



Metaheurísticas

Algoritmos heurísticos

Los algoritmos heurísticos son los más fáciles de utilizar, ya que se basan en el conocimiento de una heurística que guía el proceso de búsqueda. El conocimiento del problema usualmente ayuda a encontrar una heurística razonable que encontrará rápidamente una solución aceptable. Un algoritmo de este tipo sólo buscará dentro de un subespacio del área total a una solución buena (**que no necesariamente es la mejor**) **que satisfaga las restricciones impuestas**. La principal limitación es su incapacidad para escapar de óptimos locales (encontrar soluciones parcialmente óptimas).

Algoritmos metaheurísticos

Una metaheurística es un proceso iterativo maestro que guía y modifica las operaciones de una heurística subordinada para producir eficientemente soluciones de alta calidad. Las metaheurísticas pueden manipular una única solución completa (o incompleta) o una colección de soluciones en cada iteración. La heurística subordinada puede ser un procedimiento de alto o bajo nivel, una búsqueda local, o un método constructivo. Entre los algoritmos metaheurísticos más conocidos están, el recocido simulado y los algoritmos genéticos.



Metaheurísticas

Los cuatro tipos fundamentales:

1. Las metaheurísticas **de relajación** se refieren a procedimientos de resolución de problemas que utilizan relajaciones del modelo original (es decir, modificaciones del modelo que hacen al problema más fácil de resolver), cuya solución facilita la solución del problema original.
2. Las metaheurísticas **constructivas** se orientan a los procedimientos que tratan de obtener una solución a partir del análisis y selección paulatina de las componentes que la forman.
3. Las metaheurísticas de **búsqueda** guían los procedimientos que usan transformaciones o movimientos para recorrer el espacio de soluciones alternativas y explorar las estructuras de entornos asociadas.
4. Las metaheurísticas **evolutivas** están enfocadas a los procedimientos basados en conjuntos de soluciones que evolucionan sobre el espacio de soluciones.



Metaheurísticas

Metaheurísticas Evolutivas

Las metaheurísticas evolutivas establecen estrategias para **conducir** la evolución en el espacio de búsqueda de conjuntos de soluciones (usualmente llamados **poblaciones**) con la intención de acercarse a la solución óptima **con sus elementos**.

El aspecto fundamental de las heurísticas evolutivas consiste en la **interacción** entre los miembros de la población frente a las búsquedas que se guían por la información de soluciones individuales.

Las diferentes metaheurísticas evolutivas se distinguen por la forma en que **combinan** la información proporcionada por los elementos de la población para hacerla evolucionar mediante la obtención de **nuevas soluciones**.

Los algoritmos genéticos y meméticos y los de estimación de distribuciones emplean fundamentalmente **procedimientos aleatorios**, mientras que las metaheurísticas de búsqueda dispersa o de reencadenamiento de caminos (*Path-Relinking*) emplean **procedimientos sistemáticos**.



Metaheurísticas – ALGORITMOS GENÉTICOS

FUNDAMENTOS

En ocasiones la computación se basa en procesos observados de la naturaleza para resolver ciertos problemas: por ejemplo, las redes neuronales que replican los procesos de sinapsis entre las neuronas. En este caso, los algoritmos genéticos replican el modelo de selección natural propuesto por Darwin, y que resume la famosa frase ‘la supervivencia del más fuerte o adaptado’.

Este modelo básicamente dice que, dentro de una población, los individuos que sobreviven son aquellos que están más adaptados al medio, por lo tanto, las generaciones futuras de estos estarán mejor adaptadas ya que serán combinaciones de los mejores genes de sus antepasados. Además, esta teoría de la evolución introduce un concepto muy interesante que son las mutaciones. Una mutación es un pequeño cambio que se produce de manera aleatoria en ciertos individuos e introduce de esta manera versatilidad en las poblaciones. Habrá mutaciones que den lugar a cambios favorables y otros desfavorables.



Metaheurísticas – ALGORITMOS GENÉTICOS

USOS

Se utilizan para resolver problemas de Búsqueda y Optimización, ya que se basan en evolucionar poblaciones de soluciones hacia valores óptimos del problema.

ESTRUCTURA DE UN ALGORITMO GENÉTICO

Conceptos previos

Individuo: los individuos de la población son las posibles soluciones al problema que se intenta resolver.

Población: conjunto de individuos.

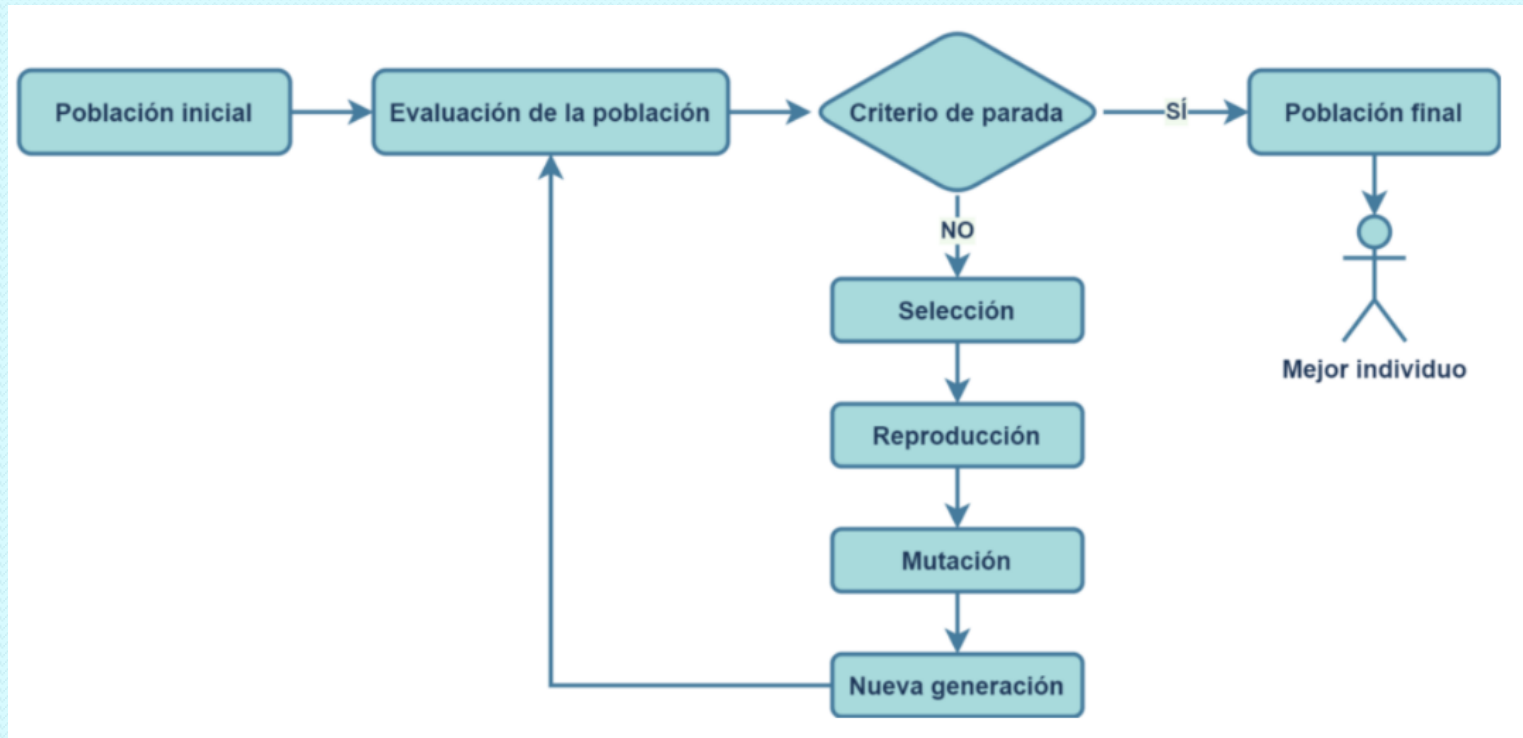
Función fitness o de adaptación: función que evalúa a los individuos y les asigna una puntuación en función de que tan buenas sean las soluciones para el problema.

Función de cruce: función que dados dos individuos genera dos descendientes a partir de la combinación de genes de sus padres, esta función depende del problema en cuestión.



Metaheurísticas – ALGORITMOS GENÉTICOS

ESTRUCTURA DE UN ALGORITMO GENÉTICO





Metaheurísticas – ALGORITMOS GENÉTICOS

ESTRUCTURA DE UN ALGORITMO GENÉTICO

1. **Fase inicial:** se genera una población inicial de individuos (soluciones)
2. **Fase de evaluación:** se evalúan los individuos de la población con la función fitness
3. **Fase de selección :** se seleccionan los mejores individuos
4. **Fase de reproducción:** se cruzan los individuos seleccionados mediante la función de cruce, dando lugar a una nueva generación que va a sustituir a la anterior
5. **Fase de mutación:** se introducen mutaciones (pequeños cambios) en ciertos individuos de la nueva población de manera aleatoria o un entrecruzamiento cromosómico (también llamado crossover o recombinación)
6. **Se obtuvo una nueva generación, en general, con soluciones mejores que la anterior. Se vuelve al punto 2**

Los algoritmos genéticos pueden finalizar cuando se alcanza un número de generaciones concreto o cuando cumplen una condición de parada.



Metaheurísticas – ALGORITMOS GENÉTICOS

VENTAJAS

- ✓ Se desenvuelven bien en problemas con un paisaje adaptativo complejo: aquéllos en los que la función de aptitud es ruidosa, cambia con el tiempo, o tiene muchos óptimos locales, gracias a los cuatro componentes principales de los algoritmos genéticos, paralelismo, selección, mutación y cruzamiento, los que trabajan juntos para conseguir su buen desempeño.
- ✓ Pueden explorar el espacio de soluciones en múltiples direcciones a la vez, por lo que, los algoritmos genéticos funcionan particularmente bien resolviendo problemas cuyo espacio de soluciones potenciales es realmente grande, demasiado vasto para hacer una búsqueda exhaustiva en un tiempo razonable
- ✓ Los algoritmos genéticos realizan cambios aleatorios en sus soluciones candidatas y luego utilizan la función de aptitud para determinar si esos cambios producen una mejora. Como sus decisiones están basadas en la aleatoriedad, todos los caminos de búsqueda posibles están abiertos; en contraste a cualquier otra estrategia de resolución de problemas que dependa de un conocimiento previo



Metaheurísticas – ALGORITMOS GENÉTICOS

DESVENTAJAS

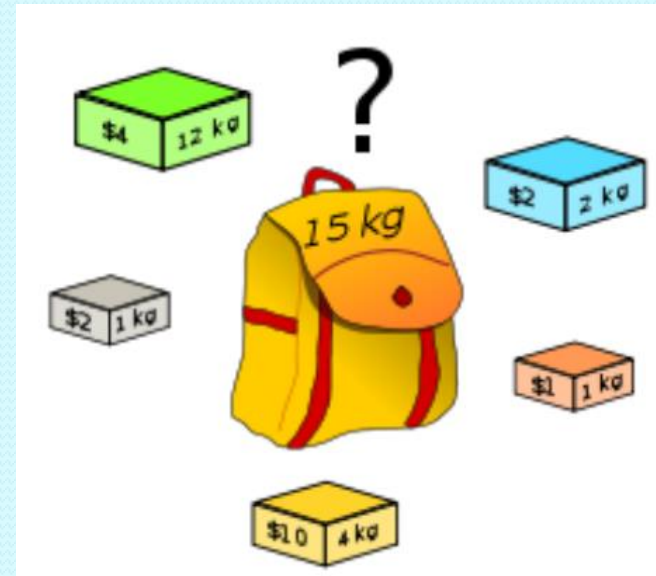
- Si se elige mal una función de aptitud o se define de manera inexacta, puede que el algoritmo genético sea incapaz de encontrar una solución al problema, o puede acabar resolviendo el problema equivocado.
- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen tamaño de la población, el ritmo de mutación y cruzamiento, el tipo y fuerza de la selección.
- El lenguaje utilizado para especificar soluciones candidatas debe ser robusto; es decir, debe ser capaz de tolerar cambios aleatorios que no produzcan constantemente errores fatales o resultados sin sentido. Una de las formas mas usadas es definir a los individuos como listas de números -binarios, enteros o reales- donde cada número representa algún aspecto de la solución candidata.

Metaheurísticas – ALGORITMOS GENÉTICOS

EJEMPLO PROPUESTO: EL PROBLEMA DE LA MOCHILA

Dada una mochila con una capacidad de 15 kg que se puede llenar con cajas de distinto peso y valor. ¿Qué cajas elijo de modo de maximizar mis ganancias y no exceder el peso permitido?

Una representación de una solución puede ser una matriz de bits, donde cada bit representa un objeto diferente, y el valor del bit (0 o 1) representa si el objeto está o no en la mochila. No todas las representaciones son válidas, ya que el tamaño de los objetos puede exceder la capacidad de la mochila. La aptitud de la solución es la suma de valores de todos los objetos en la mochila si la representación es válida, o 0 de lo contrario.





Metaheurísticas – ALGORITMOS GENÉTICOS

EJEMPLO PROPUESTO: EL PROBLEMA DE LA MOCHILA

Se tienen **n** objetos y una mochila

- El objeto **i** tiene peso **pi** y la inclusión del objeto **i** en la mochila produce un beneficio **bi**
- El objetivo es llenar la mochila, de capacidad **C**, de manera que se maximice el beneficio.

$$\text{maximizar } \sum_{1 \leq i \leq n} b_i x_i$$


$$\text{sujeto a } \sum_{1 \leq i \leq n} p_i x_i \leq C$$

$$\text{con } x_i \in \{0, 1\}, b_i > 0, p_i > 0, 1 \leq i \leq n$$

Metaheurísticas – ALGORITMOS GENÉTICOS

EJEMPLO PROPUESTO: EL PROBLEMA DE LA MOCHILA

Capacidad Mochila = 15



CUALES OBJETOS LLEVAR

Binario

OBJETO 1 Utilidad = 4
Peso = 7

OBJETO 2 Utilidad = 5
Peso = 6

OBJETO 3 Utilidad = 6
Peso = 8

OBJETO 4 Utilidad = 3
Peso = 2

CROMOSOMAS

ACTIVIDAD

- 1 gen por variable de decisión binaria
- Cromosoma compuesto por los genes
- Función de fitness == Igual a la Función Objetivo

$Z = 4x_1 + 5x_2 + 6x_3 + 3x_4$

Restricción : $7x_1 + 6x_2 + 8x_3 + 2x_4 \leq 15$

Line = 4 bits Un bit por cada objeto [0,1]

Cromosoma:

X_1	X_2	X_3	X_4
b_1	b_2	b_3	b_4

Tamaño de la población: = 4



Metaheurísticas – ALGORITMOS GENÉTICOS

EJEMPLO PROPUESTO: EL PROBLEMA DE LA MOCHILA

1. Población inicial aleatoria

Cromosomas

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

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

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

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

Pcr: 0,98
Pmut: 0,1



Metaheurísticas – ALGORITMOS GENÉTICOS

EJEMPLO PROPUESTO: EL PROBLEMA DE LA MOCHILA

Evalúa cada individuo de la población

$$Z = 4x_1 + 5x_2 + 6x_3 + 3x_4$$

$$\text{Pcr: } 0,98 \quad \text{Pmut: } 0,1 \quad P = 7x_1 + 6x_2 + 8x_3 + 2x_4 \leq 15$$

Ind	x1	x2	x3	x4	Z	Peso	Probab	Prob Acum
1	0	1	1	0	11	14	0,29730	0,29730
2	1	0	1	0	10	15	0,27027	0,56757
3	0	1	0	0	5	6	0,13514	0,70270
4	0	1	1	0	11	14	0,29730	1,00000

PRIMERA ITERACIÓN (a)

Aleat1: Padr1 0,4771 Ind 2

Aleat2: Padr2 0,9341 Ind 4

Cruzan? 0,8448 **<0,98 SI CRUZAN**

PuntoCorte 0,4376 Entre gen 2 y 3

Aleatorios para **Mutación**

0,2940	0,3741	0,0257	0,6164
0,0509	0,6250	0,4106	0,1718

<0,1

37

1	0	1	0
0	1	1	0

1	0	1	0
0	1	1	0

1	0	0	0
1	1	1	0

P=
7

21

Hijo 1

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

Metaheurísticas – ALGORITMOS GENÉTICOS

EJEMPLO PROPUESTO: EL PROBLEMA DE LA MOCHILA

PRIMERA ITERACIÓN (b)

Ind	x1	x2	x3	x4	Z	Peso	Probab	Prob Acum
1	0	1	1	0	11	14	0,29730	0,29730
2	1	0	1	0	10	15	0,27027	0,56757
3	0	1	0	0	5	6	0,13514	0,70270
4	0	1	1	0	11	14	0,29730	1,00000

Aleat1:Padr1 0,8024 Ind 4

Aleat2:Padr2 0,2657 Ind 1

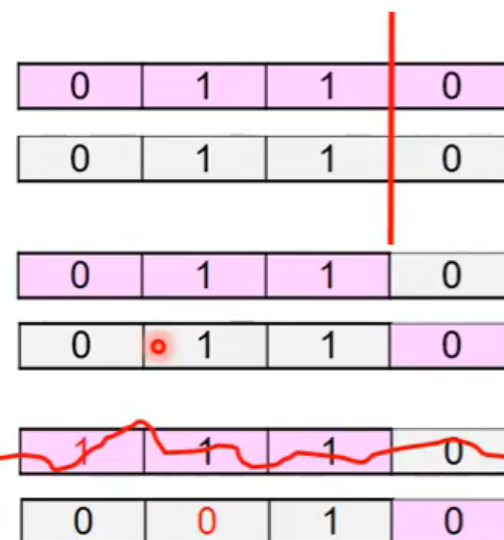
Cruzan? 0,0343 **<0,98 SI CRUZAN**

PuntoCorte 0,7402 Entre gen 3 y 4

Aleatorios para **Mutación**

0,0426	0,2529	0,3018	0,6218
0,3377	0,0844	0,5885	0,5171

<0,1



P=

21

8

Hijo 2

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



Metaheurísticas – ALGORITMOS GENÉTICOS

EJEMPLO PROPUESTO: EL PROBLEMA DE LA MOCHILA

PRIMERA ITERACIÓN (c)

Ind	x1	x2	x3	x4	Z	Peso	Probab	Prob Acum
1	0	1	1	0	11	14	0,29730	0,29730
2	1	0	1	0	10	15	0,27027	0,56757
3	0	1	0	0	5	6	0,13514	0,70270
4	0	1	1	0	11	14	0,29730	1,00000

Aleat1:Padr1 0,6578 Ind 3

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

Aleat2:Padr2 0,2580 Ind 1

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

Cruzan? 0,9890 **<0,98 NO CRUZAN**

Aleatorios para **Mutación**

0,1674	0,3015	0,0926	0,8496
0,9182	0,9214	0,1670	0,1554

<0,1

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

P=

14

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

14

Hijo 3

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

Hijo 4

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



Metaheurísticas – ALGORITMOS GENÉTICOS

EJEMPLO PROPUESTO: EL PROBLEMA DE LA MOCHILA

GENERACION t+1

Fin de primera iteración

Ind	x1	x2	x3	x4	Z	Peso	Probab	Prob Acum
1	1	0	0	0	4	7	0,12500	0,12500
2	0	0	1	0	6	8	0,18750	0,31250
3	0	1	1	0	11	14	0,34375	0,65625
4	0	1	1	0	11	14	0,34375	1,00000
					32	1		

Continuar con la siguiente iteración ...

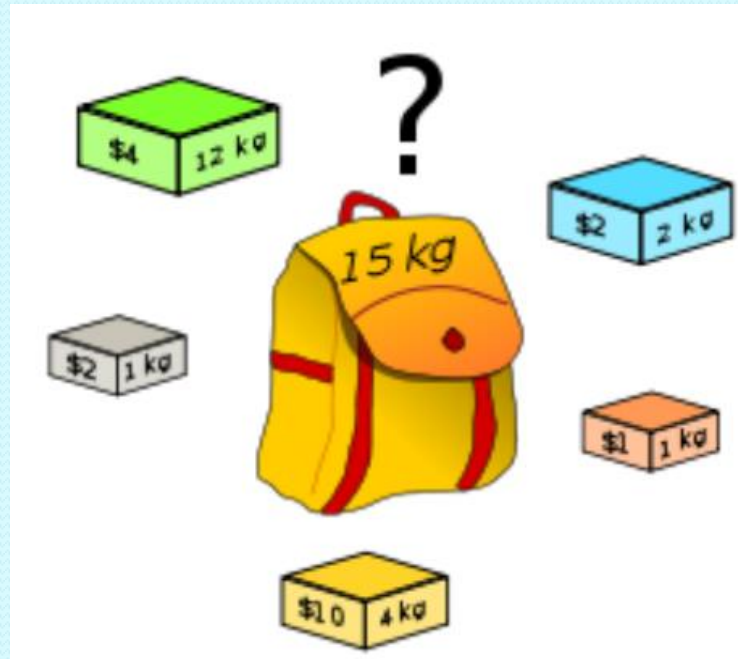
Pero si ya fuera la ultima iteración... Entonces la solución entregada sería

Llevar los objetos X2 y X3
Con una utilidad 11
y un peso de 14



Metaheurísticas – ALGORITMOS GENÉTICOS

EJEMPLO PROPUESTO: EL PROBLEMA DE LA MOCHILA



https://drive.google.com/file/d/14wGUfQhQODYXEXvepV-Xh3YghadQpkJO/view?usp=drive_link