

Complejidad y sistemas basados en la naturaleza

Fernando Elizalde Ramírez

Diseño de algoritmos matemáticos bioinspirados
Departamento de Matemáticas
Tecnológico de Monterrey

August 17, 2023

Algoritmo Genético

Un algoritmo genético (AG) es una técnica de resolución de problemas que imita a la evolución biológica como estrategia para resolver problemas, englobándose dentro de lo que antes hemos denominado técnicas basadas en poblaciones.

Entonces

- Son algoritmos de optimización, búsqueda y aprendizaje **inspirados** en los procesos de **evolución natural** y **evolución genética**, esto es, imitan el proceso de evolución biológica de sobrevivencia del más apto.
- Son **estocásticos**.
- **Eficaces** para **explorar** diversas partes de la **región factible** y evolucionar de manera gradual hacia las **mejores soluciones** factibles.
- Cada **solución** factible se considera como un **cromosoma** codificado por un conjunto de genes.
- Un **conjunto** de N **soluciones** (cromosomas) factibles se conoce como **población**.
- **La aptitud** de un individuo mide su capacidad para sobrevivir en su entorno. Por ello, se mide en términos de una función objetivo apropiada (**función fitness**), por lo que un cromosoma más apto da un mejor valor a la función fitness.

Evaluación de la descendencia

$$f(\circ)$$

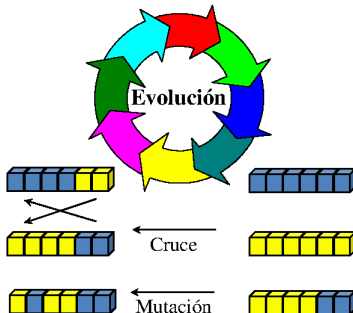
Nueva población de descendientes

Individuo	genotipo	fitness
x_1	100100	1296
x_2	010010	324
x_3	010110	484
x_4	000001	1

Selección de Progenitores



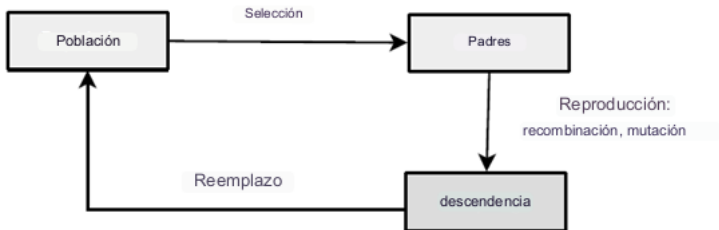
Reproducción

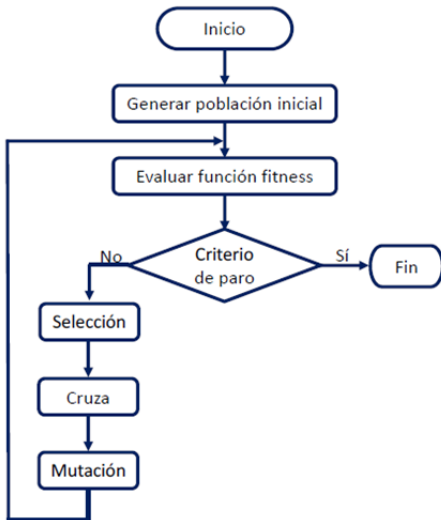


- En general, estos algoritmos codifican una solución a un problema de optimización en un simple cromosoma y mediante operadores de recombinación de estos cromosomas preservan la información crítica promoviendo la diversificación de soluciones.

- En general, estos algoritmos codifican una solución a un problema de optimización en un simple cromosoma y mediante operadores de recombinación de estos cromosomas preservan la información crítica promoviendo la diversificación de soluciones.
- El reemplazo (selección de sobreviviente) es generacional, es decir, los padres son reemplazados, en la población, por los descendientes.

- En general, estos algoritmos codifican una solución a un problema de optimización en un simple cromosoma y mediante operadores de recombinación de estos cromosomas preservan la información crítica promoviendo la diversificación de soluciones.
- El reemplazo (selección de sobreviviente) es generacional, es decir, los padres son reemplazados, en la población, por los descendientes.
- IDEA: Se seleccionan dos padres a partir de una población factible. Los genes de los dos padres se cruzan y (posiblemente) mutan para producir dos hijos. La descendencia reemplaza a los cromosomas más débiles (menos aptos) en la población, y el proceso de seleccionar nuevos padres se repite.





Algorithm 3.2 Template of an evolutionary algorithm.

```
Generate( $P(0)$ ) ; /* Initial population */  
 $t = 0$  ;  
While not Termination_Criterion( $P(t)$ ) Do  
    Evaluate( $P(t)$ ) ;  
     $P'(t)$       = Selection( $P(t)$ ) ;  
     $P'(t)$       = Reproduction( $P'(t)$ ); Evaluate( $P'(t)$ ) ;  
     $P(t + 1)$   = Replace( $P(t)$ ,  $P'(t)$ ) ;  
     $t = t + 1$  ;  
End While  
Output Best individual or best population found.
```

Generación de la población y representación del cromosoma

- Se establece un número N de soluciones factibles y es finito. Se puede hacer de forma aleatoria o usar algún procedimiento constructivo (heurístico).
- Cada individuo S_i dentro de la población P tiene asociado un cromosoma que es una cadena de genes, los cuales representan las características individuales de cada individuo.
- El cromosoma está asociado al diseño de las variables de decisión.
- Existen diferentes maneras para representar la solución mediante un cromosoma y se debe elegir la más relevante para el problema en cuestión. La más común, representación binaria.

Ejemplo: Problema de la mochila

Item	1	2	3	4	5	6	7	8	9	10
Peso	20	25	20	20	35	40	35	30	35	25
Beneficio	50	35	45	20	35	40	50	50	45	40

- Para el problema de la mochila, ¿cómo representarías un cromosoma?

Ejemplo: Problema de la mochila

Item	1	2	3	4	5	6	7	8	9	10
Peso	20	25	20	20	35	40	35	30	35	25
Beneficio	50	35	45	20	35	40	50	50	45	40

- Para el problema de la mochila, ¿cómo representarías un cromosoma?
Un vector de tamaño igual al número de items

Ejemplo: Problema de la mochila

Item	1	2	3	4	5	6	7	8	9	10
Peso	20	25	20	20	35	40	35	30	35	25
Beneficio	50	35	45	20	35	40	50	50	45	40

- Para el problema de la mochila, ¿cómo representarías un cromosoma?
Un vector de tamaño igual al número de items
- ¿Qué características tendrían los genes?

Ejemplo: Problema de la mochila

Item	1	2	3	4	5	6	7	8	9	10
Peso	20	25	20	20	35	40	35	30	35	25
Beneficio	50	35	45	20	35	40	50	50	45	40

- Para el problema de la mochila, ¿cómo representarías un cromosoma?
Un vector de tamaño igual al número de items
- ¿Qué características tendrían los genes? Sería binarios

Ejemplo: Problema de la mochila

Item	1	2	3	4	5	6	7	8	9	10
Peso	20	25	20	20	35	40	35	30	35	25
Beneficio	50	35	45	20	35	40	50	50	45	40

- Para el problema de la mochila, ¿cómo representarías un cromosoma?
Un vector de tamaño igual al número de items
- ¿Qué características tendrían los genes? Sería binarios
- ¿Cuál es el tamaño de tu población?

Ejemplo: Problema de la mochila

Item	1	2	3	4	5	6	7	8	9	10
Peso	20	25	20	20	35	40	35	30	35	25
Beneficio	50	35	45	20	35	40	50	50	45	40

- Para el problema de la mochila, ¿cómo representarías un cromosoma?
Un vector de tamaño igual al número de items
- ¿Qué características tendrían los genes? Sería binarios
- ¿Cuál es el tamaño de tu población? Se decide por experimentación

Ejemplo: Problema de la mochila

Item	1	2	3	4	5	6	7	8	9	10
Peso	20	25	20	20	35	40	35	30	35	25
Beneficio	50	35	45	20	35	40	50	50	45	40

- Para el problema de la mochila, ¿cómo representarías un cromosoma?
Un vector de tamaño igual al número de items
- ¿Qué características tendrían los genes? Sería binarios
- ¿Cuál es el tamaño de tu población? Se decide por experimentación
- ¿Cómo generas la población?

Ejemplo: Problema de la mochila

Item	1	2	3	4	5	6	7	8	9	10
Peso	20	25	20	20	35	40	35	30	35	25
Beneficio	50	35	45	20	35	40	50	50	45	40

- Para el problema de la mochila, ¿cómo representarías un cromosoma?
Un vector de tamaño igual al número de items
- ¿Qué características tendrían los genes? Sería binarios
- ¿Cuál es el tamaño de tu población? Se decide por experimentación
- ¿Cómo generas la población? Puede ser de forma aleatoria o mediante un heurístico

Función fitness

La función fitness nos ayudará a evaluar a todos los individuos. Con ella, podremos asignar un valor a cada individuo de la generación de forma que, si una solución obtiene un valor más elevado que otra, representará que es una mejor solución, o una solución más cercana a la solución óptima que se busca. Así pues, podremos saber qué individuos son mejores dentro de una generación. Necesitamos una función fitness adecuada a cada problema.

- Recordemos que la población se modifica en términos de los individuos mejores adaptados (función fitness)
- ¿Qué podemos usar como función fitness?
 - Función objetivo (comúnmente)
 - Se pueden utilizar funciones aproximadas para reducir el costo de evaluación.
 - Cuando hay restricciones, éstas se pueden introducir en el costo como penalización.
 - Con objetivos múltiples se busca una solución de compromiso
- Ejemplo: Para el problema de la mochila, ¿qué función fitness podrías utilizar para medir el desempeño del individuo?

Función fitness

La función fitness nos ayudará a evaluar a todos los individuos. Con ella, podremos asignar un valor a cada individuo de la generación de forma que, si una solución obtiene un valor más elevado que otra, representará que es una mejor solución, o una solución más cercana a la solución óptima que se busca. Así pues, podremos saber qué individuos son mejores dentro de una generación. Necesitamos una función fitness adecuada a cada problema.

- Recordemos que la población se modifica en términos de los individuos mejores adaptados (función fitness)
- ¿Qué podemos usar como función fitness?
 - Función objetivo (comúnmente)
 - Se pueden utilizar funciones aproximadas para reducir el costo de evaluación.
 - Cuando hay restricciones, éstas se pueden introducir en el costo como penalización.
 - Con objetivos múltiples se busca una solución de compromiso
- Ejemplo: Para el problema de la mochila, ¿qué función fitness podrías utilizar para medir el desempeño del individuo?
 - La suma de los beneficios si no superan la capacidad

-
- Cada individuo en la población tiene la misma probabilidad de ser elegido:

$$p_i = \frac{f(S_i)}{\sum_{i=1}^N f(S_i)}$$

, donde S_i representa una solución en la población.

- La idea es elegir las soluciones más adecuadas en términos de la función fitness.
- Puedes elegir de forma aleatoria, elegir los dos individuos más aptos, etc.
- Incluir material genético útil en el proceso de reproducción para que los individuos menos buenos puedan reproducirse.
- A veces simplemente, haremos un clon de padre a hijo, para asegurar que preservamos la genética original.

- La implementación del cruzamiento es dependiente del problema y se debe diseñar de acuerdo con la representación. Cada par de padres, al combinarse, generan uno o dos hijos, de tal forma que hereden características de cada padre.
- Puede ser aleatoria o determinista.
- Es importante diseñar un operador de cruzamiento que produzca soluciones factibles.

Operador	Característica
1-point	Dados dos cromosomas $x_1, x_2, \dots, x_r, y_1, y_2, \dots, y_r$ y un número entero $p \in \{1, \dots, r-1\}$ los hijos serán $x_1, \dots, x_p, y_{p+1}, \dots, y_r$ y $y_1, \dots, y_p, x_{p+1}, \dots, x_r$
2-point	Dados dos cromosomas $x_1, x_2, \dots, x_r, y_1, y_2, \dots, y_r$ y dos números enteros $p, q \in \{1, \dots, r-1\}$ con $p < q$ los hijos serán $x_1, \dots, x_p, y_{p+1}, \dots, y_q, x_{q+1}, \dots, x_r$ y $y_1, \dots, y_p, x_{p+1}, \dots, x_q, y_{q+1}, \dots, y_r$
Uniforme	Dados dos cromosomas el hijo se forma de tal manera que cada gen se elige de forma aleatoria de los padres.



Ejemplo

Para el problema de la mochila, ¿cómo definirías un operador de cruzamiento de cada tipo (1-point, 2-point, uniforme)?

- Algún descendiente, aleatoriamente, será modificado.
- La mutación se da con cierta probabilidad. No siempre se muta, no todos mutan.
- Se debe permitir alcanzar cualquier parte del espacio de búsqueda y mantener siempre una solución factible.
- El tamaño de la mutación debe ser controlado.

Para el problema de la mochila, ¿cómo definirías un operador de mutación de cada tipo?

El algoritmo termina cuando:

- Se alcanza el óptimo.
- Recursos limitados de CPU (tiempo límite)
- Máximo número de iteraciones.
- Después de un número de iteraciones sin mejora.

Preguntas importantes

- Antes de implementar el algoritmo, es necesario considerar lo siguiente:
- ¿Cuál debe ser el tamaño de la población?
- ¿Cómo deben seleccionarse los miembros de la población actual para convertirse en padres?
- ¿Cuáles deben ser las características de los hijos derivadas de las características de los padres?
- ¿Cómo deben insertarse las mutaciones en las características de los hijos?
- ¿Cuál es la regla de detención que debe usarse?