UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Artificial Intelligence

Lecture04 – Metaheuristic Search: Genetic Algorithms for Planning and Scheduling

Contenido

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

1. Introducción a la Búsqueda Metaheurística
2. Fundamentos de Algoritmos Genéticos
3. Aplicación a planificación
4. Aplicación a programación

Introducción a la Búsqueda Metaheurística

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

¿Qué es una metaheurística?

Se obtiene de anteponer al término heurística el sufijo "meta" que significa más allá o a un nivel superior.

- Son estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento.

Introducción a la Búsqueda Metaheurística

UNIVERSIDAD EAFIT

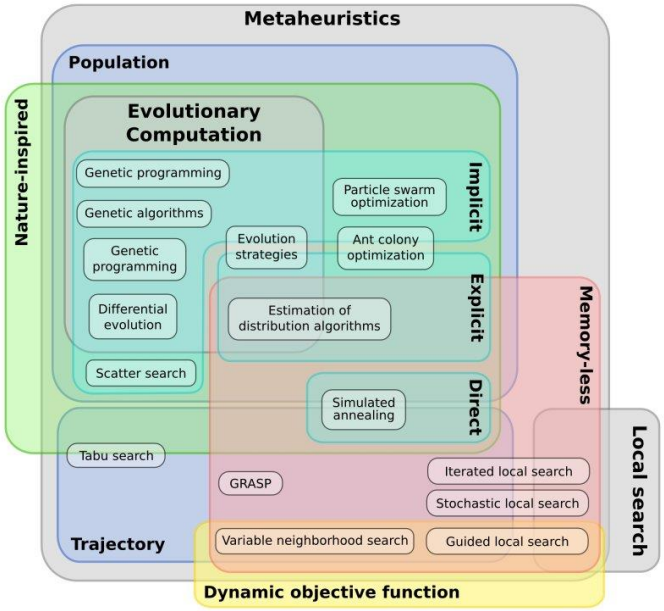
Escuela de Ciencias Aplicadas e Ingeniería

Característica	Heurística	Metaheurística
Definición	Regla práctica específica para encontrar una solución rápida.	Estrategia de alto nivel que combina y guía heurísticas para buscar mejores soluciones.
Objetivo	Obtener una solución aceptable de forma rápida.	Explorar el espacio de soluciones para encontrar soluciones cercanas al óptimo.
Alcance	Enfoque local (solución puntual).	Enfoque global (múltiples soluciones, iteración y mejora).
Tiempo de cómputo	Bajo (rápido).	Medio/alto (depende del problema y técnica).
Garantía de optimalidad	No garantiza la mejor solución.	Tampoco garantiza óptimo, pero mejora la calidad significativamente.

Tipos de metaheurísticas

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- Metaheurísticas de relajación.
- Metaheurísticas constructivas.
- Metaheurísticas de búsqueda.
- Metaheurísticas evolutivas.

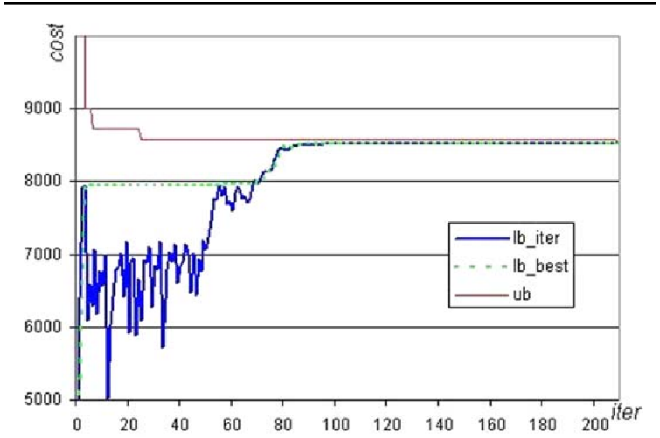


Metaheurísticas de relajación.

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Una relajación de un problema es un modelo simplificado obtenido al eliminar, debilitar, o modificar restricciones del problema real.

Los modelos muy ajustados a la realidad suelen ser muy difíciles de resolver, y sus soluciones difíciles de implementar exactamente, por lo que se acude a modelos relajados.



Metaheurísticas Constructivas.

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

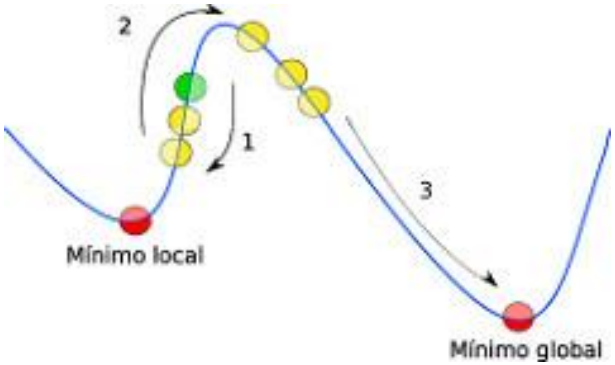
Son estrategias de optimización que construyen soluciones paso a paso, añadiendo iterativamente componentes a una solución inicialmente vacía. Estas estrategias guían el proceso de construcción mediante reglas o criterios que balancean la calidad inmediata de cada decisión con el potencial a largo plazo de la solución parcial.

Metaheurísticas de Búsqueda.

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Son estrategias de optimización que exploran el espacio de soluciones de un problema mediante transformaciones iterativas sobre soluciones iniciales, guiadas por principios de mejora local y mecanismos para evitar estancamientos en óptimos locales.

Su objetivo es encontrar soluciones de alta calidad sin necesidad de explorar todo el espacio de manera exhaustiva.



Propiedades deseables de una metaheurística

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Simple
- Precisa
- Coherente
- Efectiva
- Eficaz
- Eficiente

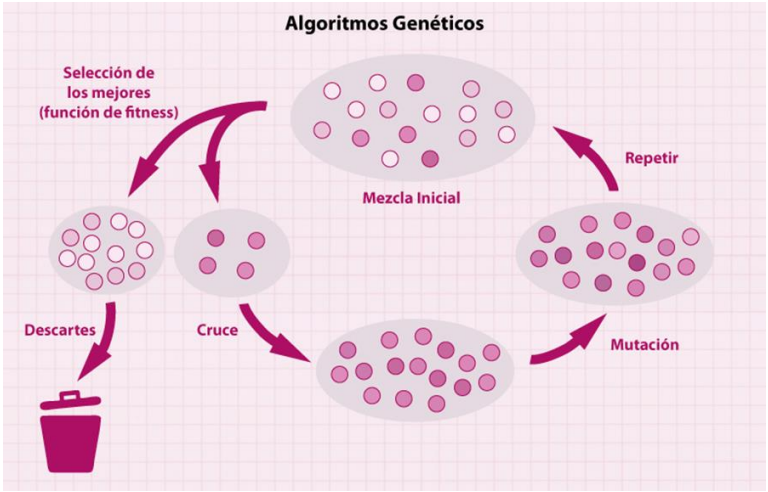
- General
- Adaptable
- Robusta
- Iterativa
- Múltiple
- Autónoma

Algoritmos Genéticos (GA): una metaheurística evolutiva

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

¿Qué son los GA?

- Inspirados en los principios de la evolución natural
- Operan sobre una población de soluciones (individuos)
- Utilizan operadores como selección, cruce y mutación
- Evolucionan soluciones a través de generaciones
- Búsqueda estocástica, guiada por una función de aptitud (fitness)



Algoritmos Genéticos (GA): una metaheurística evolutiva

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

¿Por qué son metaheurísticas?

- Realizan una búsqueda iterativa en el espacio de soluciones
- Combinan y modifican soluciones para explorar nuevas regiones
- No garantizan el óptimo, pero aproximan soluciones de alta calidad
- Se adaptan a diversos tipos de problemas: planificación, programación, diseño, optimización combinatoria

Fundamentos de Algoritmos Genéticos

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

¿Cómo funcionan los GA?

1. Población inicial
   - Conjunto de soluciones candidatas (individuos)
   - Generadas aleatoriamente o con heurísticas

2. Evaluación de aptitud (fitness)
   - Se mide la calidad de cada solución según un objetivo (e.g. tiempo, costo, distancia)

3. Selección
   - Se eligen los mejores individuos para reproducirse
   - Métodos: ruleta, torneo, ranking…

Fundamentos de Algoritmos Genéticos

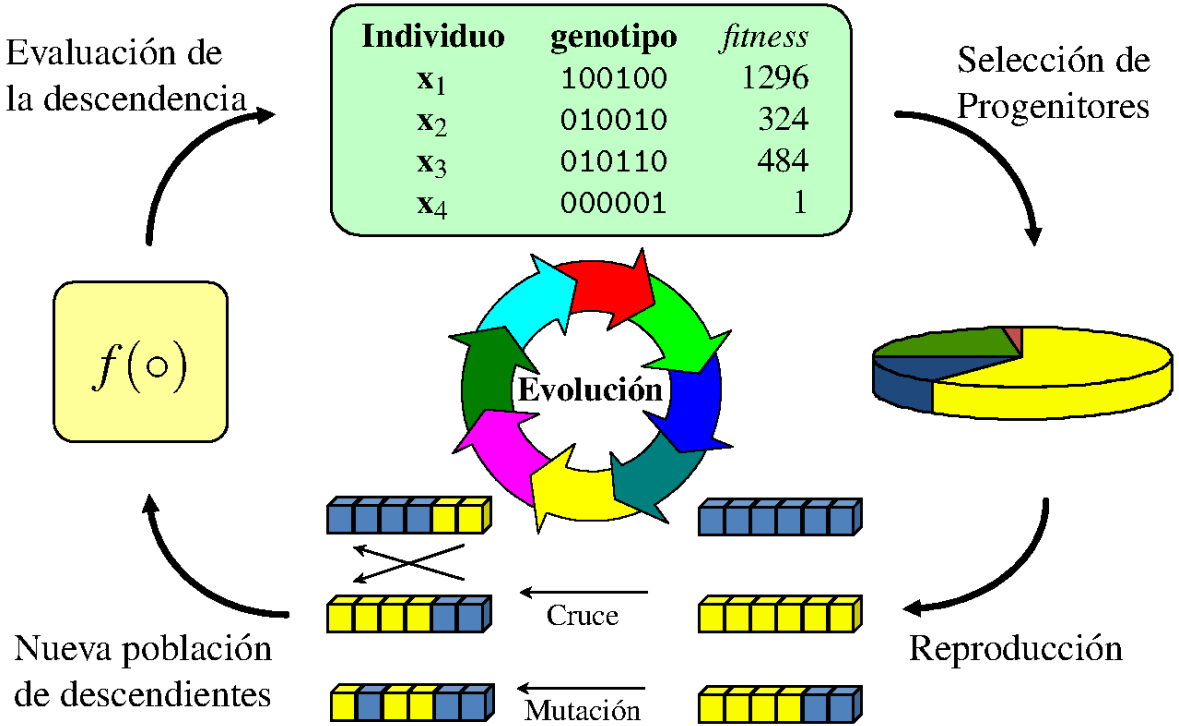
4. Cruce (crossover)
    - Se combinan partes de dos padres para crear nuevos hijos
    - Objetivo: heredar buenas características

5. Mutación
    - Se altera aleatoriamente una parte del individuo
    - Introduce diversidad en la población

6. Reemplazo
    - Se forma una nueva generación, que reemplaza total o parcialmente a la anterior

7. Criterios de parada
    - Número de generaciones, tiempo límite, mejora mínima…

Fundamentos de Algoritmos Genéticos



Pseudocódigo de GA

Podemos dar una primera formalización del proceso anterior por medio del siguiente pseudocódigo

1. *Crea población inicial*
2. *Evalúa los cromosomas de la población inicial*
3. *Repite hasta que se cumpla la condición de parada:*
    - *Selección de los cromosomas más aptos en la nueva población*
    - *Cruzamiento de los cromosomas de la población*
    - *Mutación de los cromosomas de la poblaciónE*
    - *Evaluación de los cromosomas de la población*
4. *Devuelve la mejor solución (la más apta) en la población*

Código general de GA


```
function GENETIC-ALGORITHM(population, fitness) returns an individual
  repeat
    weights ← WEIGHTED-BY(population, fitness)
    population2 ← empty list
    for i = 1 to SIZE(population) do
      parent1, parent2 ← WEIGHTED-RANDOM-CHOICES(population, weights, 2)
      child ← REPRODUCE(parent1, parent2)
      if (small random probability) then child ← MUTATE(child)
      add child to population2
    population ← population2
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to fitness

function REPRODUCE(parent1, parent2) returns an individual
  n ← LENGTH(parent1)
  c ← random number from 1 to n
  return APPEND(SUBSTRING(parent1, 1, c), SUBSTRING(parent2, c + 1, n))
```

Aplicación: GA para planificación y GA para programación

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Concepto	Planificación	Programación
Qué responde	¿Qué se debe hacer? ¿En qué orden?	¿Cuándo se debe hacer? ¿Dónde y cómo se ejecuta?
Enfoque	Definir las tareas a realizar y su secuencia lógica	Asignar recursos y tiempos específicos a cada tarea
Nivel	Más estratégico / abstracto	Más operativo / detallado
Ejemplo	Crear un plan de proyecto con dependencias entre tareas	Asignar fechas y turnos concretos a esas tareas en un cronograma
Analogía	Como el “qué y en qué orden” del viaje	Como el “cuándo, cómo y con qué recursos” del viaje

Aplicación: GA para planificación

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Problema:

Una empresa debe planificar la ejecución de 10 tareas distintas: A, B, C, D, E, F, G, H, I, J.

Estas tareas tienen restricciones de precedencia, es decir, **ciertas tareas no pueden comenzar hasta que otras hayan finalizado**.

Aplicación: GA para planificación

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Las restricciones son las siguientes:

C depende de A
D depende de A
E depende de B
F depende de C y D
G depende de E y F
H depende de G
I depende de F
J depende de H e I

Objetivo

Encontrar un orden de ejecución válido de las tareas que:

- Respete todas las restricciones de precedencia
- Minimice el desfase entre cada tarea y sus predecesoras, es decir, que las tareas que dependen de otras no se programen excesivamente tarde

Aplicación: GA para planificación

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

CODE AND CLASS EXERCISE 1

Aplicación: GA para planificación

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Preguntas

¿Por qué muchas soluciones válidas tienen el mismo fitness?

¿Qué parte del fitness penaliza más: violar precedencias o la separación entre tareas?

¿La convergencia fue rápida o lenta? ¿Por qué?

¿El algoritmo se estancó en un óptimo local o siguió explorando?

Aplicación: GA para programación

En planificación, se busca un orden válido de tareas respetando restricciones. En programación (scheduling), además de eso, se asignan tiempos y recursos. Es decir, ahora se resuelve cuándo y con qué recursos ejecutar cada tarea, buscando optimizar criterios como:

- Minimizar el tiempo total
- Reducir el número de retrasos
- Balancear carga entre recursos

Job-Shop Scheduling

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Una empresa tiene **3 máquinas (M1, M2, M3)** y debe ejecutar **5 trabajos (J1 a J5).**
Cada trabajo consiste en una **secuencia de operaciones**, que deben ejecutarse en **orden fijo** y cada una requiere una máquina específica y un tiempo.

Trabajo	Secuencia de operaciones
J1	(M1, 3) → (M2, 2) → (M3, 2)
J2	(M2, 2) → (M1, 4) → (M3, 3)
J3	(M3, 3) → (M2, 3) → (M1, 2)
J4	(M1, 2) → (M3, 1) → (M2, 4)
J5	(M2, 4) → (M3, 3) → (M1, 3)

Job-Shop Scheduling

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Objetivo:

Encontrar una programación válida de todas las operaciones que:

- Respete el orden de operaciones dentro de cada trabajo.
- No solape el uso de máquinas (una operación por máquina a la vez).
- Minimice el tiempo total de ejecución.

Job-Shop Scheduling

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

CODE AND CLASS EXERCISE 2

Aplicación: GA para programación

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Preguntas

¿Por qué se utiliza el makespan como función de aptitud?

¿Qué tipo de representación se usó para el cromosoma?

¿Qué ventajas y limitaciones tiene este tipo de representación?

Agrega un J6 con 4 operaciones. ¿Cómo afecta la dificultad del problema?

Modifica el tamaño de la población o las generaciones. ¿Cuándo el algoritmo encuentra mejores soluciones?

Ejercicio: Problema de IA

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Establecer una situación de tu vida diaria o entorno cercano que implique tomar decisiones bajo restricciones. Por ejemplo:

- Organizar horarios semanales
- Elegir qué materias cursar y cuándo
- Optimizar tu entrenamiento o tiempo libre
- Armar equipos o turnos en un grupo de trabajo
- Planificar un viaje con múltiples destinos

La tarea consiste en:

- Describir el problema en lenguaje claro (1 párrafo).
- Indicar si se trata de un problema de planificación (qué hacer y cuándo) o de programación (ordenar tareas u operaciones con recursos).
- Explicar por qué sería adecuado usar un algoritmo genético para resolverlo.
- Describir cómo representar una posible solución (cromosoma) y cómo evaluar su calidad (fitness).
- Describir los elementos del GA en tu problema: selección, cruce, mutación y criterio de parada.

References

Sancho-Caparrini, F. (s.f.). Metaheurísticas. Recuperado julio de 2025, de https://www.cs.us.es/~fsancho/Blog/posts/Metaheuristicas.md

Sancho-Caparrini, F. (s.f.). Algoritmos Genéticos. Recuperado julio de 2025, de https://www.cs.us.es/~fsancho/Blog/posts/Algoritmos_Geneticos.md.html

Berkeley University (CS188). (2022). *Completeness and Optimality – A Search**. En *CS188 Lecture Notes*. Recuperado de https://inst.eecs.berkeley.edu/~cs188/fa22/assets/notes/cs188-fa22-note02.pdf

Russell, S. J., & Norvig, P. (2020). Artificial intelligence: A modern approach (4th ed.). Pearson.