin Desde
17: seleccionar individuo  $i \leftarrow N$  como padre
```

---

# Cruza

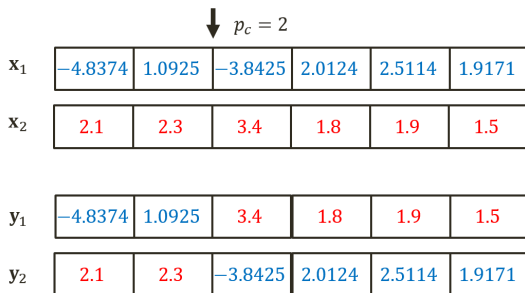
En la operación de Cruza, dos padres son seleccionados en base a su aptitud para generar dos hijos. Importante: Un padre no se puede cruzar consigo mismo, pero si puede ser seleccionado varias veces.



Se deben seleccionar tantos padres necesarios para crear hijos suficientes y mantener el tamaño de la población  $N$ .

## Cruza (continuación)

En la operación Cruza, los padres intercambian información del cromosoma a partir de un punto de cruce  $p_c$ . Dados dos padres  $\mathbf{x}_1$  y  $\mathbf{x}_2$ , generamos dos hijos  $\mathbf{y}_1$  y  $\mathbf{y}_2$  como sigue:



el punto de cruce  $p_c$  se calcula aleatoriamente tal que  $p_c \in \{1, D\}$  con  $D$  como la dimension del problema, en este caso  $D = 6$ .

# Mutación

En la operación de Mutación, cada elemento del cromosoma  $j \in \{1, D\}$  de cada hijo generado  $i \in \{1, N\}$ , tiene la probabilidad de ser mutado.

La Mutación se calcula de la siguiente manera

$$y_{ij} = \begin{cases} y_{ij} & \text{si } r_a \geq p_m \\ x_{lj} + (x_{uj} - x_{lj}) r_b & \text{si } r_a < p_m \end{cases}$$

donde  $p_m$  es la probabilidad de mutación y  $r_a, r_b \in [0, 1]$  son números aleatorios. La probabilidad de mutación es seleccionada por el usuario tal que  $p_m \in [0, 1]$ .

Además,  $x_{lj}$  y  $x_{uj}$  son elementos de los límites de trabajo inferior  $\mathbf{x}_l$  y superior  $\mathbf{x}_u$ , respectivamente.



# Mutación (continuación)

---

**Algorithm 2** Mutación

---

```
1:  $N \leftarrow$  definir población total
2:  $D \leftarrow$  definir dimensión del problema
3:  $p_m \leftarrow$  definir probabilidad de mutación
4: Desde  $i = 1$  Hasta  $N$ 
5:     Desde  $j = 1$  Hasta  $D$ 
6:          $r_a \leftarrow$  generar número aleatorio tal que  $r \in [0, 1]$ 
7:         Si  $r_a < p_m$  Entonces
8:              $r_b \leftarrow$  generar número aleatorio tal que  $r \in [0, 1]$ 
9:              $y_{ij} \leftarrow x_{lj} + (x_{uj} - x_{lj}) r_b$ 
10:        Si No
11:             $y_{ij} \leftarrow y_{ij}$ 
12:        Fin Si
13:    Fin Desde
14: Fin Desde
```

---

# Algoritmo GA

Finalmente, el algoritmo GA es descrito a continuación:

---

**Algorithm 3** Algoritmo GA

---

- 1:  $N \leftarrow$  definir población total
  - 2:  $\mathbf{x}_i \leftarrow$  inicializar  $N$  padres aleatoriamente,  $i \in \{1, N\}$
  - 3: **Hacer**
  - 4:      $aptitud_i \leftarrow$  calcular la aptitud de cada individuo (padres)  $i$
  - 5:      $\mathbf{y}_i \leftarrow \{\emptyset\}$  generar conjunto vacío de hijos,  $i \in \{1, N\}$
  - 6:     **Mientras** que la población  $\mathbf{y}$  sea menor que  $N$
  - 7:         selección de padres  $r_1$  y  $r_2$  por ruleta tal que  $r_1 \neq r_2$
  - 8:         padres  $\{\mathbf{x}_{r_1}, \mathbf{x}_{r_2}\}$  generan hijos  $\{\mathbf{y}_1, \mathbf{y}_2\}$
  - 9:          $\mathbf{y} \leftarrow \mathbf{y} \cup \{\mathbf{y}_1, \mathbf{y}_2\}$
  - 10:     **Fin Mientras**
  - 11:     Mutar aleatoriamente a los hijos  $\mathbf{y}_i$
  - 12:      $\mathbf{x} \leftarrow \mathbf{y}$
  - 13: **Mientras** que se cumpla el total de generaciones  $G$
-

# Algoritmo Genético Elitista

El problema con el GA clásico es que los padres mueren y los hijos sobreviven para la siguiente generación. El algoritmo no evoluciona adecuadamente porque muchos de los padres con buenas soluciones no se mantienen para futuras generaciones.

El Elitismo asegura que los mejores individuos se mantienen para la siguiente generación.

Algunas características para implementar Elitismo son:

- Se produce solo una cantidad de  $(N - E)$  hijos por generación, donde  $N$  es el tamaño de la población y  $E$  es el número de Elitistas.
- El número de Elitistas  $E$  lo define el usuario.
- La población para la siguiente generación se determina por la cantidad de hijos  $(N - E)$  y la cantidad de Elitistas  $E$ .

# Algoritmo Genético Elitista (continuación)

El algoritmo GA Elitista es descrito a continuación:

---

**Algorithm 4** Algoritmo GA Elitista

---

- 1:  $N \leftarrow$  definir población total
  - 2:  $E \leftarrow$  definir número de individuos Elite
  - 3:  $\mathbf{x}_i \leftarrow$  inicializar  $N$  padres aleatoriamente,  $i \in \{1, N\}$
  - 4: **Hacer**
    - 5:  $aptitud_i \leftarrow$  calcular la aptitud de cada individuo (padres)  $i$
    - 6:  $\mathbf{y}_i \leftarrow \{\emptyset\}$  generar conjunto vacío de hijos,  $i \in \{1, N - E\}$
    - 7: **Mientras** que la población  $\mathbf{y}$  sea menor que  $N - E$ 
      - 8: selección de padres  $r_1$  y  $r_2$  por ruleta tal que  $r_1 \neq r_2$
      - 9: padres  $\{\mathbf{x}_{r_1}, \mathbf{x}_{r_2}\}$  generan hijos  $\{\mathbf{y}_1, \mathbf{y}_2\}$
    - 10:  $\mathbf{y} \leftarrow \mathbf{y} \cup \{\mathbf{y}_1, \mathbf{y}_2\}$
  - 11: **Fin Mientras**
  - 12: Mutar aleatoriamente a los hijos  $\mathbf{y}_i$
  - 13:  $\mathbf{x}_E \leftarrow$  seleccionar mejores  $E$  padres
  - 14:  $\mathbf{x} \leftarrow \mathbf{y} \cup \mathbf{x}_E$
  - 15: **Mientras** que se cumpla el total de generaciones  $G$
-

# Selección por Ranking

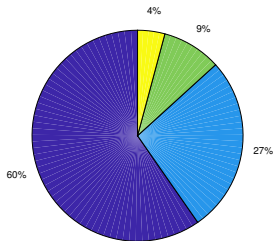
La selección por ranking ordena a los individuos del mejor al peor y realiza la selección con base a un ranking en vez del valor de la aptitud.

Individuo	Mejor individuo	...	...	Peor Individuo
Rank	4	3	2	1
Probabilidad	40%	30%	20%	10%

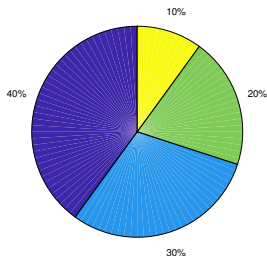
Una vez ordenados los individuos, se le asigna a cada uno un peso. Al mejor individuo se le asigna un peso de  $N$  y al peor individuo un peso de 1. Se calcula entonces la probabilidad del ranking en vez de la probabilidad con base en la aptitud.

## Selección por Ranking (continuación)

A continuación se muestra una comparativa entre la probabilidad con base en Aptitud y Ranking.



(a) Probabilidad con base en Aptitud



(b) Probabilidad con base en Ranking

Se utiliza el algoritmo de la ruleta para la etapa de selección, pero con la probabilidad con base en Ranking.

# Selección por Torneo

La selección por ruleta para grandes poblaciones requiere alto costo computacional.

La selección por torneo reduce el costo computacional de la etapa de selección.

Los pasos para la selección por torneo son:

- Se seleccionan  $\tau$  individuos aleatoriamente de la población, donde  $\tau$  es el tamaño de participantes en el torneo y este valor lo elige el usuario.
- Después se selecciona al individuo con la mejor aptitud entre los participantes en el torneo.

# Cruza plana y aritmética

La cruce plana se utiliza para crear un hijo  $\mathbf{y}$ , con base en la información de dos padres  $\mathbf{x}_1$  y  $\mathbf{x}_2$ . Se utiliza la siguiente ecuación

$$\mathbf{y} = \alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2$$

donde  $\alpha$  es un número aleatorio tal que  $\alpha \in [0, 1]$ .

En la cruce aritmética, dos hijos se crean con base en la siguiente regla

$$\begin{aligned}\mathbf{y}_1 &= \alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2 \\ \mathbf{y}_2 &= (1 - \alpha) \mathbf{x}_1 + \alpha \mathbf{x}_2\end{aligned}$$



# Mutación - Distribución normal

En la operación de Mutación con distribución normal, cada elemento del cromosoma  $j \in \{1, D\}$  de cada hijo generado  $i \in \{1, N\}$ , tiene una probabilidad distribuida normalmente para ser mutado.

La Mutación se calcula de la siguiente manera

$$y_{ij} = \begin{cases} y_{ij} & \text{si } r_a \geq p_m \\ y_{ij} + N(0, 1) & \text{si } r_a < p_m \end{cases}$$

donde  $p_m$  es la probabilidad de mutación,  $r_a, r_b \in [0, 1]$  son números aleatorios y  $N(0, 1)$  representa un número aleatorio con una distribución normal de media 0 y varianza 1.

# Gracias por tu atención!

Información de contacto:

Dr. Javier Enrique Gómez Avila

E-mail: [jenrique.gomez@academicos.udg.mx](mailto:jenrique.gomez@academicos.udg.mx).