

Panorámica de los procedimientos metaheurísticos

ABRAHAM DUARTE

Optimización

- En lenguaje coloquial, **optimizar** significa mejorar
- En el contexto científico, es el proceso de tratar de encontrar la **mejor** solución posible para un **problema** determinado



Problema de optimización

- El **objetivo** de un problema de optimización es encontrar la solución **óptima** dado un criterio para **discriminar** entre dos soluciones
- Es decir, se trata de encontrar el valor de unas **variables** de decisión (sujetas a **restricciones**) para los que una determinada función objetivo alcanza su valor **máximo** o **mínimo**

Optimización

- **Representación:** codifica las soluciones factibles para su representación. Determina el tamaño del espacio de búsqueda del problema
- **Objetivo:** modelo matemático que expresa la tarea a realizar
- **Función de evaluación:** asocia a cada solución factible un valor que determina su calidad.

Optimización

- Dado un **dominio** X y una **función**

$$f(X): x \in X \rightarrow R$$

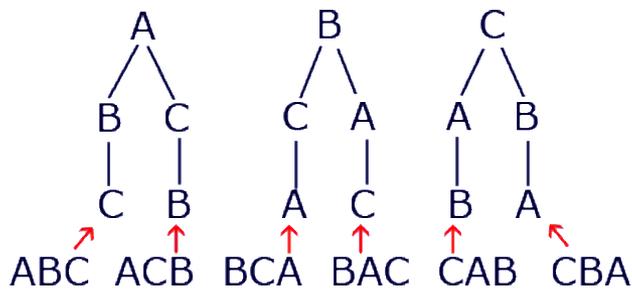
el **objetivo** es encontrar un x que verifique

$$x^* \in X: f(x^*) \geq f(x), \forall x \in X$$

- **Optimización combinatoria:** consiste en encontrar un objeto en un conjunto finito de soluciones

Tipos de problemas combinatorios

- Según la representación de la solución
 - Permutaciones: problemas de ordenación
 - Binarios: problemas de pertenencia
 - Enteros: problemas de cardinalidad
 - Otros (mistos, subgrafos, árboles, etc.)



00111111
11111010
01110101
11011100
10101011
01011110

1221334
4433112
2214432
1112234

Tipos de problemas

- Fáciles de resolver
 - **Lineales:** función objetivo y restricciones lineales (método Simplex)
- Difíciles de resolver (*NP*-difícil)
 - No podemos garantizar encontrar la **mejor** solución en un tiempo **razonable**
 - La **mayoría** de problemas de aplicación **práctica**
 - Desarrollamos procedimientos **eficientes** para encontrar soluciones de **calidad**©

Tipos de problemas

Máximo $Z = 3X_1 + 2X_2$; sujeta a las restricciones :

$$X_1 + X_2 \leq 6 \quad \dots (1)$$

$$2X_1 - X_2 \geq 0 \quad \dots (2)$$

$$X_1 = 2 \quad \dots (3)$$

$$X_2 \geq 0$$

Forma estándar

$$\text{Máximo } Z = 3X_1 + 2X_2$$

$$\begin{array}{rclcl} X_1 + X_2 + 1H_1 & = & 6 & X_1 + X_2 + H_1 & = & 6 \dots (1) \\ 2X_1 - X_2 - 1S_2 & = & 0 & 2X_1 - X_2 - S_2 + W_2 & = & 0 \dots (2) \\ X_1 & = & 2 & X_1 + W_3 & = & 2 \dots (3) \end{array}$$

$$X_2 \geq 0, H_1 \geq 0, S_2 \geq 0$$

Base artificial

$$X_2 \geq 0, H_1 \geq 0, S_2 \geq 0, W_2 \geq 0, W_3 \geq 0$$

Penalización de la función objetivo :

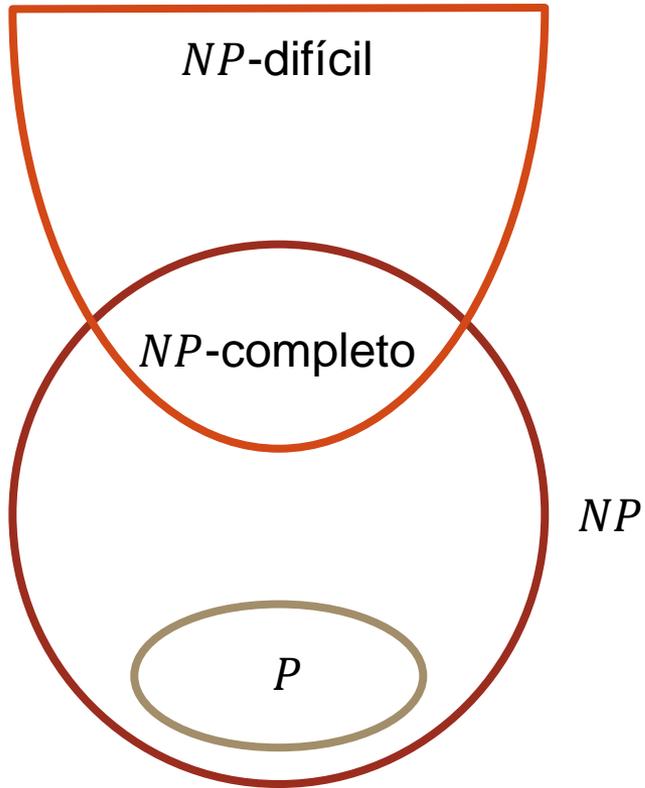
$$\text{Máximo } Z = 3X_1 + 2X_2 - MW_2 - MW_3$$

$$\text{Máximo } Z - 3X_1 - 2X_2 + MW_2 + MW_3 = 0$$

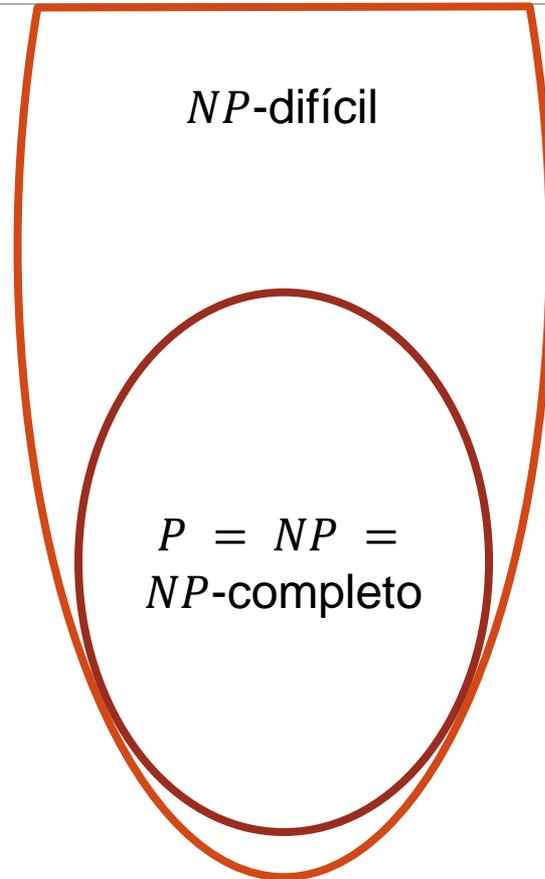
$$\text{Matriz unitaria con holgura } H_1 \text{ y las artificiales } W_i : I = \begin{pmatrix} H_1 & W_2 & W_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Optimalidad →	Base	Z	DECISIÓN		HOLG	SUPER.	ARTIFICIAL		Factibilidad	
			X_1	X_2	H_1	S_2	W_2	W_3	Sol.	
	Z	1	-3	-2	0	0	M	M	0	
	H ₁	0	1	1	1	0	0	0	6	↓
	W ₂	0	2	-1	0	-1	1	0	0	
	W ₃	0	1	0	0	0	0	1	2	
	Z'	1	-2M-3	M-2	0	M	0	M	0	
	Z''	1	-3M-3	M-2	0	M	0	0	-2M	
	H ₁	0	↑ VE 1	1	1	0	0	0	6	6/1=6
	W ₂	0	2 P ←	-1	0	-1	1	0	0	0/2=0
	W ₃	0	1	0	0	0	0	1	2	2/1=2
	Z	1	0	$-\frac{1}{2}M - \frac{7}{2}$	0	$-\frac{1}{2}M - \frac{3}{2}$	$\frac{3}{2}M + \frac{3}{2}$	0	-2M	
	H ₁	0	0	↑ VE $\frac{3}{2}$	1	$\frac{1}{2}$	$-\frac{1}{2}$	0	6	6/3=4
	X ₁	0	1	$-\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	No vát.
	W ₃	0	0	$\frac{1}{2}$ P ←	0	$\frac{1}{2}$	$-\frac{1}{2}$	1	2	2/1/2=4
	Z	1	0	0	0	2	M-2	M+7	14	Sol.
	H ₁	0	0	0	1	-1	1	-3	0	Dege-
	X ₁	0	1	0	0	0	0	1	2	nerada
	X ₂	0	0	1	0	1	-1	2	4	←

Tipos de problemas

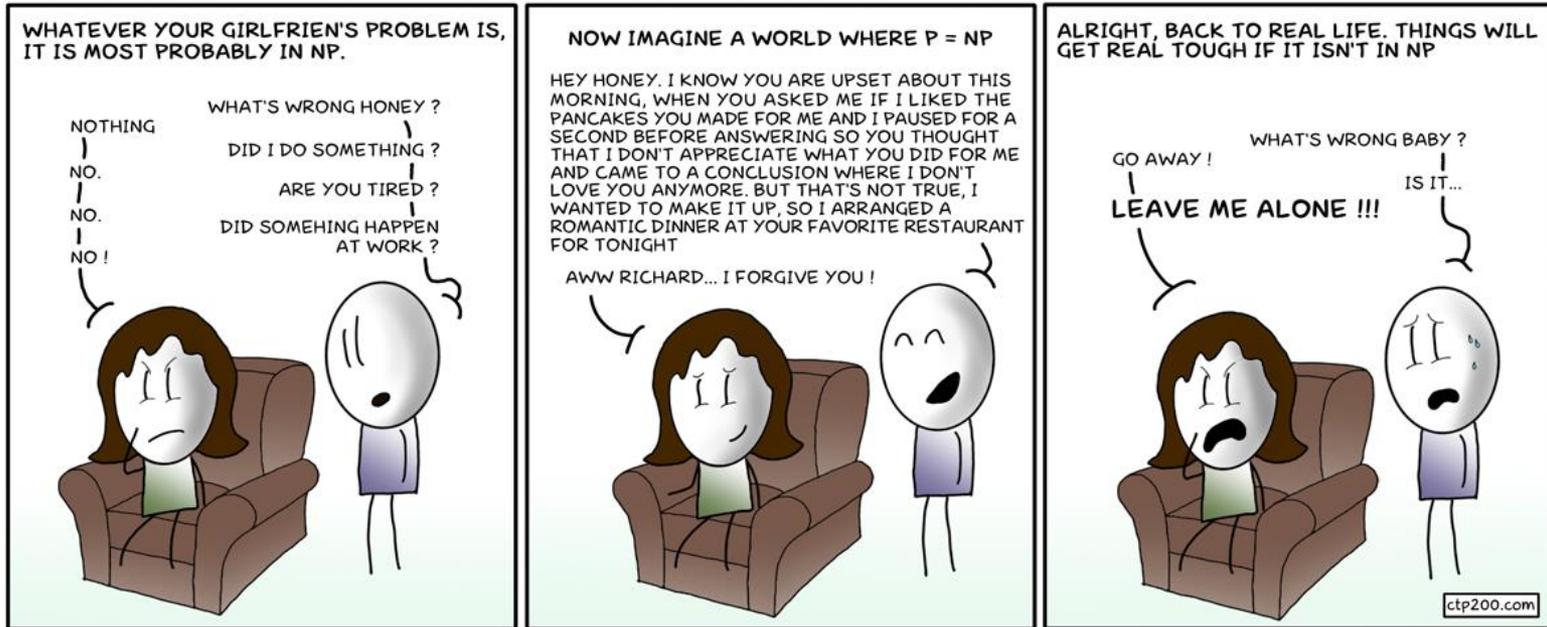


$P \neq NP$



$P = NP$

Tipos de problemas

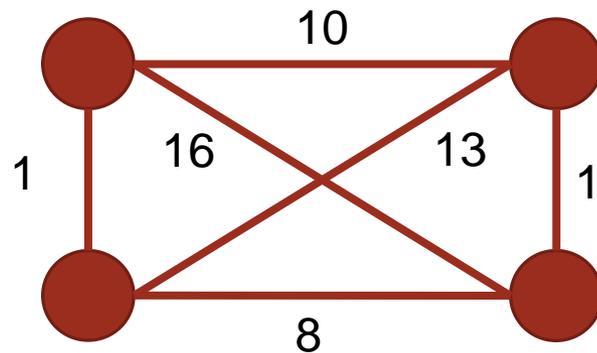


Ejemplos de problemas

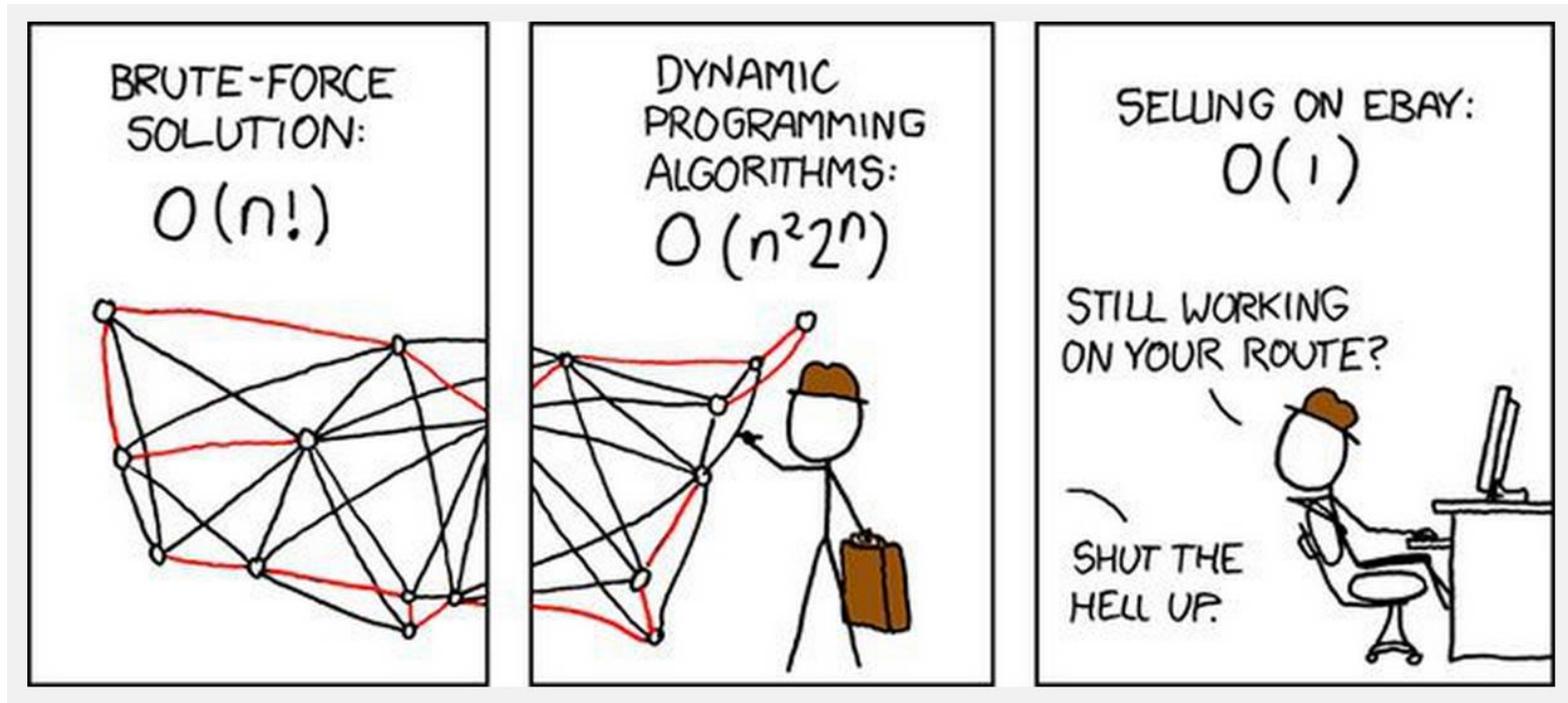
- Viajante
- Mochila
- Coloreado de grafos
- Partición de conjuntos
- Ordenación lineal
- Diversidad
- Enrutado de vehículos

Problema del viajante

- *Travelling Salesman Problem*, TSP. Un comerciante tiene que visitar n ciudades, comenzando y finalizando en su propia ciudad. Conociendo el coste de ir de una ciudad a otra, encontrar el recorrido de coste mínimo.



Problema del viajante



Problema del viajante

- Número de soluciones en el TSP
 - 10 ciudades. 10^5 (aprox.)
 - 20 ciudades. 10^{12} (aprox.)
 - 50 ciudades. 10^{62} (aprox.)
 - Edad del universo.
 - 15000 millones de años $\rightarrow 10^{25}$ nanosegundos

Problema de la mochila

- *Knapsack Problem*. Dados n objetos, cada uno con un peso w_i y un valor v_i , se debe seleccionar el conjunto de objetos que maximicen el valor, sin exceder el peso máximo W de la mochila

4€
12 kg

10€
4 kg

15 kg

2€
2 kg

1€
3 kg

2€
1 kg

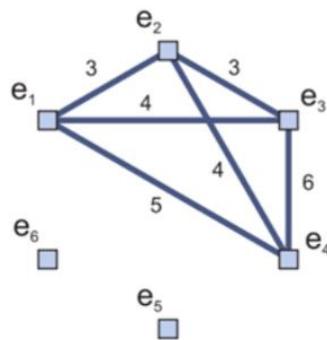
Problemas de la diversidad

- Separación de antenas en redes de telecomunicaciones
 - Determinar **dónde** colocar la antena para maximizar la cobertura de una región
 - Si disponemos de N **localizaciones**, dónde colocamos las M **antenas**
 - Objetivo: aumentar la distancia **mínima** entre cada par de antenas

Problemas de la diversidad

○ Ejemplo de modelado

- $N = 6$ localizaciones y hay que colocar $M = 4$ antenas
- Se eligen las localizaciones e_1, e_2, e_3 y e_4 , donde la mínima distancia es $d(e_1, e_2) = d(e_2, e_3) = 3$
- El objetivo del problema es aumentar la distancia



	e_1	e_2	e_3	e_4	e_5	e_6
e_1	-	3	4	5	7	5
e_2	3	-	3	4	8	1
e_3	4	3	-	6	4	7
e_4	5	4	6	-	5	9
e_5	7	8	4	5	-	8
e_6	5	1	7	9	8	-

Problema de la diversidad

- Seleccionar un conjunto de elementos de una colección de forma que los elementos seleccionados tengan las características más variadas entre sí

$$\text{Maximizar } z_{MM}(x) = \min_{i < j} d_{ij} x_i x_j$$

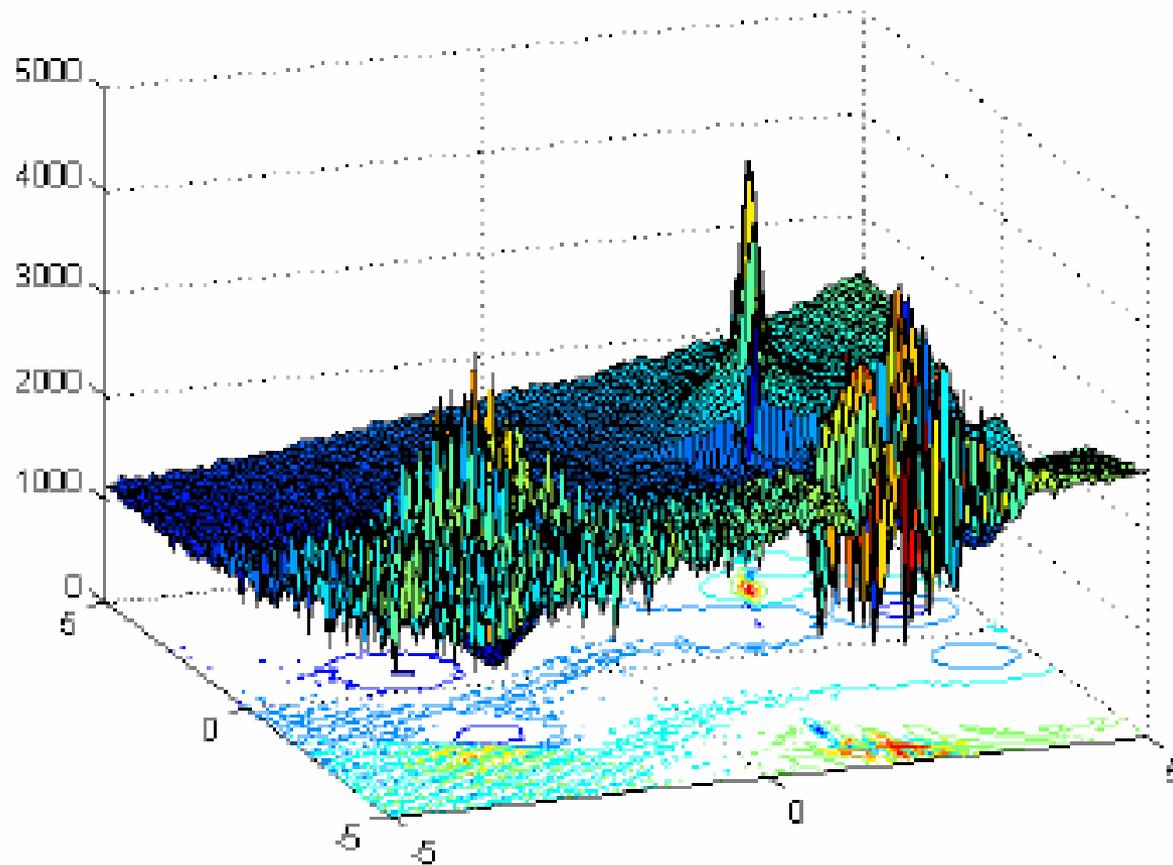
$$\text{Sujeto a } \sum_{i=1}^n x_i = m$$

$$x_i = \{0, 1\}, \quad i = 1, \dots, n.$$

Tipos de óptimos

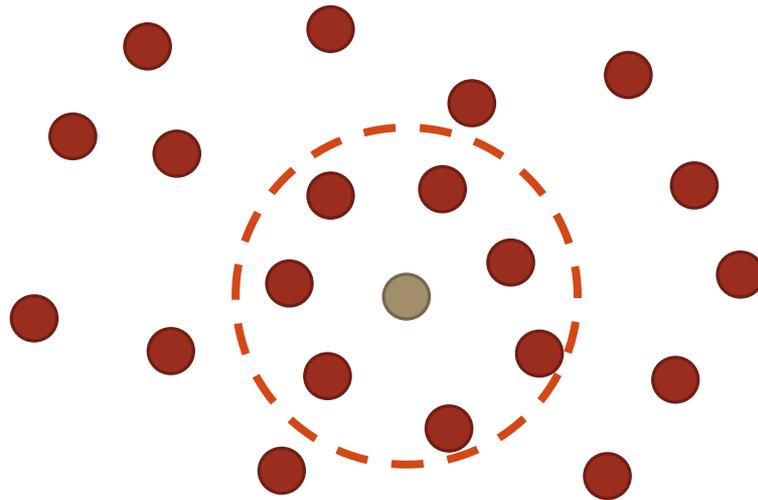
- **Óptimo global:** se trata de la mejor solución que se puede encontrar a un problema
- **Óptimo local:** es la mejor solución de un problema dentro de una vecindad determinada
- Un óptimo **global** es óptimo local de **cualquier vecindad**

Tipos de óptimos



Vecindad

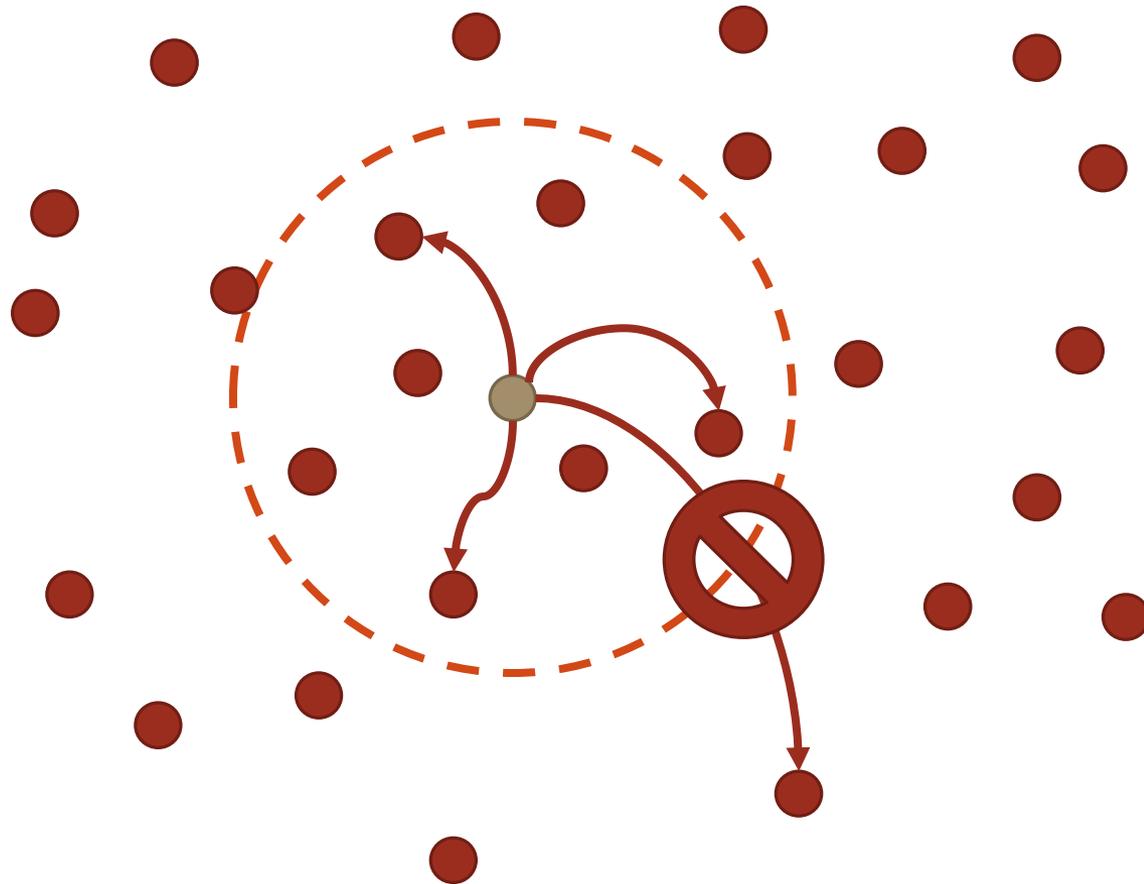
- Dada una **solución** x , la vecindad $N(x)$ de esa solución es un **subconjunto** del espacio de soluciones que contiene soluciones **próximas** a la original



Movimiento

- En una **vecindad** $N(x)$ de una solución x se encuentran todas aquellas soluciones x' accesibles desde x a través de un **movimiento**
- Dicho de otro modo, dada una solución x , cada solución de su vecindad $x' \in N(x)$ puede obtenerse **directamente** desde x mediante una operación llamada **movimiento**

Movimiento



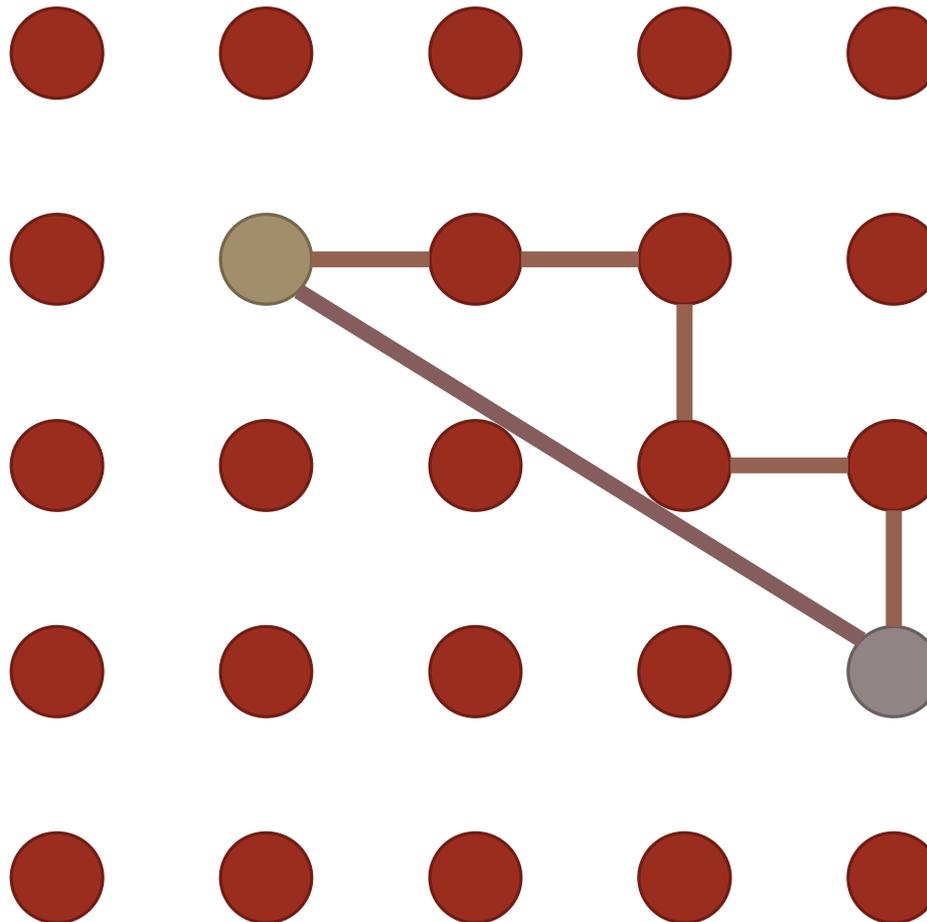
Distancia

- Es posible definir la **distancia** $dist(x, x')$ entre dos soluciones x, x'
- De la misma forma, podemos **redefinir** la vecindad como

$$N(x) = \{x' \in SS: dist(x, y) \leq \epsilon\}$$

- **Ejemplos:** distancia Euclídea, distancia de Hamming, etc.

Distancia



Heurísticas y metaheurísticas

- Para la mayoría de los problemas con interés práctico, los **algoritmos exactos** no son una alternativa realista
 - El **tamaño** del problema hace que éste sea **ineficiente** computacionalmente
 - Ejemplo del viajante

Heurísticas y metaheurísticas

- Hay condiciones **difíciles** de modelar que exigen **flexibilidad**
- Se necesitan soluciones **aproximadas** como parte de un procedimiento global que garantice el **óptimo**
 - El **heurístico** proporciona una **buena** solución inicial y participa en el paso intermedio del procedimiento

Heurísticas y metaheurísticas

- **Heurísticas:** procedimientos simples basados en el sentido común que obtienen una buena solución a problemas difíciles de un modo sencillo y rápido
- **Metaheurísticas:** procedimiento iterativo maestro que guía y modifica las operaciones de una heurística subordinada para producir eficientemente soluciones de alta calidad

Exacto vs heurístico

Exactos	Heurísticos
Solución óptima	Solución de calidad (óptimo no garantizado)
El tiempo invertido para obtener el óptimo de un problema difícil puede ser desproporcionado	El tiempo suele ser reducido
Útiles para problemas pequeños	Útiles cuando el problema es grande y necesitamos generar soluciones en un corto espacio de tiempo

Medidas de calidad

- **Eficiencia:** un esfuerzo computacional realista para obtener la solución
- **Bueno:** la solución debe estar, en promedio, cerca del óptimo
- **Robusto:** la probabilidad de obtener una mala solución debe ser baja

Medidas de calidad

- Se mide, para cada ejemplo analizado, la **desviación** porcentual de la **solución** heurística frente a una **referencia**, calculando el promedio de las desviaciones
 - Comparación con el óptimo
 - Comparación con una cota
 - Comparación con un método exacto truncado
 - Comparación con otros heurísticos
 - Análisis del peor caso

Medidas de calidad

- La **rapidez** de un heurístico es tan importante como la **calidad** de la solución obtenida
- Un método heurístico es un procedimiento para **resolver** un problema de optimización bien definido mediante una **aproximación intuitiva**, en la que la **estructura** del problema se utiliza de forma **inteligente** para obtener una **buena** solución.

Clasificación de heurísticos

- **Heurísticos constructivos:** construyen una solución a un problema dado sin disponer de una solución previa. La construcción depende de la estrategia elegida
- **Heurísticos de búsqueda:** partiendo de una solución factible dada, intentan mejorarla

Heurísticos constructivos

- **Estrategia voraz:** Partiendo de una semilla, construyen iterativamente una solución
- En cada paso, se **añade** a la solución el elemento que produzca la **mayor mejora** en la solución parcial
- **Visión miope:** eligen la mejor opción para la siguiente iteración sin importar lo que suceda en el futuro

Heurísticos constructivos

- **Estrategia de descomposición:** Se divide el problema en subproblemas más pequeños hasta que el tamaño del subproblema es trivial
- El algoritmo **combina** las soluciones en cada paso hasta obtener la solución al problema original
- Ejemplo: divide y vencerás

Heurísticos constructivos

- **Estrategia de reducción:** identifican características que contienen las soluciones buenas conocidas y se asume que la solución óptima también las tendrá
 - Reducción drástica del espacio de búsqueda
- **Manipulación del modelo:** simplifican el modelo del problema original, se soluciona el modelo simplificado y se extrapola al problema original

Heurísticos constructivos

- **Estrategia aleatorizada:** dada una solución factible y una vecindad asociada, se seleccionan aleatoriamente soluciones dentro de la vecindad
- Su **éxito** depende de la selección de vecindades **prometedoras**

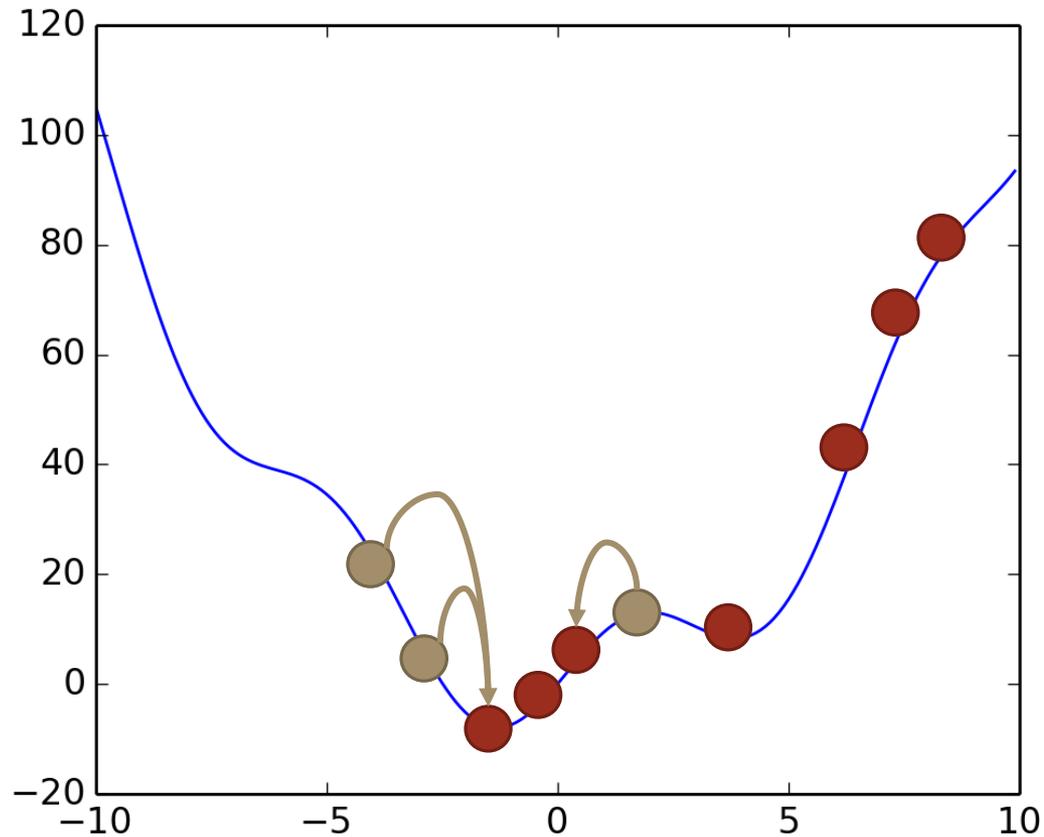
Heurísticos de búsqueda

- **First improvement:** examina la vecindad de una solución y selecciona el primer movimiento que produce una mejora
- **Best improvement:** examina todos los posibles movimientos en una vecindad, seleccionando el que produce la mayor mejora

Limitaciones de los heurísticos

- **Dependen** en gran medida del problema **concreto** para el que se han diseñado
- Las técnicas e ideas aplicadas a la resolución de un problema son **específicas** de éste
- Es difícil **trasladar** el aprendizaje a otros problemas
 - Han de particularizarse para cada caso

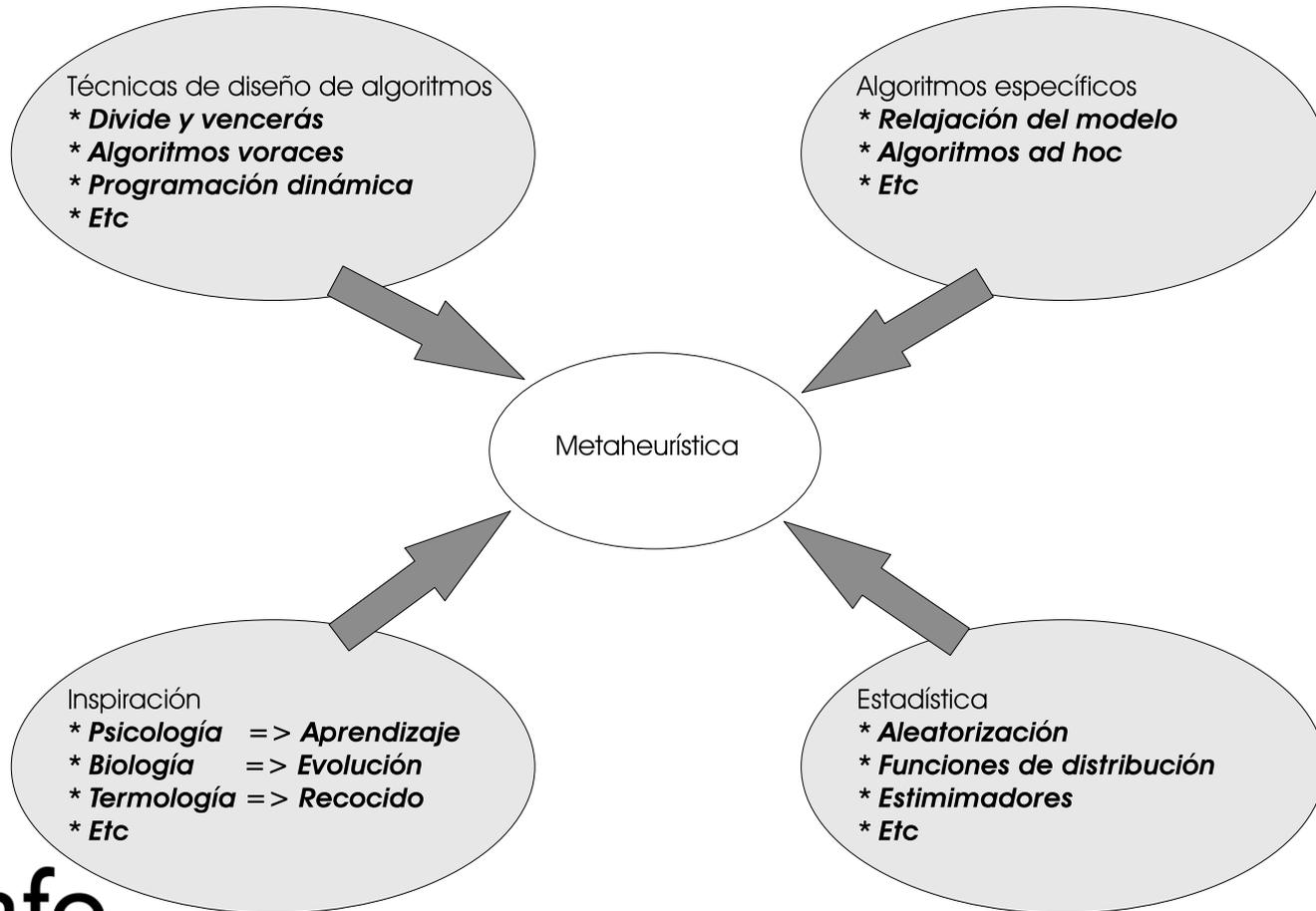
Limitaciones de los heurísticos



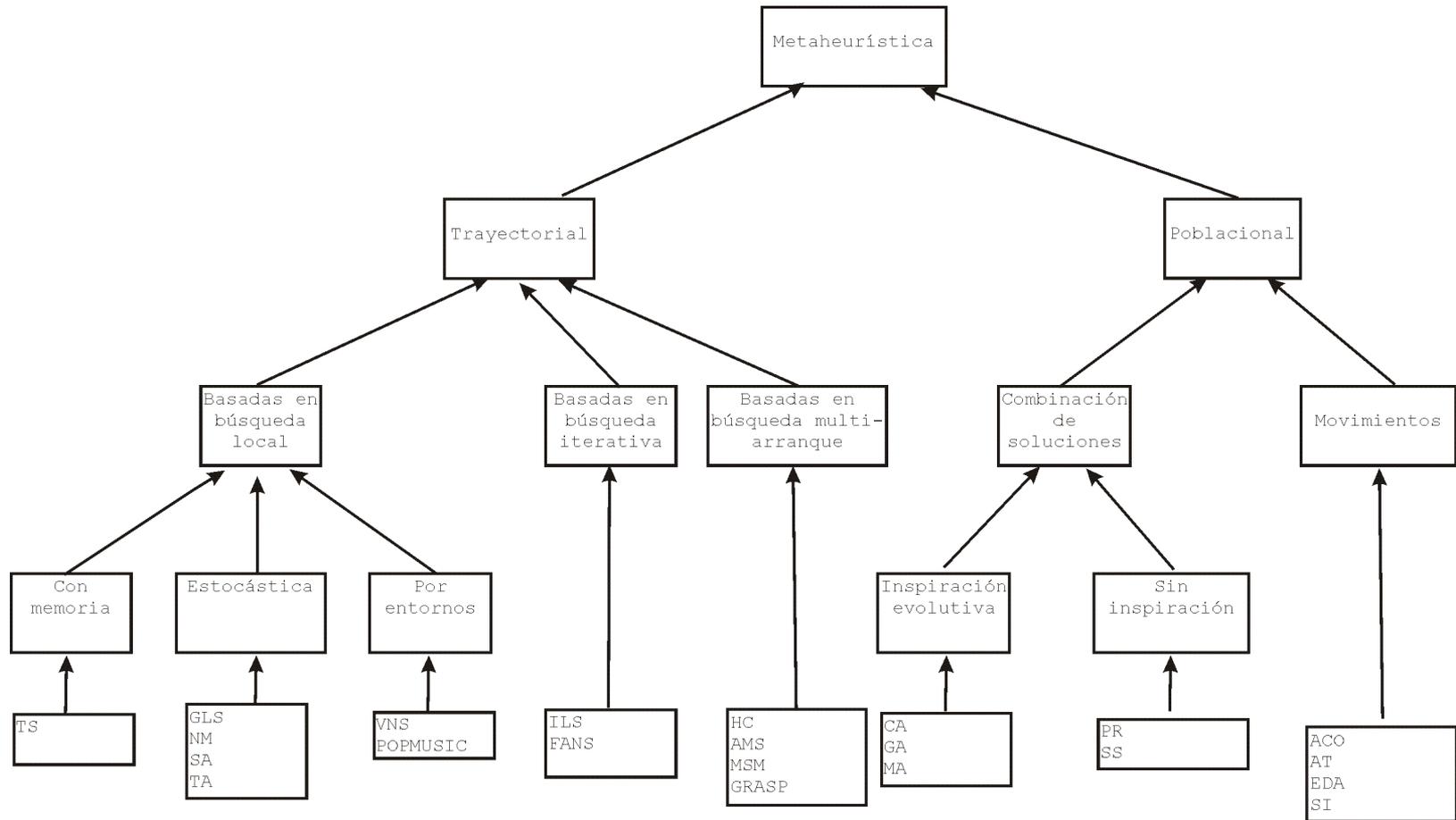
Metaheurísticas

- Propósito: obtener **mejores** resultados que los obtenidos por los **heurísticos** tradicionales
- El término fue introducido por **Fred Glover** en 1986
- Los procedimientos se sitúan “**por encima**” de los heurísticos, **guiando** su comportamiento

Metaheurísticas



Taxonomía

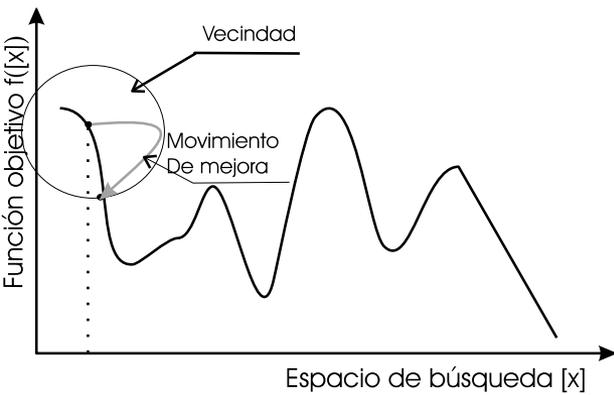


Ejemplos trayectoriales

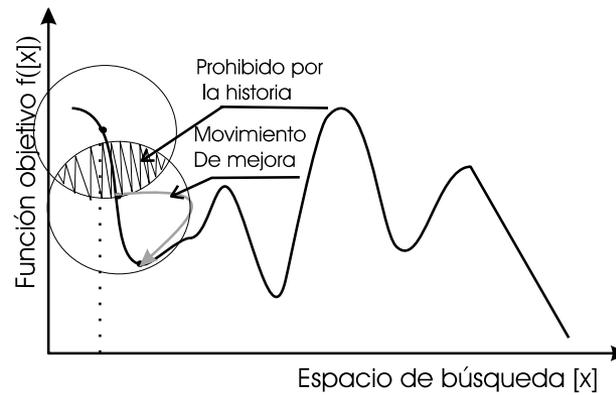
- Búsqueda Tabu (*Tabu Search* – TS)
- Búsqueda de Vecindad Variable (*Variable Neighborhood Search* – VNS)
- GRASP - *Greedy Randomized Adaptive Search Procedures*
- Búsqueda local iterada (*Iterated Local Search* – ILS)
- Recocido Simulado (*Simulated Annealing* – SA)

Ejemplos trayectoriales

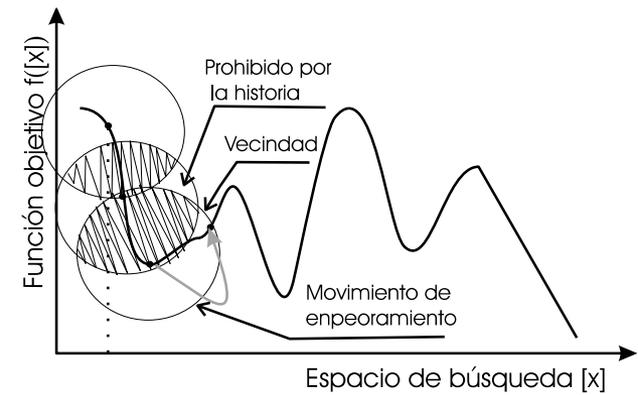
○ *Tabu Search* - TS



(a)



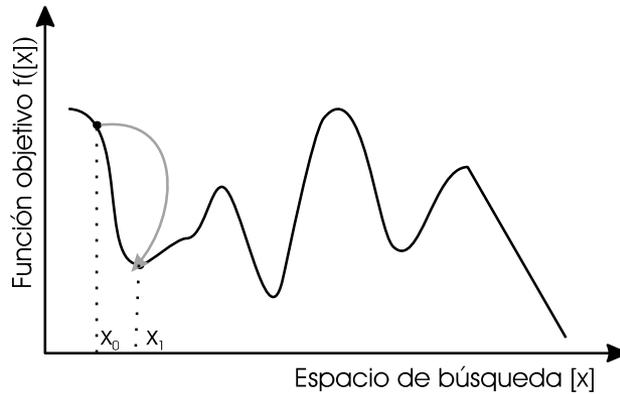
(b)



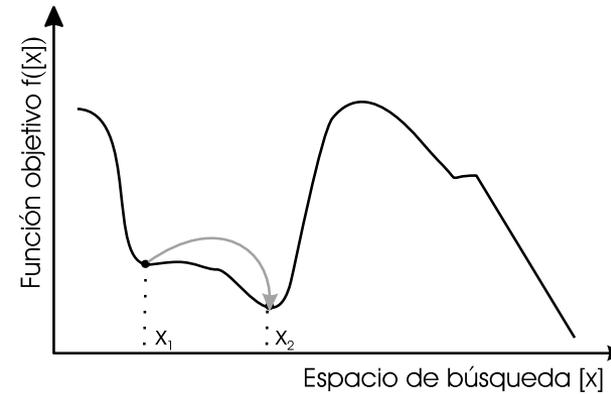
(c)

Ejemplos trayectoriales

- *Variable Neighborhood Search – VNS*



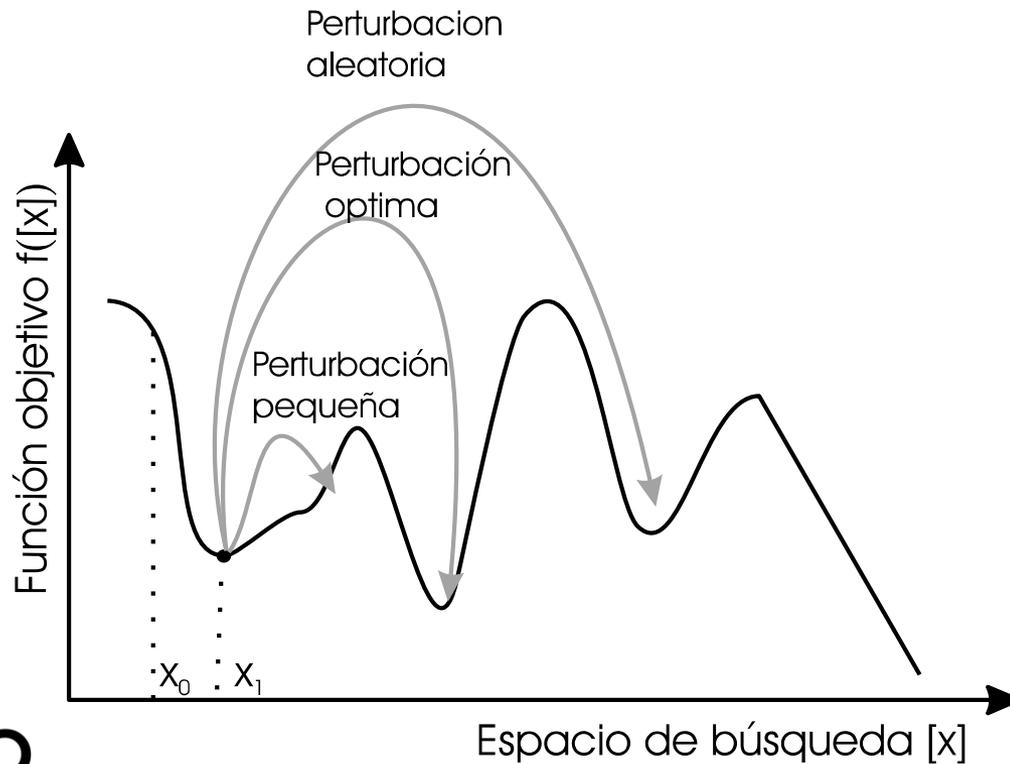
(a)



(b)

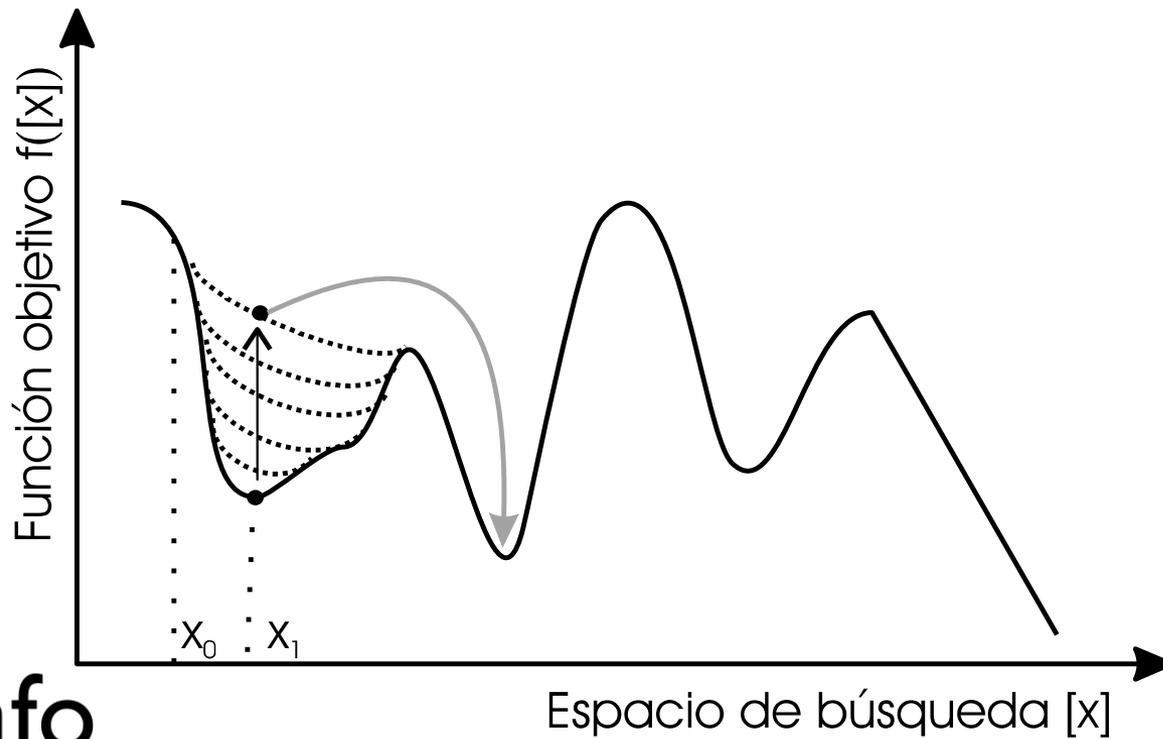
Ejemplos trayectoriales

- *Iterated Local Search – ILS*



Ejemplos trayectoriales

- *Guided Local Search – ILS*

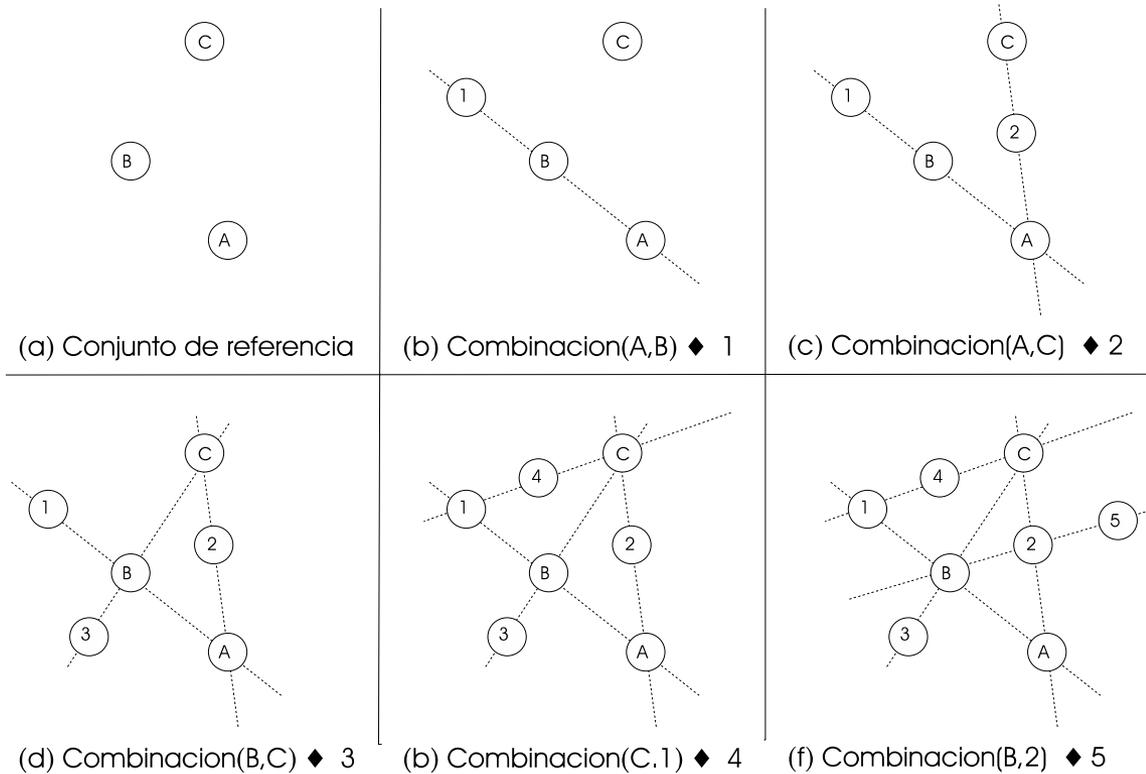


Ejemplos poblacionales

- Búsqueda dispersa (*Scatter Search – SS*)
- Reencadenamiento de trayectorias (*Path Relinking – PR*)
- Algoritmos evolutivos (genéticos – GA y meméticos – MA)
- Optimización por colonias de hormigas (*Ant Colony Optimization – ACO*)
- Inteligencia de enjambre (*Swarm Intelligence –*

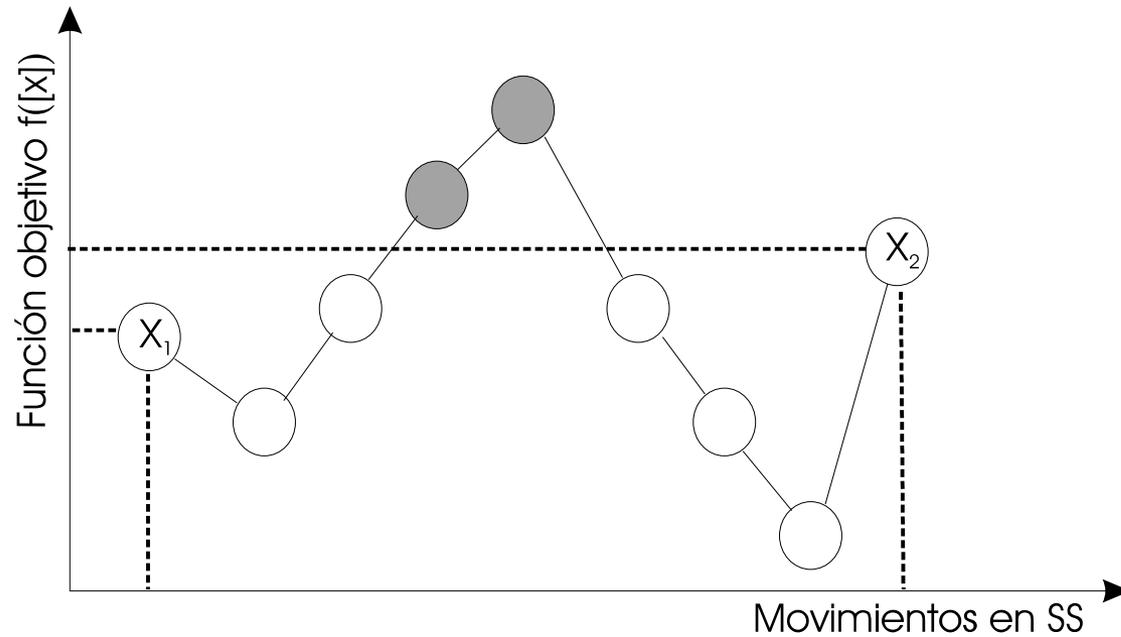
Ejemplos poblacionales

○ Scatter Search - SS



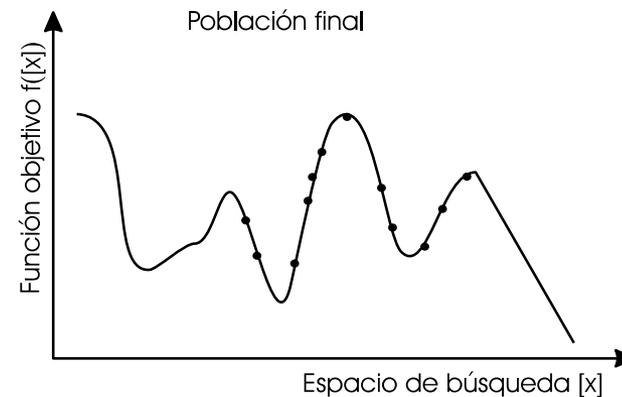
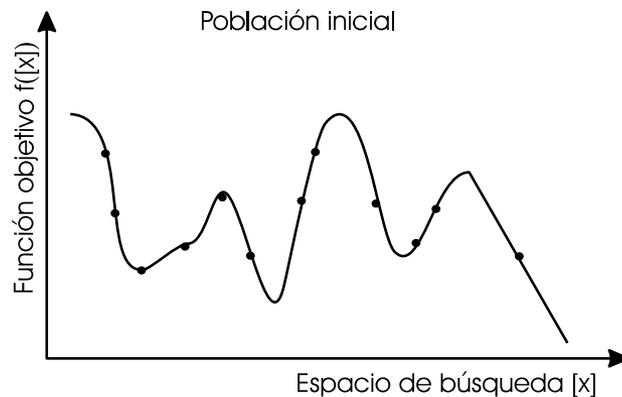
Ejemplos poblacionales

- *Path Relinking – PR*



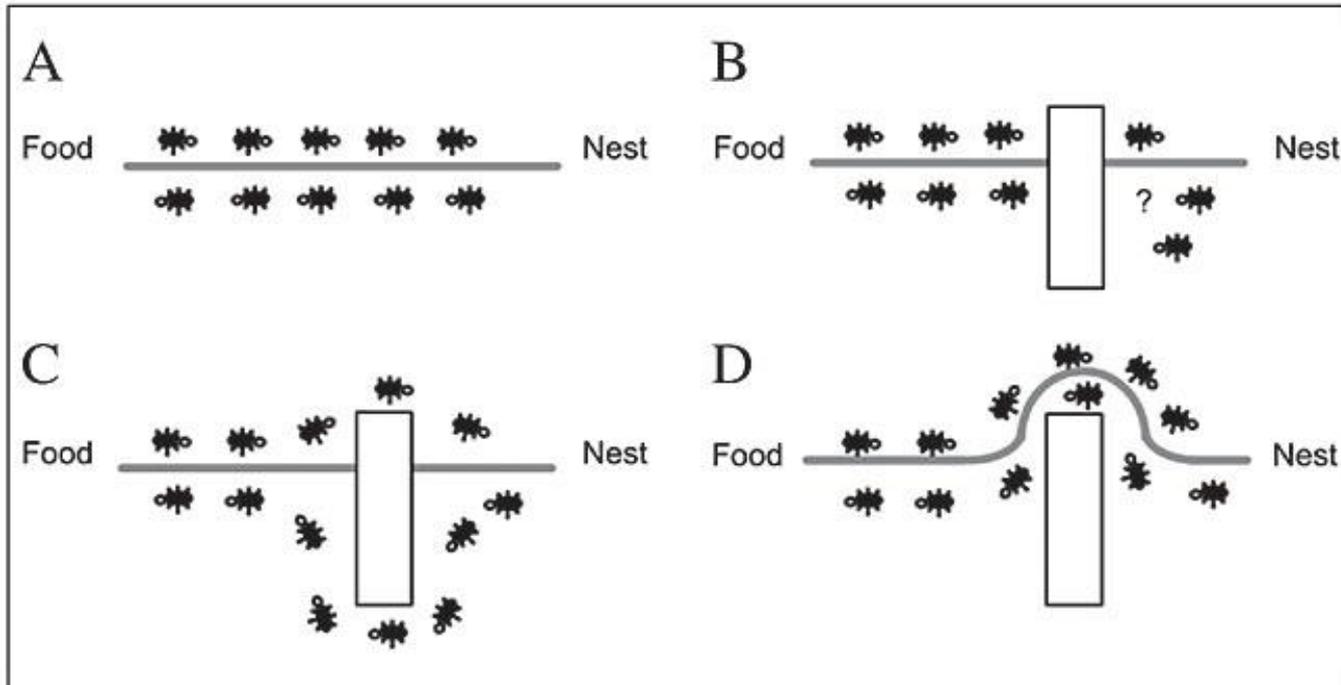
Ejemplos poblacionales

- Algoritmos evolutivos



Ejemplos poblacionales

- Algoritmos por colonias de hormigas – ACO



Nuevos métodos bioinspirados

- Hipótesis de partida
 - La similitud **biológica/evolutiva** es una fuente de inspiración para nuevos métodos metaheurísticos
 - Si ha funcionado en la naturaleza, ¿por qué no va a **funcionar en optimización?**

Nuevos métodos bioinspirados

- Estrategias establecidas
 - Optimización por Colonias de abejas
 - Algoritmo de la luciérnaga
 - Búsqueda harmónica
 - Búsqueda del cuco

Nuevos métodos bioinspirados

○ Inspiración física

- ***Magnetic Inspired optimization Algorithms.*** Inspirado en el movimiento de partículas en campos magnéticos
- ***Quantum Evolutionary Algorithms.*** Basados en bits cuánticos y superposición de estados
- ***Intelligent water drop.*** Basada en cómo los ríos encuentran casi el mejor camino a su desembocadura
- ***Gravitational search algorithm .*** Basado en teoría de gravitación Universal
- ***Charged System Search.*** Basado en la ley de Coulomb

Nuevos métodos bioinspirados

- Inspiración de enjambre
 - ***Invasive weed optimization algorithms.*** Inspirada cómo las semillas crecen en unos determinados terrenos
 - ***Glowworms Colonies Optimization.*** Equivalente a ACO pero con larvas
 - ***Monkey Search.*** Inspirada en cómo los monos trepan a los árboles para buscar comida
 - ***Shuffled frog-leaping algorithm.*** Búsquedas locales en nenúfares (memeplexes) e intercambios de ranas

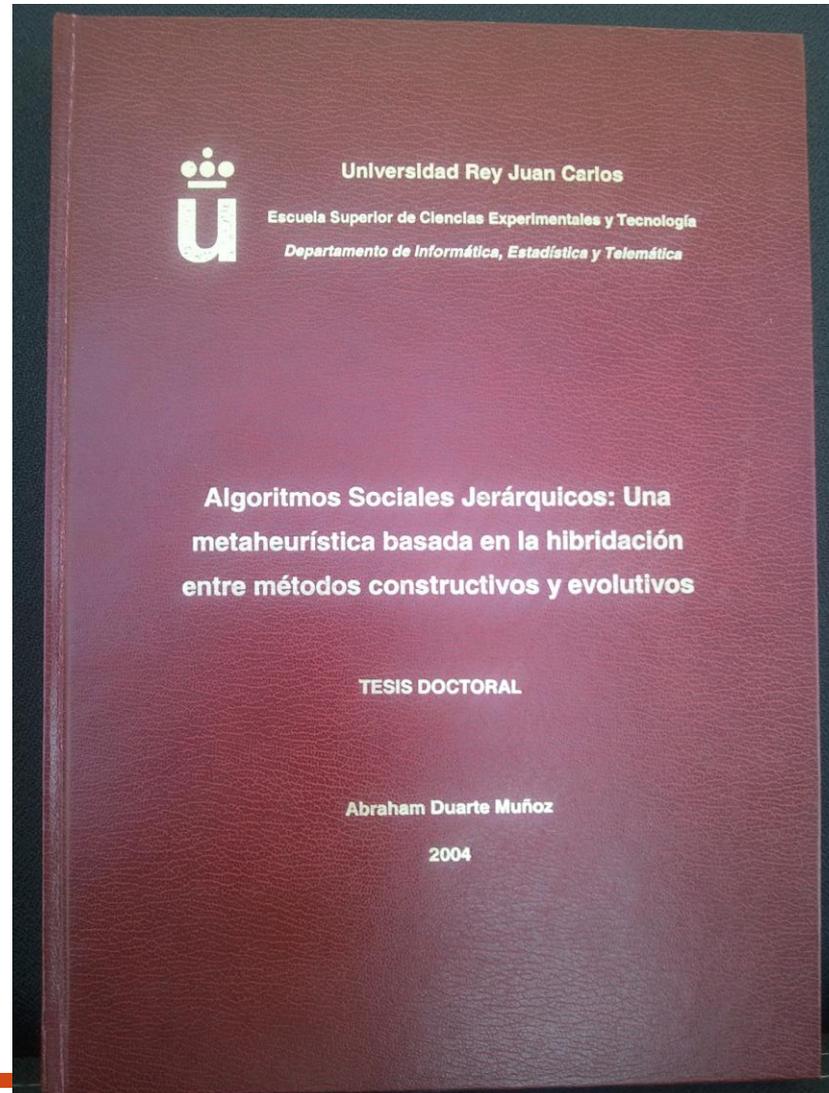
Conclusiones

- Optimización por colonias de hormigas: Inclusión de ***memoria tipo “soft”***
- Optimización por enjambre de partículas: Nuevas formas de ***generar caminos*** entre soluciones
- Algoritmos genéticos híbridos: ***búsquedas locales, nuevos métodos de combinación***, etc.
- Otros métodos sin inspiración biológica (*Iterated Greedy*)

Conclusiones

- Limitaciones
 - *No free lunch theorem*
 - Algoritmos competitivos
- Solución
 - Hibridaciones
 - Implementaciones novedosas
 - Paralelización
 - Etc.

Conclusiones



Panorámica de los procedimientos metaheurísticos

ABRAHAM DUARTE
