Institución Universitaria
Reacreditada en Alta Calidad

Optimización

Optimización Heurística

Docente: Cristian Guarnizo Lemus

Somos Innovación Tecnológica con Sentido Humano

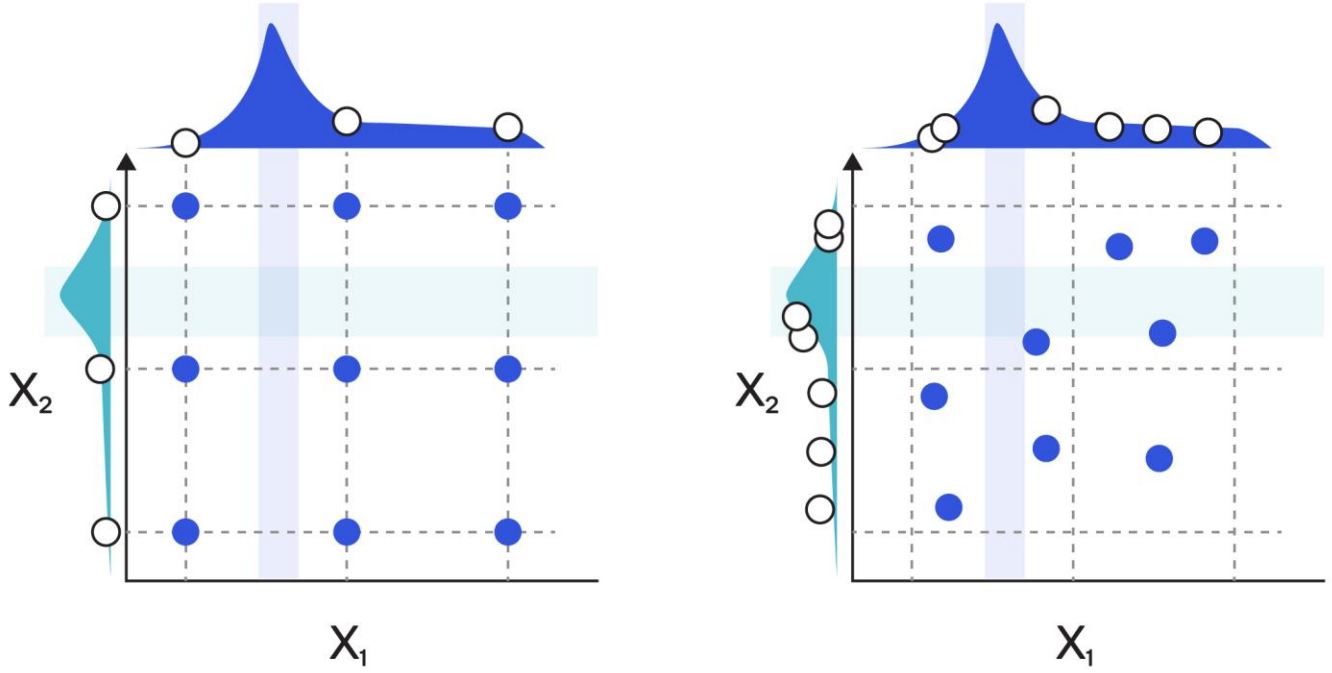
Alcaldía de Medellín

Contenido

1. Introducción a la optimización Heurística.
2. Algoritmos Genéticos.
3. Cumulo de Partículas.

Optimización Heurística

Se basa en búsquedas aleatorias combinadas con la experiencia o comportamientos presentes en la naturaleza. Por lo general emplea una población para buscar el valor optimo.

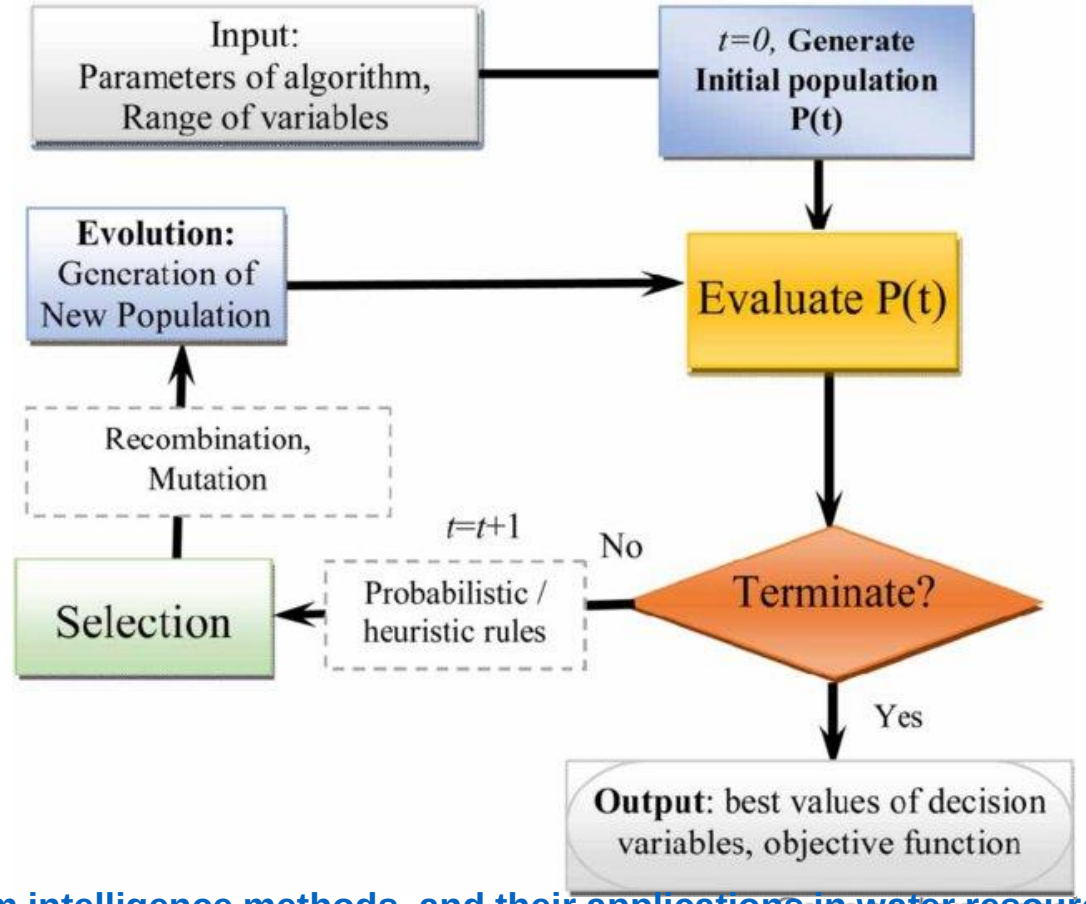


(a) Standard Grid Search

(b) Random Search

Actualización de la Población

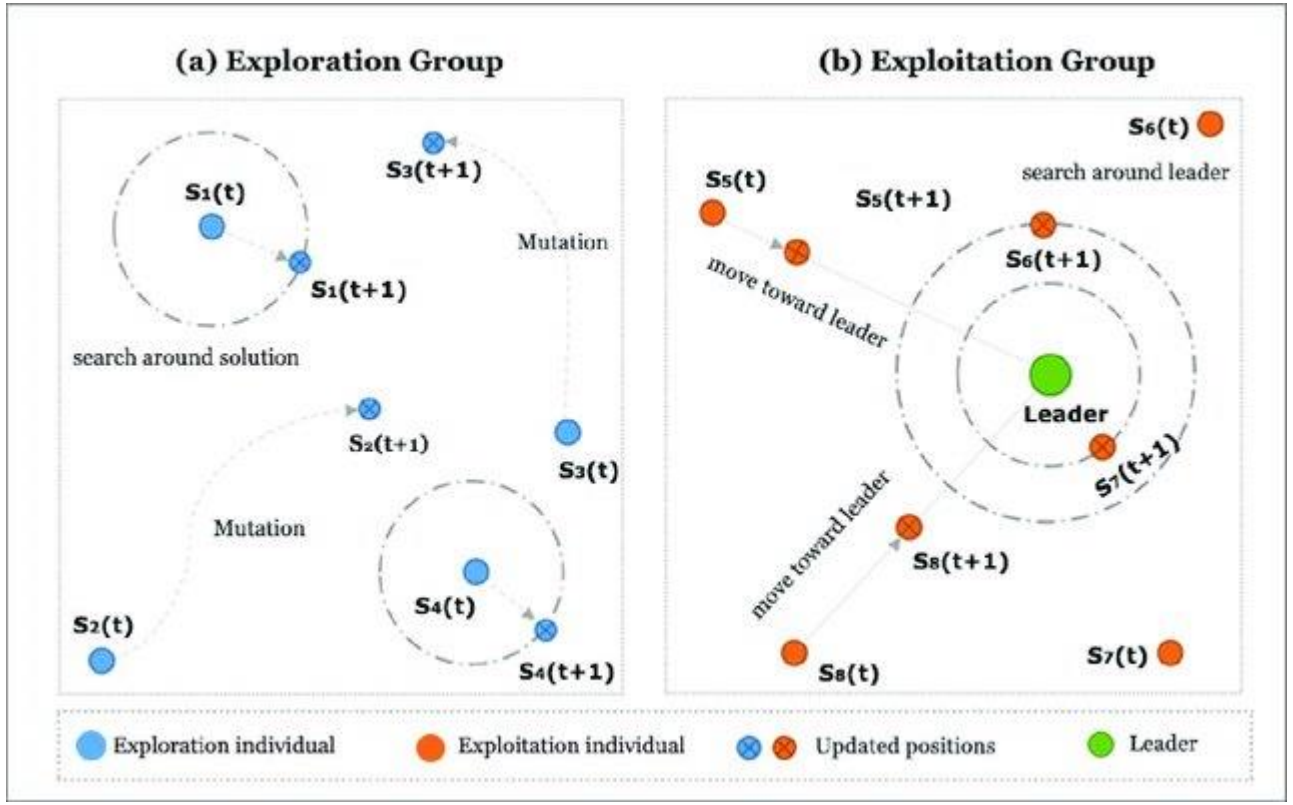
La población se actualiza con el fin de explorar posibles solución al problema de optimización.
El conjunto de reglas para realizar la actualización es el corazón del método heurístico.



Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review

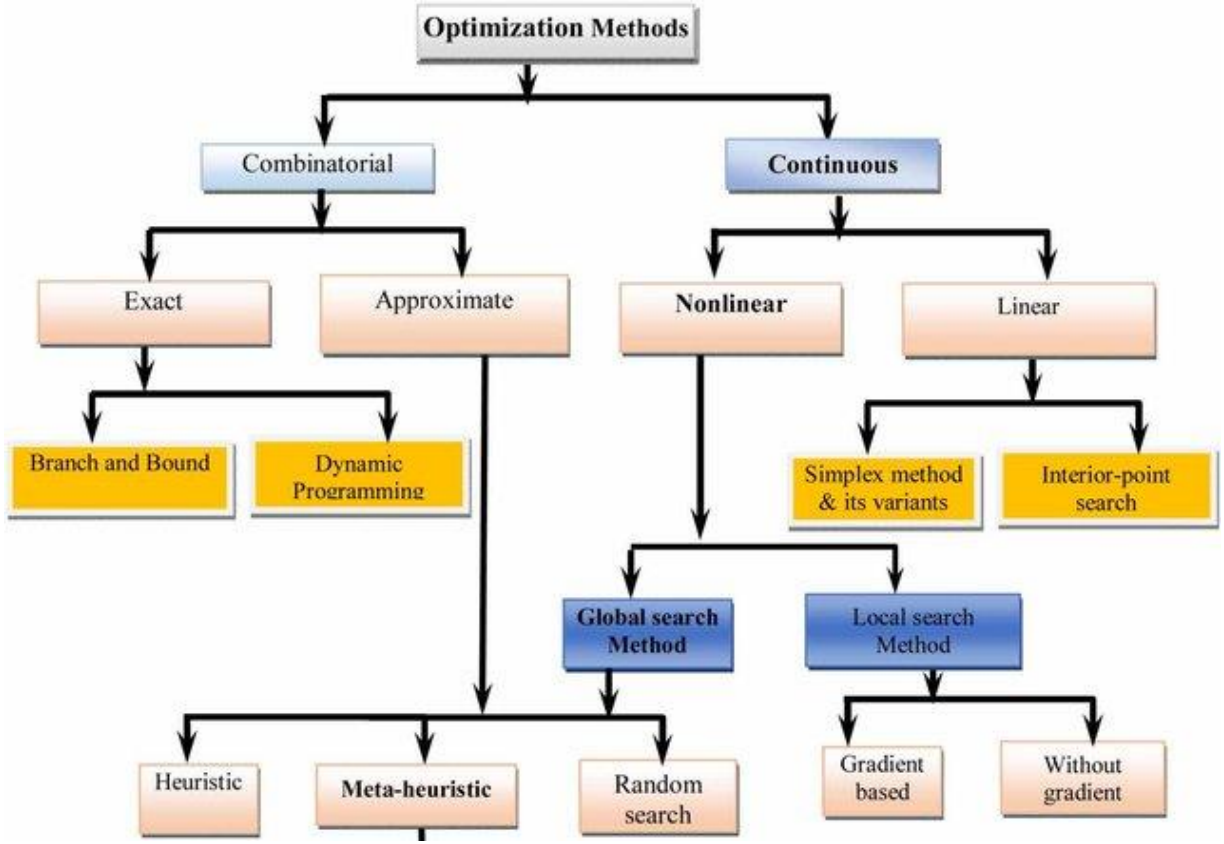
Exploración y Explotación

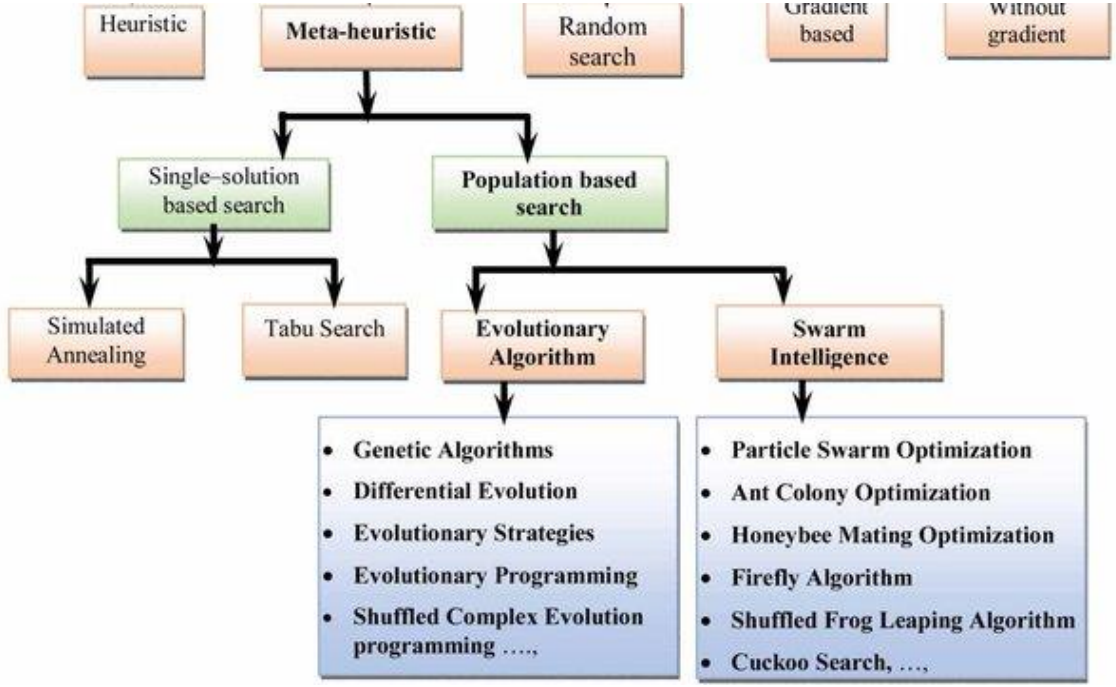
Por lo general los métodos de búsqueda aleatoria emplean los estos dos conceptos.
Exploración: Crear individuos de la población en zonas poco exploradas del espacio de soluciones.
Explotación: Generar nuevos individuos alrededor del optimo local actual.



https://ieeexplore.ieee.org/document/9164971/

Métodos





Ventajas y Aplicaciones

Ventajas:

- No requieren conocer la expresión matemática de la función objetivo. Por tal motivo son técnicas libres de gradiente.
- Convergen a un optimo local o cerca de él.
- Funcionan bien y producen buenas soluciones en casos muy complejos.
- Son altamente paralelizables para funciones objetivo de bajo costo.

Aplicaciones:

- Problemas de Job Scheduling o Timetables (reorganizar tareas de una fabrica, oficina, etc., de modo de minimizar algo, normalmente tiempo o costo).
- Problemas de diseño automatico o Asistido (diseño de antenas, de dispositivos electronicos).
- Seleccionar parametros de un controlador PID.

Chequeo

- Cual es la idea principal de los algoritmos de búsqueda heurística en optimización?
- Que es la población? Hay algoritmos heurísticos que realicen búsqueda sin población?
- Que es exploración y explotación?

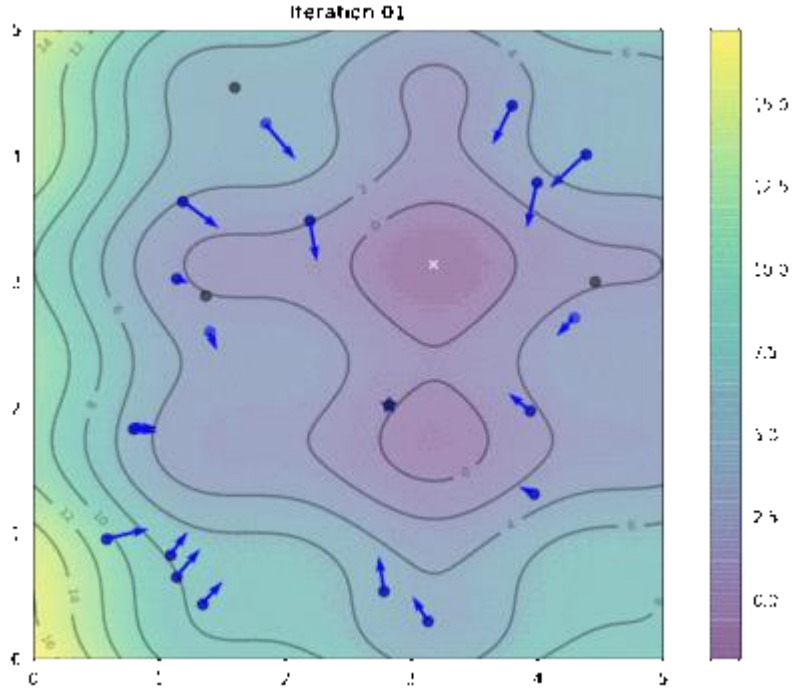
Institución Universitaria
Reacreditada en Alta Calidad

Contenido

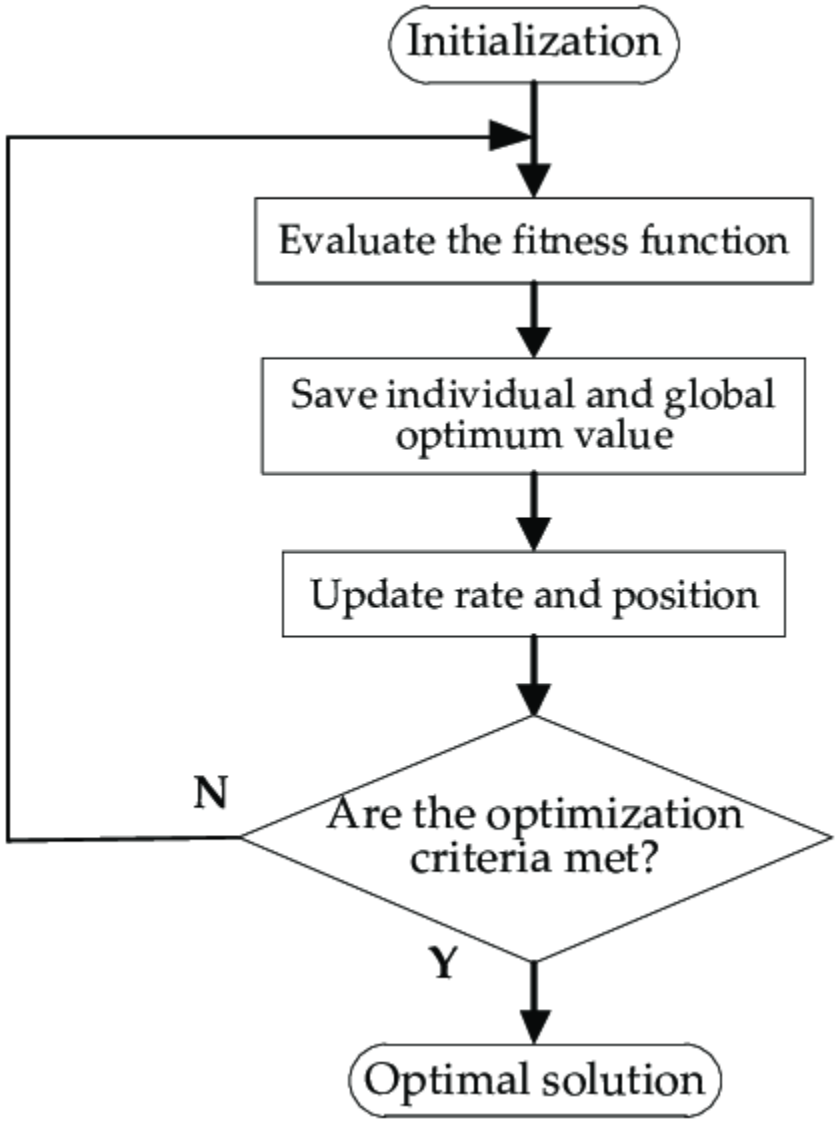
1. Introducción a la optimización Heurística.
2. Enjambre de Partículas.
3. Algoritmos Genéticos.

Enjambre de partículas - PSO

- Se basa en Swarm intelligence (inteligencia de Enjambre), la cual almacena evaluaciones pasadas y actuales de la función objetivo, y los individuos, para así generar reglas de actualización de los individuos (particulas).



Enjambre de partículas - PSO



Evaluar la función de aptitud sobre cada individuo $\boldsymbol{f_X} = f(\boldsymbol{X}) \in R^N$

$$P_b^t, g_b^t$$

$$V_i^{t+1} = W.V_i^t + c_1 U_1^t\left(P_{b_1}^t - P_i^t\right) + c_2 U_2^t\left(g_b^t - P_i^t\right)$$

$$P_i^{t+1} = P_i^t + V_i^{t+1}$$

Enjambre de partículas - PSO

```
1: procedure PSO
2:     for each particle p in swarm S  do
3:         initialize particle with a feasible random position
4:         evaluate the fitness F_i of particle
5:     end for
6:     evaluate p_best , g_best
7:     for i = 1 to maximum_iterations do
8:         for each particle p in swarm S  do
9:             v_ij = v_ij + c_1 r_1j [ p_best,i - x_ij ] + c_2 r_2j [ g_best - x_ij ]
10:            x_ij = x_ij + v_ij
11:        end for
12:        update p_best , g_best
13:        choose best fitness as solution
14:        End if termination conditions met
15:    end for
16: end procedure
```

https://www.sciencedirect.com/science/article/pii/S2214579617302915

Enjambre de partículas - PSO

El parámetro W es el peso de inercia, este parámetro es importante para balancear la búsqueda de la solución global. Valor grande explora, valor pequeño explota.

Diversificación: busca nuevas soluciones, encuentra regiones con mejores soluciones potenciales.

Intensificación: explora las soluciones previas, encuentra la mejor solución de una región dada.

$$V_i^{t+1} = W.V_i^t + c_1 U_1^t \left(P_{b_1}^t - P_i^t\right) + c_2 U_2^t \left(g_b^t - P_i^t\right)$$

Inercia: hace que la partícula se mueva en la misma dirección y con la misma velocidad.

Influencia personal: mejora al individuo. Hace que la partícula vuelva a una posición previa, mejor que la actual.

Influencia social: hace que la partícula siga la mejor solución de los vecinos.

https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-particle-swarm-optimization-algorithm/

Institución Universitaria
Reacreditada en Alta Calidad

Contenido

1. Introducción a la optimización Heurística.
2. Enjambre de Partículas.
3. Algoritmos Genéticos.

Algoritmos Genéticos

- Basado en la teoría de la evolución de Darwin, “reproducción y supervivencia del mas apto”.
- Los individuos de la población (candidatos) se denotan como cromosomas. Las nuevas generaciones con mutaciones representan mejores solución al problema.
- Cada cromosoma se le asocia un valor de aptitud (función objetivo o fitness function).
- Se organizan de mayor a menor (Ranking) y se seleccionan algunos individuos como padres (Selection).
- De los padres se genera una nueva generación (Crossover) de hijos, los cuales pueden presentar evoluciones (Mutation).

Codificación Individuos

Codificación real

- Si el vector solución $\boldsymbol{x} \in \boldsymbol{R}^P$ es flotante o real, entonces el individuo i-ésimo de la población

$$\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{iP}]$$

Codificación Entera

- Cada elemento de $\boldsymbol{x}$ puede ser definido por enteros.

$$\boldsymbol{x}_i = [9,4, \dots, 120]$$

- A partir de un conjunto de elementos binarios, p.e. si el numero esta entre 0 y 255 se requiere una cadena binaria de 8 bits para cada valor entero.

$$\boldsymbol{x}_i = [0,0,0,1,0,1,1,1,1,1] \Rightarrow [0,5,15]$$

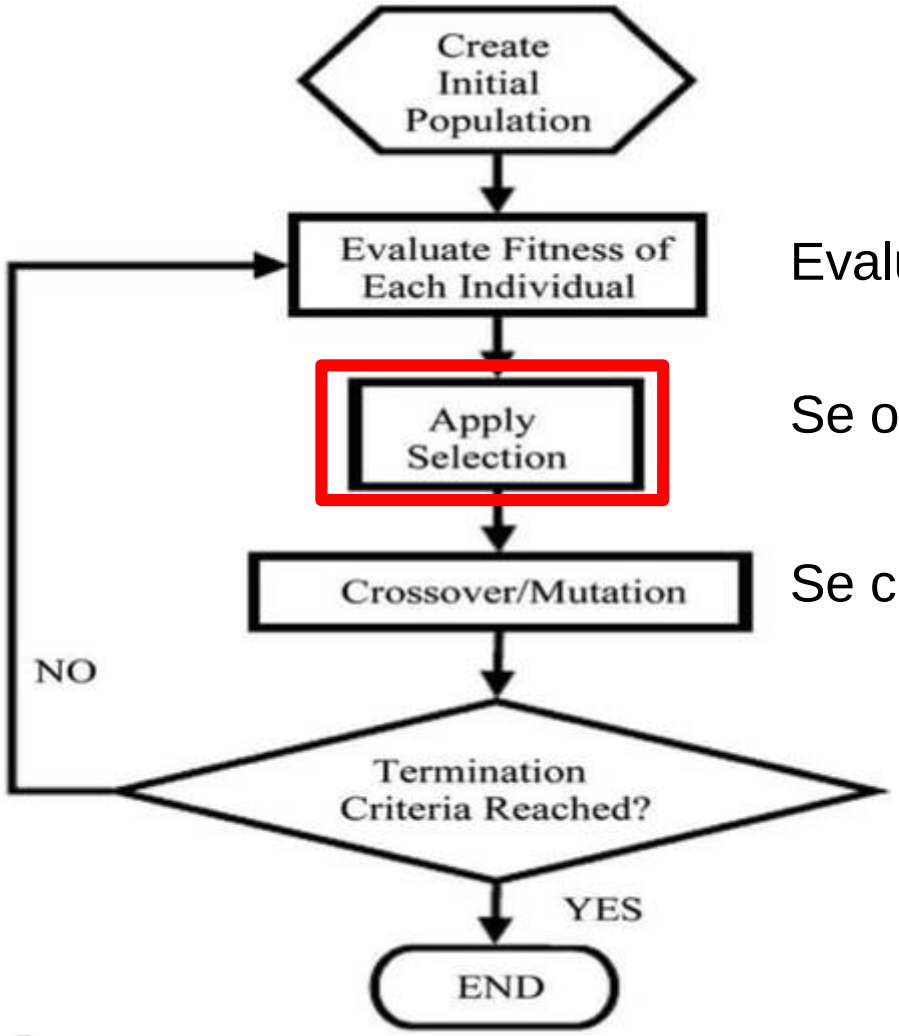
Codificación Binaria

- Para problemas donde la solución sean variables binarias (activación o desactivación).

$$\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{iP}], \forall x_{ij} \in \{0,1\}$$

Codificación mixta

Algoritmos Genéticos



Crear una matriz con N individuos de la población $\boldsymbol{X} \in R^{N \times P}$

Evaluar la función de aptitud sobre cada individuo $\boldsymbol{f_X} = f(\boldsymbol{X}) \in R^{N}$

Se organiza por aptitud y seleccionan los padres mas aptos.

Se cruzan los padres y los hijos resultan se mutan.

Lo criterios de parada pueden ser por iteración o cuando en una cantidad de iteraciones no se mejore la solución optima.

https://www.mdpi.com/2078-2489/10/12/390

AG - Selección

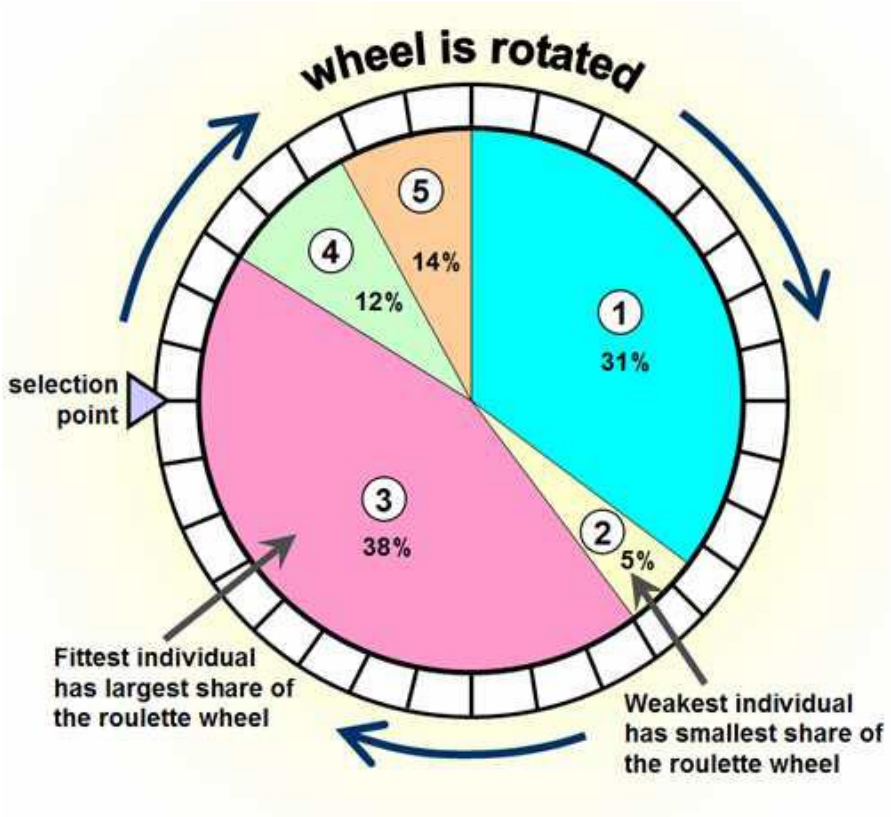
Consiste en el criterio para seleccionar los padres. Según Darwin los mejores deberían sobrevivir para crear la nueva generación:

– Ruleta
– Rango: Primero se realiza un ranking de acuerdo a la aptitud. Posteriormente se inicia en 1 para el que tenga la menor aptitud, 2 para el segundo menor, hasta llegar a N para el mayor.
– Torneo: Se realiza una ruleta K veces. De los K cromosomas seleccionados se crea un subgrupo. Se selecciona el mejor del subgrupo como padre.
– Aleatorio

AG – Selección: Ruleta

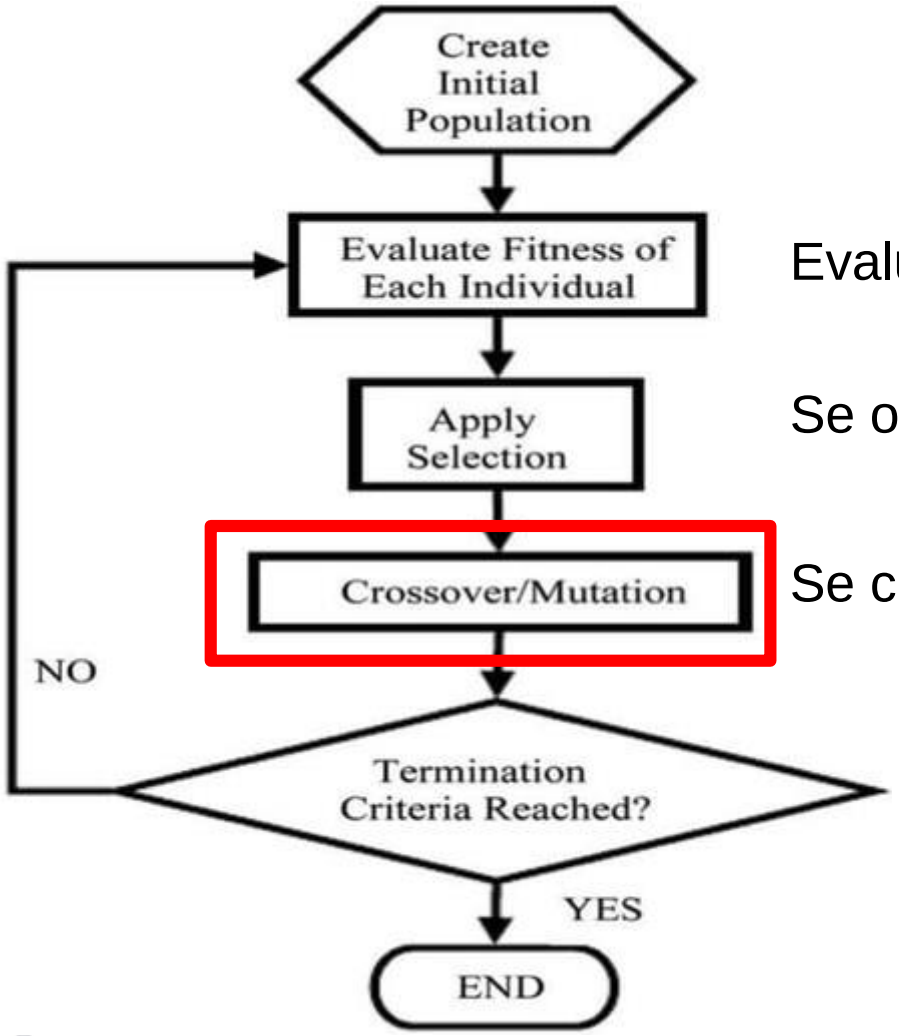
Los padres se seleccionan de acuerdo a su “aptitud”. Los mejores cromosomas son los que tienen mas probabilidad de ser elegidos.

– Calcule la suma de las aptitudes de todos los cromosomas (S).
– Genere un numero aleatorio entre [0-S] → r.
– Realice sumas parciales de las aptitudes de los cromosomas, pare cuando se supere r.



Cromosoma	Aptitud	% Total
1	6.82	31
2	1.11	5
3	8.48	38
4	2.57	12
5	3.08	14
Total	22.05	100

Algoritmos Genéticos



Crear una matriz con N individuos de la población $\boldsymbol{X} \in R^{N \times P}$

Evaluar la función de aptitud sobre cada individuo $\boldsymbol{f_X} = f(\boldsymbol{X}) \in R^{N}$

Se organiza por aptitud y seleccionan los padres mas aptos.

Se cruzan los padres y los hijos resultan se mutan.

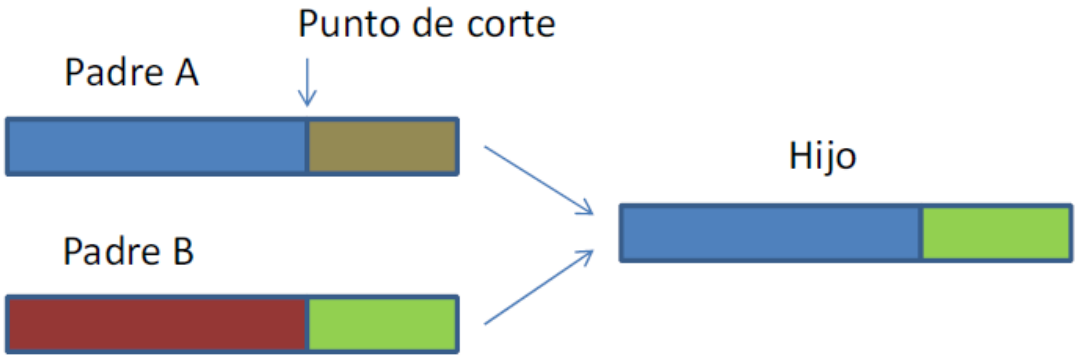
Lo criterios de parada pueden ser por iteración o cuando en una cantidad de iteraciones no se mejore la solución optima.

https://www.mdpi.com/2078-2489/10/12/390

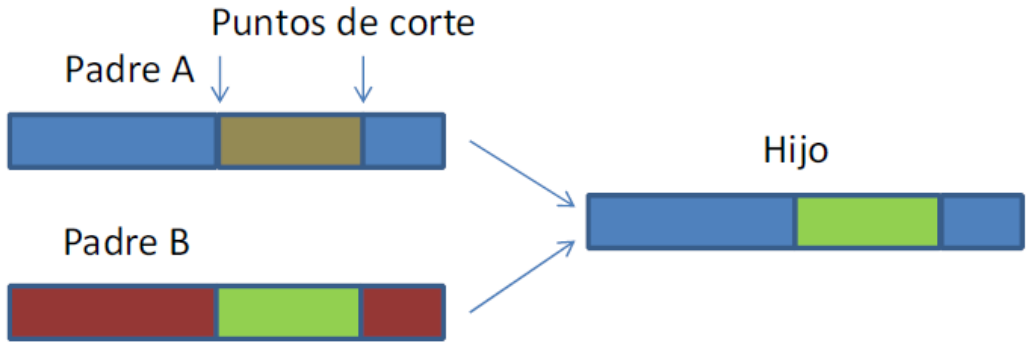
AG - Crossover

Es un operador genético utilizado para actualizar la configuración de los cromosoma de cada generación. Hereda particularidades de los padres.

– Un punto

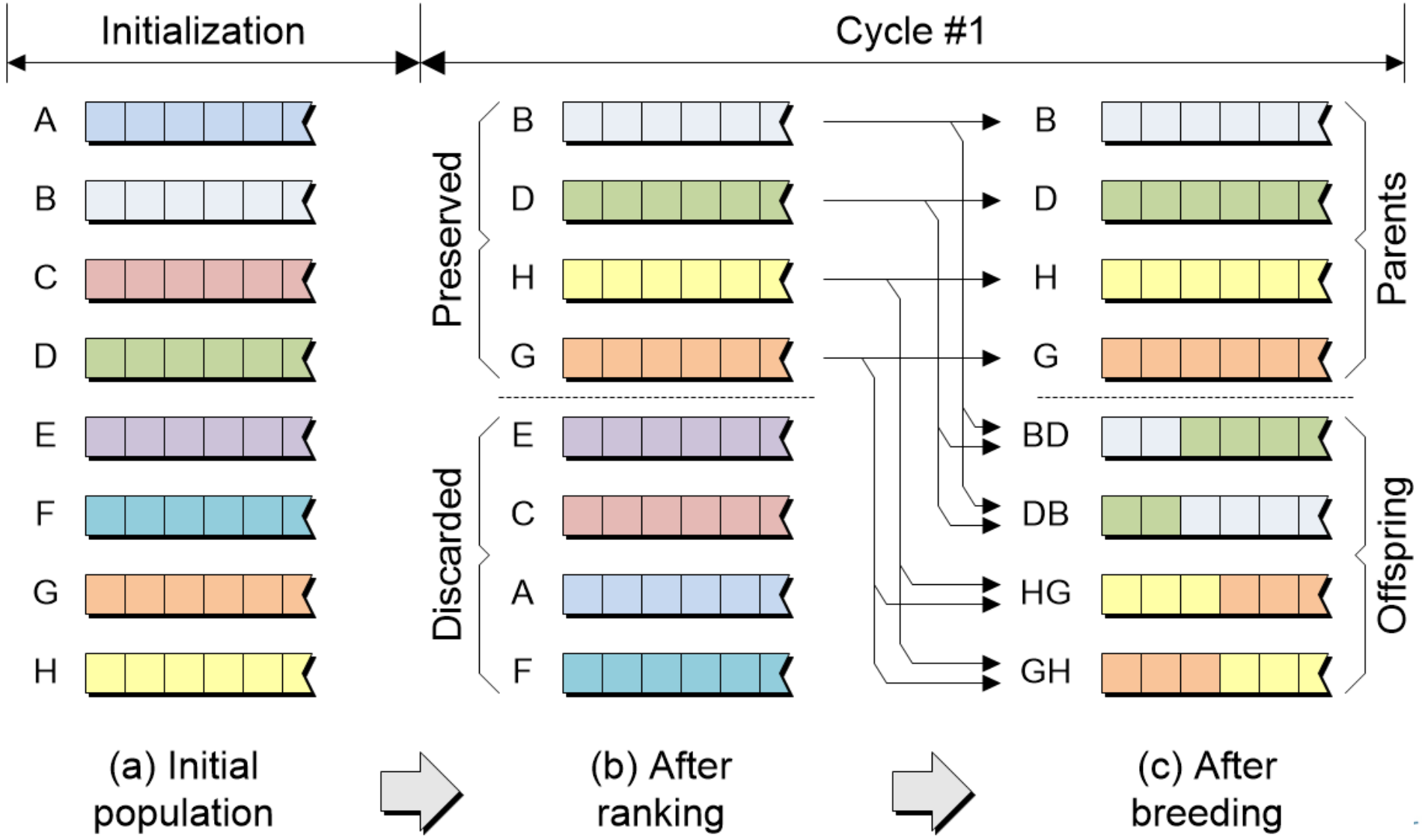


- Dos puntos



– Uniforme
– Permutación (codificación entera)
– Aritmético (codificación real)

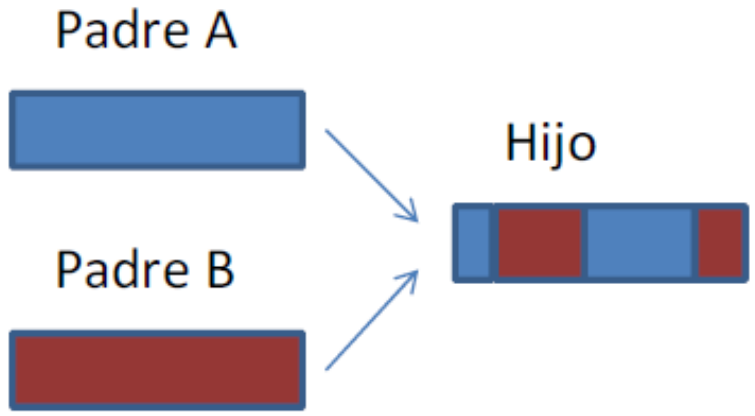
AG - Crossover



AG – Crossover: Uniforme

Se copian aleatoriamente los genes de los padres:

```
for i=1:10,
    r=rand();
    If r>=0.5,
        hijo(i)=padrea(i);
    else
        hijo(i)=padreb(i);
    end
end
```



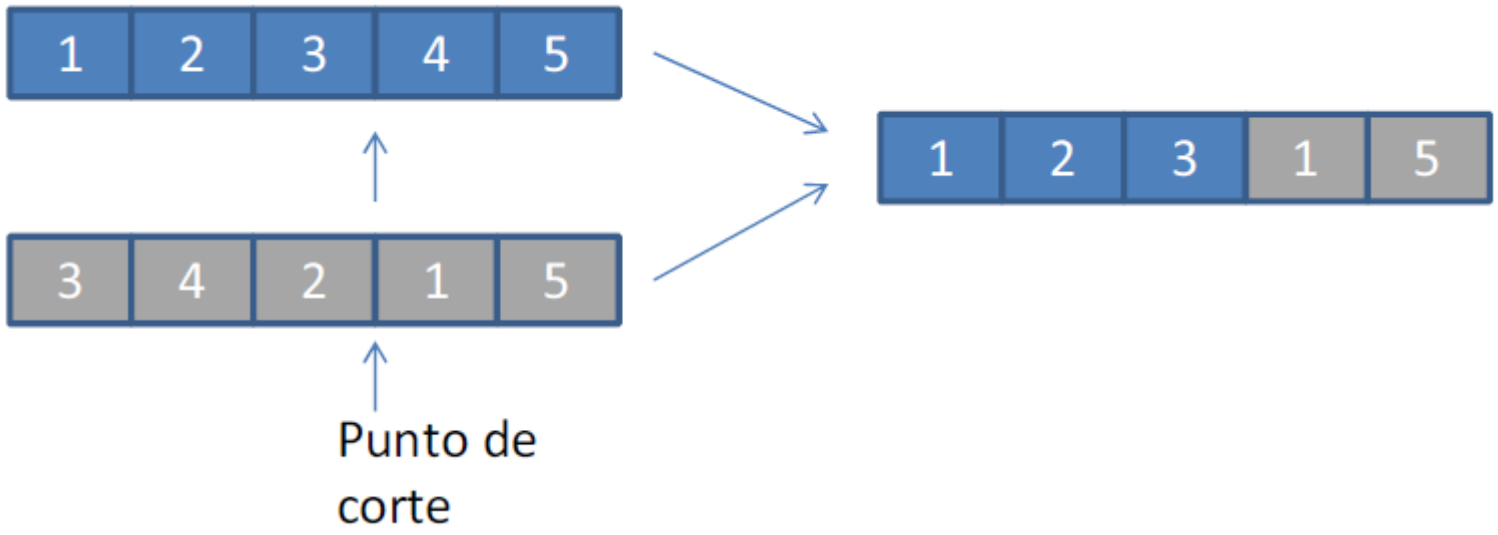
AG – Crossover: Permutación

Cuando los genes son enteros:

- Normal
- Un punto
- Orden 1
- Parcialmente mapeado
- Ciclo

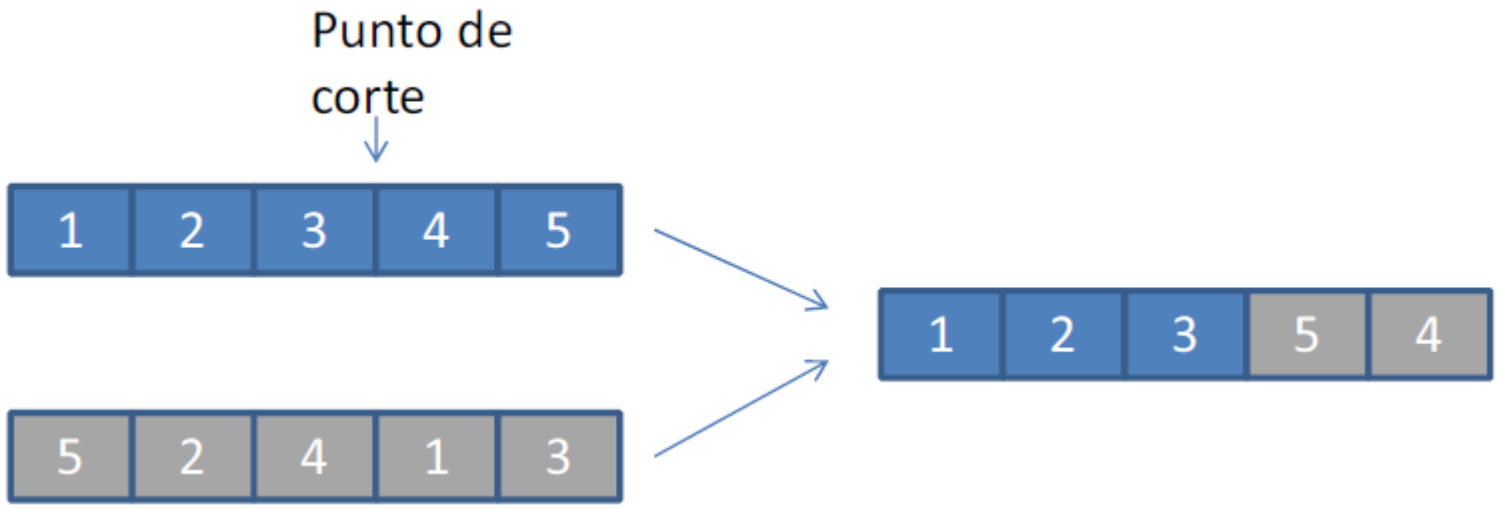
AG – Crossover: Permutación - Normal

Se selecciona un punto de corte, y se intercambian los alelos. Se aplica cuando no importa si se repiten los números:



AG – Crossover: Permutación – Un punto

Se selecciona un punto de corte, se copia del padre A los genes hasta el corte, del padre B se toman en orden los elemento que no estén:

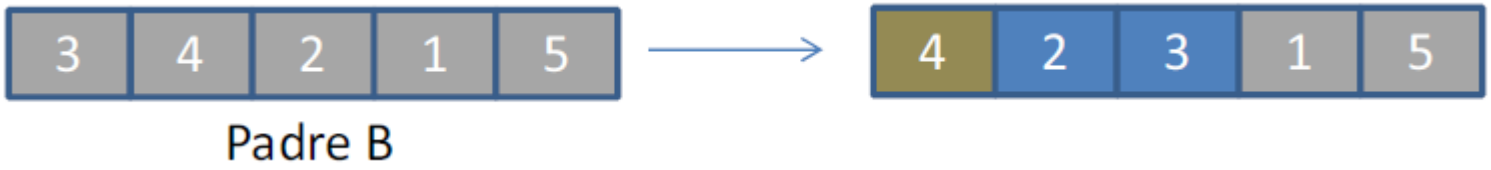


AG – Crossover: Permutación – Orden 1

La idea es preservar el orden relativo de los genes.
Procedimiento:

• Seleccione un segmento del padre A.

• Copie el segmento al hijo.
• Copie los números que no están en el segmento, al primer hijo:
– Empezando a la derecha del punto de corte.
– Usando el orden del padre B.



AG – Crossover: Permutación – Parcialmente mapeado

1. Seleccione al azar un segmento de padre A y cópielo al hijo.

2. Comenzando en el primer punto de corte busque elementos en ese segmento del padre B que no han sido copiados.

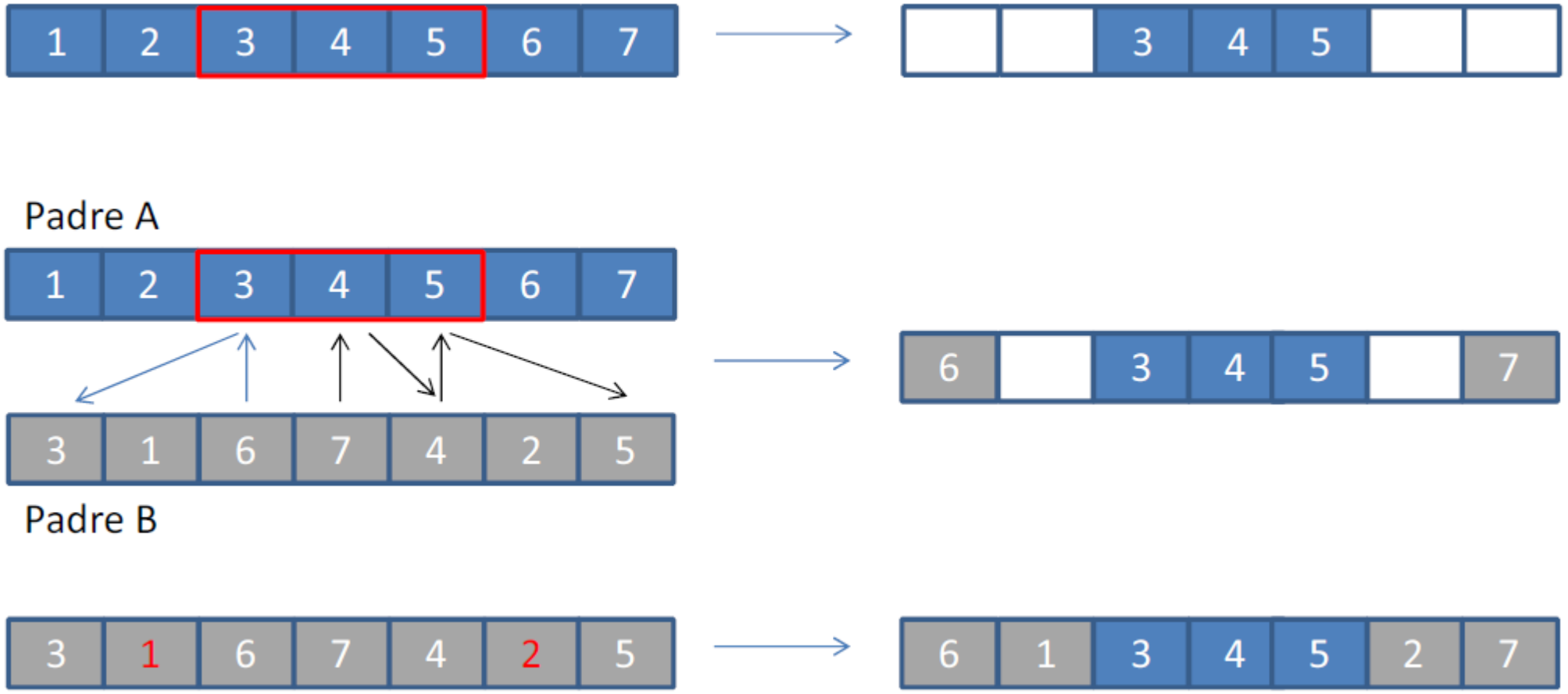
3. Para cada gen i busque en el hijo que elemento j se copio en su lugar desde padre A.

4. Ubique a i en la posición ocupada por j en padre B.

5. Si la posición ocupada por j en padre B ya esta ocupada por k en el hijo, ponga a i en la posición ocupada por k en el padre B.

6. Después de ubicar los genes del segmento, el resto se llena con los genes del padre B.

AG – Crossover: Permutación – Parcialmente mapeado



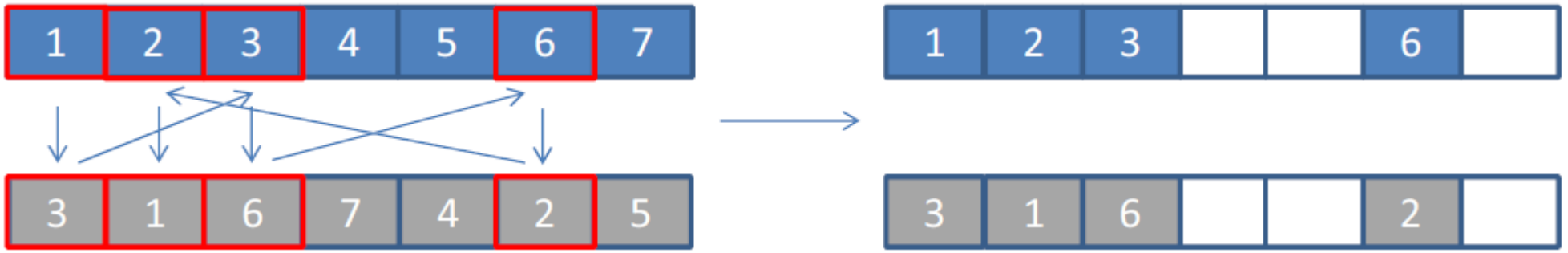
AG – Crossover: Ciclo

Realizar un ciclo de alelos desde Padre A de las siguiente forma:

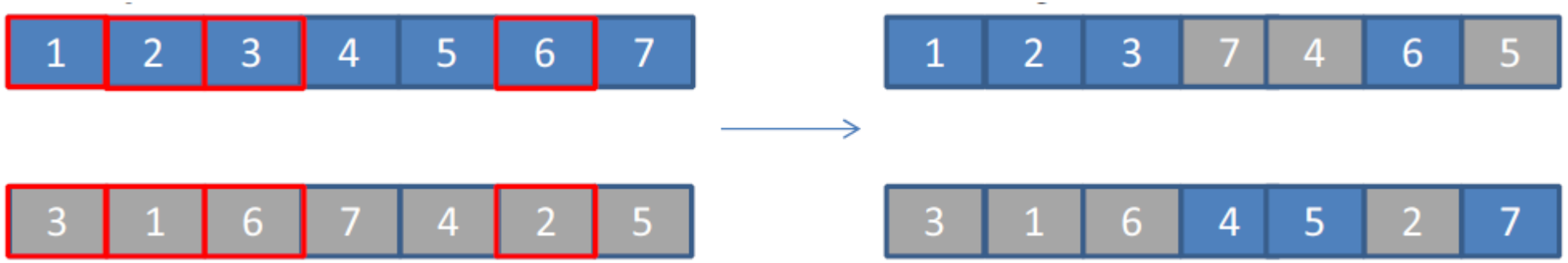
- Empezar con el primero alelo de Padre A.
- Mirar el alelo en la misma posición en Padre B.
- Vaya a la posiciones con el mismo alelo en A.
- Agregue este alelo al ciclo.
- Repita el procedimiento hasta encontrar el primero alelo de padre A.

AG – Crossover: Ciclo

- Identificar ciclos:



- Copiar ciclos alternados a los hijos:



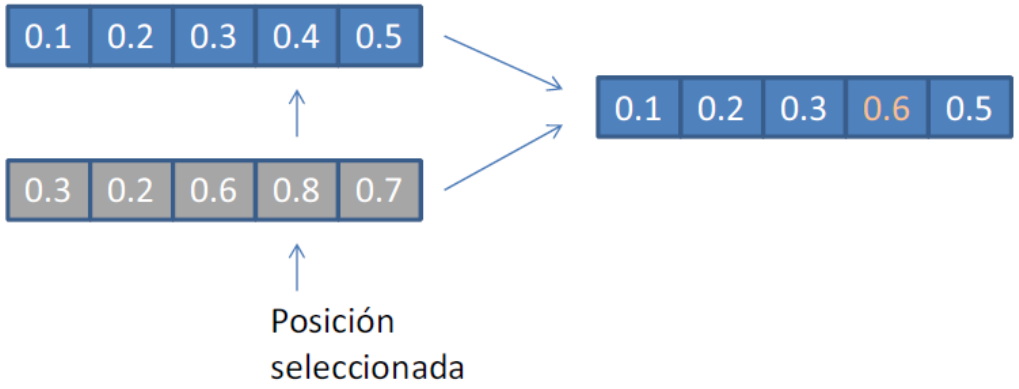
AG – Crossover: Aritmético

Utilizado cuando los genes están codificados en valores reales. Se aplica la regla

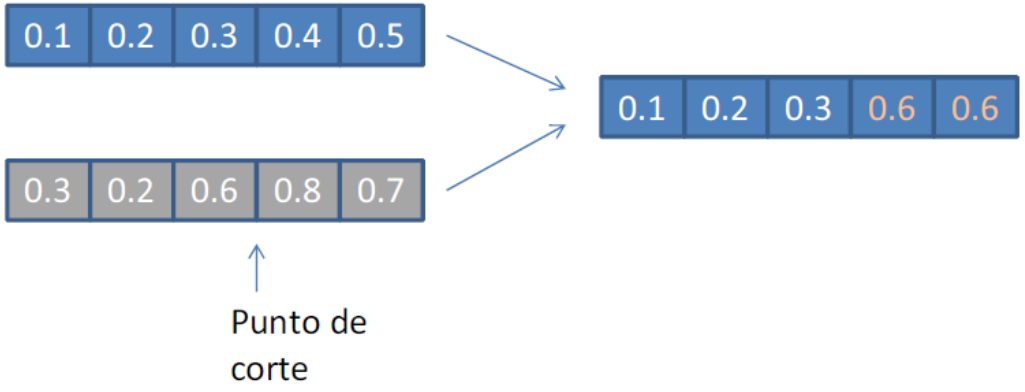
Hijo(i)=a*Padrea(i)+(1-a)*Padreb(i) con 0<a<1

Casos:

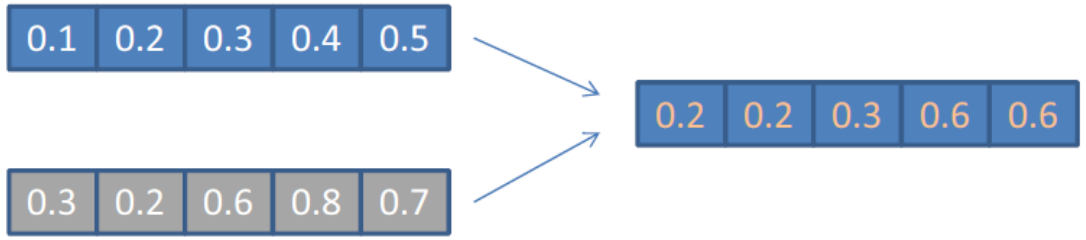
– Unitario



Simple



– Completo



AG – Crossover: Mutación

Crea pequeñas variaciones a los hijos, de esta forma se esta cerca de la herencia de los padres.

- Inversión del bit (codificación binaria)

- Cambio de orden (codificación entera)

- Suma número aleatorio (cod. Real)

AG – Inserción

- Se van agregando los hijos en una nueva población hasta que se llena.
- Elitismo: en cada iteración se almacena el cromosoma con mayor aptitud.
- Se desechan los padres menos aptos y se cambian por los hijos mas aptos.

Referencias

- M. Janga Reddy, D. Nagesh Kumar. Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review. 2020. https://doi.org/10.2166/h2oj.2020.128

Institución Universitaria
Reacreditada en Alta Calidad

¡Gracias!

Somos Innovación Tecnológica con Sentido Humano

Alcaldía de Medellín