

P1.1.- FUNDAMENTOS TEÓRICOS (en TEMA 4)

1

Introducción

Métodos de búsqueda local y problemas de optimización

Búsqueda por ascensión de colinas

Búsqueda por Temple Simulado

Búsqueda por haz local

Búsqueda por Algoritmos genéticos

- Codificación
- Operadores genéticos
- Algoritmo Genético

Búsqueda entre adversarios

Juegos. Decisiones óptimas en juegos

- Estrategia óptima. Minimax
- Decisiones óptimas en juegos multi-jugador
- Minimax-alfa-beta (Poda alfa-beta)

Decisiones en tiempo real imperfectas. Heurísticas

- Funciones de evaluación
- Corte de la búsqueda

Juegos que incluyen un elemento de posibilidad



Práctica 1

Sistemas de resolución de problemas en Espacio de Estados

2

Elementos principales de los Sistemas de Resolución de Problemas:

- **Representación:** “*describe*” el dominio del problema, el objetivo final que se desea alcanzar y la situación inicial.
- **Operadores:** “*transforman*” la situación del problema, o dividen un problema en varios sub-problemas (de solución más sencilla).
- **Estrategia de Control:** “*selecciona*” el operador a aplicar en cada situación del problema, de entre todos los aplicables. Al aplicar un operador, la situación actual del problema se transforma de acuerdo al sentido característico de ese operador.

Uno de los métodos de representación de problemas está basado en la formulación en **Espacio de Estados**.

- Estados: representación completa de la situación del problema en un momento dado (contiene la información relevante).
- Operadores: acciones que pueden transformar un estado en otro.

Un elemento importante en la estrategia de control, desde el punto de vista de los sistemas inteligentes, es el uso de “heurísticas”. Un tipo particular de heurística es la **Función Heurística** la cual estima, en una formulación mediante estados, la proximidad de un estado a un estado objetivo.

Relacionado con la estrategia de control tenemos el método de búsqueda entendido como el proceso para examinar y encontrar qué secuencia de acciones permite obtener un estado objetivo. Un procedimiento de búsqueda utiliza como entrada el problema y devuelve una secuencia de acciones que conduce a la solución. En la estrategia de control se suelen definir distintos criterios que hacen característicos los distintos métodos de búsqueda.

Métodos de Búsqueda: Si no importa el camino al objetivo, podemos considerar los **métodos de búsqueda local**. Estos tienen dos ventajas clave:

- (1) usan muy poca memoria - por lo general una cantidad constante; y
- (2) pueden encontrar soluciones razonables en espacios grandes o infinitos (continuos) donde los algoritmos sistemáticos son inadecuados.

Muy útiles en problemas de optimización puros (encontrar el mejor estado según una función objetivo).

En particular, los métodos de búsqueda por haz local guardan la pista de k estados. Variante interesante: búsqueda de haz estocástica o “Algoritmos Genéticos” - los estados sucesores se generan combinando dos estados, más que modificar un solo estado. Este método, además de utilizar una función heurística para estimar la proximidad de un estado a un objetivo, puede entenderse como una técnica heurística ya que define de manera heurística los pasos para identificar soluciones de alta calidad con pocos recursos aunque no podamos garantizar encontrar la solución.

BÚSQUEDA POR ALGORITMOS GENÉTICOS (AGs)

Los elementos que utiliza un AG son los siguientes:

- ✓ Cada estado (individuo o fenotipo) está codificado como una cadena sobre un alfabeto finito (la codificación de un estado se denomina cromosoma o genotipo).
- ✓ Cada elemento de un cromosoma es denominado gen.
- ✓ El valor heurístico de un estado es denominado fitness o idoneidad. La función fitness mide la calidad (respecto a alguna característica) de los estados.
- ✓ Se necesita de un conjunto de operadores (denominados operadores genéticos) que permitan generar nuevos estados.

La codificación define el tamaño del espacio de búsqueda y el tipo de operadores.

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos

5

- **Codificación**

El AG codifica la información de cada estado en una cadena que llamamos cromosoma. Existen tipos de cadenas que se han utilizado con éxito en el diseño de un AG, aunque podemos diseñar cadenas más acordes o adecuadas a algunos tipos de problemas. Entre las cadenas más utilizadas, nos encontramos con cadenas de 0s y 1s, cadenas de enteros, cadenas de reales, etc.

Codificación binaria

1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

gen

Por ejemplo, puede representar al número:

$$1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 163$$

ó $n^0 \in [2.5, 20.5] \rightarrow 2.5 + (163/255) \times (20.5 - 2.5) = 14.0058$

Codificación entera

1	2	6	8	2	3	1	4
---	---	---	---	---	---	---	---

Entre otros, se utiliza en problemas de localización

Codificación de orden

7	3	6	8	2	4	1	5
---	---	---	---	---	---	---	---

Los individuos se representan como permutaciones. Entre otros, se utiliza para problemas de secuenciación

Codificación real

Una forma natural de codificar una solución es utilizando valores reales como genes. Muchas aplicaciones tienen esta forma natural de codificación.

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos

6

- **Operadores genéticos**

El AG dispone de 3 operadores genéticos básicos:

Selección: encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no. Habrá individuos que aparezcan más de una vez e individuos que no aparezcan.

Cruce: una vez seleccionados los individuos, éstos son recombinados para producir la descendencia que se insertará en la siguiente generación.

Mutación: la mutación provoca que algunos de los genes de un individuo varíe su valor.

A la hora de definir estos operadores genéticos debemos tener en cuenta que ellos son dependientes del tipo de codificación utilizada. Por tanto, existen operadores ya definidos para las codificaciones más utilizadas (codificación binaria, entera, de orden, real, etc). Si diseñamos nuestro tipo particular de codificación, es necesario definir los operadores genéticos adecuados.

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos

7

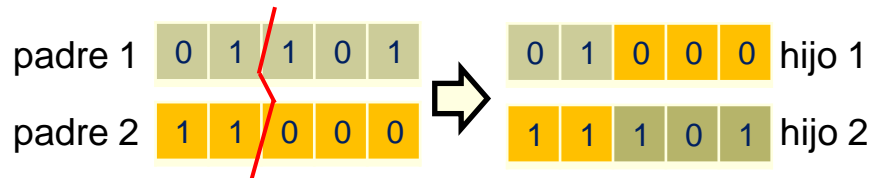
Algunos de los operadores genéticos utilizados son los siguientes:

Selección:

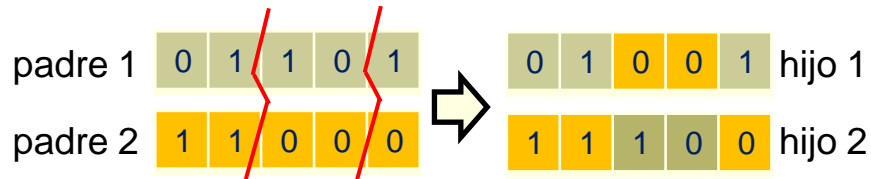
- Ruleta: Se eligen con probabilidad proporcional a su función de idoneidad.
- Torneo: Se establecen k *torneos* aleatorios entre parejas de individuos y se eligen los que ganan en cada torneo (mejor función idoneidad).

Cruce:

- Cruce por un punto.

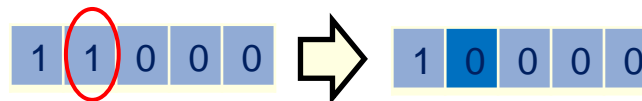


- Cruce por dos puntos.

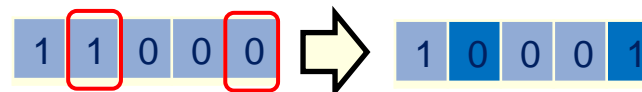


Mutación:

- Cambio de un gen aleatorio.



- Intercambio entre dos genes.



Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos

8

- **Algoritmo Genético**

Esquemáticamente, el proceso de un Algoritmo Genético (AG) es el siguiente:

Se comienza con un conjunto (población) de k estados generados aleatoriamente.
Iterar hasta que la población converge o alcanza una condición de parada.

1. Se seleccionan k individuos de la población actual para crear la población intermedia.
2. Se eligen los individuos con una probabilidad (p_c) para ser cruzados.
Se emparejan, y para cada pareja se aplica el operador de cruce.
Se obtienen los nuevos individuos, que sustituyen a los padres.
3. Con una probabilidad (p_m) se mutan los genes de los individuos de la población actual.
4. Esta nueva población obtenida (mediante selección, cruce y mutación) forman la población inicial de la siguiente generación.

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos

9



función Algoritmo Genético

$t = 0$

inicializar $P(t)$

evaluar $P(t)$

mientras (no condición de parada) **hacer**

$t = t + 1$

seleccionar $P(t)$ desde $P(t-1)$

cruzar $P(t)$

mutación $P(t)$

evaluar $P(t)$

Un algoritmo genético más detallado (con el operador de cruce por un punto y la mutación de un gen), es el siguiente:

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos

10

función ALGORITMO-GENÉTICO(población, IDONEIDAD) **devuelve** un individuo

entradas: población; función idoneidad

repetir

nueva_población \leftarrow conjunto vacío

nueva_población \leftarrow SELECCIÓN(población, IDONEIDAD)

población_cruzar \leftarrow conjunto vacío

para x desde 1 **hasta** TAMAÑO(nueva_población) **hacer**

si ($r < p_c$) **entonces** población_cruzar \leftarrow población_cruzar + x

para i desde 1 **hasta** TAMAÑO(población_cruzar) **hacer**

 x, y \leftarrow elegir aleatoriamente 2 individuos de población_cruzar

 hijo₁, hijo₂ \leftarrow CRUCE(x, y)

 sustituir x e y por hijo₁ e hijo₂ en nueva_población

para x desde 1 **hasta** TAMAÑO(nueva_población) \times LONGITUD(individuo) **hacer**

si ($r < p_m$) **entonces** gen \leftarrow MUTAR(gen) y modificar gen del hijo de nueva_población

población \leftarrow nueva_población

hasta que algún individuo es bastante adecuado, o ha pasado bastante tiempo

devolver el mejor individuo en la población, de acuerdo con la IDONEIDAD

función CRUCE(x, y) **devuelve** un individuo

entradas: x, y, padres individuales

 n \leftarrow LONGITUD(x)

 c \leftarrow número aleatorio de 1 a n

devolver individuo₁ \leftarrow AÑADIR(SUBCADENA(x, 1, c), SUBCADENA(y, c+1, n))

 individuo₂ \leftarrow AÑADIR(SUBCADENA(y, 1, c), SUBCADENA(x, c+1, n))

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos

11

Por tanto, para construir un sistema basado en un AG necesitamos realizar:

■ El diseño y construcción del AG.

1. Diseñar una representación.
2. Decidir cómo inicializar una población.
3. Diseñar una correspondencia entre genotipo y fenotipo.
4. Diseñar una forma de evaluar un individuo.
5. Decidir cómo seleccionar los individuos para ser padres.
6. Diseñar un operador de cruce adecuado.
7. Diseñar un operador de mutación adecuado.
8. Decidir la condición de parada.

DEPENDE DEL
PROBLEMA

COMPONENTES
DEL ALGORITMO

■ El ajuste de parámetros.

- a. Una vez diseñado y construido el AG, es necesario, y útil, ajustar sus parámetros al problema concreto al que se le aplicará.

■ Validación del sistema.

Veámoslos con más detenimiento guiados por el problema de las n-reinas.

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Diseño

12

1. Diseñar una representación

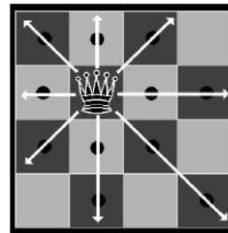
Debemos disponer de un mecanismo para codificar un individuo como un cromosoma (genotipo). Existen muchas maneras de hacer esto y se ha de elegir la más relevante para el problema en cuestión.

IMPORTANTE:

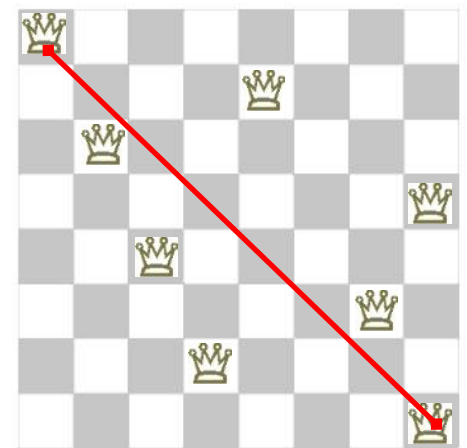
Una vez elegida una representación, debemos de tener en cuenta cómo evaluar los cromosomas y qué operadores genéticos utilizar.

Ejemplo: Problema de las n reinas: Colocar n reinas en un tablero de ajedrez $n \times n$ sin que se amenacen.

- Posibles movimientos de una reina \rightarrow 8 direcciones



- Una reina amenaza a las fichas que se encuentren en su misma fila, columna o diagonal



Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Diseño

13

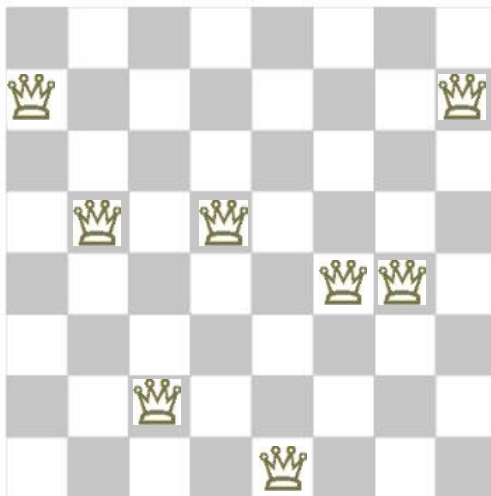
Ejemplo: Problema de las n reinas

- Codificación binaria de un estado: Se expresa la posición (fila) en la que está la reina de cada columna
- Se basa en que sólo puede haber una reina en cada columna

Individuo (para el caso de las 8-reinas)

Col. 1			Col. 2			Col. 3			Col. 4			Col. 5			Col. 6			Col. 7			Col. 8		
0	0	1	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	0	0	0	0	1

1	000
2	001
3	010
4	011
5	100
6	101
7	110
8	111



No es estado final

Se necesitan $8 \times \log_2 8 = 24$ bits

- Para abreviar, podemos expresarlo en decimal (representación entera):

2	4	7	4	8	5	5	2
---	---	---	---	---	---	---	---

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Diseño

14

2. Inicialización

Uniforme sobre el espacio de búsqueda ... (si es posible)

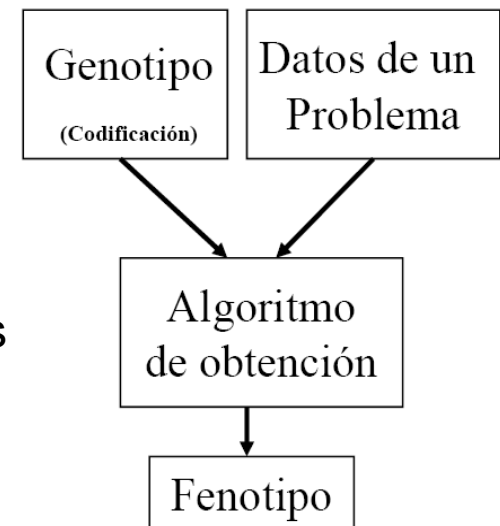
- Cadena binaria: 0 ó 1 con probabilidad 0.5.
- Representación entera: uniforme sobre un conjunto de números enteros.
- Representación real: uniforme sobre un intervalo dado (para valores acotados)

Elegir la inicialización a partir de los resultados de una heurística previa.

3. 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.



Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Diseño

15

4. Evaluación de un individuo (función idoneidad o fitness)

Este es el paso más costoso para una aplicación real. Se debe definir una función que asigne una medida de calidad a cada individuo (estado)

- 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

Ejemplo: Problema de las n reinas

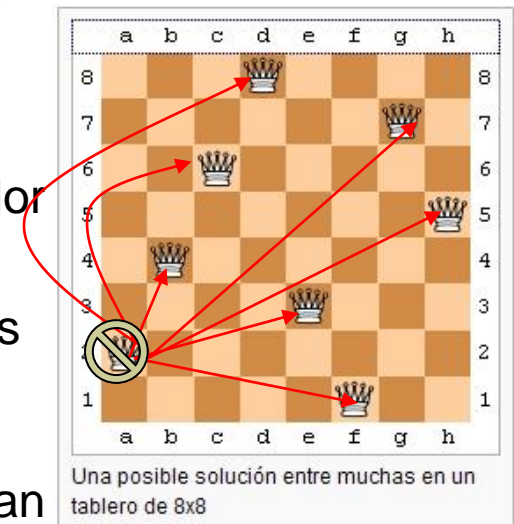
Función idoneidad: N° de pares de reinas no atacadas

Para una solución (caso de las 8-reinas), se tiene el valor máximo: 28 pares ($7+6+5+4+3+2+1 = 28$ pares)

Para un individuo, se resta a 28 el n° de pares atacados

2	4	7	4	8	5	5	2
---	---	---	---	---	---	---	---

Idoneidad = $28 - 4 = 24$, pues hay 4 pares que se atacan



Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Diseño

16

¿CÓMO CONSTRUIMOS UN AG?

Operadores genéticos:

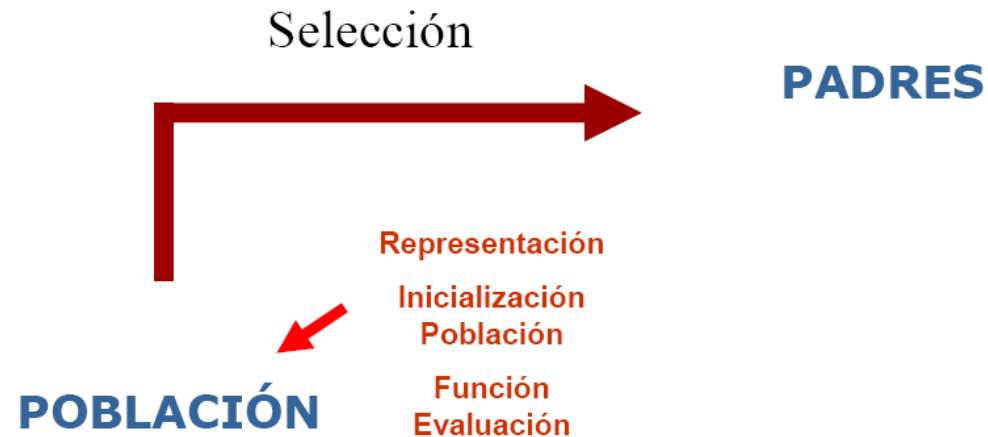
5. Estrategia de Selección

Escoge qué individuos van a disponer de oportunidades de reproducirse y cuáles no. Habrá individuos que aparezcan más de una vez e individuos que no aparezcan.

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

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.



Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Diseño

17

Ya hemos comentado dos métodos de selección como son:

- Selección por Ruleta (modelo clásico).
- Selección por Torneo.

Pero hay muchos otros como por ejemplo:

- Orden Lineal (LR): La población se ordena en función de su fitness y se asocia una probabilidad de selección a cada individuo que depende de su orden
- Selección Aleatoria (RS)
- Emparejamiento Variado Inverso (NAM): Un padre lo escoge aleatoriamente y para el otro selecciona aleatoriamente N_{nam} padres y escoge el más lejano al primer padre ($N_{nam}=3,5, \dots$). Está orientado a generar diversidad

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Diseño

18

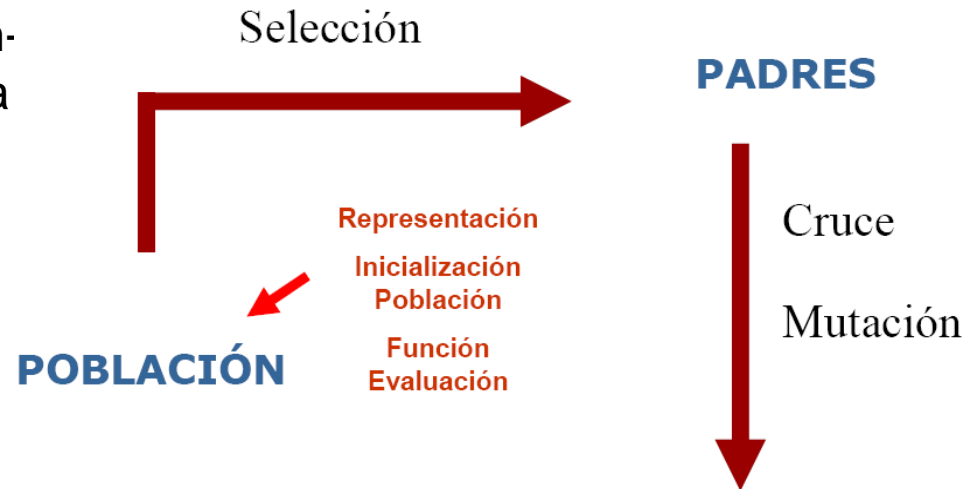
6. Operador de Cruce

Los individuos seleccionados son recombinados para producir la descendencia que formará la siguiente generación.

Podríamos tener uno o más operadores de cruce para nuestra representación.

Algunos aspectos importantes a 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 a la representación.
- La recombinación debe producir cromosomas válidos.
- Se utiliza con una probabilidad alta (p_c) de actuación sobre cada pareja de padres a cruzar ($p_c \in [0.6, 0.9]$), y si no actúa, los padres son los descendientes del proceso de recombinación de la pareja.



Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Diseño

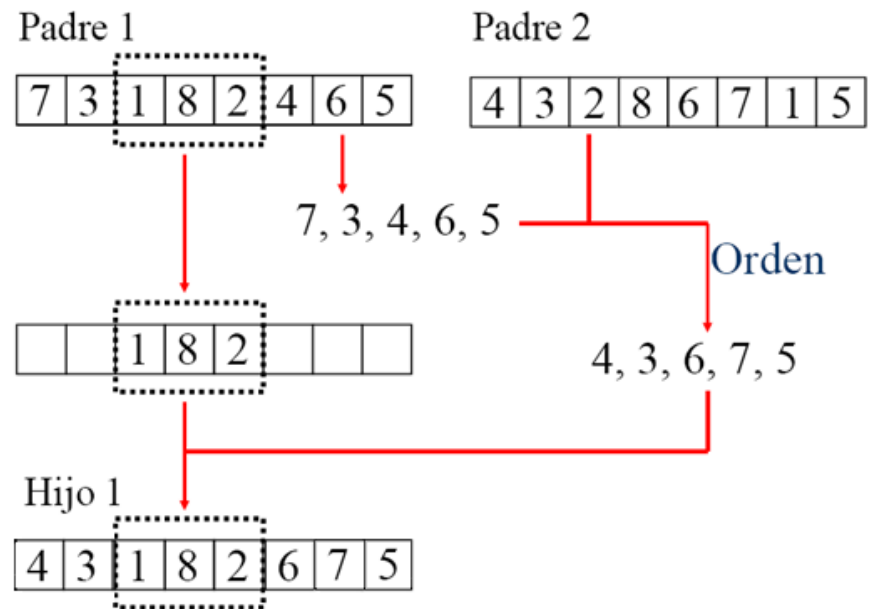
19

Ya hemos comentado dos operadores de cruce como son:

- Cruce por un punto.
- Cruce por dos puntos.

Pero hay muchos otros como por ejemplo:

- Para representación de orden:



Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Diseño

20

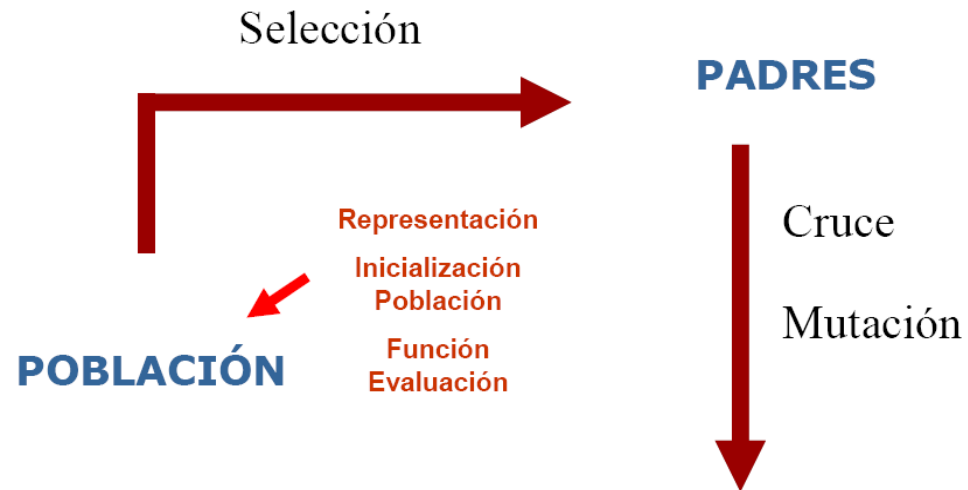
7. Operador de Mutación

Cambia aleatoriamente el valor de algunos de los genes.

Podríamos tener uno o más operadores de mutación para nuestra 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 (p_m) sobre cada descendiente obtenido tras aplicar el operador de cruce (incluidos los descendientes que coinciden con los padres cuando el operador de cruce no actúa)



Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Diseño

21

Ya hemos comentado dos operadores de mutación como son:

- Cambio de un gen aleatorio.
- Intercambio entre dos genes

Pero hay muchos otros como por ejemplo:

- Para representación real

Perturbación de los valores, frecuentemente, mediante una distribución Gaussiana $N(0, \sigma)$, donde

- 0 es la media
 - σ es la desviación típica
- $$\left. \begin{array}{l} \bullet 0 \text{ es la media} \\ \bullet \sigma \text{ es la desviación típica} \end{array} \right\} x'_i = x_i + N(0, \sigma_i) \text{ para cada gen}$$

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Sobre el ajuste de parámetros

22

Cuando utilizamos un AG para resolver un problema, es muy útil (y la mayoría de las veces, crucial) ajustar sus parámetros con el objetivo de obtener un comportamiento "razonable" en la resolución del problema (los parámetros más utilizados en un ajuste son la " p_c ", " p_m " y el "tamaño de la población", aunque cualquier otro puede también ser incluido en el ajuste).

El objetivo es disponer de las mejores combinaciones de operadores genéticos y valores para que las soluciones que obtenga sean las "óptimas" o las "mejores posibles". Este proceso se denomina "Ajuste de Parámetros" y consiste en:

- Elegir los distintos operadores genéticos aplicables y un conjunto de posibles valores para los parámetros numéricos,
- Seleccionar un conjunto significativo, suficientemente grande, de casos del problema,
- Resolver todos los casos utilizando todas las combinaciones de operadores-valores,
- Analizar las soluciones obtenidas. Por lo general, se buscan combinaciones que obtengan la solución óptima (o "muy buenas") para todos los casos.

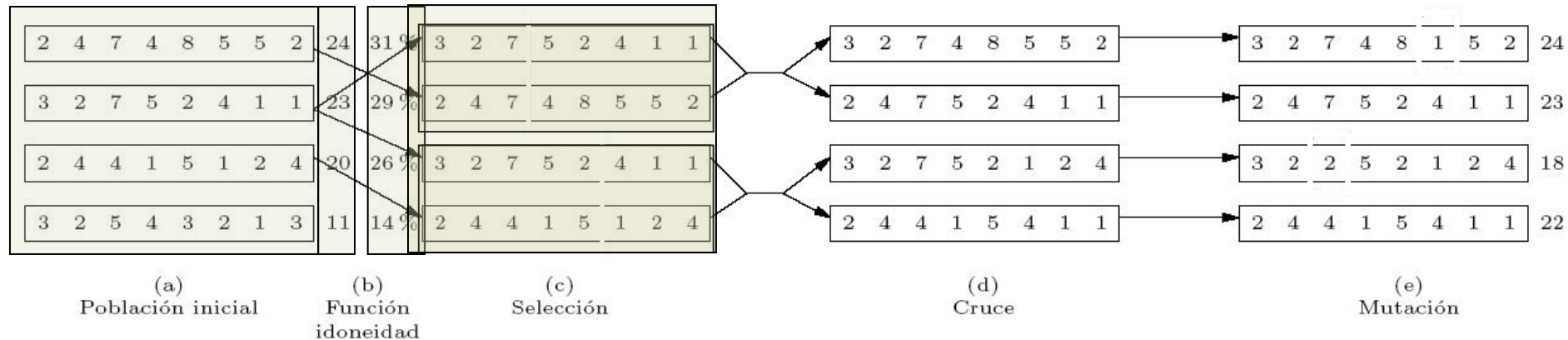
A partir de este proceso, debemos obtener unos determinados operadores-valores que hacen al AG robusto, eficiente y óptimo (o lo mejor posible) ante el problema.

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Ejemplo

23

Problema de las 8 reinas



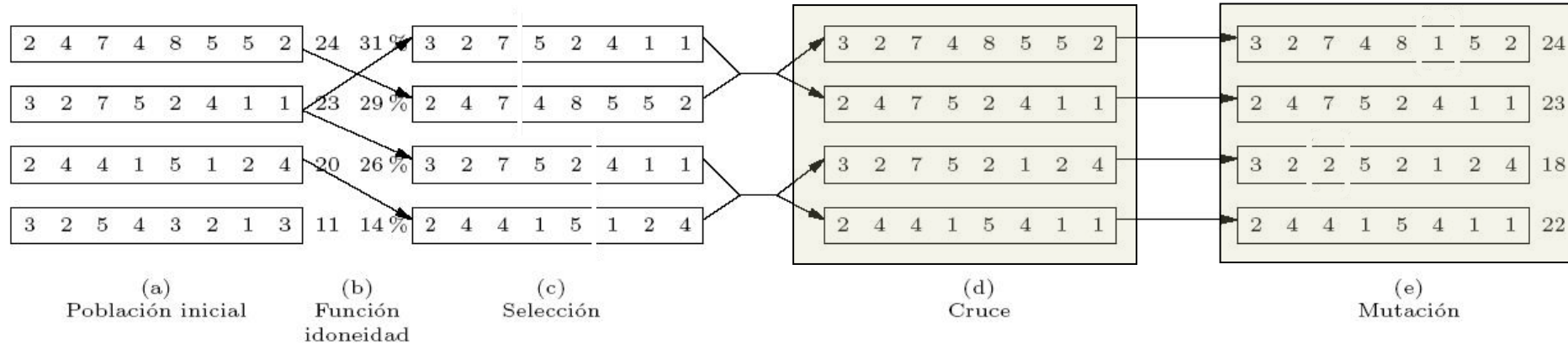
- Se parte de una población de, por ejemplo, $k = 4$ individuos
- Se calcula una función idoneidad de cada individuo
- Se calcula para cada individuo una probabilidad de ser seleccionado: proporcional a la función idoneidad (ruleta)
- En función de esta probabilidad, se van seleccionando individuos mediante el operador de selección hasta seleccionar y completar una población
- Se marcan individuos, en función de la probabilidad de cruce, para ser cruzados. Estos individuos se agrupan en pares con algún criterio

Métodos de búsqueda local y problemas de optimización

Búsqueda por Algoritmos Genéticos – Ejemplo

24

Problema de las 8 reinas



- Se aplica el operador de cruce (por un punto), obteniendo dos pares de descendientes (en algunas situaciones existirán individuos que no se crucen)
- Se aplica el operador de mutación:
 - Cada posición está sujeta a mutación aleatoria de acuerdo a la prob. de mutación
 - Equivale a escoger una reina y moverla a una casilla aleatoria de su columna
- Resulta una nueva generación de cuatro individuos, con sus funciones idoneidad, que forma la población para el siguiente ciclo