

Algoritmos Genéticos

Dr. Misael López Ramírez

Mayo 2023

Contenido

1. Introducción
2. Esquema básico
3. Codificación
4. Evaluación
5. Selección
6. Operadores
7. Ejemplo

Introducción

- Propuestos por Holland, mediados 70, computación evolutiva
- Popularizados por Goldberg, mediados 80, solución de problemas del mundo real
- Inspirados en el modelo de evolución biológica
- Aplicables a problemas de búsqueda y optimización complejos

Aproximación a la evolución biológica

- Método de búsqueda y optimización inspirados en la evolución biológica
- Posibles soluciones: población
- Selección de los individuos más *aptos*
- Generación de nuevos candidatos: **reproducción**
 - **Recombinación (cruce)**
 - **Mutación**

¿Cómo funcionan?

- Esquema de funcionamiento de un AG:
 - Se crea una población inicial generando individuos aleatoriamente.
 - Repetimos hasta que se alcance el individuo óptimo o el número máximo de generaciones:
 - *Asignar* un valor de supervivencia a cada miembro de la población.
 - *Seleccionar* a un conjunto de individuos que actuarán como padres usando como criterio su probabilidad de supervivencia.
 - *Emparejar* un grupo de padres para crear descendencia.
 - *Combinar* la descendencia con la población actual para crear nueva población.

Esquema básico

función ALGORITMO-GENÉTICO(*poblaciónInicial*) **returns** una población

entrada: *poblaciónInicial*, una población

static: *población(.)*, un array de población

begin

$t \leftarrow 0$

$población(t) \leftarrow poblaciónInicial$

EVALUAR($población(t)$)

while (not condiciónTerminación) **do**

$t \leftarrow t + 1$

$población1 \leftarrow SELECCIONAR(población(t-1))$

$población2 \leftarrow CRUZAR(población1)$

$población3 \leftarrow MUTAR(población2)$

$población(t) \leftarrow REMPLAZAR(población3)$

end

return($población(t)$)

end

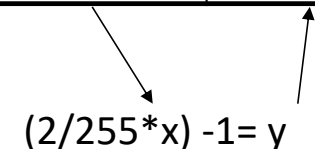
Codificación

- Individuo: cromosoma
- Cromosoma: cadena de caracteres
 - En principio, cualquier representación es válida
- Codificación óptima: alfabeto binario (teorema de los esquemas)
- Codificación habitual: cadena de bits

Ejemplos de codificación maximización función

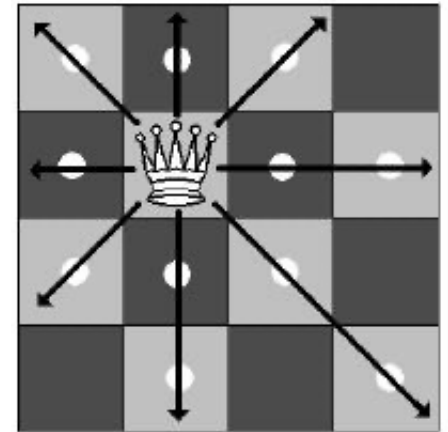
- $f(x)=1-x^2$, parábola invertida con máximo en $x=0$
- Único parámetro o atributo: variable x
- Codificamos el valor de la variable mediante un byte $[0,255]$, ajustado al intervalo real $[-1,1]$, donde queremos hallar el máximo de la función

Valor binario	Descodificación	Valor real
10010100	148	0,161
10010001	145	0,137
00101001	41	-0,678
01000101	65	-0,490


$$(2/255 * x) - 1 = y$$

8-reinas

- Atributo: posición de una dama en una columna (3 bits)
- Cromosoma: secuencia de atributos, 24 bits



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

Regla de los bloques de construcción

- **La codificación** es clave en la resolución del problema
- Heurística: parámetros relacionados entre sí (*genes*) deben de estar cercanos en el cromosoma
- Gran flexibilidad
 - Cromosomas bi, tridimensionales
 - Longitud variable

Evaluación

- En esta etapa hay que cuantificar la calidad de los individuos de la población
- Generalmente
 - Decodificar el cromosoma
 - Calidad de la solución
 - Evaluación mediante función *fitness* o *aptitud*

Ejemplos de funciones de aptitud

- Para $f(x)=1-x^2$, la función de aptitud es la misma
- Para 8-reinas: número total de pares de damas no amenazadas
 - En cualquier solución:
 $7+6+5+4+3+2+1=28$

Valor binario	Descodificación	Valor real	Aptitud
10010100	148	0,161	0.974
10010001	145	0,137	.981
00101001	41	-0,678	.540
01000101	65	-0,490	.760

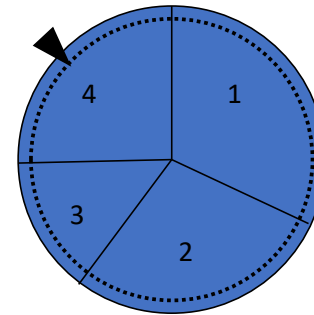
Selección

Selección de los elementos que se reproducen

- A partir de la función de aptitud
- Varios métodos
 - Rueda de ruleta
 - Basado en el rango
 - Selección de torneo
- Cambio de generación
 - Manteniendo el tamaño de la población
 - Aumentando el tamaño de la población

Rueda de ruleta

- Se asigna a cada individuo la probabilidad:
$$\Pr(x) = \frac{\textit{aptitud}(x)}{\sum_{y \in \textit{población}} \textit{aptitud}(y)}$$
- Si algún individuo domina la población, se escala o normaliza.
- Se elijen parejas aleatorias de individuos de acuerdo a su probabilidad.
- Inconveniente: los individuos con más aptitud tiende a dominar la población en pocas generaciones.

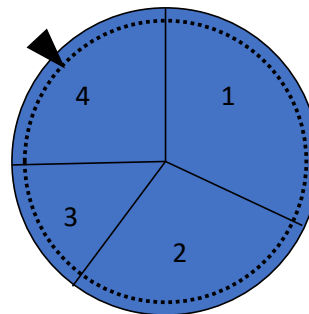


Rueda

Valor binario	Descodificación	Valor real	Aptitud	Probabilidad	Probabilidad acumulada
10010100	148	0,161	0.974	0.299	0.299
10010001	145	0,137	0.981	0.301	0.600
00101001	41	-0,678	0.540	0.166	0.766
01000101	65	-0,490	0.760	0.233	1.000

3.255

$$f(x)=1-x^2$$



Basado en el rango

- Se ordena la población por orden creciente de aptitud
- Se eliminan los M primeros (menor aptitud)
- Se eligen de forma aleatoria, con probabilidad dada por el rango, pares de individuos y sus descendientes se añaden a la población

Torneo

- Se seleccionan dos individuos aleatoriamente
- Se elije el más apto con una probabilidad P y el menos apto con una probabilidad $(1-P)$
- Introduce más diversidad en la población

cambio de generación

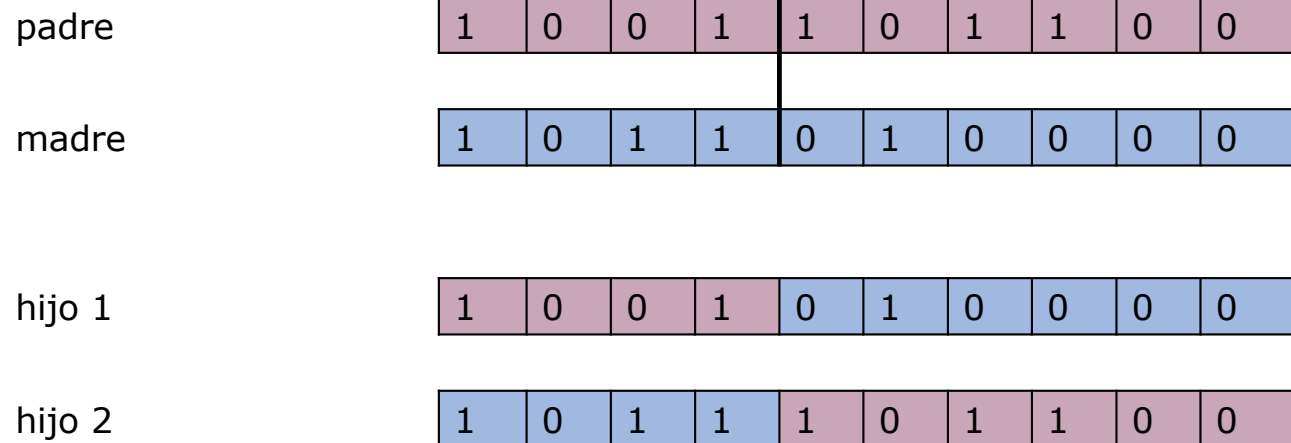
- Manteniendo el tamaño de la población intermedia
 - Reemplazar padres por hijos
 - Reemplazar un par de individuos elegidos aleatoriamente por los hijos
 - Otros
- Aumentando el tamaño de la población intermedia
 - Crear población temporal con padres e hijos, seleccionando los mejores
 - Dados n padres generar m ($m > n$) hijos y de ellos seleccionar los n mejores

Operador de cruce (crossover)

- Principal operador genético
- Simula el intercambio de material genético o genes
- Se aplica con probabilidad p_c a individuos seleccionados
- Cruce ideal: recombina buenos bloques de construcción de sus progenitores
- Operadores
 - Cruce de n-puntos
 - Cruce multipunto
 - Cruce especializado

Cruce de un punto

- Seleccionar aleatoriamente una posición en el cromosoma
- Intercambiar el final del cromosoma a partir de dicho punto



Cruce de dos puntos

padre

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

madre

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

hijo 1

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

hijo 2

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

GRACIAS!!!