

WUOLAH



TEAM_GETPPID__
www.wuolah.com/student/TEAM_GETPPID__



Metahuristica Tema 3 y 4.pdf

Resúmenes Metaheurística



3º Metaheurísticas



Grado en Ingeniería Informática



**Escuela Politécnica Superior de Córdoba
UCO - Universidad de Córdoba**

 **escuela
de negocios**
CÁMARA DE SEVILLA

MÁSTER EN DIRECCIÓN Y GESTIÓN DE RECURSOS HUMANOS

www.mastersevilla.com

Inscríbete



BECAS

Tema 3 y 4: Metaheurísticas basadas en poblaciones. Algoritmos evolutivos

**LA ÚNICA BEBIDA ENERGÉTICA
CON UN GRAN SABOR A COCA-COLA
EXPANDE TU ENERGÍA POSITIVA**



Alto contenido en Cafeína. Ver envase. ©2019 The Coca-Cola Company. Todos los derechos reservados.
COCA-COLA es una marca registrada de The Coca-Cola Company.

Contenido

1. Introducción a la computación evolutiva.....	3
2. Algoritmos genéticos	4
2.1 Modelos: Generacional vs Estacionario	6
2.2 Primeros pasos en la construcción.....	6
2.2.1 Representación.....	6
2.2.2 Inicialización	6
2.2.3 Evaluación.....	7
2.3 Estrategias de selección.....	7
2.3.1 Selección aleatoria.....	7
2.3.2 Selección por torneo	7
2.3.3 Selección por ruleta.....	8
2.3.4 Selección basada en ranking	8
2.3.5 Selección por emparejamiento variado inverso (NAM)	9
2.3 Operadores de cruce	9
2.3.1 Cruce en un punto.....	9
2.3.2 Cruce en n puntos	10
2.3.3 Cruce uniforme.....	10
2.3.4 Cruce aritmético simple (Representación real).....	10
2.3.5 Cruce BLX- α	10
2.3.6 Cruces para permutaciones (Representación ordinal)	11
2.4 Operadores de mutación	12
2.4.1 Mutación clásica	12
2.4.2 Mutación uniforme.....	12

2.4.3 Mutación basada en intercambios y reordenación.....	12
2.4.4 Mutación basada en inserción	12
2.4.5 Mutación basada en inversión	13
2.5 ¿Cruce o mutación?	13
2.5 Estrategias de reemplazo	13
2.6 Consideraciones finales.....	14
2.6.1 Criterio de parada	14
2.6.2 Fases de un algoritmo genético	14
2.6.3 Evolución del fitness.....	14
3. Diversidad vs convergencia.....	15
3.1 Inclusión de diversidad en la evolución	16
3.1.1 Diversidad con la mutación	16
3.1.2 Diversidad con el cruce.....	16
3.1.3 Separación espacial.....	17
3.1.4 Adaptación, autoadaptación, metaevolución.....	17
3.1.5 Estrategias de reemplazamiento	17



LA ÚNICA BEBIDA
ENERGÉTICA CON UN
GRAN SABOR A COCA-COLA
EXPANDE TU ENERGÍA POSITIVA

1. Introducción a la computación evolutiva

En la naturaleza, los procesos evolutivos ocurren cuando se dan las siguientes condiciones:

- Un individuo tiene la habilidad de reproducirse
- Existe una población de individuos que **pueden reproducirse**
- Existe alguna **variedad o habilidad** entre los individuos que se reproducen. Para evolucionar hacia habilidades mejores es necesario que exista variabilidad genética.
- Algunos individuos **sobreviven** en el entorno gracias a esa habilidad. Debe existir una compensación, pues algunas características mejores hacen que otras empeoren.
- El proceso **no tiene memoria**

En el proceso de evolución, el entorno (medio ambiente) juega un papel fundamental y los individuos se van adaptando a dicho entorno.

Podemos entonces definir la computación evolutiva como simulación del proceso evolutivo en un ordenador. Esta técnica de optimización probabilística mejora, con cierta frecuencia, a otros métodos clásicos en problemas difíciles. Se compone de modelos de evolución basados en poblaciones cuyos individuos representan soluciones a problemas.

El pensamiento evolutivo actual gira en torno al Neo-Darwinismo, el cual establece que toda la vida en el planeta puede ser explicada a través de sólo 4 procesos:

- Reproducción
- Mutación
- Competencia
- Selección

Podemos definir cuatro paradigmas básicos:

- **Algoritmos Genéticos.** Utilizan operadores genéticos sobre cromosomas.
- **Estrategias de Evolución.** Enfatizan los cambios de comportamiento al nivel de los individuos.
- **Programación Evolutiva.** Enfatizan los cambios de comportamiento al nivel de las especies.

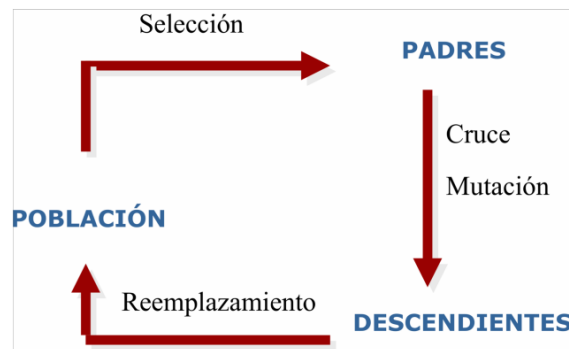
- **Programación Genética.** Evoluciona expresiones que representan programas.

2. Algoritmos genéticos

Son algoritmos de optimización, búsqueda y aprendizaje inspirados en los procesos de evolución natural y evolución genética. El algoritmo genético enfatiza la importancia de la cruce sexual (operador principal) sobre el de la mutación (operador secundario) y usa selección probabilística.

Utilizan los siguientes **mecanismos**:

- Sobreviven los organismos con mejor capacidad dentro de una población
- Secuencias de caracteres para representar el ADN
- Métodos aleatorios para generar la población y su reproducción



Existen una serie de **condiciones** que son necesarias para la evolución:

- Entidad capaz de reproducirse
- Población de dichas entidades
- Variedad en la reproducción
- Diferencias en la habilidad de las entidades que permiten sobrevivir en base al medio ambiente
 - Individuos con mayor éxito tienen mayor probabilidad de reproducirse y de generar un mayor número de descendientes
 - Genes de individuos mejor adaptados se propagarán en sucesivas generaciones

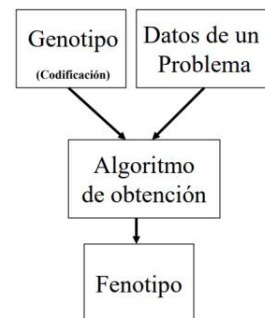
El algoritmo básico es el siguiente:

- Generar (aleatoriamente) una población inicial.
- Calcular la aptitud de cada individuo.
- Seleccionar (probabilísticamente) con base a la aptitud.
- Aplicar operadores genéticos (cruza y mutación) para generar la siguiente población.
- Ciclar hasta que cierta condición se satisfaga.

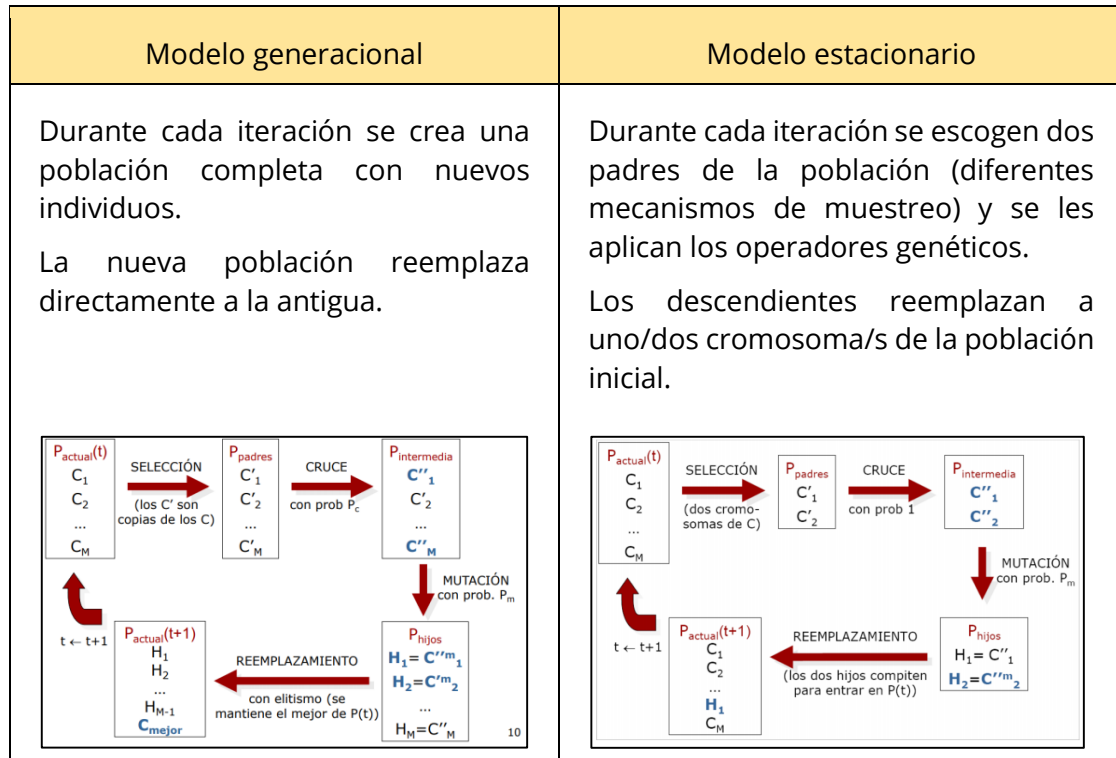
```
Inicio
  t = 0;
  Inicializar P(t);
  Evaluar P(t);
  mientras (no se cumpla la condición de parada) hacer
    Seleccionar P(t+1) desde P(t)
    Cruzar P(t+1)
    Mutar P(t+1)
    Evaluar P(t+1)
    Reemplazar P(t) por P(t+1)
    t=t+1
  fin mientras
fin
```

Para construir un algoritmo genético debemos de seguir los siguientes pasos:

- Establecer un tipo de **representación** (Binaria, real, ordinal...)
- Decidir cómo **inicializar la población** (normalmente un proceso aleatorio)
- Diseñar una correspondencia entre **genotipo y fenotipo**:
 - Algunas veces la obtención del fenotipo a partir del genotipo es un proceso obvio.
 - En otras ocasiones el genotipo puede ser un conjunto de parámetros para algún algoritmo, el cual trabaja sobre los datos de un problema para obtener un fenotipo
- Definir una forma de **evaluar** individuos: Es el paso más costoso (fitness)
- Diseñar un operador de **mutación** adecuado
- Diseñar un operador de **cruce** adecuado
- Decidir un operador de **selección** de padres
- Establecer un **reemplazo** de individuos
- Definir una **condición de parada**



2.1 Modelos: Generacional vs Estacionario

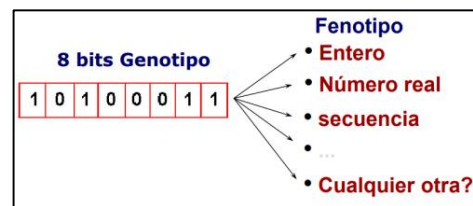


El modelo estacionario es **elitista** (No reemplazará al mejor cromosoma de la población), sin embargo, el modelo generacional puede serlo o no. Además produce una **presión selectiva alta** (convergencia rápida) cuando se reemplazan los peores cromosomas de la población.

2.2 Primeros pasos en la construcción

2.2.1 Representación

Debemos disponer de un mecanismo para codificar un individuo como un genotipo. Los algoritmos genéticos han sido generalmente representados con codificación binaria, aunque también hay: Ordinal, Entera y Real.



Es importante saber diferenciar entre un genotipo y un fenotipo.

2.2.2 Inicialización

Debemos establecer claramente cómo vamos a inicializar la población.

- Uniforme sobre el espacio de búsqueda ... (si es posible)
 - Cadena binaria: 0 ó 1 con probabilidad 0.5
 - Representación real: uniforme sobre un intervalo dado (para valores acotados)
- Elegir la población a partir de los resultados de una heurística previa.



**LA ÚNICA BEBIDA
ENERGÉTICA CON UN
GRAN SABOR A COCA-COLA**
EXPANDE TU ENERGÍA POSITIVA

2.2.3 Evaluación

Representación de la aptitud de cada uno de los individuos.

- Este es el paso más costoso para una aplicación real
- Puede ser una subrutina, un simulador, o cualquier proceso externo (ej. Experimentos en un robot, ...)
- Se pueden utilizar funciones aproximadas para reducir el costo de evaluación.
- Cuando hay restricciones, éstas se pueden introducir en el costo como penalización.
- Con múltiples objetivos se busca una solución de compromiso.

2.3 Estrategias de selección

Debemos de garantizar que los mejores individuos tienen una mayor posibilidad de ser padres (reproducirse) frente a los individuos menos buenos.

Además, también debemos de ser cuidadosos para dar una oportunidad de reproducirse a los individuos menos buenos. Éstos pueden incluir material genético útil en el proceso de reproducción.

Esta idea nos define la presión selectiva que determina en qué grado la reproducción está dirigida por los mejores individuos.

2.3.1 Selección aleatoria

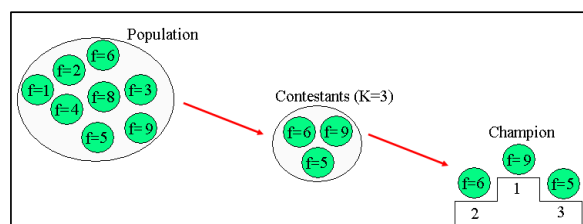
Se escogen padres de manera aleatoria de entre la población:

- Con reemplazamiento
- Sin reemplazamiento

2.3.2 Selección por torneo

Para cada padre a seleccionar:

- **Escoger aleatoriamente k individuos**, con reemplazamiento
- **Seleccionar el mejor** de ellos k se denomina tamaño del torneo. A mayor k, mayor presión selectiva y viceversa. (Si tomamos $k=1$, es una selección aleatoria)



Dependiendo del valor de k que escojamos podemos tener:

- **Torneos grandes:** hay una presión selectiva elevada y los peores individuos apenas tienen oportunidades de reproducirse
 - **Caso particular:** elitismo global, donde participan todos los individuos y la selección se vuelve totalmente determinista
- **Torneos pequeños:** la presión selectiva disminuye y los peores individuos tienen más oportunidades de ser seleccionados

2.3.3 Selección por ruleta

Es uno de los métodos más utilizados. Se asigna una probabilidad de selección proporcional al valor del fitness del cromosoma. La suma de todas las probabilidades es igual a la unidad. Por ende, los mejores individuos tendrán una probabilidad mayor que los peores. En términos matemáticos, la probabilidad asociada a la selección de N individuos es:

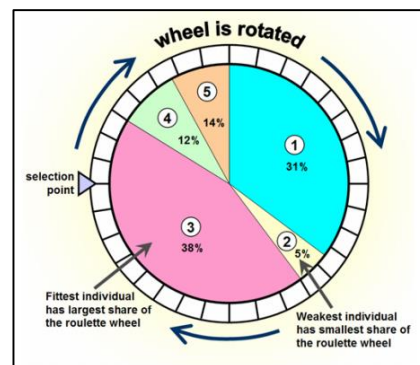
$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}$$

Donde:

- p_i : Probabilidad asociada a la selección de un individuo i
- f_i : fitness del individuo i

Generalmente la población está ordenada en base a la función de aptitud por lo que las proporciones más grandes están al inicio de la ruleta.

La selección se realiza con un aleatorio en el intervalo [0, 1] y devuelve un individuo en esa posición de la ruleta.



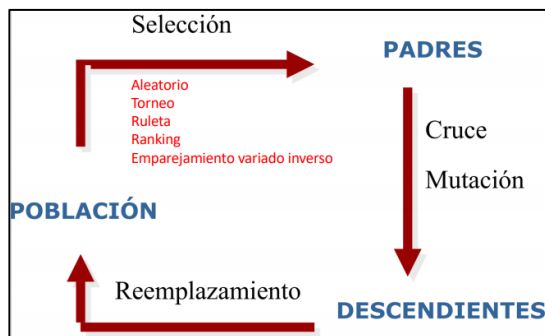
2.3.4 Selección basada en ranking

La población se ordena en función de su fitness y se asocia una probabilidad de selección a cada individuo que depende de su orden.

- Puede evitar una convergencia prematura
- Puede ser computacionalmente costosa (ordenar la población)
- El individuo peor tiene la posición 1 y el mejor la posición N
- Se aplica una selección por ruleta normal, pero aplicada a los rankings

2.3.5 Selección por emparejamiento variado inverso (NAM)

- Se selecciona aleatoriamente un padre
- Para el otro padre, se seleccionan N padres y se escoge el más lejano al primero
- Está orientado a generar diversidad



Ya hemos definido el proceso de selección, en nuestro esquema del algoritmo genético por lo que ahora tocan los operadores de cruce y mutación.

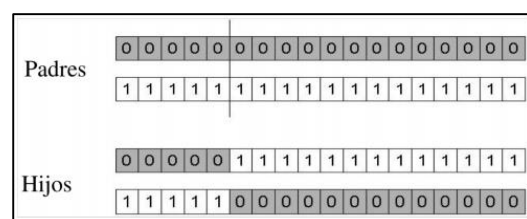
2.3 Operadores de cruce

Podríamos tener uno o más operadores de cruce para nuestra representación. Algunos aspectos importantes para tener en cuenta son:

- Los hijos deberían heredar algunas características de cada padre. Si éste no es el caso, entonces estamos ante un operador de mutación.
- Se debe diseñar de acuerdo con la representación.
- La recombinación debe producir cromosomas válidos.

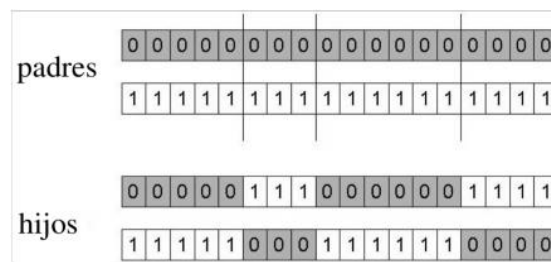
2.3.1 Cruce en un punto

- Seleccionar un punto al azar en ambos padres
- Cortar los padres en este punto
- Crear hijos intercambiando las colas
- Probabilidad de cruce típica en el rango (0.6, 0.9)



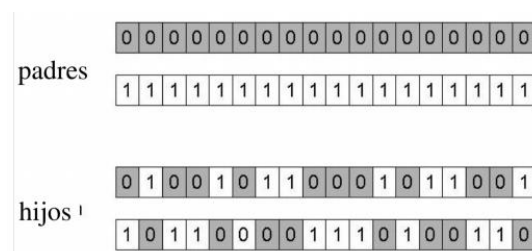
2.3.2 Cruce en n puntos

- Seleccionar n puntos al azar en ambos padres
- Cortar los padres en este punto
- Pegar las partes alternando entre los padres
- Generalización del cruce en 1 punto



2.3.3 Cruce uniforme

- Lanzar un aleatorio $[0, 1]$ para cada gen de los padres
- En base a dicho aleatorio, tomar el gen de uno u otro padre
- El otro hijo será el inverso



2.3.4 Cruce aritmético simple (Representación real)

- El gen de la descendencia toma el valor medio de los genes de los padres
- Tiene la desventaja de que únicamente se genera un descendiente

a b c d e f

A B C D E F

$(a+A)/2$ $(b+B)/2$ $(c+C)/2$ $(d+D)/2$ $(e+E)/2$ $(f+F)/2$

2.3.5 Cruce BLX- α

Dados 2 cromosomas

$$C_1 = (c_{11}, \dots, c_{1n}) \text{ y } C_2 = (c_{21}, \dots, c_{2n}) ,$$

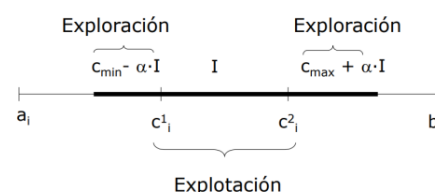
BLX- α genera dos descendientes

$$H_k = (h_{k1}, \dots, h_{kn}) , k = 1, 2$$

donde h_{ki} se genera aleatoriamente en el intervalo:

$$[C_{\min} - I \cdot \alpha, C_{\max} + I \cdot \alpha]$$

- $C_{\max} = \max \{c_{1i}, c_{2i}\}$
- $C_{\min} = \min \{c_{1i}, c_{2i}\}$
- $I = C_{\max} - C_{\min} , \alpha \in [0, 1]$





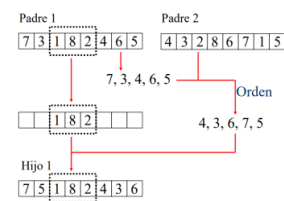
**LA ÚNICA BEBIDA
ENERGÉTICA CON UN
GRAN SABOR A COCA-COLA**
EXPANDE TU ENERGÍA POSITIVA

2.3.6 Cruces para permutaciones (Representación ordinal)

Los cruces estándar generalmente conllevan a soluciones inadmisibles en este tipo de representaciones, por lo que se plantean los cruces OX1 y OX2.

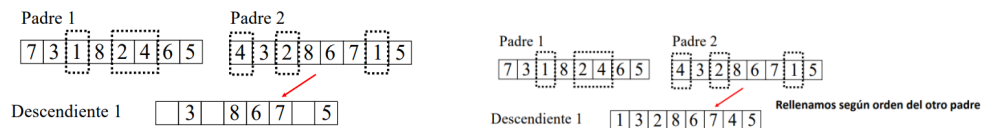
2.3.6.1 OX1

El **OX1** construye descendientes escogiendo una serie de genes de un padre y preservando el orden relativo de los genes del otro padre.



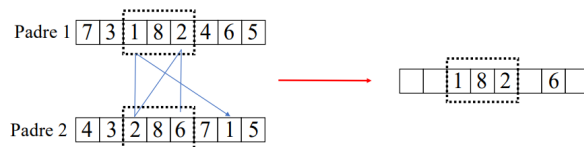
2.3.6.2 OX2

El **OX2** es una modificación del OX1 en el que se escogen al azar varios genes de uno de los padres para imponer en el otro padre el orden de los elementos en las posiciones seleccionadas.



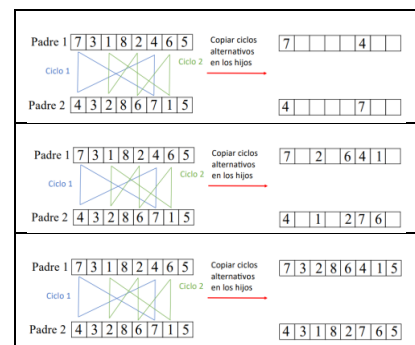
2.3.6.3 PMX

En él, una parte de los genes representa a uno de los padres, se hace corresponder con una parte, de igual tamaño, de los genes del otro padre, intercambiándose la información restante.



2.3.6.4 CX

Crea un descendiente a partir de los padres, de tal manera que cada posición se ocupa por el correspondiente elemento de uno de los padres.



2.4 Operadores de mutación

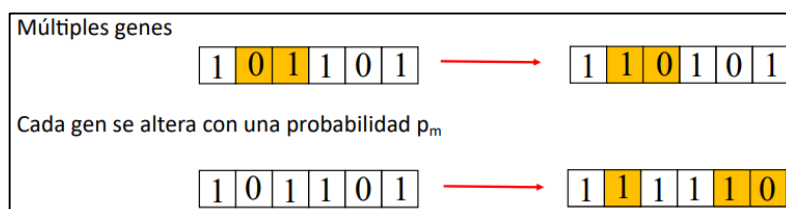
Podría tenerse uno o más operadores de mutación para una representación.

Algunos aspectos importantes a tener en cuenta 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).

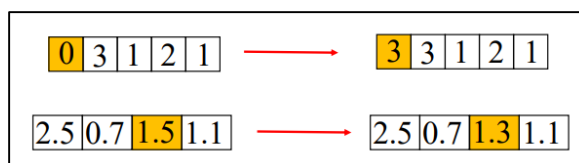
2.4.1 Mutación clásica

Se selecciona uno o varios genes y se cambian sus valores.



2.4.2 Mutación uniforme

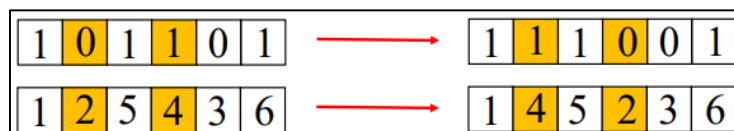
Se selecciona un gen y su valor (entero o real) es escogido aleatoriamente entre los valores de un rango.



2.4.3 Mutación basada en intercambios y reordenación

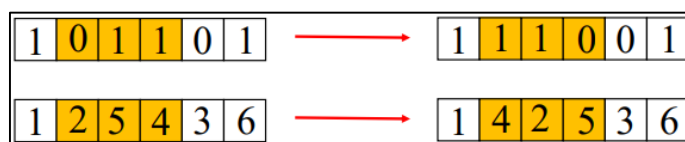
2.4.3.1 Intercambios

Se seleccionan dos genes y se intercambian sus valores



2.4.3.2 Reordenación

Se selecciona un subconjunto de genes y sus valores son reordenados de manera aleatoria



2.4.4 Mutación basada en inserción

Selecciona aleatoriamente un gen del $geno1po$ y lo inserta en otro lugar al azar, pudiendo realizarse esto también sobre conjuntos de genes.

2.4.5 Mutación basada en inversión

Selecciona un conjunto de genes del genotipo y los inserta de manera inversa, pudiendo variar la posición en la que se insertan.

2.5 ¿Cruce o mutación?

Depende del problema que se desea resolver pero, en general, suele ser bueno utilizar ambos operadores. Se puede utilizar un algoritmo que sólo emplee mutación, pero no un algoritmo que sólo use cruce.

- **Cruce:** es un método para compartir información entre cromosomas que juega un papel central en los algoritmos genéticos ya que explota la información disponible de muestras previas para influir en futuras búsquedas. Sólo el cruce puede combinar lo mejor de dos padres
- **Mutación:** es un método para explorar el espacio de búsqueda con el objetivo de descubrir áreas prometedoras lejos de los padres. Sólo la mutación puede introducir nuevos genes en la población.

2.5 Estrategias de reemplazo

La presión selectiva se ve también afectada por la forma en que los cromosomas de la población son reemplazados por los nuevos descendientes. Podemos utilizar métodos de reemplazamiento aleatorios, o determinísticos.

El elitismo se aconseja en modelos generacionales para no perder la mejor solución.

En los modelos estacionarios se pueden ver diferentes propuestas de reemplazo:

- **Reemplazar al peor de la población:** genera alta presión selectiva
- **Torneo restringido:** se reemplaza al más parecido de entre N individuos. Mantiene una cierta diversidad
- **Peor entre semejantes:** se parte de un descendiente generado y éste reemplaza al peor individuo de un conjunto de los N más parecidos (padres o individuos de la población anterior). Busca un equilibrio entre diversidad y presión selectiva
- **Crowding determinístico:** el hijo reemplaza a su padre más parecido. Mantiene diversidad.

El esquema final de un algoritmo genético sería el siguiente:



2.6 Consideraciones finales

2.6.1 Criterio de parada

Podemos establecer como criterio de parada diferentes elementos como por ejemplo: Al alcanzar el óptimo, limitaciones de recursos (Con un número máx de iteraciones o generaciones), o basado en la evolución, es decir, cuando no mejora después de x iteraciones.

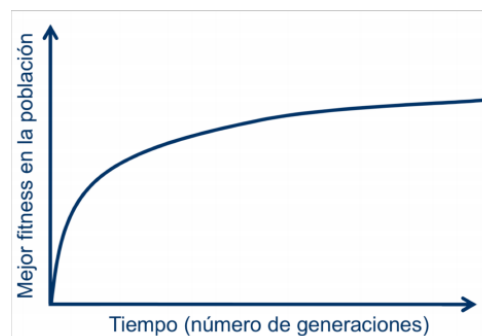
2.6.2 Fases de un algoritmo genético

Se divide en 3 fases distintas:

- **Generaciones iniciales:** Distribución aleatoria de la población
- **Generaciones intermedias:** Población cerca de las colinas
- **Generaciones finales:** Población concentrada en las cimas de las colinas más altas

2.6.3 Evolución del fitness

La evolución típica del fitness tiende a mejorar considerablemente e ir estabilizándose hasta llegar a un punto en el que la mejora sea insignificante.



Otros detalles:

- Nunca sacar conclusiones de una única ejecución. Utilizar medidas estadísticas (medias, medianas, etc.) con un número suficiente de ejecuciones independientes
- No se debe comprobar la actuación de un algoritmo sobre ejemplos simples si se desea trabajar con casos reales
- Diferentes enfoques/diseños:
 - Encontrar una solución muy buena al menos una vez
 - Encontrar al menos una solución muy buena en cada ejecución



LA ÚNICA BEBIDA
ENERGÉTICA CON UN
GRAN SABOR A COCA-COLA
EXPANDE TU ENERGÍA POSITIVA

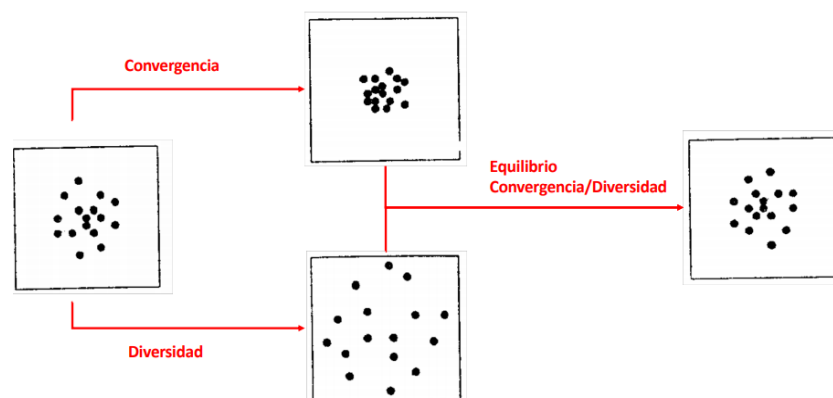
3. Diversidad vs convergencia

Estos dos factores son contrapuestos, por lo que es necesario establecer un equilibrio entre ambos:

- Exploración del espacio de soluciones, para realizar una búsqueda en amplitud, localizando así zonas prometedoras.
- Explotación espacio de búsqueda, para hacer una búsqueda en profundidad en dichas zonas, obteniendo así las mejores soluciones.

Dos factores contrapuestos influyen sobre la efectividad de un algoritmo genético:

- **Convergencia:** Centrar la búsqueda en regiones prometedoras mediante presión selectiva
 - Presión selectiva: permite que los mejores individuos sean seleccionados para reproducirse. Es necesaria para que el proceso de búsqueda no sea aleatorio
- **Diversidad:** Evitar la convergencia prematura (rápida convergencia hacia zonas que no contienen el óptimo global). Está asociada a las diferencias entre los individuos de la población. La falta de diversidad genética significa que todos los individuos en la población son parecidos (Convergencia hacia óptimos locales).
 - Se aplican técnicas de inclusión de mecanismos de diversidad y reinicialización para evitar convergencias prematuras.



3.1 Inclusión de diversidad en la evolución

3.1.1 Diversidad con la mutación

Para ello debemos de adaptar la probabilidad de mutación, reduciéndola durante la ejecución: 1º Diversidad 2º Convergencia. También aplicando probabilidades altas sobre soluciones malas y pequeñas sobre las buenas.

3.1.2 Diversidad con el cruce

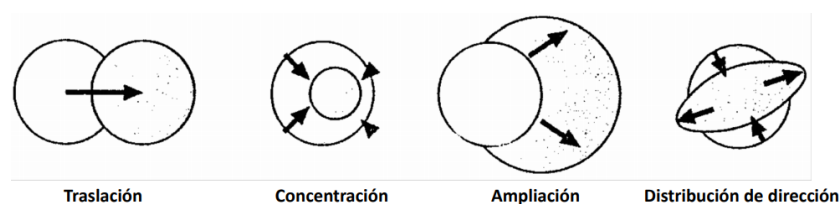
3.1.2.1 Técnicas de emparejamiento

Los padres se pueden seleccionar de forma que se mantenga la diversidad de la población

- **Prohibición de cruce basada en ascendencia.** Un individuo no puede emparejarse con él mismo, ni con sus padres, ni con sus hijos, ni con sus hermanos
- **Prohibición de incesto.** Dos padres se cruzan si su distancia Hamming está por encima de cierto umbral
- **Emparejamiento variado.** Un cromosoma se cruza con otro que es bastante diferente

3.1.2.2 Técnicas para generar los hijos

Es importante establecer un equilibrio entre los siguientes atributos.



3.1.2.3 Número de padres

Aumentar el número de padres en el cruce ayuda a generar una mayor diversidad.



3.1.2.4 Número de hijos

Aumentar el número de hijos en el cruce ayuda a generar una mayor diversidad.



3.1.3 Separación espacial

Este tipo de métodos se proponen para evitar la convergencia prematura. Los más representativos son:

- **Algoritmos genéticos distribuidos:** La población se divide en subpoblaciones que son tratadas como islas independientes en cada una de las cuales se ejecuta una instancia distinta del algoritmo. Estas instancias se comunican entre sí mediante unas cadenas de conexión.
- **Algoritmos genéticos celulares:** Una única instancia del algoritmo genético opera sobre la población en forma de vecindarios; es decir, para cada individuo, realiza las operaciones correspondientes entre él y sus vecinos.

3.1.4 Adaptación, autoadaptación, metaevolución

3.1.4.1 Adaptación

Ajuste de determinados elementos (parámetros o componentes) de los algoritmos genéticos a lo largo de la ejecución en función de su estado o información disponible sobre el espacio de búsqueda (Funciones de evaluación, operadores genéticos, representación...)

3.1.4.2 Autoadaptación

Evolución de los parámetros de acuerdo con el comportamiento del algoritmo.

3.1.4.3 Meta-evolución

Considera la búsqueda del mejor AG para resolver un problema como un problema de optimización y se utiliza otro AG para resolverlo.

3.1.5 Estrategias de reemplazamiento

Introducir mayor diversidad o convergencia en la población en función del mecanismo de reemplazamiento de los cromosomas de la población en curso por los descendientes.