

# Práctica #2

## Diseño de un AG Estándar vs AG simple

---



16 NOVIEMBRE

---

Algoritmos Genéticos  
María Elena Cruz Meza



# Práctica #2 Diseño de un AG Estándar vs AG simple

## Objetivo:

Mostrar el conocimiento, la habilidad, y destreza para identificar y manipular los parámetros y procesos involucrados en los AG; específicamente en el diseño de un AG Estándar e identificar los métodos básicos de selección, cruce y mutación para un AG simple, identificando las bondades entre estos dos tipos de Algoritmos Genéticos.

Objetivos particulares:

- Derivado de que en la práctica No. 1 se ha solicitado que el usuario elija el tipo de SG (Generacional o Estacionario o elitista), ahora el usuario podrá elegir el tipo de Selección de la población, así como los tipos de operadores para la Cruza y Mutación
- Diseñar un Algoritmo Genético Estándar
  - Selección discreta
  - Cruza por 1 punto
  - Mutación aleatoria o por intercambio de bit
- Algoritmo Genético simple
  - Selección proporcional Ruleta
  - Cruza o recombinación
    - Cruza de 1 punto
    - Cruza de 2 puntos
  - Mutación
    - Básica: Por intercambio de bit
    - Permutación por:
      - Inserción
      - Desplazamiento
- Gráfica para la evaluación del algoritmo y conclusiones respecto al desempeño de ambos tipos de AG (Estándar vs Simple)

## Instrucciones

- **Conocimientos previos**
  - a) **Haber desarrollado la práctica No.1**
  - b) Habilidades en la implementación de vectores,
  - c) Habilidades en la implementación e interpretación de histogramas
- **Sistema operativo y lenguaje de programación**
  - a) En los lenguajes y sistema operativo elegidos en la práctica No.1, continuar desarrollando la práctica.
- **Entrega de evidencia**
  - a) Trabajo colaborativo: a partir de esta práctica, el resto de las prácticas y el proyecto serán desarrolladas y entregadas por equipo. En caso de trabajar solo un integrante, deberás indicar a la maestra.
  - b) Medio para su entrega: Entregar la tarea en la plataforma Microsoft Teams una vez terminada toda la práctica completa. Elaborar un informe del desarrollo de esta práctica en un archivo con formato de video (consultar rúbrica para conocer los indicadores de evaluación).
  - c) Crear un archivo en formato de video, elija la herramienta de su preferencia y subirlo a alguna plataforma para compartir el link para visualizarlo (drive, Dropbox, youtube, etc). El archivo debe enviarse con una nomenclatura específica y con el siguiente contenido:
    - Nombre al archivo: Si se elige una plataforma donde se tiene acceso al archivo, este deberás nombrarlo como: P2-ABC-DEF.FOR, donde “P2” indica el número de práctica reportada, “ABC” corresponderá a iniciales al nombre del primer integrante (empezando por el apellido), y “DEF” corresponden al segundo integrante y así sucesivamente con el resto de los integrantes separados por un guión. “FOR” es el formato correspondiente al archivo tipo video, etc. Si optas por compartirlo en youtube o alguna herramienta similar, entonces el nombre deberá ser: “Practica #2-AG Estándar vs AG Simple”, y en la descripción correspondiente indicar los nombres de los participantes. Como solo es para fines académicos se sugiere compartirlo en forma privada, pero, tú decides su publicación libre.
    - Portada: Es importante agregar algunos datos al inicio del vídeo: identidad politécnica (Nombre o logos de la unidad académica: IPN y Escom), Nombre de la unidad de aprendizaje, Número de la práctica, Objetivo de la práctica, Nombre de quien presenta, es opcional mostrar el Nombre del(a) facilitador(A) (profesora) y Fecha de entrega.
    - Marco teórico: El vídeo debe incluir una breve información formal que sustenta el trabajo que se desarrolla (para este caso puede

tomar como base o referencia lo que se anexa al final de este documento, notas del curso facilitadas en la plataforma o consultar algún autor), si consulta alguna página, esta debe ser validada por alguna universidad u organismo educativo (paper o artículos, cursos en línea, presentaciones de docentes o investigadores, libros gratuitos en línea o por alguna editorial, etc.).

- Programa y pruebas: En el vídeo debe explicarse que lenguaje de programación es utilizado para implementar el algoritmo, explicar los segmentos de código principales, y posteriormente el funcionamiento de este con las ff acorde a los métodos que se le aplicaran para optimizar. Incluir las conclusiones respecto al desempeño de ambos tipos de AG (Estándar vs Simple)
- Al entregar el vídeo también anexa un archivo en formato PDF o TXT, con el código principal completo para su verificación y ponderación respectiva conforme los indicadores marcados en la rúbrica. El nombre de este archivo deberá nombrarse como el archivo de vídeo.
  - En caso de no terminar todos o bien crea que aun tenga fallas en la programación, presentar los que se hayan realizado. Esto es con fin de evitar que en esta tarea quede en ausente (vacío o cero), repercutiendo en tu evaluación final.
- Créditos: Al final del video agregar las referencias consultadas para el desarrollo de la actividad.
- Para trabajos con uno o dos integrantes la duración del vídeo es recomendable entre 5 a 10 minutos. Para equipos de mas de dos integrantes es recomendable no más de 15 minutos.

# Fundamentos

El diseño de un Algoritmo Genético Simple, también conocido como Canónico (ver figura 1) involucra varios procesos, necesita una codificación o representación del problema, requiere una función de ajuste o adaptación al problema, la cual asigna un número real a cada posible solución codificada. Luego, durante la ejecución del algoritmo, los padres deben ser seleccionados para la reproducción (recombinación o cruce), para luego estos padres seleccionados cruzarlos y generen dos hijos, uno por cada padre, y finalmente se elige sobre cuáles actuará un operador de mutación. Al terminar el primer ciclo también conocido como generación, la combinación de todos estos operadores proveerán un conjunto de individuos (posibles soluciones al problema), los cuales conforme evolucione el Algoritmo Genético, formarán parte de la siguiente población.

```

t ← 0
población(t) ← poblaciónInicial
EVALUAR(población(t))

while not (condición de terminación)
    t ← t + 1
    población(t) ← SELECCIONAR(población(t-1))
    población(t) ← CRUZAR(población(t))
    población(t) ← MUTAR(población(t))
    EVALUAR(población(t))
    EVALUAR(población(t))

return población(t)

```

Imagen 1. Estructura de un algoritmo genético clásico

La función de selección de padres más utilizada es la denominada función de selección proporcional a la función objetivo, en la cual cada individuo tiene una probabilidad de ser seleccionado como padre que es proporcional al valor de su función objetivo. Denotando por  $p_{i,t}^{prop}$  la probabilidad de que el individuo  $I_{j,t}$  sea seleccionado como padre, se tiene que:

$$p_{i,t}^{prop} = \frac{g(I_i^t)}{\sum_{j=1}^{\lambda} g(I_j^t)} \quad \left. \vphantom{\frac{g(I_i^t)}{\sum_{j=1}^{\lambda} g(I_j^t)}} \right\} \begin{array}{l} \text{Esta función es invariante ante un} \\ \text{cambio de escala, pero no ante} \\ \text{una traslación} \end{array}$$

Sin embargo, un problema frecuente durante este proceso es la convergencia prematura causada por los super individuos. Para resolver el problema al aplicar la anterior función

de selección, usualmente se efectua la selección proporcional al rango del individuo, ello produce una repartición más uniforme de la probabilidad de selección entre los individuos.

Por otro lado, si denotamos por rango  $g(I_i^t)$  el rango de menor a mayor (es decir el peor individuo tiene rango 1, mientras que el individuo con una mejor función objetivo tiene rango  $\lambda$ ), y sea  $p_{i,t}^{rango}$  la probabilidad de que el individuo  $I_{j,t}$  sea seleccionado como padre cuando la selección se efectua proporcionalmente al rango del individuo, se tiene que

$$p_{i,t}^{rango} = \frac{rango [g(I_i^t)]}{\lambda(\lambda + 1)/2}$$

La suma de los rangos,  $\lambda(\lambda+1)/2$ , constituye la constante de normalización. La función de selección basada en el rango es invariante frente a la translación y al cambio de escala.

Otro posible refinamiento del modelo de selección proporcional, es el modelo de selección del valor esperado. En este modelo, para cada individuo  $I_{j,t}$  se introduce un contador, inicializado en  $g(I_{it})/\bar{g}_t$ , donde  $\bar{g}_t$  denota la media de la función objetivo en la generación  $t$ . Cada vez que el individuo  $I_{j,t}$  es seleccionado para el cruce, dicho contador decrece en una cantidad ( $c \in (0, 5; 1)$ ). El individuo en cuestión dejaría de poder ser seleccionado en esa generación, cuando su contador sea negativo.

Una vez que los individuos denominados padres son elegidos, se procede a la recombinación o cruce entre ellos. Y uno de los métodos base es el presentado por Holland, en el cual solo se opta por elegir un punto de cruce entre la pareja de padres. Sin embargo, otros investigadores han propuesto otros operadores de cruce, tal es el caso resentedo por De Jong (1975), quien propone elegir mas de un punto de cruce. Él investigó el comportamiento del operador de cruce basado en multiples puntos, concluyendo que el cruce basado en dos puntos, representaba una mejora mientras que anadir más puntos de cruce no beneficiaba el comportamiento del algoritmo. La ventaja de tener más de un punto de cruce radica en que el espacio de búsqueda puede ser explorado más fácilmente, siendo la principal desventaja el hecho de aumentar la probabilidad de ruptura de buenos esquemas. En el operador de cruce basado en dos puntos, los cromosomas (individuos) pueden contemplarse como un circuito en el cual se efectua la selección aleatoria de dos puntos. Visto desde este esquema, el cruce basado en un punto, puede verse como un caso particular del cruce basado en dos puntos, en el cual uno de los puntos de corte se encuentra fijo al comienzo de la ristra que representa al individuo.

El operador secundario a la cruce es la mutación, este operador es de suma importancia ya que proporciona un pequeño elemento de aleatoridad en la vecindad (entorno) de los individuos de la población. Aunque el operador de cruce es el responsable de efectuar la búsqueda a lo largo del espacio de posibles soluciones, al parecer la mutación se ha vuelto de gran interés por algunos investigadores, principalmente a medida que la población de individuos va convergiendo (Davis, 1985). En este sentido, Schaffer y col. (1989) encuentran que el efecto del cruce en la búsqueda es inferior al que previamente se esperaba y utilizan la denominada evolución primitiva, en la cual, el proceso evolutivo consta tan sólo de selección y mutación, en este estudio afirman que se supera a una evolución basada exclusivamente en la selección y el cruce. También describen que la determinación del valor óptimo de la probabilidad de mutación es mucho más crucial que el relativo a la probabilidad de cruce.

Con lo anterior, es notorio que la búsqueda del valor óptimo para la probabilidad de mutación ha sido muy estudiada y encontramos que De Jong (1975) recomienda la utilización de una probabilidad de mutación del bit de  $1/l$ , donde  $l$  es la longitud de la cadena. Schaffer y col. (1989) utilizan resultados experimentales para estimar la tasa óptima proporcional a  $1/\lambda 0.9318 \ln 0.4535$ , donde  $\lambda$  denota el número de individuos en la población. Si bien en la mayoría de las implementaciones de Algoritmos Genéticos se asume que tanto la probabilidad de cruce como la de mutación permanecen constantes, algunos autores han obtenido mejores resultados experimentales modificando la probabilidad de mutación a medida que aumenta el número de iteraciones.

Cuando se opta por diseñar un algoritmo genético simple, este se diferenciará del algoritmo estándar, principalmente por el uso de distintas técnicas con los operadores de selección, cruce y mutación. Ahora veamos como diseñar un AG Estándar y un Algoritmo Genético Simple, con los operadores descritos, basándonos en el ejemplo siguiente.

***“Sea la función fitness  $f(x) = \text{ABS} \mid x - 5 \mid / 2 + \text{sen}(x)$ , con un rango de datos que va de [0-15], y con una representación binaria, eligiendo el tipo de AG generacional. Siendo los individuos (fenotipos)  $x_1 = 7$  y  $x_2 = 4$  los seleccionados para iniciar la población”***



# Algoritmo genético estándar

1. Seleccionar aleatoriamente a dos padres.

**Del ejemplo, suponiendo que los individuos elegidos son los siguientes cromosomas**

$$x_1(7) = 0\ 1\ 1\ 1 \quad \text{y} \quad x_2(4) = 0\ 1\ 0\ 0\ 0$$

2. Con probabilidad de cruce  $pc$ , cruzar los padres. Usualmente se propone a  $pc$  entre 0.6 y 0.9

- a. Aleatoriamente entre un rango de 0.0 a 1.0, elegir a un  $va$
- b. Si  $va \geq pc$  habrá cruce en esta generación, si no, entonces los cromosomas simplemente se copian.

- Para cada individuo  $x_i$  verificar si es apto para la cruce

- a) Evaluar la función de aptitud

- b) Obtener una  $va_i$  y verificar si cae dentro del rango de su probabilidad proporcional a su fitness  $pf(x)$ :

- i.  $pf(x_i) = \frac{f(x_i)}{f(x_i) + f(x_j)}$ , con  $i \neq j$

- ii. Si  $va_i \in [pisop(f_i) - techo(f_i)] \rightarrow$  El individuo es apto para la cruce

- iii. Sino  $\rightarrow$  el cromosoma simplemente se copia. El individuo no podrá cruzarse aunque haya cruce en esa generación.

**Con los fenotipos elegidos del ejemplo: Si se tiene una  $Pc$  del 70%,  $pc=0.7$ , habrá cruce en esta generación, por lo tanto hay que validar si los padres elegidos son aptos para cruzarse.**

**Aptitud de los Individuos elegidos:**

Individuo 1 ( $x=7$ ):  $f(7) = ABS \left| \frac{7-5}{2+Sen(7)} \right| = 0.75$

Individuo 2 ( $x=4$ ):  $f(4) = ABS \left| \frac{4-5}{2+Sen(4)} \right| = 0.8$

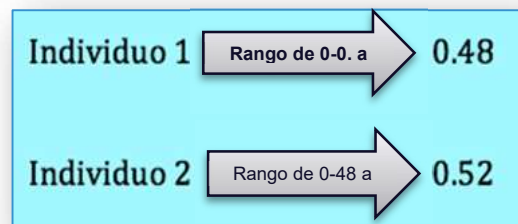


Evaluación fitness de cada individuo y verificando si son aptos para la cruce

$$\text{Individuo 1} = \frac{0.75}{0.75 + 0.8} = 0.48$$

$$\text{Individuo 2} = \frac{0.8}{0.75 + 0.8} = 0.52$$

- Si sacamos un número aleatorio dentro del intervalo [0-0.48) el individuo 1 se emparejará y
- Si otro número aleatorio está en el intervalo [0.48-1] también será apto el individuo 2



Aleatoriamente se tienen:  $va(x_i)=0.41$  y  $va(x_j)=0.74$

Rango del Individuo 1 [0, 0.48]	
<b>0.41</b>	Esta dentro del rango
El individuo 1 es apto para la cruce	

Rango del Individuo 2 [0.48, 1]	
<b>0.74</b>	Esta dentro del rango
El individuo 2 es apto para la cruce	

Ambos individuos son aptos, ellos se podrán cruzar conforme la valoración de la  $P_c$  se cumple.

Cruza en la generación: proceso del operador cruza de un punto con la pareja de padres

Si suponemos que se obtiene el  $va$  para la  $P_c$  como 0.15, verificamos los rangos de la  $P_c$ :

Rango de cruza [0, 0.7]

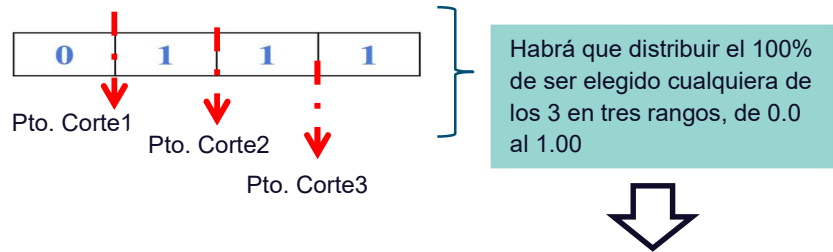
Rango de no cruza [0.7-1]

Dado que  $pc=0.7$ :

0.15 → Número aleatorio obtenido  
Habrá cruza en esta generación!

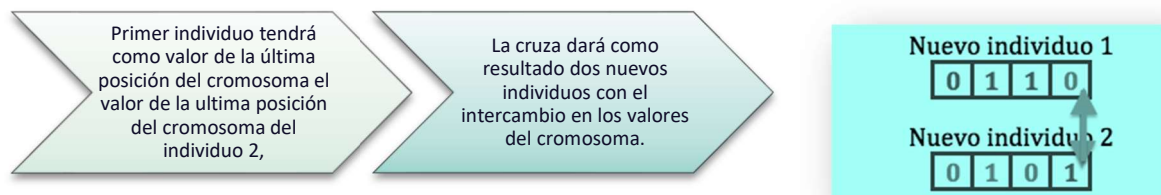
Ahora mediante un método para cruza, deberemos cruzar a los dos padres elegidos

Hay que calcular el "punto de corte" del cromosoma sobre el que se va a hacer el intercambio de alelos. De acuerdo con la longitud de los cromosomas  $L=4$ , la longitud de la cadena tiene  $L-1$  puntos de cruza,  $4-1=3$  puntos de cruza.



No. de punto de corte	Definici�n de los rangos de los L-1 puntos de corte en el cromosoma rango inicial- rango final))	Al generar un n�mero aleatorio entre [0.0-1.00] para elegir el punto de corte, si este es:
Rango Pto. corte1	[0 0.33]	0.85  Entonces el punto de corte es el 3� ya que va cae dentro de ese rango
Rango Pto. corte1	[0.33, 0.66]	
Rango Pto. corte1	[0.66, 1]	

De acuerdo con eso, el resultado de la cr za entre los padre  $x1(7) = 0\ 1\ 1\ 1$  y  $x2(4) = 0\ 1\ 0\ 0$



3. Para cada descendiente, se muta cada bit independientemente con probabilidad  $pm$ . Usualmente  $pm$  se propone entre  $[1/Tp - 1/l]$  donde  $Tp$  es el tama o de la poblaci n y  $l$  es la longitud del cromosoma

a) Elegir al azar el n mero de genes que intercambiar sus alelos de 0 a 1 o de 1 a 0 con base en la longitud de la cadena

- Para todos los individuos, generar n meros aleatorios para cada gen ( $G_i$ ), y cuando se cumpla que  $va_{G_i} \leq pm$ , el alelo del  $G_i$  ser  modificado de 0 a 1 o viceversa, en ese individuo.

**NOTA:** En caso de elegir un tipo de AG estacionario o elitista, es en esta fase cuando se analiza si el individuo es mejor que su padre:

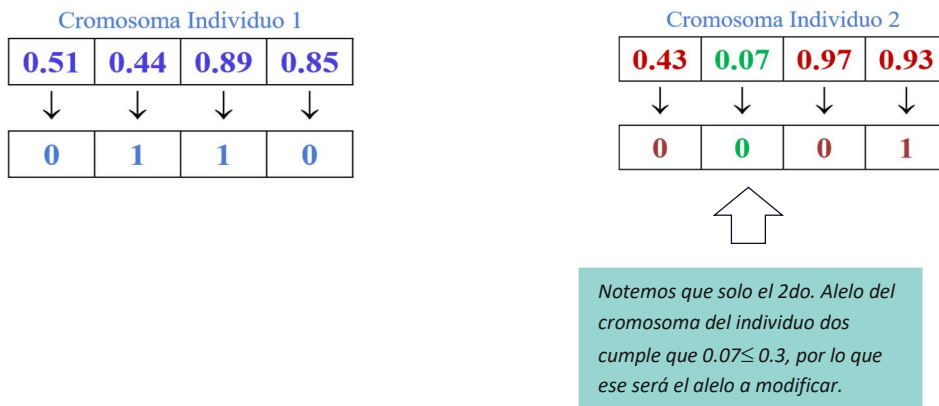
Si  $f(x)_{padre} < f(x)_{hijo} \rightarrow$  El hijo es mejor que el padre y se elimina a este, el cromosoma que pasa a la siguiente generaci n, es la que corresponde al hijo

Sino  $\rightarrow$  el cromosoma que pasa a la siguiente generaci n, es la que corresponde al padre

### Mutación de un punto en uno de los padres elegidos

Consideremos la probabilidad de mutación del 30%,  $pm=0.3$ . Entonces debemos generar números aleatorios para cada alelo en cada cromosoma ( $A_i$ ), y cuando se cumpla que  $va_{A_i} \leq pm$ , el alelo  $A_i$  que cumpla tal condición será modificado de 0 a 1 o viceversa:

Si suponemos que las  $va_{A_i}$  para cada alelo en cada cromosoma es:



Tenemos una nueva generación de individuos creada que es una generación en la cual tenemos un individuo (individuo 2 con  $x=1$ ) que es un individuo "mejor" que los de la generación anterior. Su valor aplicando la función de fitness es de 1,41

Individuo 1 ( $x=6$ ):	$f(6) = ABS \left  \frac{6-5}{2+Sen(6)} \right  = 0.58$
Individuo 2 ( $x=1$ ):	$f(1) = ABS \left  \frac{1-5}{2+Sen(1)} \right  = 1.41$

Estos individuos son los que poblarán la siguiente generación y el proceso es cíclico hasta que lograr la convergencia. En caso de elegir el tipo de algoritmo genético estacionario, se evalúa si los hijos son mejor que los padres para proceder a la extinción del cromosoma débil, eligiendo al cromosoma fuerte para la siguiente generación.

# 1. MECANISMOS DE SELECCIÓN

Un algoritmo genético puede utilizar muchas técnicas diferentes para seleccionar a los individuos que deben copiarse hacia la siguiente generación. Aquellos individuos que compiten por los recursos de forma más efectiva tienen más posibilidades de reproducirse. La selección suele considerar la población en su conjunto y la probabilidad de reproducción se define sobre la generación actual de la población. Se asignan probabilidades de selección a los individuos de la población (padres) en función de su fitness. Sin embargo, se suele emplear un mecanismo de selección estocástico:

- Los mejores individuos es más probable que se seleccionen (aunque nada lo garantiza).
- Incluso el peor individuo de la población puede ser seleccionado.
- La selección estocástica nos ayuda a escapar de óptimos locales.

A continuación, se describen brevemente los más comunes, debiéndose tomar en cuenta que algunos son mutuamente excluyentes, mientras que otros se pueden combinar:

- Selección elitista: se garantiza la selección de los miembros más aptos de cada generación. Aunque en la mayoría de los AGs no se utiliza elitismo puro, más bien se implementa una modificación por la que el individuo mejor, o algunos de los mejores, son copiados hacia la siguiente generación en caso de que no surja nada mejor.
- Selección generacional: la descendencia de los individuos seleccionados en cada generación se convierte en toda la siguiente generación. No se conservan individuos entre las generaciones.
- Selección proporcional a la aptitud: los individuos más aptos tienen más probabilidad de ser seleccionados, aunque no se garantiza.
- Selección por rueda de ruleta: es una forma de selección proporcional a la aptitud, aquí, la probabilidad de que un individuo sea seleccionado es proporcional a la diferencia entre su aptitud y la de sus competidores.
- Selección escalada: al incrementarse la aptitud media de la población, la fuerza de la presión selectiva también aumenta y la función de aptitud se hace más discriminadora. Este método puede ser útil para seleccionar más tarde, cuando todos los individuos tengan una aptitud relativamente alta y sólo les distingan pequeñas diferencias en la aptitud.
- Selección por torneo: se trata de que los individuos compitan entre sí, eligiendo subgrupos de individuos de la población, y los miembros de cada subgrupo compiten entre ellos. Sólo se elige a un individuo de cada subgrupo para la reproducción.

- Selección por rango: a cada individuo de la población se le asigna un rango numérico basado en su aptitud, y la selección se basa en este ranking, en lugar de las diferencias absolutas en aptitud. La ventaja de este método es que puede evitar que individuos muy aptos ganen dominancia al principio a expensas de los menos aptos, lo que reduciría la diversidad genética de la población y podría obstaculizar la búsqueda de una solución aceptable.
- Selección jerárquica: los individuos atraviesan múltiples rondas de selección en cada generación. Las evaluaciones de los primeros niveles son más rápidas y menos discriminatorias, mientras que los que sobreviven hasta niveles más altos son evaluados más rigurosamente. La ventaja de este método es que reduce el tiempo total de cálculo al utilizar una evaluación más rápida y menos selectiva para eliminar a la mayoría de los individuos que se muestran poco o nada prometedores, y sometiendo a una evaluación de aptitud más rigurosa y computacionalmente más costosa sólo a los que sobreviven a esta prueba inicial.

### a. Método de Selección Ruleta

El algoritmo de la Ruleta (de acuerdo a DeJong [137]) es el siguiente:

1. Calcular la aptitud de cada individuos de la lista
2. Medir el desempeño de los individuos
3. Calcular la suma del valor esperado total de los individuos en la población (T).
  - a. Repetir N veces (N es el tamaño de la población)
    - Generar un número aleatorio  $r$  entre 0.0 y T
    - Ciclar a través de los individuos de la población sumando los valores esperados hasta que la suma (Psa) sea mayor o igual a  $r$ .
    - El individuo que haga que esta suma (Psa) exceda el límite es el seleccionado.

Notemos que la selección por rueda de ruleta es más fácil de implementar pero es ruidoso debido a que la tasa de evolución depende de la varianza del fitness en la población.

**Ejemplo:**

Supongamos el problema – adaptado de (Goldberg. 1989) – de encontrar el máximo de la función  $f(x) = x^2$  sobre los enteros  $\{1, 2, \dots, 32\}$ .

Tabla 1. Aplicando el método de selección por la Ruleta:

No. Cadena	Población inicial	Valor x	Aptitud $f(x)=x^2$	Probabilidad $Prob_i$	Ve (Probabilidad acumulada)	Padres elegidos
1	01101	13	169	0.13	0.53	2,4,5,1,2,4
2	11000	24	576	0.24	1.8	
3	01000	8	64	0.08	0.2	
4	10011	19	361	0.19	1.3	
5	10111	23	529	0.23	1.65	
6	01111	15	225	0.15	0.7	
Suma			1924	1.00	6	6
Promedio			320.66	0.16	1	1.8
Max (Mejor)			576	0.24	1.85	

## 2. MECANISMOS DE CRUZA

Combinan la información de los padres para crear nuevos descendientes

- La selección de qué información de los padres se combina es estocástica.
- Muchos descendientes pueden ser peores que los padres (en términos de la función de fitness).
- Se espera que algunos de ellos sean mejores al combinar los elementos de sus genotipos que conducen a la obtención de mejores fenotipos
- Los métodos de cruce suelen aplicarse a la representación binaria, estas son generalizables a alfabetos de cardinalidad mayor.

### Tipos de métodos de cruce o recombinación:

- Basados en las frecuencias de los alelos (votación p-sexual, generalización del cruce uniforme)
- Basados en la segmentación de los padres (cruce diagonal, generalización del cruce en n puntos)
- Basados en operaciones numéricas sobre alelos con valores reales (p.ej. centro de masas)

Veamos el funcionamiento de los métodos de:

Cruza de un punto

Cruza de dos puntos

### a. Cruza de un punto

Propuesta por Holland, fue muy popular, actualmente no suele usarse en la práctica debido a sus inconvenientes.

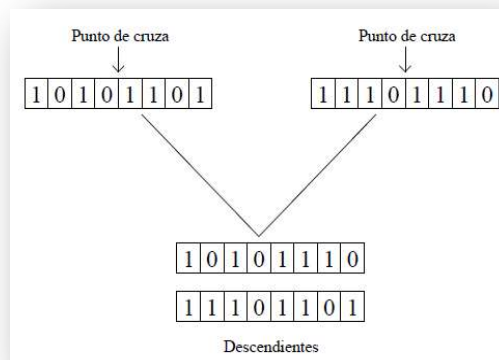
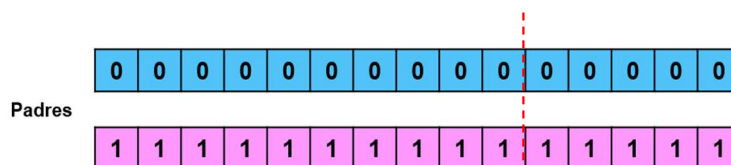


Imagen 2: Cruza de un punto.

### Algoritmo

1. Se selecciona un punto de cruce aleatoriamente, conforme la longitud del cromosoma.
2. Se dividen los padres en ese punto.
3. Se crean hijos intercambiando partes de los cromosomas de los padres, a partir del punto de corte hacia la derecha o final del cromosoma

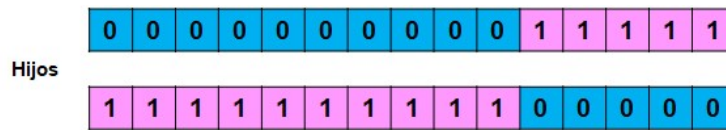
Ej. Para los padres  $P_1$  y  $P_2$ , el punto de cruce elegido es el gen 10, marcado:



→ a partir de este punto, los hijos serán creados combinando los genes de los padres.



Cada padre genera un hijo, quedando de esta forma:



### b. Cruza de dos puntos

Propuesta por DeJong (1975), es una generalización de la cruce de un punto. De Jong, propone elegir más de un punto de cruce debido a que en su investigación, notó que el comportamiento del operador de cruce basado en múltiples puntos no beneficiaba el comportamiento del algoritmo, concluyendo que el cruce basado en dos puntos, representaba una mejora. Por lo tanto, el cruce basado en un punto, puede verse como un caso particular de éste.

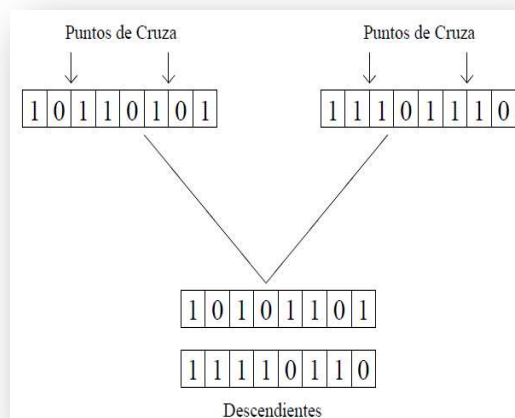
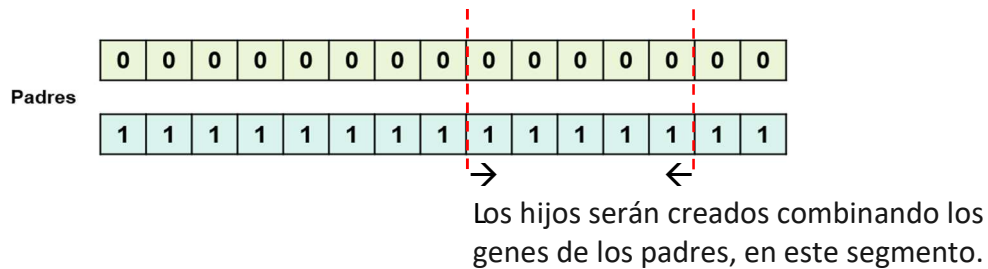


Figura 2: Cruza de dos puntos.

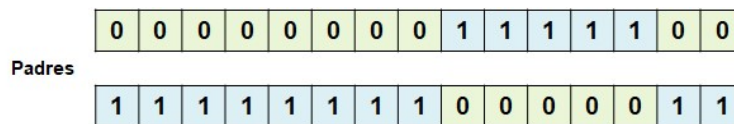
### Algoritmo

1. Se eligen 2 puntos de cruce aleatoriamente, conforme la longitud del cromosoma.
2. Se fragmentan los cromosomas en esos puntos.
3. Se crean los hijos conservando los genes de los padres e intercambiando de forma alternan los genes de los padres que se encuentren dentro del segmento marcado por los puntos de cruce.

Ej. Para los padres  $P_1$  y  $P_2$ , el punto de cruce elegido es el gen 8, marcado:



Cada padre genera un hijo, quedando de esta forma:



### 3. MECANISMOS DE MUTACIÓN

Una vez que se han creado los individuos descendientes, algunos de estos se someterán a un proceso de mutación en el que, cada una de sus posiciones, puede verse modificada con una probabilidad  $p$ . Este paso es importante para añadir diversidad al proceso y evitar que el algoritmo caiga en mínimos locales porque todos los individuos sean demasiado parecidos de una generación a otra. Así, encontramos que existen diferentes estrategias para controlar la magnitud del cambio que puede provocar una mutación:

- ✓ Distribución uniforme: la mutación de la posición  $i$  se consigue sumándole al valor de  $i$  un valor extraído de una distribución uniforme, por ejemplo una entre  $[-1, +1]$ .
- ✓ Distribución normal: la mutación de la posición  $i$  se consigue sumándole al valor de  $i$  un valor extraído de una distribución normal, comúnmente centrada en 0 y con una determinada desviación estándar. Cuanto mayor la desviación estándar, con mayor probabilidad la mutación introducirá cambios grandes.
- ✓ Aleatorio: la mutación de la posición  $i$  se consigue reemplazando el valor de  $i$  por nuevo valor aleatorio dentro del rango permitido para esa variable. Esta estrategia suele conllevar mayores variaciones que las dos anteriores.

Hay que tener en cuenta que, debido a las mutaciones, un valor que inicialmente estaba dentro del rango permitido puede salirse de él. Una forma de evitarlo es: si el valor tras la mutación excede alguno de los límites acotados, se sobrescribe con el valor del límite. Es decir, se permite que los valores se alejen como máximo hasta el límite impuesto. Los métodos para la mutación por permutaciones son una forma de corregir el problema que causa una mutación normal, esta produce permutaciones inválidas ya que se repetirán etiquetas. Estas técnicas tienen que cambiar al menos dos alelos, se emplea la probabilidad de mutación, refiriéndose en estos casos a las chances de modificar el genotipo no a las chances de modificar un gene solamente. La diferencia radical entre el método de mutación por intercambio de bit en un AG estándar es que la probabilidad de mutación ahora se referirá a cromosomas completos, no a genes individuales.

Podemos tener uno o más operadores de mutación para la representación elegida. Sin embargo, algunos aspectos importantes a tener en cuenta al momento de la elección de este en el diseño del AG simple son:

- Debe permitir alcanzar cualquier parte del espacio de búsqueda.
- El tamaño de la mutación debe ser controlado.
- Debe producir cromosomas válidos.
- Se aplica con una probabilidad muy baja de actuación sobre cada descendiente obtenido tras aplicar el operador de cruce (incluidos los descendientes que coinciden con los padres porque el operador de cruce no actúa)

Partiendo de la propuesta por John Holland en 1975, la mutación gaussiana. En este, dado un cromosoma  $x$  con un gen seleccionado para la mutación  $i$ , se le aplica una distribución normal  $N$  de media  $\mu_i$  y desviación estándar  $\sigma$  (parámetro). Dado un cromosoma  $p$  como  $j$ -ésimo de un gen seleccionado para mutación, se produce un cromosoma  $c$  de la siguiente forma:

$$C_i = N\delta_i, \theta \text{ si } j=1; P_i \text{ en caso contrario}$$

Donde  $N\delta_i, \theta$  es una distribución normal con media  $\mu_i$  y desviación estándar  $\theta$  (parámetro). Alternativamente se puede disminuir el valor de  $\theta$  a medida que aumente el número de generaciones.

### a. Mutación discreta (intercambio de bit)

#### Algoritmo

1. Se elige aleatoriamente un porcentaje de la población que mutaran, con probabilidad  $pm$ . Usualmente  $pm$  se propone como  $[1/Tp, 1/l]$  donde  $Tp$  es el tamaño de la población y  $l$  es la longitud del cromosoma.
2. Para el conjunto de individuos elegidos
  - a. Elegir al azar el número de genes que intercambiar sus alelos de 0 a 1 o de 1 a 0 con base en la longitud de la cadena
  - b. Para todos los individuos, generar números aleatorios para cada gen ( $G_i$ ), y cuando se cumpla que  $vaG_i \leq pm$ , el alelo del  $G_i$  será modificado de 0 a 1 o viceversa, en ese individuo.

NOTA: En caso de elegir un tipo de AG estacionario, es en esta fase cuando se analiza si el individuo es mejor que su padre:

- i. Si  $f(x)_{\text{padre}} < f(x)_{\text{hijo}} \rightarrow$  El hijo es mejor que el padre y se elimina a este, el cromosoma que pasa a la siguiente generación, es la que corresponde al hijo
- ii. Sino  $\rightarrow$  el cromosoma que pasa a la siguiente generación, es la que corresponde al padre

## b. Método por permutación

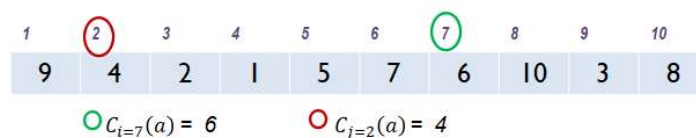
A continuación, se muestran los métodos de mutación por permutación, notemos qué en este modelo, se elige aleatoriamente un porcentaje de la población que mutara con una probabilidad pm. Recordemos que la pm se propone como  $[1/Tp, 1/l]$  donde Tp es el tamaño de la población y l es la longitud del cromosoma. Una vez que se ha elegido los individuos a mutar normalmente se ejecutan dos pasos.

### i. Mutación por inserción

1. Elegir al azar dos alelos, conforme la longitud del cromosoma
2. Mover el segundo a continuación del primero y hacer un shift del resto
  - Notar que esto preserva casi toda la información de adyacencias (orden)

Ejemplo

**Dado:** Si suponemos que elegimos la posición  $i=7$  y decidimos mover ese valor a la posición  $j=2$ ,



Donde:

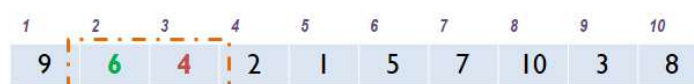
$(a)$  = Valor del alelo en el cromosoma,

$C_{i,j}$  = Posición del gen en el cromosoma, con  $i \neq j$ .

$C_i$  : es la posición o gen, del alelo a mutar (mover de posición)

$C_j$  : es la posición o gen, del alelo en la que se insertará el alelo del gen  $C_i$ .

Tendríamos:



## ii. Mutación por desplazamiento

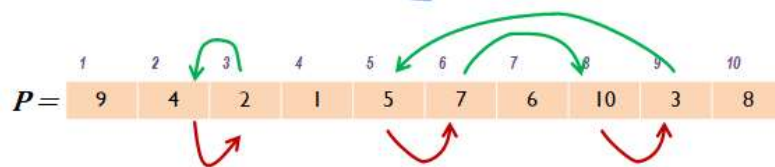
Es una generalización de la mutación por inserción en la que, en vez de mover un solo valor, se cambian de lugar varios a la vez.

1. Elegir al azar  $n$  alelos, conforme la longitud del cromosoma
2. Mover por parejas, el segundo a continuación del primero y hacer un shift del resto

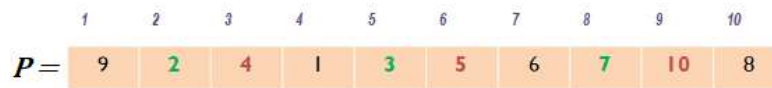
Ejemplo

Si se eligen tres posiciones a mutar; tendremos tres posiciones a mover y tres posiciones a desplazar; suponemos que elegimos:

1er. Alelo a mutar:  $C_{i=3}(\alpha) = 2, C_{j=2}(\alpha) = 4$   
 2do. Alelo a mutar:  $C_{i=9}(\alpha) = 3, C_{j=5}(\alpha) = 5$   
 3er. Alelo a mutar:  $C_{i=6}(\alpha) = 7, C_{j=8}(\alpha) = 10$



Tendríamos:



## 4. Ejemplo: Diseñando un AG simple [Tomado de Goldberg 89']

Ejecutar una iteración, ciclo o generación tomando en cuenta las siguientes consideraciones.

Problema simple		$\max x^2$ sobre $\{0,1,\dots,31\}$
Enfoque AG:	Tamaño de la población	4
	Representación	Código binario $01101 \rightarrow 13$
	Recombinación	Cruza de 1 punto
	Mutación	Cambio a nivel de bit
	Selección de padres	Por Ruleta
	Selección de supervivencia	Generacional

## 1.- Inicialización, evaluación y selección de padres.

No. Cadena	Población inicial	Valor x	Aptitud $f(x)=x^2$	Probabilidad $Prob_i$	Ve	Conteo real
1	0 1 1 0 1	13	169	0.14	0.57	1
2	1 1 0 0 0	24	576	0.49	1.96 $\approx$ 1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Suma			1170	1.00	4.00	4
Promedio			293	0.25	1.00	1
Mejor			576	0.49	1.97	2

## 2.- Cruza y evaluación de la descendencia

No. Cadena	Cruza	Punto de cruce	Descendencia	Valor x	Aptitud $F(x)=x^2$
1	0 1 1 0   1	4	0 1 1 0 0	12	144
2	1 1 0 0   0	4	1 1 0 0 1	25	625
3	1 1   0 0 0	2	1 1 0 1 1	27	729
4	1 0   0 1 1	2	1 0 0 0 0	16	256
Suma					1754
Promedio					439
Max					729

### 3.- Mutación y evaluación de descendencia

No. Cadena	Descendencia después del cruce	Descendencia después de la mutación	Valor x	Aptitud $F(x)=x^2$
1	0 1 1 0 0	1 1 1 0 0	26	626
2	1 1 0 0 1	1 1 0 0 1	25	625
3	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Suma				2354
Promedio				588.5
Max				729

Nota: De acuerdo con los métodos elegidos en este ejemplo, en las tablas 2 y 3, vemos el proceso aumentando dos columnas: valor de x (conversión del cromosoma al fenotipo) y la valoración de la ff para ese valor de x, estos procesos son únicamente para verificar que tan bueno ha sido con respecto a su padre. Este evento es realizado solo en la prueba de escritorio, en la programación, únicamente se desarrolla el proceso de la cruce y se pasa al siguiente proceso de la mutación, una vez realizada la mutación, se pasa a la valoración de la ff porque en este momento si es necesario averiguar si la nueva generación es mejor que los padres, ya que serán los pobladores de la nueva generación, observando quien de estos individuos fue el mejor.

## Fuentes consultadas

- Sancho C. F. Blogs: "Temas selectos". Profr. en el Dpto. Ciencias de la Computación e Inteligencia Artificial, Universidad de Sevilla. Fecha de consulta: Agosto/2018. Disponible en: <http://www.cs.us.es/~fsancho/?e=65>.
- Beasley et al., 1993. Beasley, D., Bull, D. R., and Martin, R. R. (1993). An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, 15(2):58-69.
- Blickle, T. and Thiele, L. (1995). A comparison of selection schemes used in genetic algorithms. Technical Report 11, Computer Engineering and Communication Network Lab (TIK), Gloriastrasse 35, 8092 Zurich, Switzerland.



- Darwin, C. (1859). On the Origin of Species by Means of Natural Selection. John Murray, London.
- Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). Artificial Intelligence through Simulated Evolution. John Wiley & Sons, New York.
- Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor. Republished by the MIT press, 1992.
- Jong, K. A. D. (1975). An analysis of the behavior of a class of genetic adaptive systems. PhD thesis, University of Michigan.
- Koza, J. R. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems). The MIT Press.
- Rechenberg, I. (1973). Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart.
- Tomassini, M. (1995). A survey of genetic algorithms. Annual Reviews of Computational Physics, III:87-118.