Algoritmos Bioinspirados

Unidad I : Metaheurísticas

Programa: Ingeniería en Inteligencia Artificial

- Actualmente los problemas de optimización se han vuelto extremamente complejos.
- Los métodos convencionales emplearían demasiado tiempo para resolver el problema.

Método de optimización	Naturaleza de la solución
Programación lineal y no lineal	Exacta
Branch and Bound	Exacta
Métodos heurísticos	Inexacta, cercana al óptimo
Métodos metaheurísticos	Inexacta, cercana al óptimo

- En los métodos heurísticos y metaheurísticos, se realiza un *trade-off* entre la calidad de la solución y el tiempo computacional.
- La calidad de la solución se sacrifica para obtener una solución rápida.

	Métodos heurísticos	Métodos metaheurísticos
Ejecución	Determinista	Aleatoriedad
Tipo	Iterativos	Iterativos
Ejemplo	Vecinos más cercanos	Genéticos, colonia de hormiga
Solución	Inexacta, cercana al óptimo	Inexacta, mejor probabilidad de alcanzar el óptimo

Metaheurística

Una metaheurística es un marco algorítmico de alto nivel independiente del problema que proporciona un conjunto de pautas o estrategias para desarrollar algoritmos de optimización heurística.

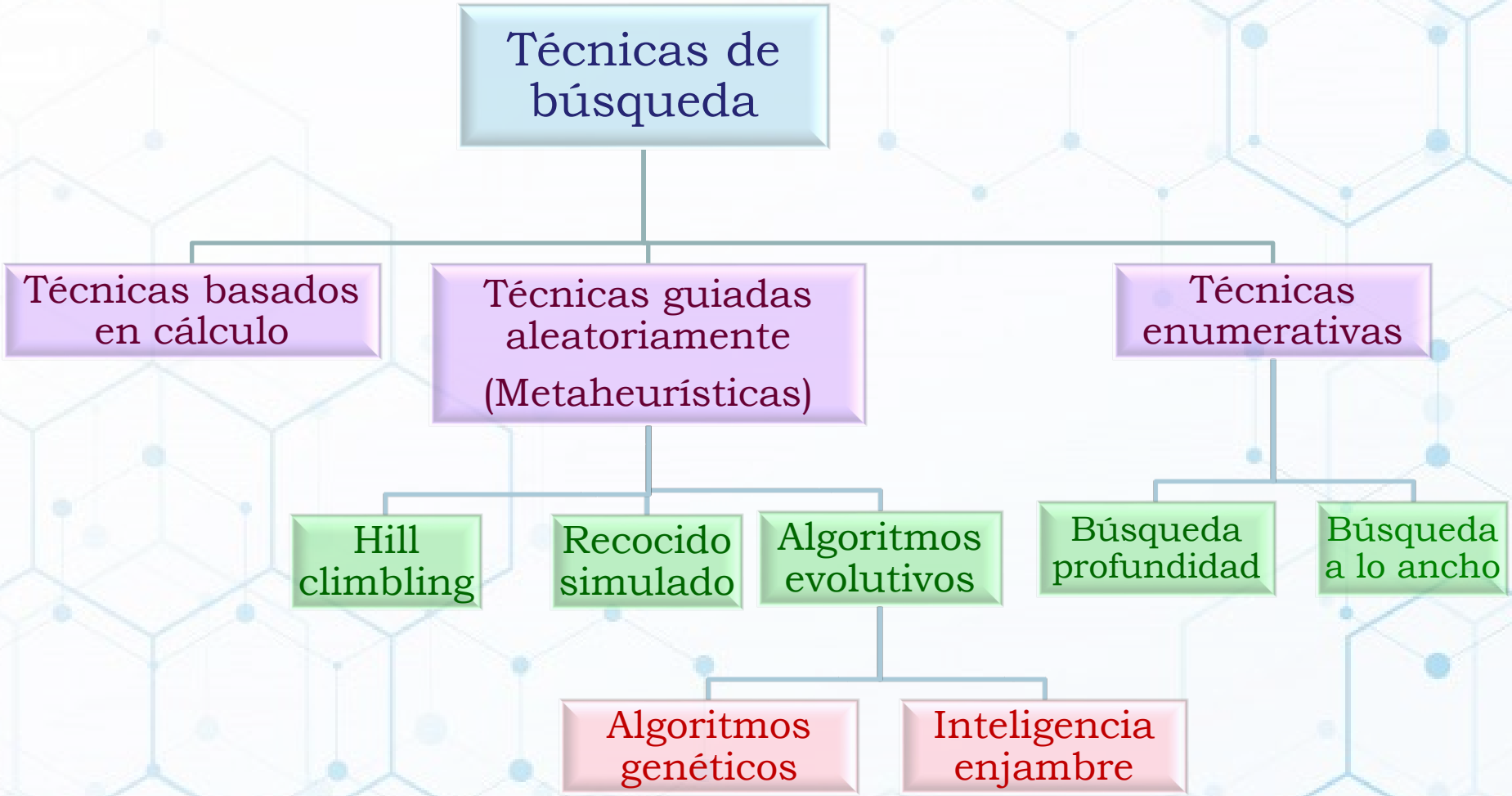
Meta = por encima, más allá (nivel más alto).

Heurística = del griego *heuriskein* o *euriskein*, buscar.

El término fue acuñado por Fred Glover en 1986.

Una metaheurística:

- Guía una heurística subordinada de forma iterativa.
- Técnicas de búsqueda guiadas aleatoriamente.
- Usa exploración y explotación del espacio de búsqueda.
- Generalmente incluyen métodos para evitar quedar atrapados en óptimos locales.
- Encuentra soluciones cercanas al óptimo.


Técnicas de búsqueda
Técnicas basados en cálculo
Técnicas guiadas aleatoriamente (Metaheurísticas)
Técnicas enumerativas
Hill climbling
Recocido simulado
Algoritmos evolutivos
Búsqueda profundidad
Búsqueda a lo ancho
Algoritmos genéticos
Inteligencia enjambre

Ascenso de Colina (Hill climbing)

Idea:

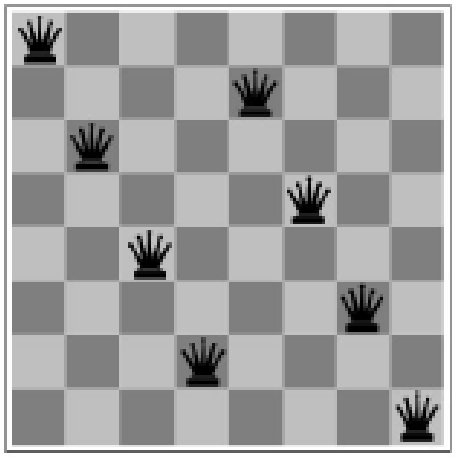
- Iniciar desde algún estado S.
- Moverse a un vecino v con un mejor score.

Pregunta: ¿Qué es un vecino?

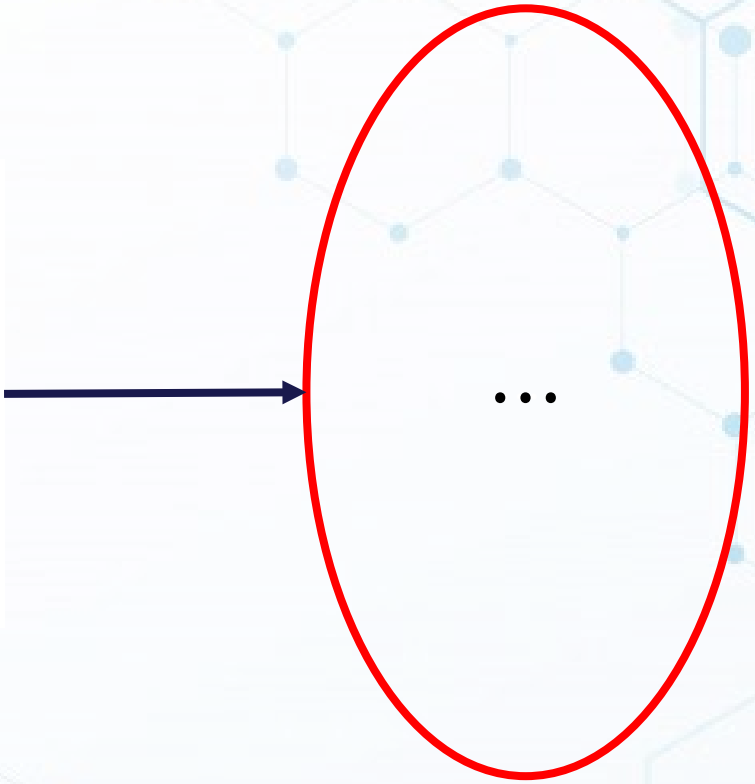
- Definido para cada problema.
- El vecindario de un estado es el conjunto de vecinos.
- También llamado conjunto de movimiento.

Ejemplo: problema de las 8 reinas.

s



$f(S) = 1$

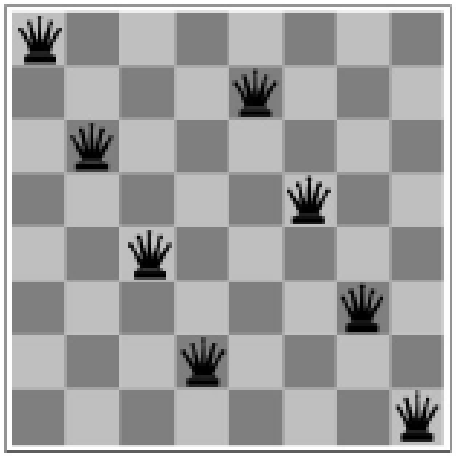


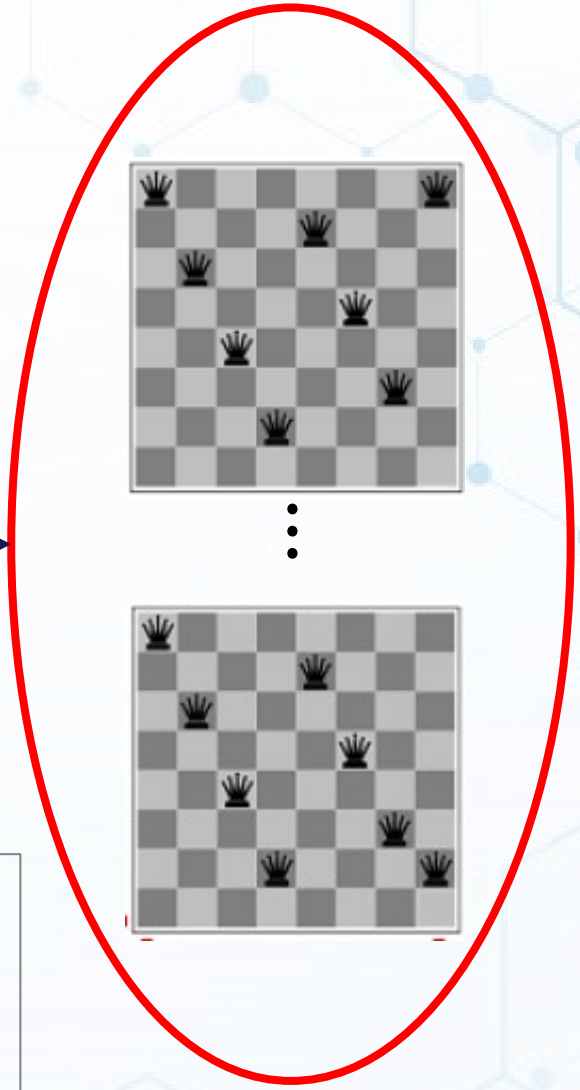
Vecindario de S

Ejemplo: problema de las 8 reinas.

s

$f(S) = 1$





$f(S) = 1$

$f(S) = 2$

Vecindario de S

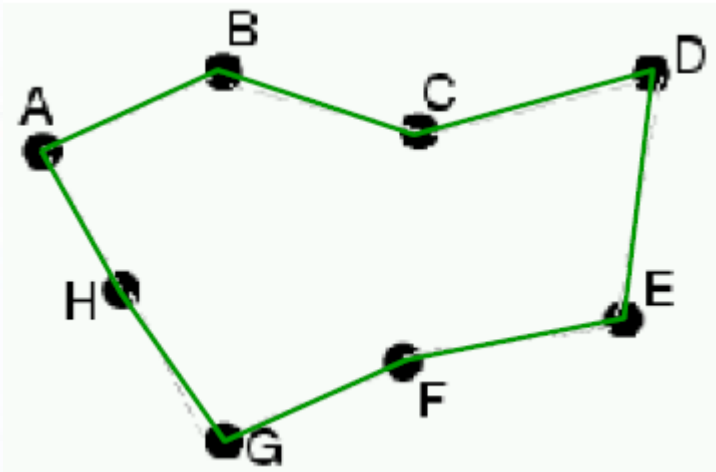
Heurística:

1. Elegir la columna más a la derecha que tenga conflicto.
2. Mover la reina en esa columna sobre la vertical un número dado de posiones.

Ejemplo: agente viajero.

$S = (A, B, C, D, E, F, G, H, A) \rightarrow$ estado

$f(S) =$ longitud del camino



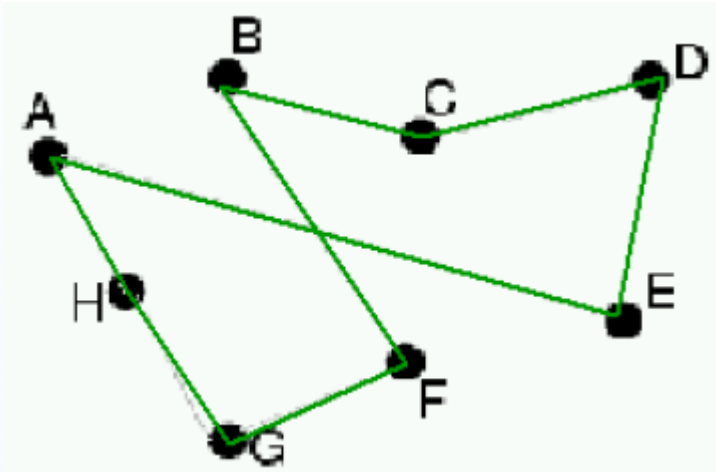
Heurística:

1. Seleccionar segmento del camino.
2. Cambiar los elementos en los extremos.

A-B-C-D-E-F-G-H-A

Intercambiar

A-E-C-D-B-F-G-H-A



Pregunta: ¿Qué es un vecino?

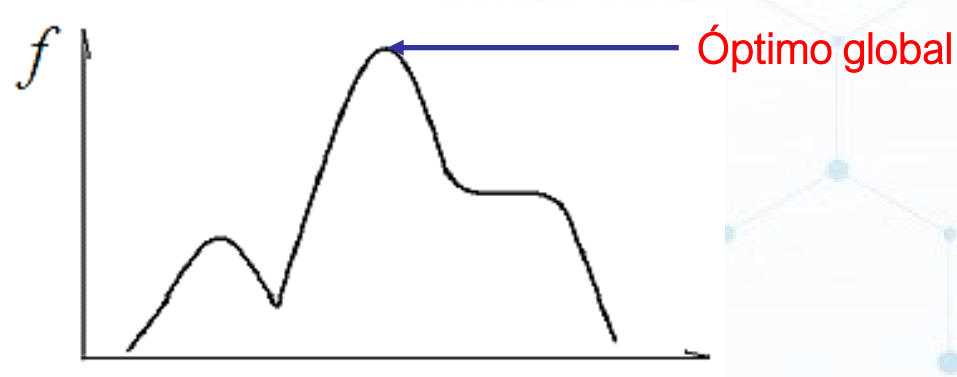
- Los problemas tienden a tener estructuras. Un pequeño cambio produce un estado vecino.
- El vecindario debe ser lo suficientemente pequeño para lograr eficiencia.
- Diseñar el vecindario es fundamental. Éste es el verdadero ingenio, no la decisión de usar ascenso de colina.
- ¿Qué vecino elegir? El mejor (codicioso)
- ¿Qué pasa si ningún vecino es mejor que el estado actual? STOP. (¡Doh!)

Algoritmo de Ascenso de Colina

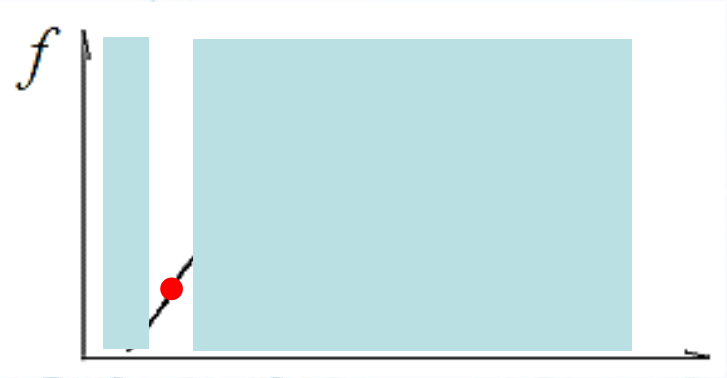
1. Elegir estado inicial S.
2. Elegir el mejor vecino v en vencidario(S)
3. Si $f(S)\ es\ mejor\ que\ f(v)$ entonces STOP, retornar S.
4. Sino $S = v$. Ir al paso 2.

- No es el algoritmo más sofisticado del mundo
- Muy codicioso.
- Se queda estancado fácilmente .

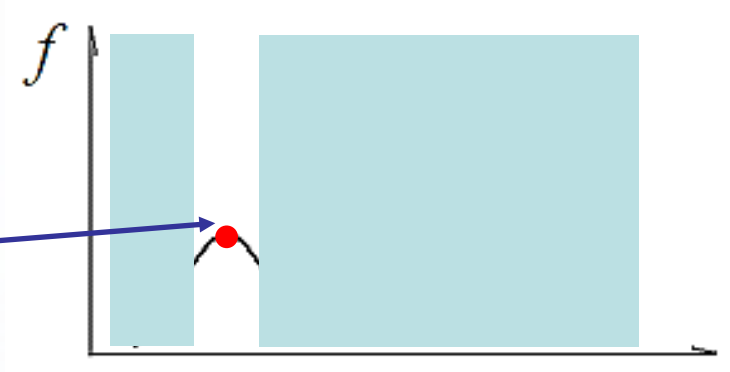
- Supongamos la siguiente función f:



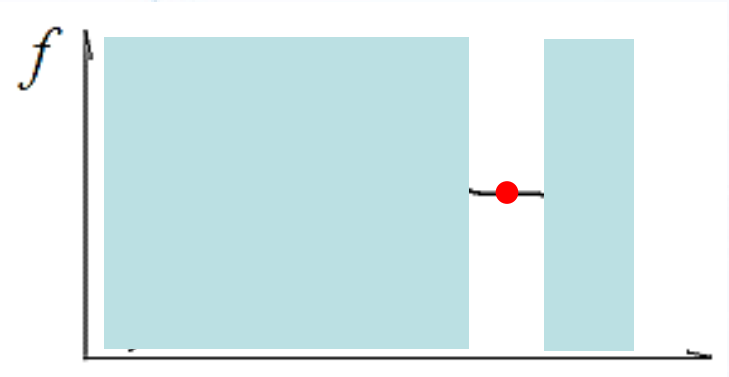
- Pero el algoritmo no es capaz de ver todo el entorno, solo su vecindario (agreguemos algo de niebla).



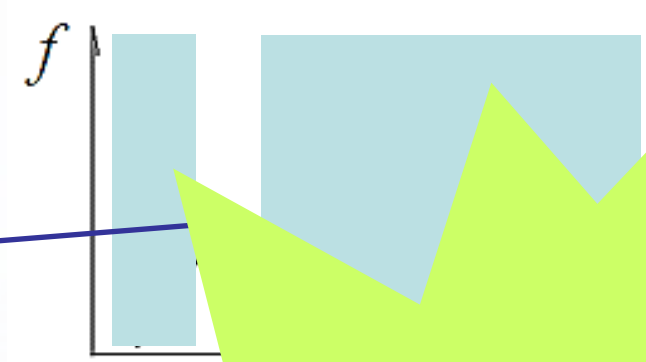
- Óptimo local



- Y ahora ¿ A dónde voy?



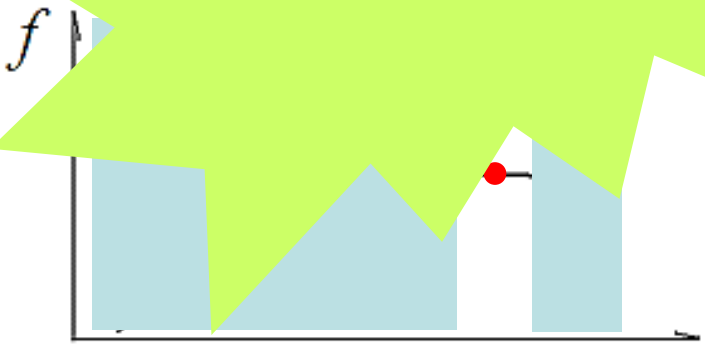
- Óptimo local



Declaro la cima del mundo

- Y ahora ¿ A dónde vo

Debemos escapar de los óptimos locales



Algoritmo de Ascenso de Colina con reinicios aleatorios

Modificación sencilla:

- Cuando estés atascado, elige un estado inicial aleatorio y ejecuta el ascenso de colina básico a partir de ese estado.
- Repetir k veces.
- Seleccionar como solución el mejor de los k óptimos locales.

Repetir k veces:

1. Elegir estado inicial S.
2. Elegir el mejor vecino v en vecidario(S).
3. Si $f(S)\ es\ mejor\ que\ f(v)$ entonces guardar S e ir al paso 1
4. Sino $S = v$. Ir al paso 2.

Retornar el mejor entre los k óptimos locales

- Aún con modificaciones el Ascenso de Colina sigue siendo un **algoritmo voraz que sólo se mueve hacia arriba**.

- Observación importante:

A veces uno necesita dar un paso atrás temporalmente para poder avanzar.

=

A veces es necesario desplazarse hacia un vecino inferior para escapar de un óptimo local.

¡ Gracias !

Thanks !

Obrigado

Xie xie ni

Domo arigatou

Спасибо

Merci

Grazie