

XIII

Conferencia de la Asociación Española para la Inteligencia Artificial

Introducción a los Algoritmos Metaheurísticos

caepia
ttia
09

Francisco Herrera

Grupo de Investigación

“Soft Computing and Intelligent Information Systems”

Dpto. Ciencias de la Computación e I.A.

Universidad de Granada

18071 – ESPAÑA

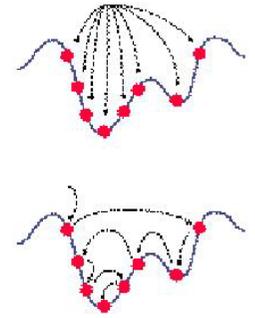
herrera@decsai.ugr.es

<http://sci2s.ugr.es>



DECSAI
Universidad de Granada

Introducción a los Algoritmos Metaheurísticos

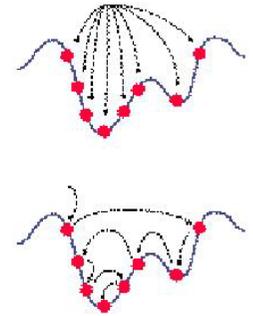


En múltiples problemas de IA nos encontramos modelos asociados a problemas de optimización y búsqueda.

Los algoritmos metaheurísticos son algoritmos aproximados de optimización y búsqueda de propósito general.

Son procedimientos iterativos que guían una heurística subordinada combinando de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda.

Introducción a los Algoritmos Metaheurísticos



En múltiples problemas de IA nos encontramos modelos asociados a problemas de optimización y búsqueda.

Los algoritmos metaheurísticos son algoritmos aproximados de optimización y búsqueda de propósito general.

Son procedimientos iterativos que guían una heurística subordinada combinando de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda.

El objetivo de este tutorial presentar las líneas básicas de desarrollo de algoritmos metaheurísticos y su aplicabilidad en el ámbito de la Inteligencia Artificial.

Introducción a los Algoritmos Metaheurísticos



1. Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda
2. MH I: Búsqueda Basada Trayectorias
3. MH II: Swarm Intelligence
4. MH III: Algoritmos Evolutivos
5. Limitaciones de los Algoritmos MH: Teorema NFL
6. MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias
7. Consideraciones Sobre el Uso de las Metaheurísticas
8. Algunos Estudios/Extensiones de Interés
9. Comentarios Finales

Introducción a los Algoritmos Metaheurísticos



1. **Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda**
2. **MH I: Búsqueda Basada Trayectorias**
3. **MH II: Swarm Intelligence**
4. **MH III: Algoritmos Evolutivos**
5. **Limitaciones de los Algoritmos MH: Teorema NFL**
6. **MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias**
7. **Consideraciones Sobre el Uso de las Metaheurísticas**
8. **Algunos Estudios/Extensiones de Interés**
9. **Comentarios Finales**

Introducción a las Metaheurísticas

Resolución de Problemas Mediante Algoritmos de Búsqueda: Introducción

Metaheurísticas: Definición

Funcionamiento de las Metaheurísticas

Clasificación de las Metaheurísticas

Ejemplo: El problema del viajante de comercio

Resolución de Problemas Mediante Algoritmos de Búsqueda: Introducción

- Existe una serie de problemas reales (de optimización combinatoria, de ingeniería, ..) de difícil solución que requieren de tareas tales como encontrar:
 - el **camino más corto** entre varios puntos,
 - un **plan de mínimo coste** para repartir mercancías a clientes,
 - una **asignación óptima** de trabajadores a tareas a realizar,
 - una **secuencia óptima** de proceso de trabajos en una cadena de producción,
 - una **distribución** de tripulaciones de aviones **con mínimo coste**,
 - el **mejor enrutamiento** de un paquete de datos en Internet,
 - ...

Resolución de Problemas Mediante Algoritmos de Búsqueda: Introducción

- Estos problemas se caracterizan porque:
 - suelen requerir agrupamientos, ordenaciones o asignaciones de un conjunto discreto de objetos que satisfagan ciertas restricciones,
 - se encuentran en muchas áreas de aplicación,
 - presentan una gran complejidad computacional (son NP-duros)
 - así, los algoritmos exactos (Programación Dinámica, Backtracking, Branch and Bound, ...) son ineficientes o simplemente imposibles de aplicar,
 - en la práctica se resuelven mediante algoritmos aproximados que proporcionan buenas soluciones (no necesariamente la óptima) al problema en un tiempo razonable

Metaheurísticas: Definición

Son una familia de algoritmos aproximados de propósito general. Suelen ser procedimientos iterativos que guían una heurística subordinada de búsqueda, combinando de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda.

- **Ventajas:**

- **Algoritmos de propósito general**
- **Gran éxito en la práctica**
- **Fácilmente implementables**
- **Fácilmente paralelizables**

- **Inconvenientes:**

- **Son algoritmos aproximados, no exactos**
- **Son no determinísticos (probabilísticos)**
- **No siempre existe una base teórica establecida**

Metaheurísticas: Definición

Factores que pueden hacer interesante su uso

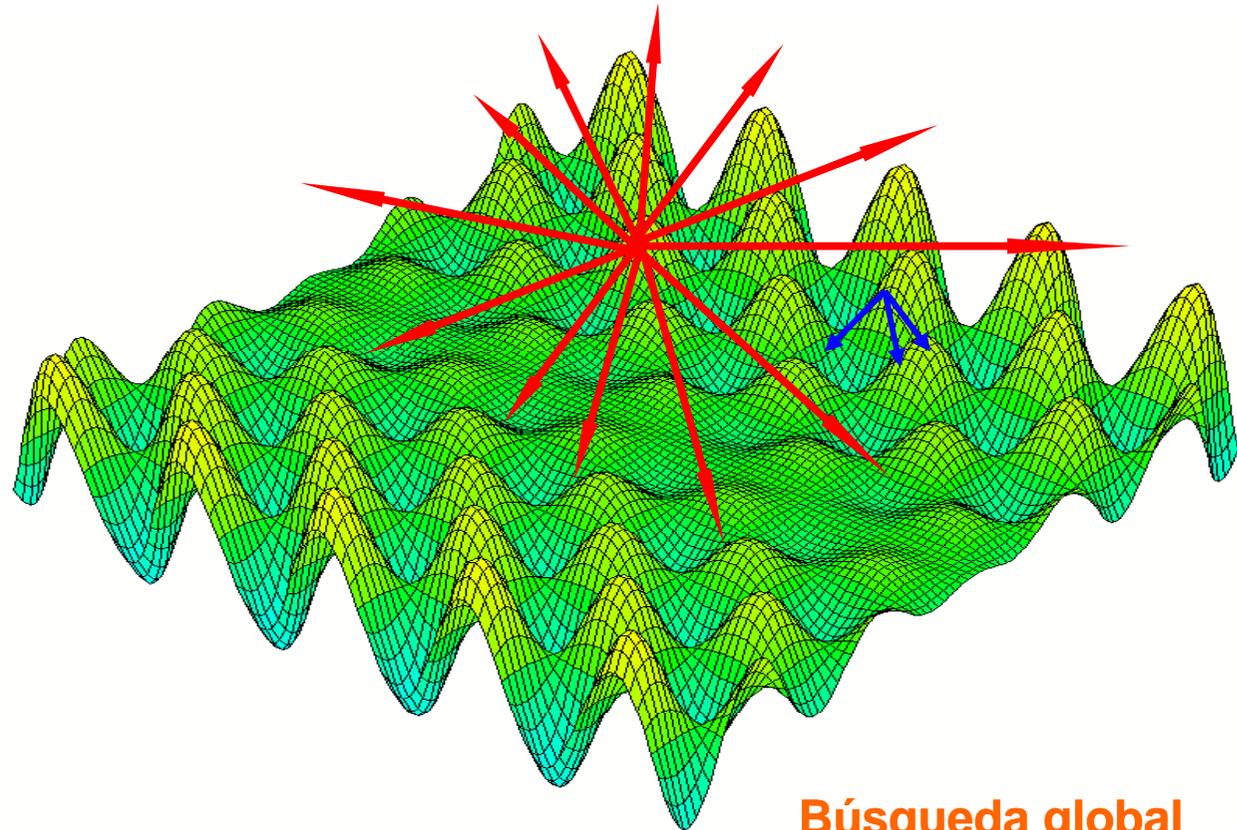
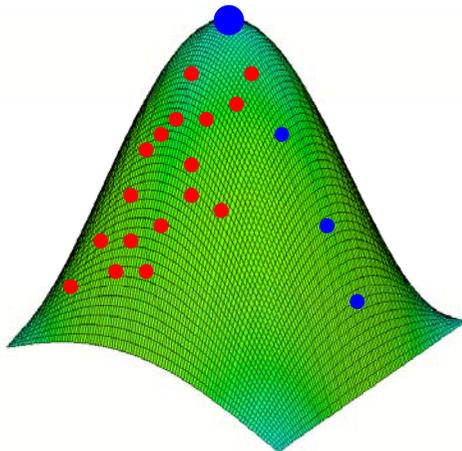
- Cuando no hay un método exacto de resolución, o éste requiere mucho tiempo de cálculo y memoria (ineficiente)
- Cuando no se necesita la solución óptima, basta con una de buena calidad

Funcionamiento de las Metaheurísticas

- Para obtener buenas soluciones, cualquier algoritmo de búsqueda debe establecer un balance adecuado entre dos características contrapuestas del proceso:
 - **Intensificación**: cantidad de esfuerzo empleado en la búsqueda en la región actual (explotación del espacio)
 - **Diversificación**: cantidad de esfuerzo empleado en la búsqueda en regiones distantes del espacio (exploración)

Funcionamiento de las Metaheurísticas

Búsqueda local
Explotación
Intensificación



Búsqueda global
Exploración
Diversificación

Funcionamiento de las Metaheurísticas

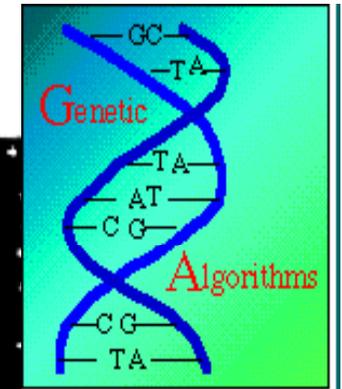
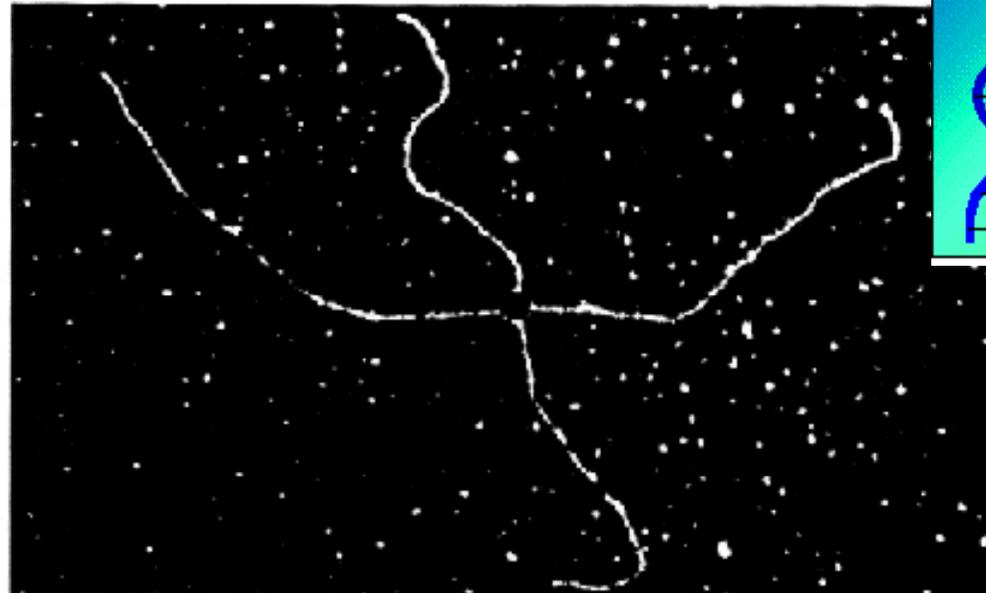
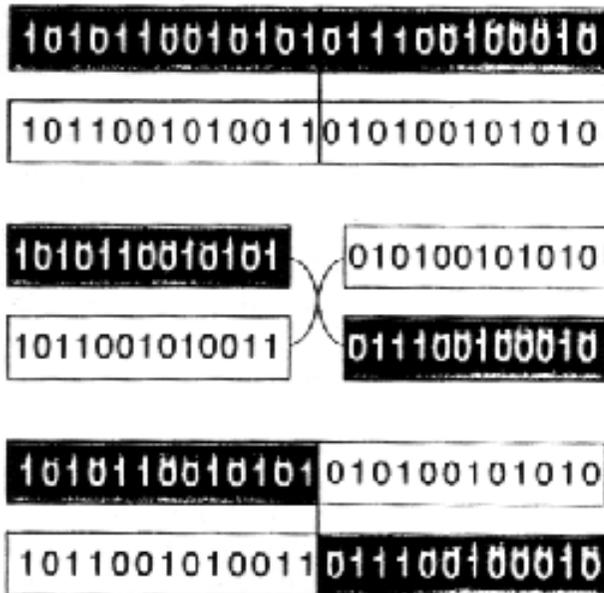
- El equilibrio entre intensificación y diversificación es necesario para:
 - Identificar rápidamente regiones del espacio con soluciones de buena calidad
 - No consumir mucho tiempo en regiones del espacio no prometedoras o ya exploradas
- Las metaheurísticas aplican distintas estrategias para obtener un buen balance entre intensificación y diversificación

Clasificación de las Metaheurísticas

- Existen distintas metaheurísticas en función de conceptos como:
 - **Seguimiento de trayectoria considerado**
 - **Uso de poblaciones de soluciones**
 - **Uso de memoria**
 - **Fuente de inspiración**

Clasificación de las Metaheurísticas

- Fuente de inspiración. Inspiración biológica. Ej. Algoritmos Genéticos

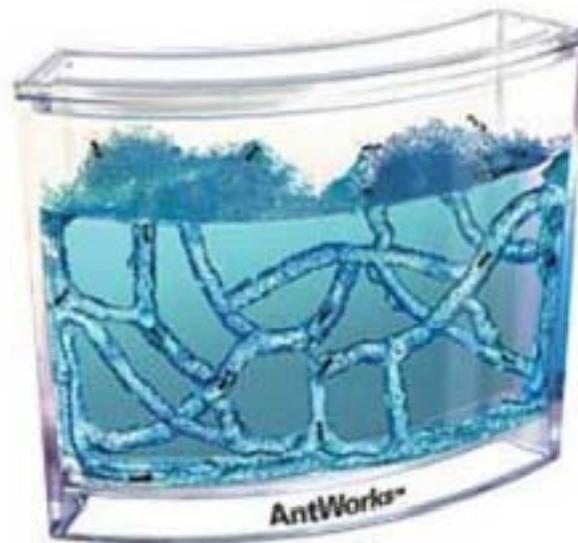


CROSSOVER is the fundamental mechanism of genetic rearrangement for both real organisms and genetic algorithms.

Chromosomes line up and then swap the portions of their genetic code beyond the crossover point.

Clasificación de las Metaheurísticas

- **Fuente de inspiración. Inspiración biológica. Ej. Algoritmos de Optimización basados en Colonias de Hormigas**



Clasificación de las Metaheurísticas

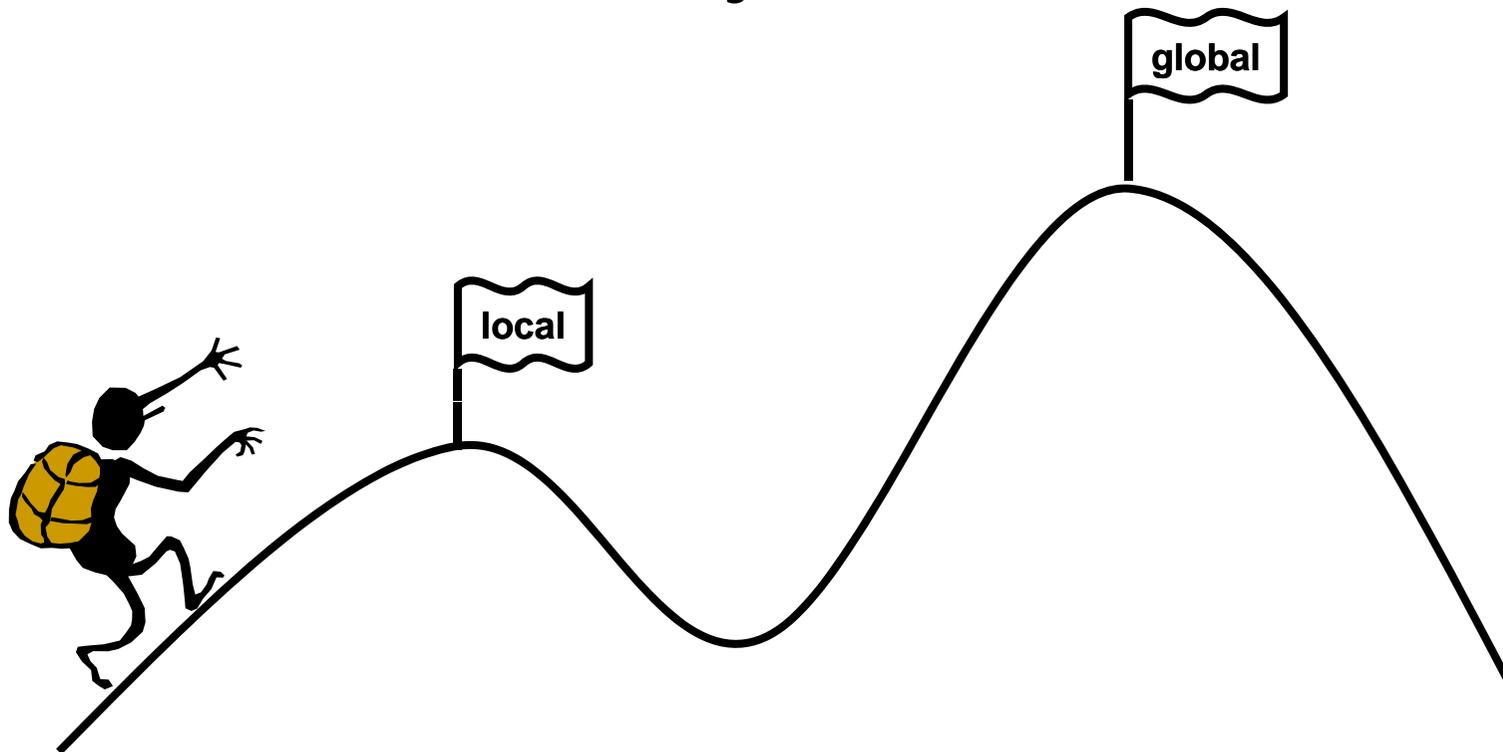
- Una posible clasificación:
 - **Basadas en métodos constructivos** (*Parten de una solución inicial vacía y van añadiéndole componentes hasta construir una solución*): GRASP, Optimización Basada en Colonias de Hormigas
 - **Basadas en trayectorias** (*la heurística subordinada es un algoritmo de búsqueda local que sigue una trayectoria en el espacio de búsqueda. Parten de una solución inicial e iterativamente tratan de reemplazarla por otra solución de con mejor calidad*): Búsqueda Local, Enfriamiento Simulado, Búsqueda Tabú, BL Iterativa, ...

Clasificación de las Metaheurísticas

- Una posible clasificación:
 - **Basadas en métodos constructivos** (*Parten de una solución inicial vacía y van añadiéndole componentes hasta construir una solución*): GRASP, Optimización Basada en Colonias de Hormigas
 - **Basadas en trayectorias** (*la heurística subordinada es un algoritmo de búsqueda local que sigue una trayectoria en el espacio de búsqueda. Parten de una solución inicial e iterativamente tratan de reemplazarla por otra solución de con mejor calidad*): Búsqueda Local, Enfriamiento Simulado, Búsqueda Tabú, BL Iterativa, ...

Clasificación de las Metaheurísticas

- Basadas en trayectorias

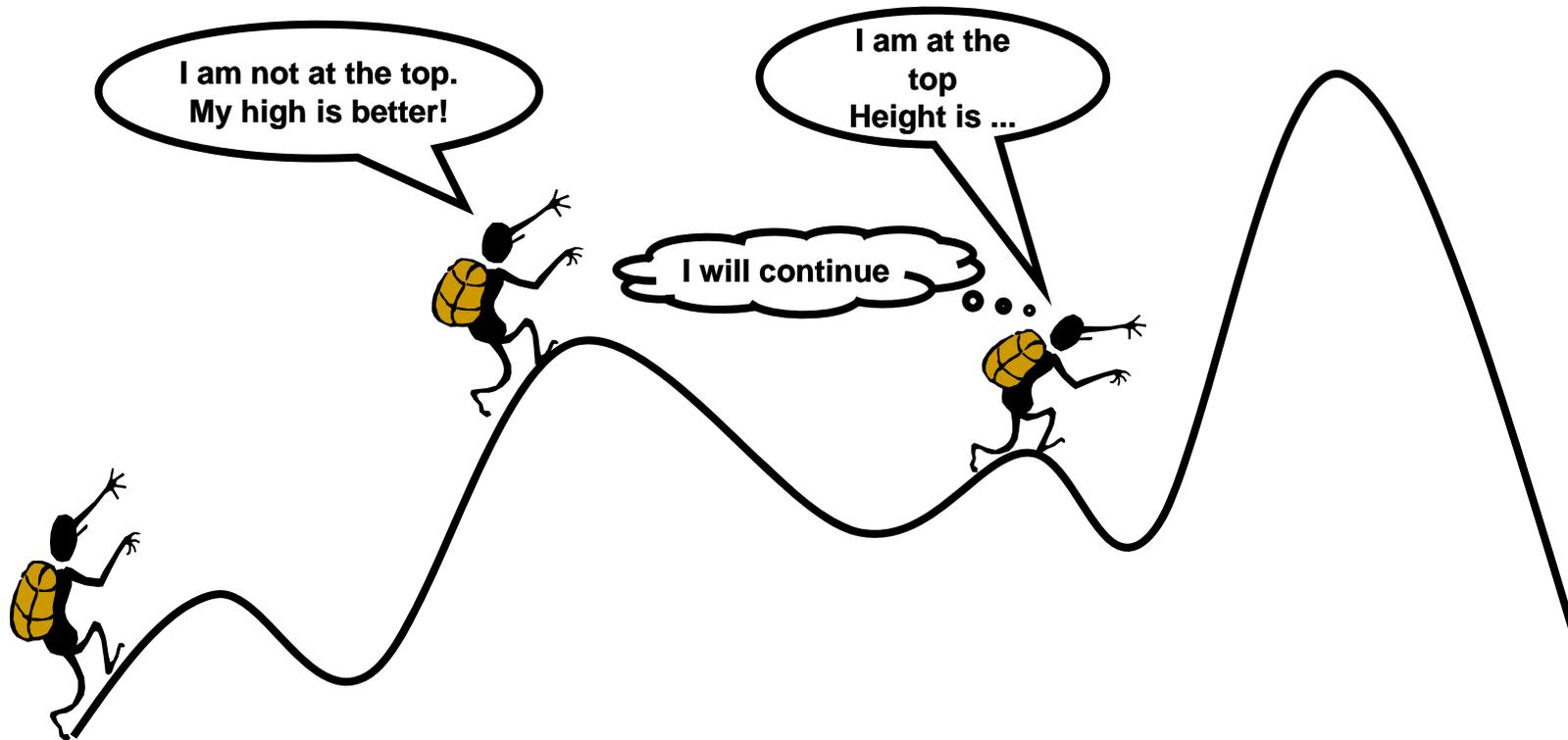


Clasificación de las Metaheurísticas

- Una posible clasificación:
 - **Basadas en métodos constructivos:** GRASP, Optimización Basada en Colonias de Hormigas
 - **Basadas en trayectorias** (*la heurística subordinada es un algoritmo de búsqueda local que sigue una trayectoria en el espacio de búsqueda*): Búsqueda Local, Enfriamiento Simulado, Búsqueda Tabú, BL Iterativa, ...
 - **Basadas en poblaciones** (*el proceso considera múltiples puntos de búsqueda en el espacio que evolucionan en paralelo*): Algoritmos Genéticos, Scatter Search, Algoritmos Meméticos, PSO, ...

Clasificación de las Metaheurísticas

- Basadas en poblaciones



Clasificación de las Metaheurísticas

- Basadas en poblaciones



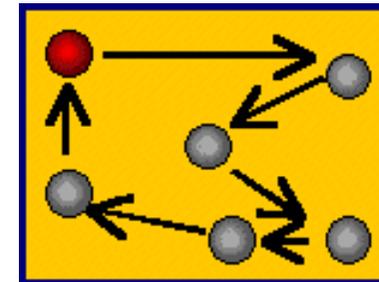
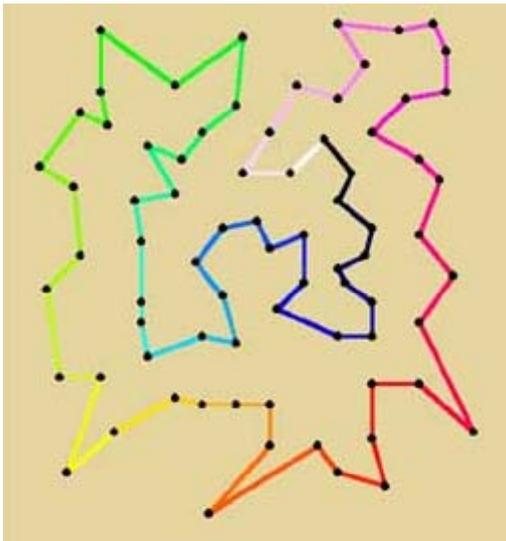
Clasificación de las Metaheurísticas

- **Metaheurísticas inspiradas en la naturaleza: *Algoritmos Bioinspirados***
 - **Swarm Intelligence** (Algoritmos basados en la inteligencia de enjambre)
 - Optimización Basada en Colonias de Hormigas (constructivos)
 - Algoritmos basados en nubes de partículas (basados en poblaciones)
 - **Algoritmos Evolutivos** (basados en poblaciones)

Clasificación de las Metaheurísticas

- Las **heurísticas constructivas** son más rápidas pero dan soluciones de peor calidad que la búsqueda basada en trayectorias
- Ambos son **procesos de búsqueda efectuados sobre un espacio de soluciones** al problema.
 - En los métodos constructivos, el espacio es de soluciones parciales,
 - mientras que en la búsqueda por trayectorias y poblaciones es de soluciones completas (candidatas)
- En este último caso el **espacio de búsqueda suele ser de un tamaño exponencial con respecto al tamaño del problema**

Ejemplo: El problema del viajante de comercio

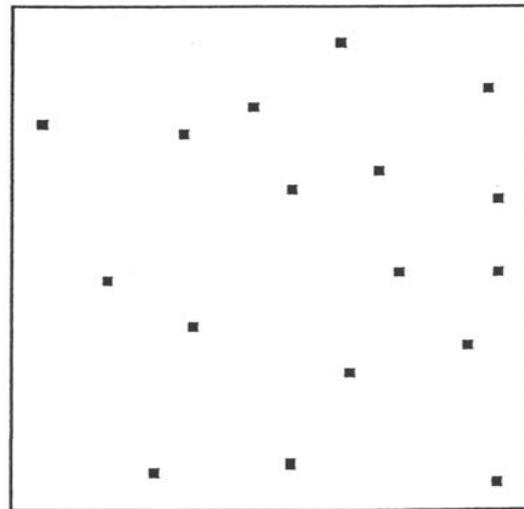


Es un problema muy estudiado al presentar aplicaciones reales tales como la fabricación en serie de tarjetas de ordenador (impresión de los buses de estaño)

En el viajante de comercio, se tiene una red de nodos, que pueden ser ciudades o simplemente lugares de una ciudad. Se parte de un lugar inicial, y deben recorrerse todos sin pasar más de una vez por cada lugar, volviendo al lugar inicial. Para cada arco, se tiene un valor C_{ij} , que indica la distancia o el costo de ir del nodo i al nodo j .

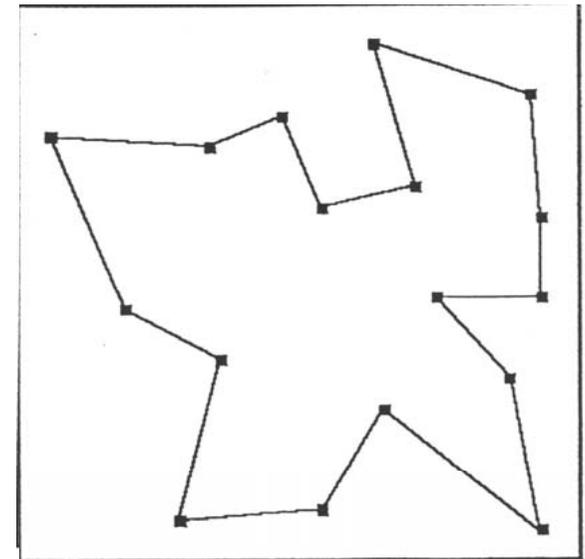
Ejemplo: El problema del viajante de comercio

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	12.87	19.71	31.56	22.70	17.26	23.33	12.16	24.71	34.51	12.58	21.38	42.37	27.43	36.51	19.10	1.18
2	0	15.80	37.51	21.52	28.57	35.43	22.70	16.78	28.57	11.13	25.26	50.62	38.16	35.97	9.04	34.56	
3	0	50.18	36.56	35.86	35.51	21.60	31.50	43.51	25.58	38.78	61.57	46.15	51.10	23.50	48.52		
4	0	20.90	21.52	37.62	38.14	33.26	31.90	27.13	13.03	15.53	18.39	19.37	35.84	8.12			
5	0	26.00	40.72	33.74	12.87	14.71	11.68	9.72	35.86	30.96	15.06	16.78	15.27				
6	0	16.99	18.53	34.51	40.20	22.34	18.53	27.70	10.80	34.94	32.08	25.24					
7	0	14.54	46.60	54.54	33.80	34.52	40.35	22.09	51.20	41.84	41.73						
8	0	36.31	46.12	24.21	30.50	45.72	28.09	46.77	30.20	39.71							
9	0	12.54	13.31	21.52	48.18	41.50	23.85	8.50	27.43								
10	0	22.43	23.33	46.67	44.80	16.31	20.53	24.58									
11	0	14.71	40.81	30.52	26.21	10.50	23.93										
12	0	27.43	21.97	17.20	23.35	10.35											
13	0	18.89	32.78	50.15	22.59												
14	0	35.88	40.51	24.71													
15	0	30.18	11.90														
16	0	31.31															
17	0																

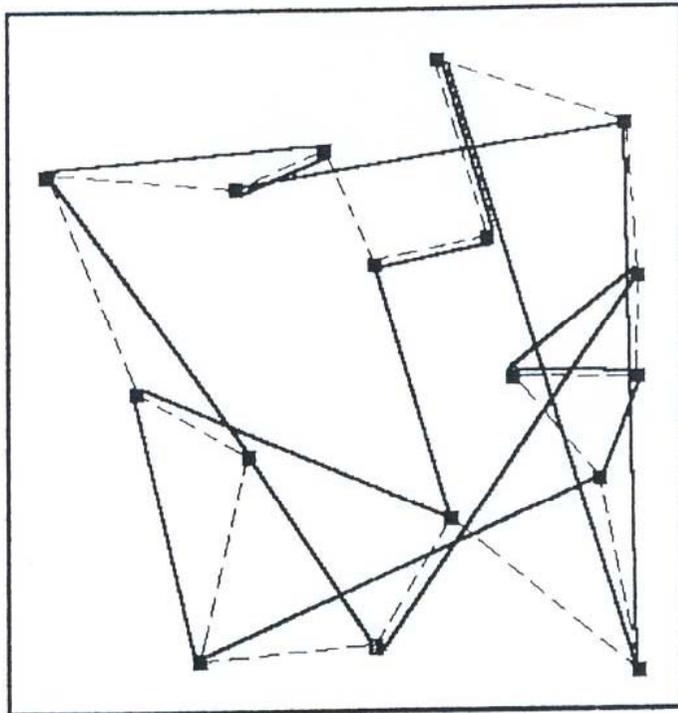


17! (3.5568734e14)
soluciones posibles

Solución óptima:
Coste=226.64

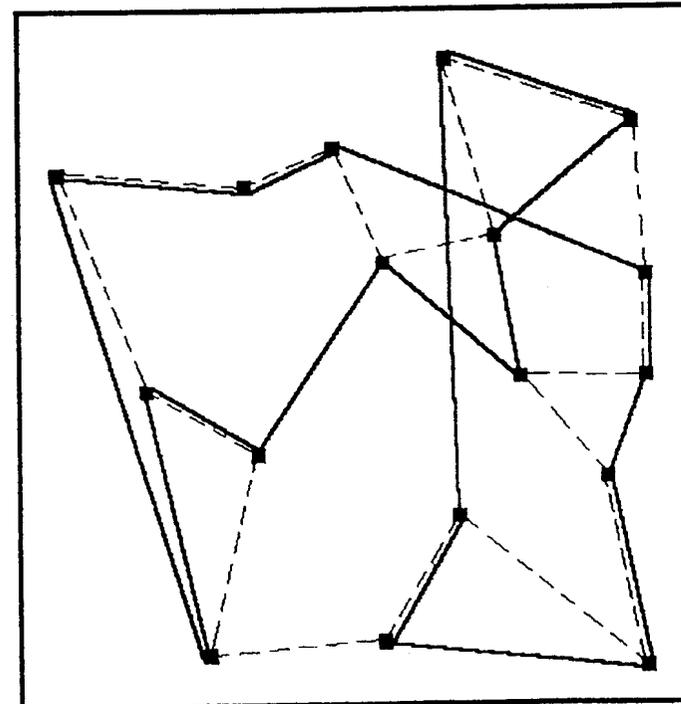


Viajante de Comercio



—— Mejor solución
----- Solución óptimal

Iteración: 0 Costo: 403.7

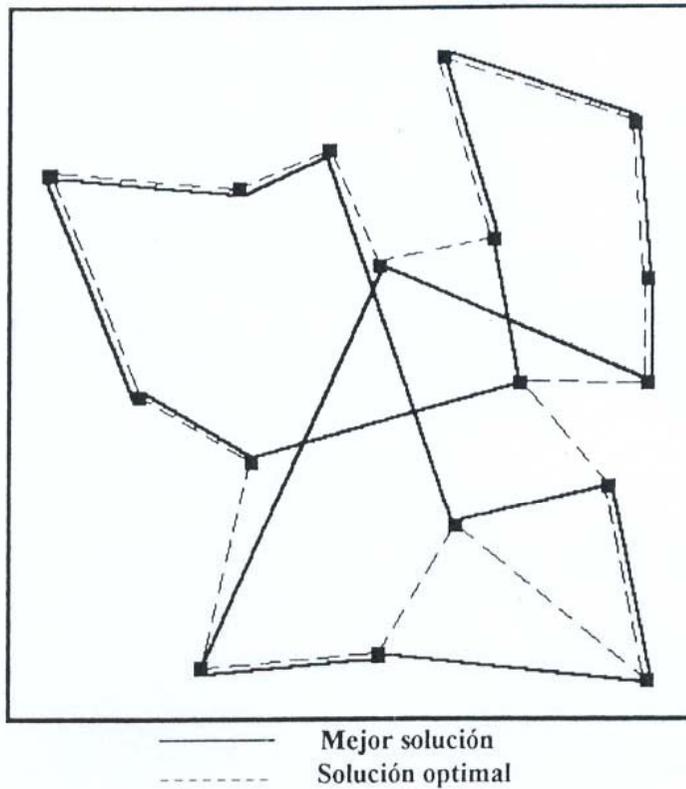


—— Mejor solución
----- Solución óptimal

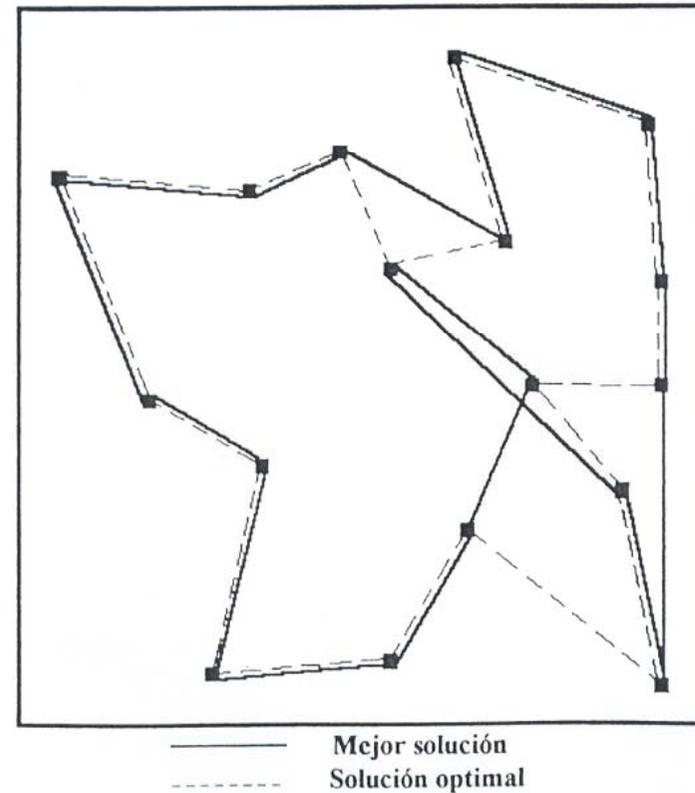
Iteración: 25 Costo: 303.86

Solución óptima: 226.64

Viajante de Comercio



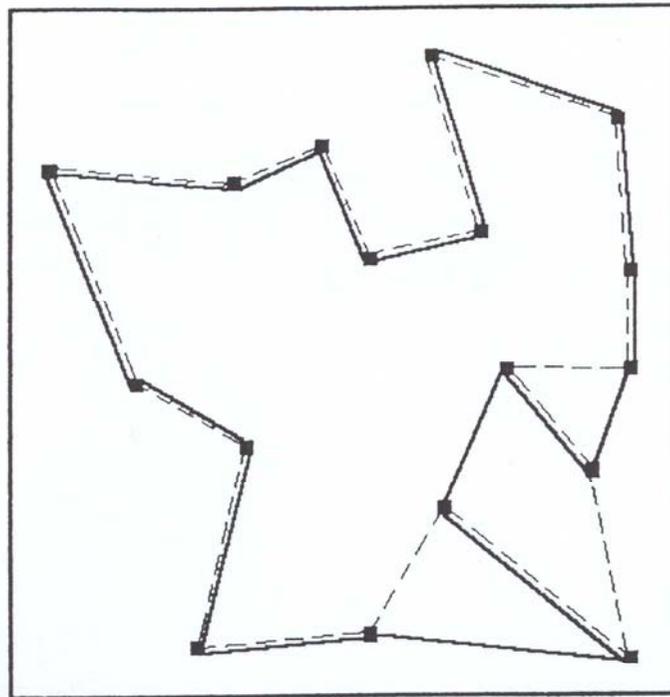
Iteración: 50 Costo: 293.6



Iteración: 100 Costo: 256.55

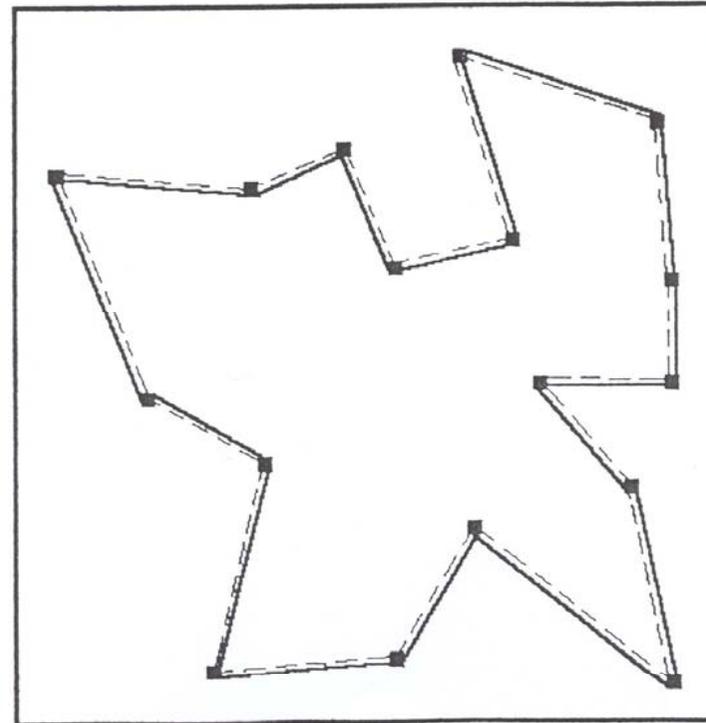
Solución óptima: 226.64

Viajante de Comercio



—— Mejor solución
- - - Solución óptima

Iteración: 200 Costo: 231.4

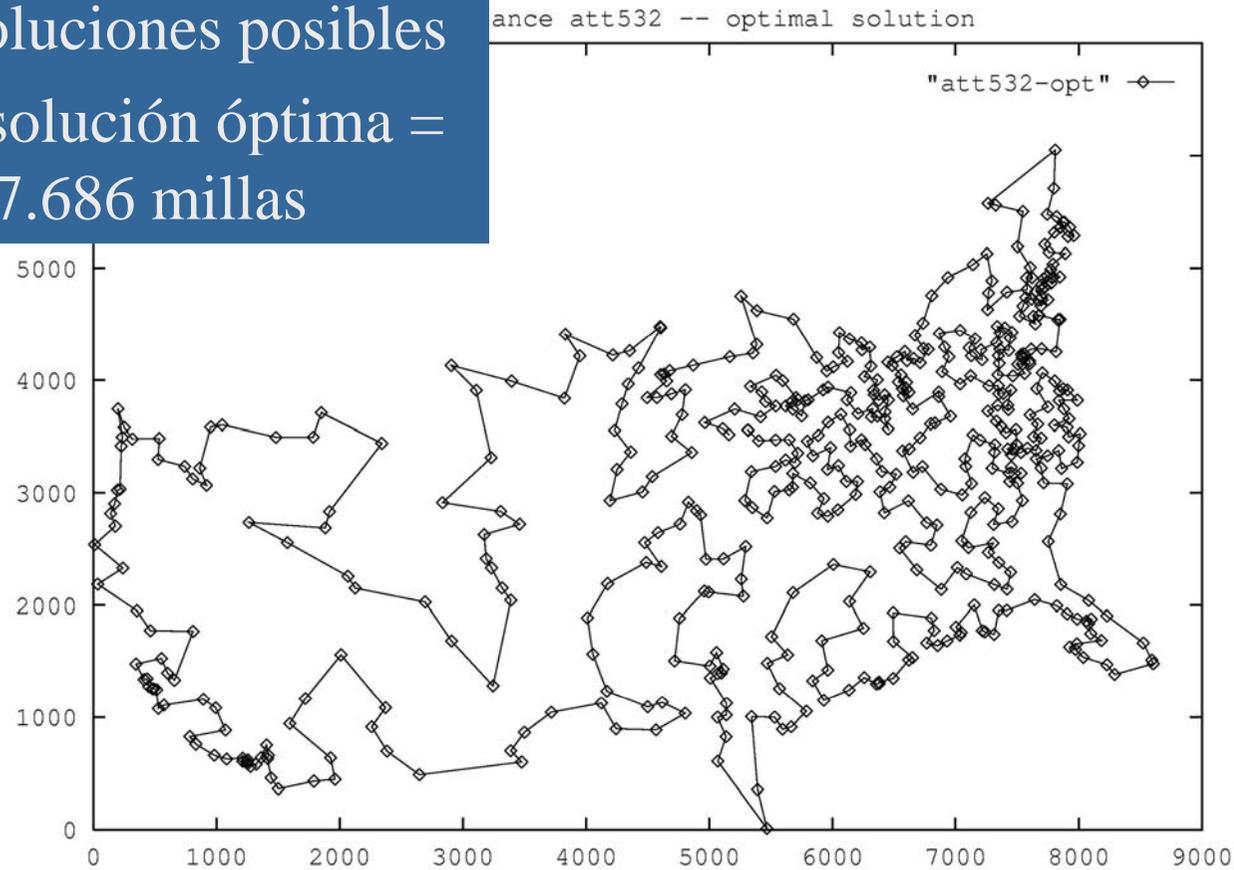


—— Mejor solución
- - - Solución óptima

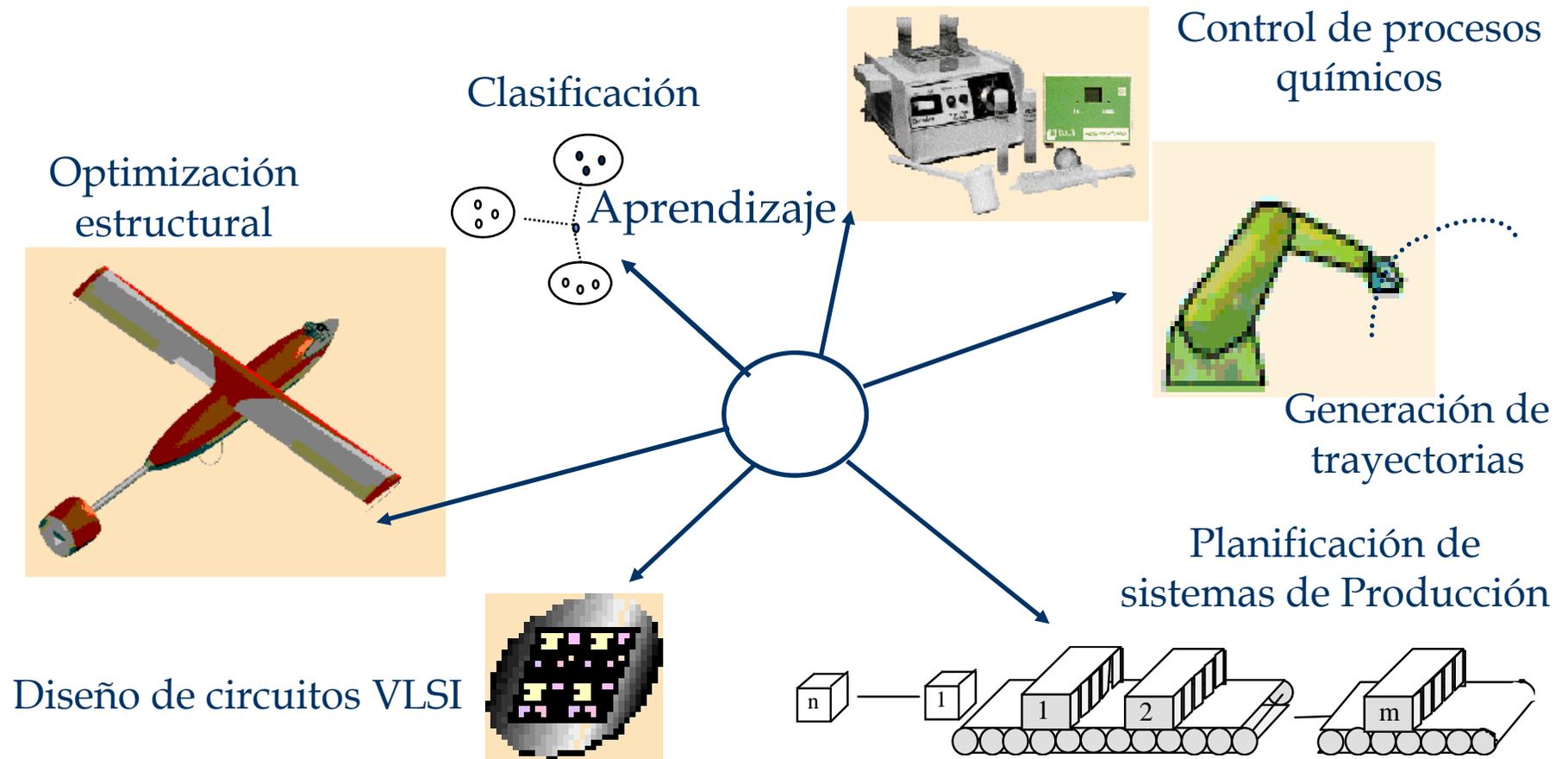
Iteración: 250 Solución
óptima: 226.64

Ejemplo: El problema del viajante de comercio

532! soluciones posibles
Coste solución óptima =
27.686 millas



Aplicaciones



Aplicaciones

DOMINIOS DE APLICACIÓN

Optimización combinatoria

Optimización en ingeniería

Modelado e identificación de sistemas

Planificación y control

Aprendizaje y minería de datos

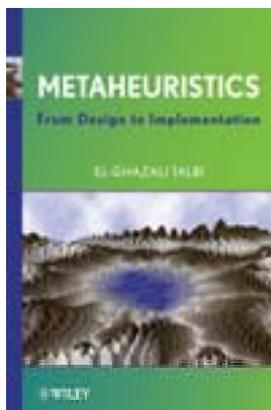
Vida artificial

Bioinformática

....

Lecturas

- B. Melián, J.A. Moreno Pérez, J.M. Moreno Vega. Metaheurísticas: un visión global. Revista Iberoamericana de Inteligencia Artificial 19 (2003) 7-28
- C. Blum, A. Roli. Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput. Surv. 35:3 (2003) 268-308.
- Colin G. Johnson A design framework for metaheuristics. Artif Intell Rev (2008) 29:163–178



Metaheuristics: From Design to Implementation

El-Ghazali Talbi

ISBN: 978-0-470-27858-1

593 pages

July 2009

<http://sci2s.ugr.es/docencia/metaheuristicas.php>

Introducción a los Algoritmos Metaheurísticos



1. Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda
2. **MH I: Búsqueda Basada Trayectorias**
3. **MH II: Swarm Intelligence**
4. **MH III: Algoritmos Evolutivos**
5. **Limitaciones de los Algoritmos MH: Teorema NFL**
6. **MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias**
7. **Consideraciones Sobre el Uso de las Metaheurísticas**
8. **Algunos Estudios/Extensiones de Interés**
9. **Comentarios Finales**

Búsqueda Basada en Trayectorias

Búsqueda Local y Estructura de Entorno

Características de Diferentes Algoritmos de Búsqueda Basada en Trayectorias

Búsqueda Basada en Múltiples Trayectorias: Métodos Multiarranque

Búsqueda Local Iterativa: ILS

Bibliografía

F. Glover, G.A. Kochenberger (Eds.). Handbook of Metaheuristics. Kluwer Academic Press, 2003.

H.H. Hoos, T. Stützle. Stochastic Local Search. Morgan Kaufmann, 2004.

Búsqueda Local y Estructura de Entorno

Término “LOCAL”

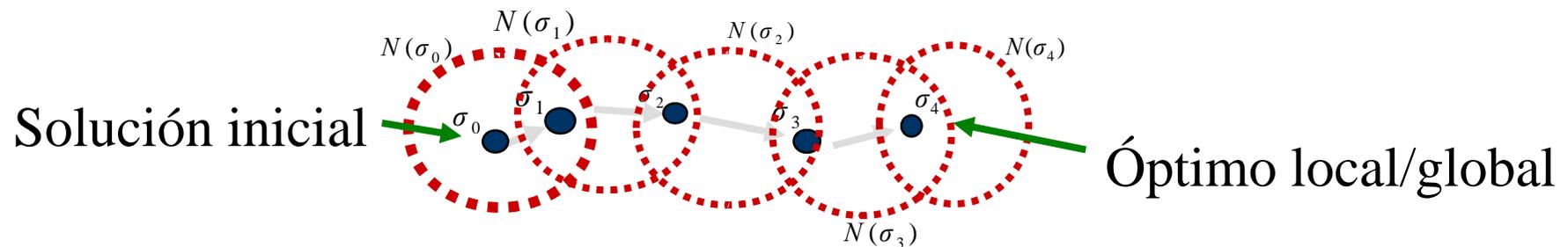
El término “*local*” se utiliza frecuentemente en los estudios teóricos y prácticos de las metaheurísticas de búsqueda. Se asocia al uso de estructuras de entorno, reflejando el concepto de proximidad o vecindad entre las soluciones alternativas del problema.

Todas las soluciones incluidas en el entorno de la solución actual, que viene delimitado por un operador de generación de soluciones, se denominan soluciones vecinas.

Búsqueda Local y Estructura de Entorno

Término “LOCAL”

Los **algoritmos basados en trayectorias** efectúan un estudio local del espacio de búsqueda, analizan el entorno de la solución actual para decidir cómo continuar el recorrido de la búsqueda.



Búsqueda Local y Estructura de Entorno

Estructura de Entorno

Basta con diseñar la estructura de entorno para obtener un modelo genérico de algoritmo de búsqueda.

DESCRIPCIÓN

- **Se fija una codificación para las soluciones.**
- **Se define un operador de generación de vecino y, en consecuencia, se fija una estructura de entorno para las mismas.**
- **Se escoge una solución del entorno de la solución actual hasta que se satisfaga el criterio de parada.**

Búsqueda Local y Estructura de Entorno

Procedimiento Búsqueda por Entornos

Inicio

GENERA(Solución Inicial)

Solución Actual ← Solución Inicial;

Mejor Solución ← Solución Actual;

Repetir

Solución Vecina ← GENERA_VECINO(Solución Actual);

Si Acepta(Solución Vecina)

entonces Solución Actual ← Solución Vecina;

Si Objetivo(Solución Actual) **es mejor que** Objetivo(Mejor Solución)

entonces Mejor Solución ← Solución Actual;

Hasta (Criterio de parada);

DEVOLVER (Mejor Solución);

Fin

Búsqueda Local y Estructura de Entorno

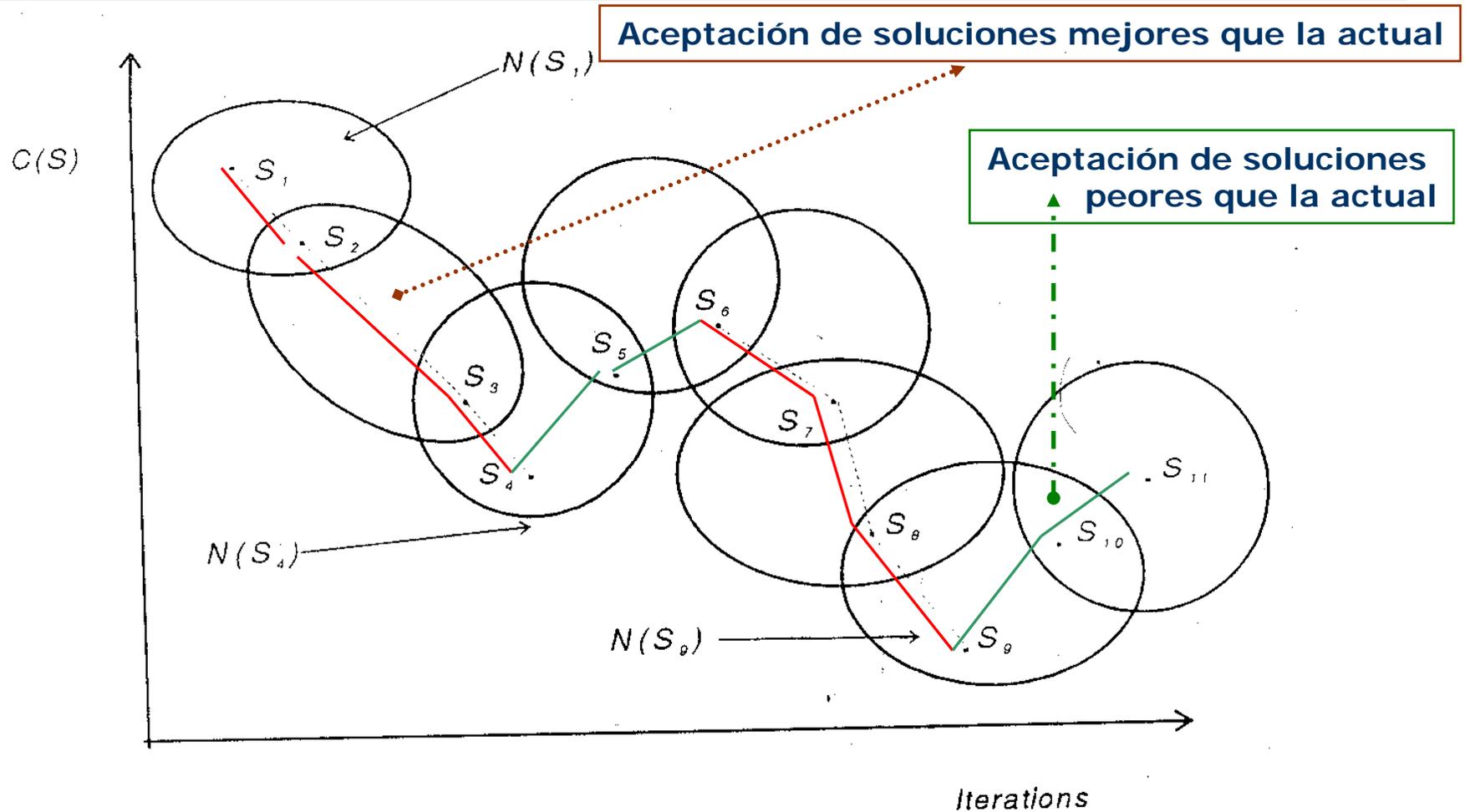


Figura que muestra una trayectoria de búsqueda basada en entornos

Características de Diferentes Algoritmos de Búsqueda Basada en Trayectorias

SOLUCIONES: 3 opciones para salir de los óptimos locales

- Permitir movimientos de empeoramiento de la solución actual (Ejemplo: Enfriamiento Simulado, Búsqueda Tabú, ...).
- Modificar la estructura de entornos (Ejemplo: Búsqueda Tabú, Búsqueda Descendente Basada en Entornos Variables: VND, Búsqueda en Entornos Variables: VNS, ...).
- Volver a comenzar la búsqueda desde otra solución inicial (Ejemplo: Búsquedas Multiarranque, ILS, VNS, GRASP, Optimización Basada en Colonia de Hormigas, ...).

Búsqueda Local y Estructura de Entorno

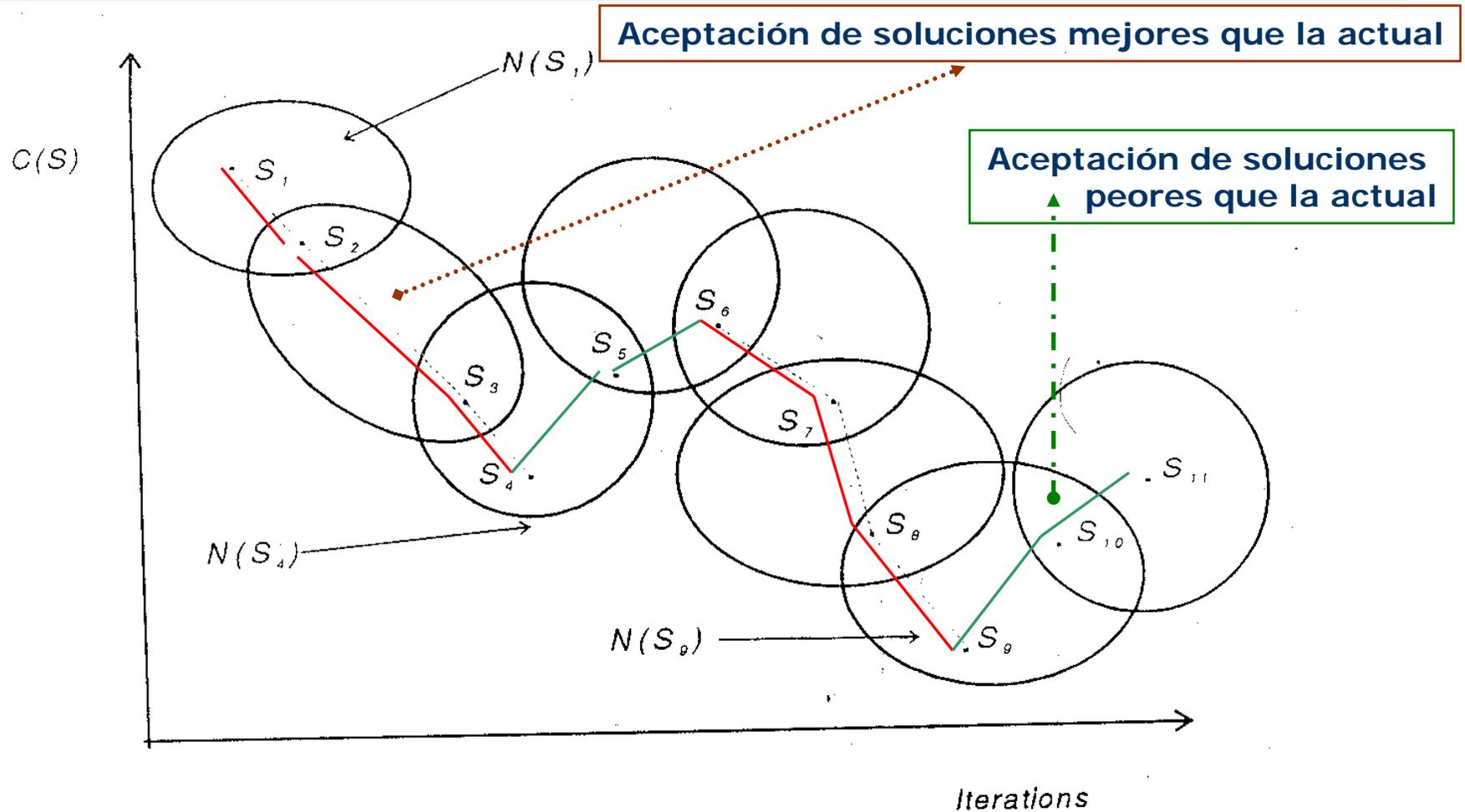


Figura que muestra una trayectoria de búsqueda basada en entornos.
Algoritmos avanzados: Simulated Annealing, Tabu Search

Lecturas

<http://sci2s.ugr.es/docencia/metaheuristicas.php>

- K.A. Dowsland, B.A. Díaz. Diseño de Heurísticas y Fundamentos del Recocido Simulado. Revista Iberoamericana de Inteligencia Artificial 19 (2003) 93-102.
- B. Suman, P. Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. Journal of the Operational Research Society 57:10 (2006) 1143-1160.
- F. Glover, B. Melián. Búsqueda Tabú. Revista Iberoamericana de Inteligencia Artificial 19 (2003) 29-48.
- M. Gendreau. Chapter 2: An Introduction to Tabu Search. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 37-54.

- M.G.C. Procedur Academi
- R. Martí. Handboo
- H.L. Lou G.A. Koc 353.
- P. Hanse Kochenb

http://sci2s.ugr.es/docencia/metaheuristicas.php

SCI²S Thematic Public Websites

Genetic Fuzzy Systems

Computing with Words in Decision Making

Statistical Inference in Computational Intelligence and Data Mining

Introducción a los Algoritmos Metaheurísticos

XIII Conferencia de la Asociación Española para la Inteligencia Artificial Sevilla (España)

Lecturas

<http://sci2s.ugr.es/docencia/metaheuristicas.php>

Introducción a los Algoritmos Metaheurísticos

XIII Conferencia de la Asociación Española para la Inteligencia Artificial Sevilla (España)

Francisco Herrera (Universidad de Granada)

Tutorial: Introducción a los Algoritmos Metaheurísticos

El objetivo de este tutorial presentar las líneas básicas de desarrollo de algoritmos metaheurísticos y su aplicabilidad en el ámbito de la Inteligencia Artificial.

Los algoritmos metaheurísticos son algoritmos aproximados de optimización y búsqueda de propósito general.

Son procedimientos iterativos que guían una heurística subordinada combinando de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda.

El tutorial está organizado de la siguiente manera:

1. Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda.
2. MH I: Búsqueda Basada Trayectorias.
3. MH II: Swarm Intelligence.
4. MH III: Algoritmos Evolutivos.
5. Limitaciones de los Algoritmos MH: Teorema NFL.
6. MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias.
7. Consideraciones Sobre el Uso de las Metaheurísticas.

Características de Diferentes Algoritmos de Búsqueda Basada en Trayectorias

SOLUCIONES: 3 opciones para salir de los óptimos locales

- Permitir movimientos de empeoramiento de la solución actual (Ejemplo: Enfriamiento Simulado, Búsqueda Tabú, ...).
- Modificar la estructura de entornos (Ejemplo: Búsqueda Tabú, Búsqueda Descendente Basada en Entornos Variables: VND, Búsqueda en Entornos Variables: VNS, ...).
- Volver a comenzar la búsqueda desde otra solución inicial (Ejemplo: Búsquedas Multiarranque, ILS, VNS, GRASP, Optimización Basada en Colonia de Hormigas, ...).

Búsqueda Basada en Múltiples Trayectorias: Modelos Multiarranque

■ Modelos Multiarranque

■ Métodos constructivos de la solución inicial

- Construcción *greedy*: Algoritmos GRASP
- Algoritmos Basados en Colonias de Hormigas: ACO

■ Métodos iterativos mediante modificación de la solución encontrada

- ILS: Búsqueda Local Iterativa
- VNS: Búsqueda de Entorno Variable

■ Hibridaciones entre técnicas poblacionales de exploración/ combinación de soluciones y métodos de búsqueda local

- Algoritmos Meméticos / Algoritmos Genéticos con BL
- *Scatter Search* (Búsqueda Dispersa)

Búsqueda Local Iterativa: ILS

Algoritmo ILS

- La ILS está basada en la aplicación repetida de un algoritmo de Búsqueda Local a una solución inicial que se obtiene por mutación de un óptimo local previamente encontrado
- Propuesta inicialmente en la Tesis Doctoral de Thomas Stützle:

T. Stützle, 1998. Local Search Algorithms for Combinatorial Problems-Analysis, Improvements and New Applications. PhD Thesis, Darmstadt, University of Technology, Department of Computer Science.

Búsqueda Local Iterativa: ILS

- La aplicación de la ILS necesita de la definición de cuatro componentes:
 1. Una *solución inicial* (usualmente, aleatoria)
 2. Un *procedimiento de modificación* (**mutación**) que aplica un cambio brusco sobre la solución actual para obtener una solución intermedia
 3. Un procedimiento de *Búsqueda Local*
 4. Un *criterio de aceptación* que decide a qué solución se aplica el procedimiento de modificación
- En la mayoría de las aplicaciones, la mutación se aplica a la mejor solución encontrada

H.R. Lourenço, O.C. Martin, T. Stützle, Iterated Local Search. En: F.Glover, G. Kochenberger (Eds.), Handbook of Metaheuristics. Kluwer Academic Publishers, 2003, pp. 321-353.

H.H. Hoos, T. Stützle. Stochastic Local Search. Morgan Kaufmann, 2004.

Búsqueda Local Iterativa: ILS

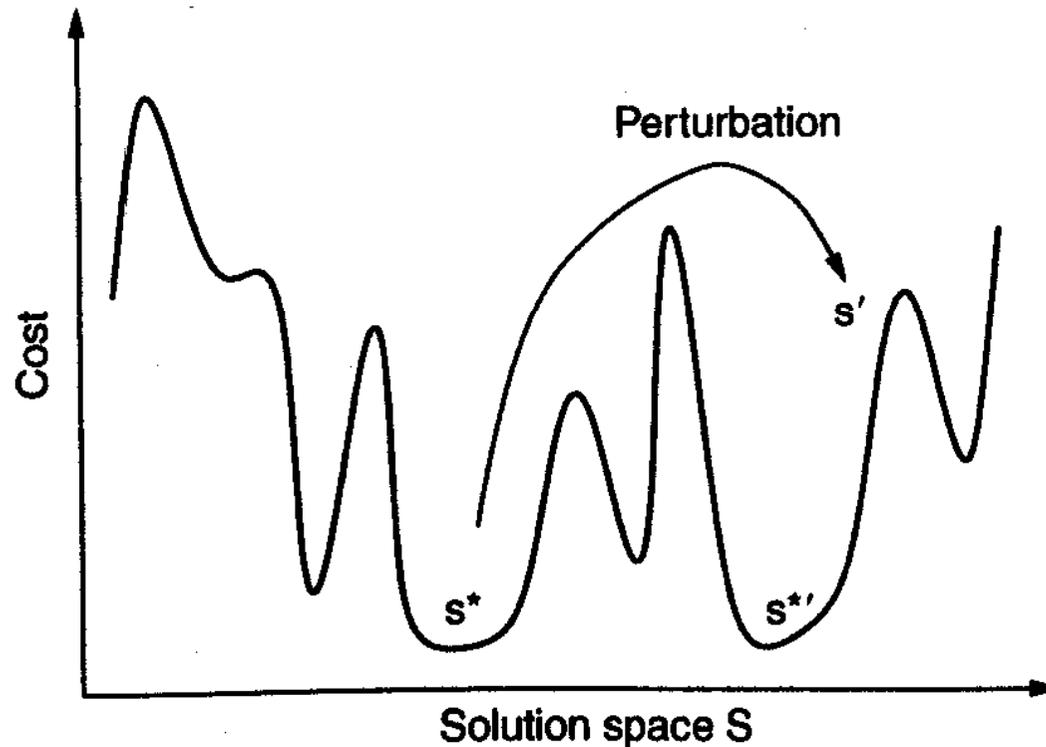


Figura: Representación del funcionamiento de la ILS

Caída en el óptimo local s^* . La perturbación/mutación conduce a s' . Después de aplicar la Búsqueda Local, se encuentra un nuevo óptimo s^{**} que es mejor que s^*

Búsqueda Local Iterativa: ILS

Procedimiento Búsqueda Local Reiterada (ILS)

Comienzo-ILS

$S_0 \leftarrow$ Generar-Solución-Inicial

$S \leftarrow$ Búsqueda Local (S_0)

Repetir

$S' \leftarrow$ Modificar (S , historia) %Mutación

$S'' \leftarrow$ Búsqueda Local (S')

$S \leftarrow$ Criterio-Aceptación (S , S'' , historia)

Actualizar (S , *Mejor_Solución*)

Hasta (Condiciones de terminación)

Devolver *Mejor_Solución*

Fin-ILS

Modelo General del Algoritmo ILS

Lecturas

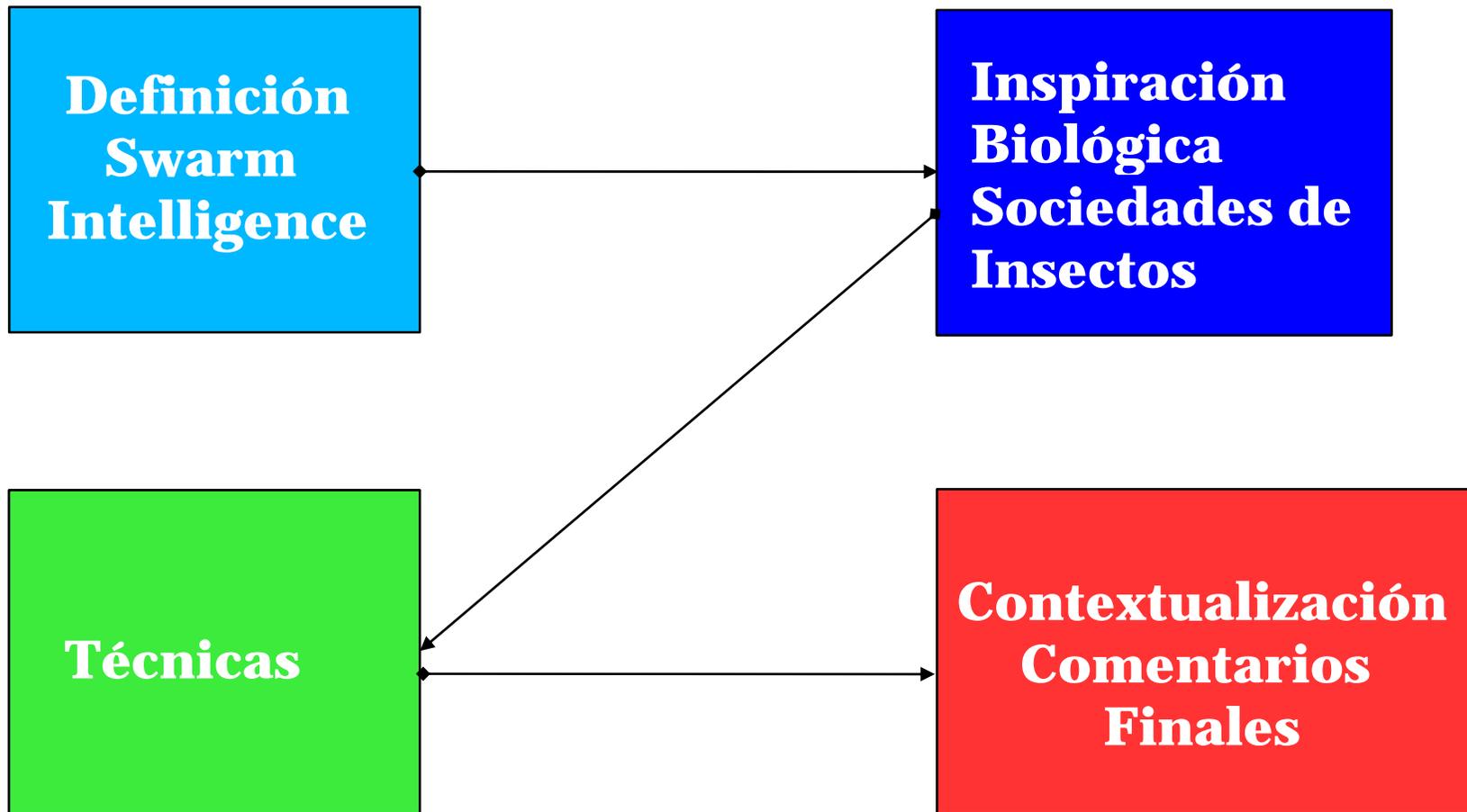
- K.A. Dowsland, B.A. Díaz. Diseño de Heurísticas y Fundamentos del Recocido Simulado. Revista Iberoamericana de Inteligencia Artificial 19 (2003) 93-102.
- B. Suman, P. Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. Journal of the Operational Research Society 57:10 (2006) 1143-1160.
- F. Glover, B. Melián. Búsqueda Tabú. Revista Iberoamericana de Inteligencia Artificial 19 (2003) 29-48.
- M. Gendreau. Chapter 2: An Introduction to Tabu Search. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 37-54.
- **M.G.C. Resende, C.S. Ribeiro. Chapter 8: Greedy Randomized Adaptive Search Procedure. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 221-249.**
- **R. Martí. Chapter 12: Multi-Start Methods. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 355-368.**
- **H.L. Lourenço, O.C. Martin, T. Stützle. Chapter 11: Iterated Local Search. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 321-353.**
- **P. Hansen, N. Mladenovic. Chapter 6: Variable Neighborhood Search. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 145-184.**

Introducción a los Algoritmos Metaheurísticos



1. Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda
2. MH I: Búsqueda Basada Trayectorias
3. **MH II: Swarm Intelligence**
4. **MH III: Algoritmos Evolutivos**
5. **Limitaciones de los Algoritmos MH: Teorema NFL**
6. **MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias**
7. **Consideraciones Sobre el Uso de las Metaheurísticas**
8. **Algunos Estudios/Extensiones de Interés**
9. **Comentarios Finales**

Swarm Intelligence



Introducción: Swarm Intelligence

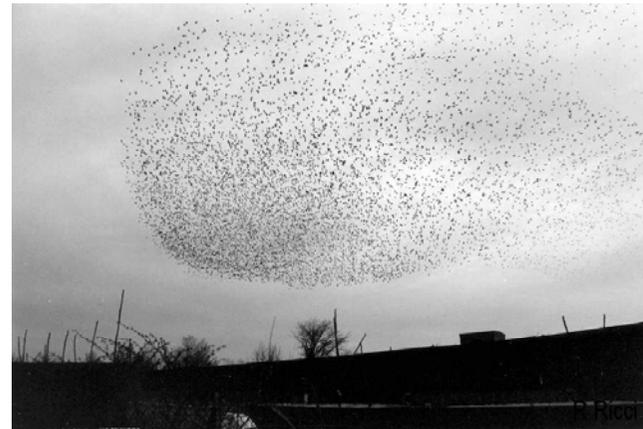
“La inteligencia colectiva emergente de un grupo de agentes simples”

“The emergent collective intelligence of groups of simple agents”

“ Algoritmos o mecanismos distribuidos de resolución de problemas inspirados en el comportamiento colectivo de colonias de insectos sociales u otras sociedades de animales”.

(Bonabeau, Dorigo, Theraulaz, 1999)

E. Bonabeau, M. Dorigo, G. Theraulaz
Swarm Intelligence. From Nature to
Artificial Systems.
Oxford University Press, 1999.



Introduction: Swarm Intelligence

Inspiración Biológica



sociedades de insectos
(bees, wasps, ants, termites)
(abejas, avisvas, hormigas, termitas)



flocks of birds
(bandadas de aves)



schools of fish
(bancos de peces)



herds of mammals
(manadas de mamíferos) 55

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Insectos

- Llevan a cabo actuaciones colectivas que no serían posibles para un único individuo
- Proporcionan comportamientos adecuados para la computación distribuida
- El repertorio de comportamientos de cada insecto es limitado
- No existe acceso individual al estado completo de la colonia
 - No pueden hacer una división efectiva de la labor a realizar
 - No pueden garantizar el progreso de la colonia

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Comportamiento emergente

- Las colonias de insectos llevan a cabo actuaciones de nivel complejo de forma inteligente, flexible y fiable, actuaciones que no serían factibles si tuviesen que ser realizadas por un insecto de forma individual (éstos son no inteligentes, no fiables, simples).
- Los insectos siguen reglas simples, y utilizan comunicación local simple
- La estructura global (nido) emerge desde las acciones de los insectos (las cuales son no fiables atendidas individualmente)

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Abejas



- **Cooperación de la colmena**
- **Regulan la temperatura de la colmena**
- **Eficiencia vía especialización: división de la labor en la colonia**
- **Comunicación: Las fuentes de comida son explotadas de acuerdo a la calidad y distancia desde la colmena**

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

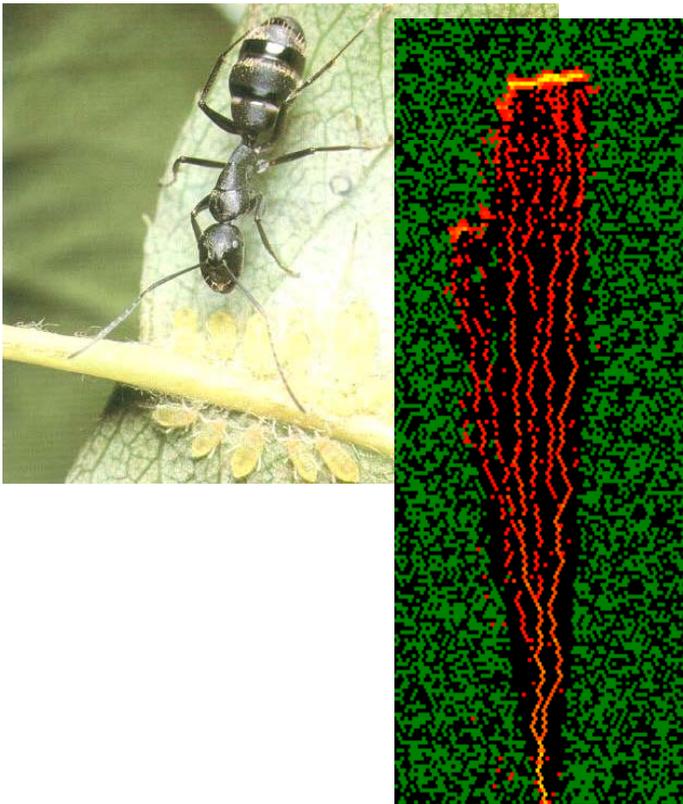
Termitas



- Nido con forma de cono con paredes externas y conductos de ventilación
- Camaras de camadas en el centro de la colmena
- Rejillas del ventilación en espirales
- Columnas de soporte

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Hormigas



- Organizan autopistas “hacia y desde” la comida por medio de rastros de feromona (pheromone)

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Sociedades de Insectos/Tipos de interacción entre Insectos Sociales

- Sistemas de toma de decisión colectiva
- Comunicación directa/interacción directa
 - Comida/intercambio de líquidos, contacto visual, contacto químico (pheromones)
- Comunicación indirecta/interacción indirecta (Stigmergy)
 - El comportamiento individual modifica el entorno, el cual a su vez modifica el comportamiento de otros individuos

⇒ **sociedades auto-organizadas**

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Características de un Enjambre

- Compuesto de agentes *simples* (Self-Organized)
- Descentralizado
 - No hay un único supervisor
- No hay un plan global (emergente)
- Robusto
 - Las actuaciones se completan aunque un individuo falle
- Flexible
 - Puede responder a cambios externos
 - Percepción del entorno (sentidos)
 - No existe un modelo explícito de entorno/abilidad para cambiarlo

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Resumen

- La complejidad y sofisticación de la auto-organización se lleva a cabo sin un líder/jefe de la sociedad
- Lo que podemos aprender de los insectos sociales lo podemos aplicar al campo del diseño de Sistemas Inteligentes
- La modelización de los insectos sociales por medio de la auto-organización puede ser de ayuda para el diseño de modelos artificiales distribuidos de resolución de problemas. Esto es conocido como:

Swarm Intelligent Systems.

Swarm Intelligence: Técnicas

- ¿Cómo puede ser coordinado el esfuerzo individual para alcanzar una tema común?

Swarm Inspired Methods

- **particle swarm optimization – PSO**

Optimización basada en nubes de partículas

- Conjunto de técnicas inspiradas en el comportamiento de las bandadas de aves o bancos de peces

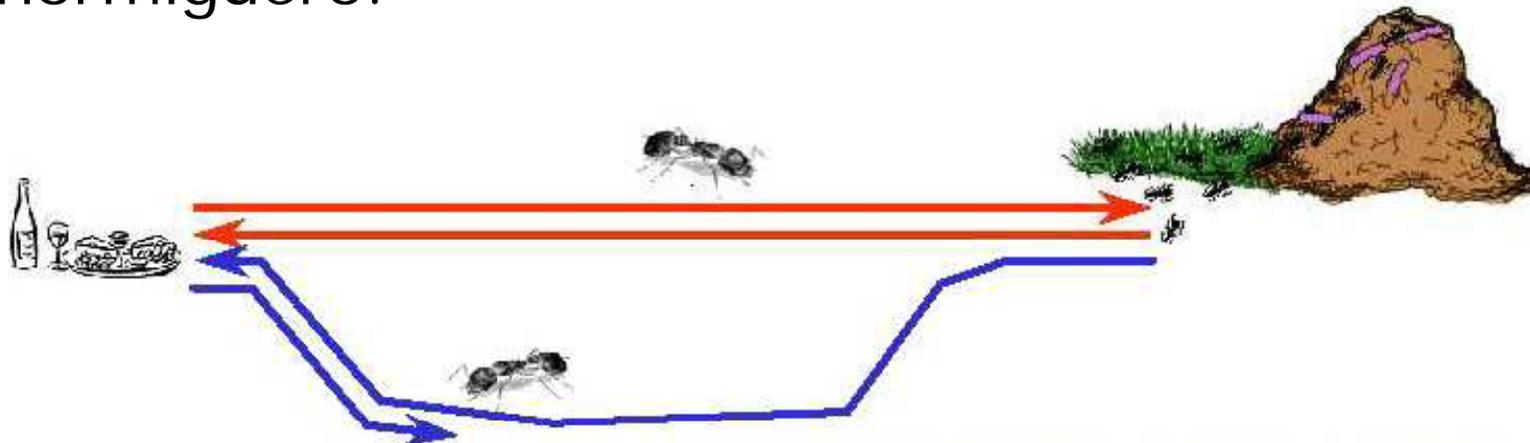
- **ant colony optimization – ACO**

Optimización basada en colonias de hormigas

- Conjunto de técnicas inspiradas por las actividades de una colonia de hormigas

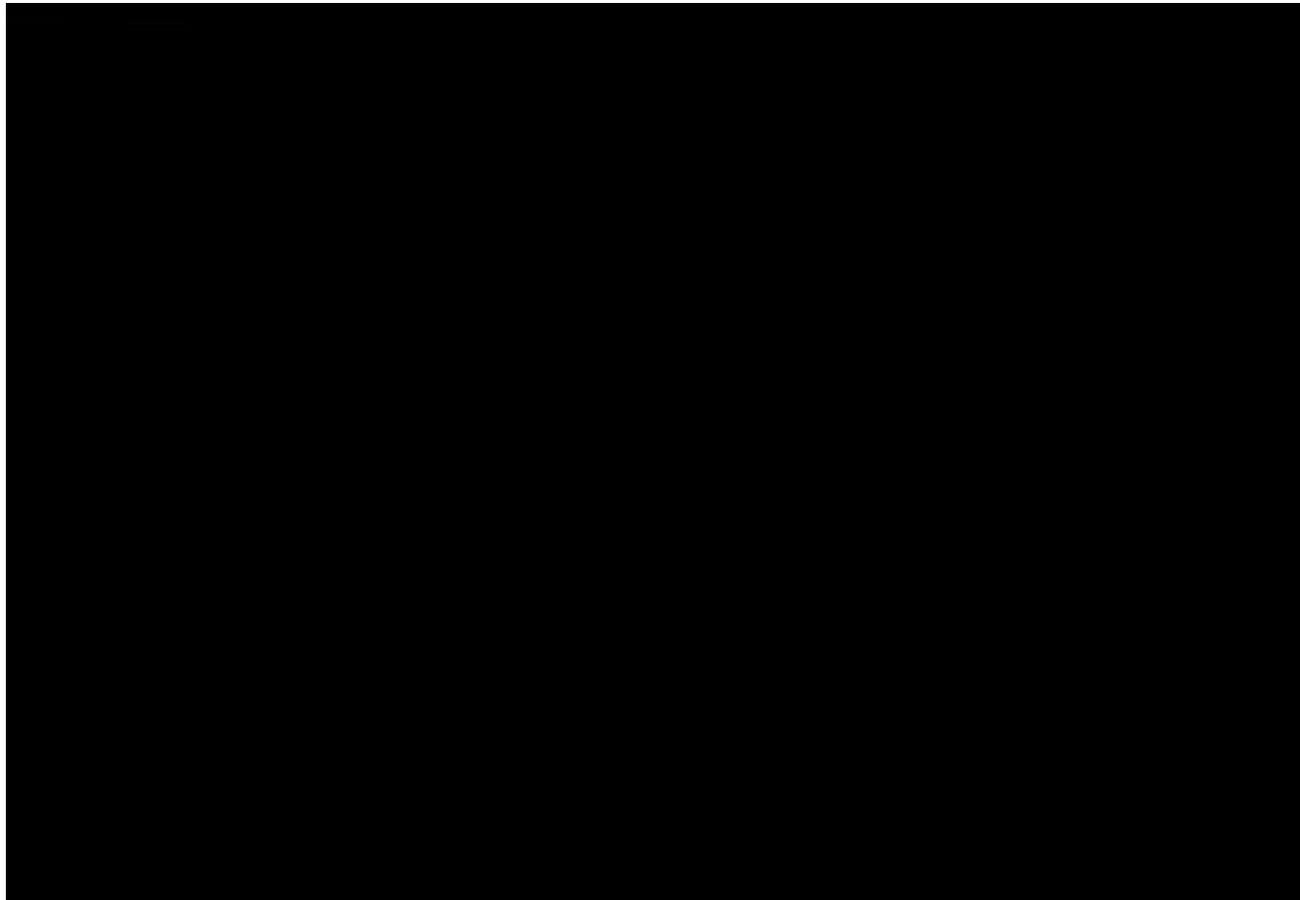
Ant Colony Optimization

- La analogía más cercana a ACO son los problemas de rutas en grafos
- Mientras las hormigas buscan comida, depositan rastros de feromona que atraen a otras hormigas. Desarrollan caminos mínimos entre la comida y el hormiguero.



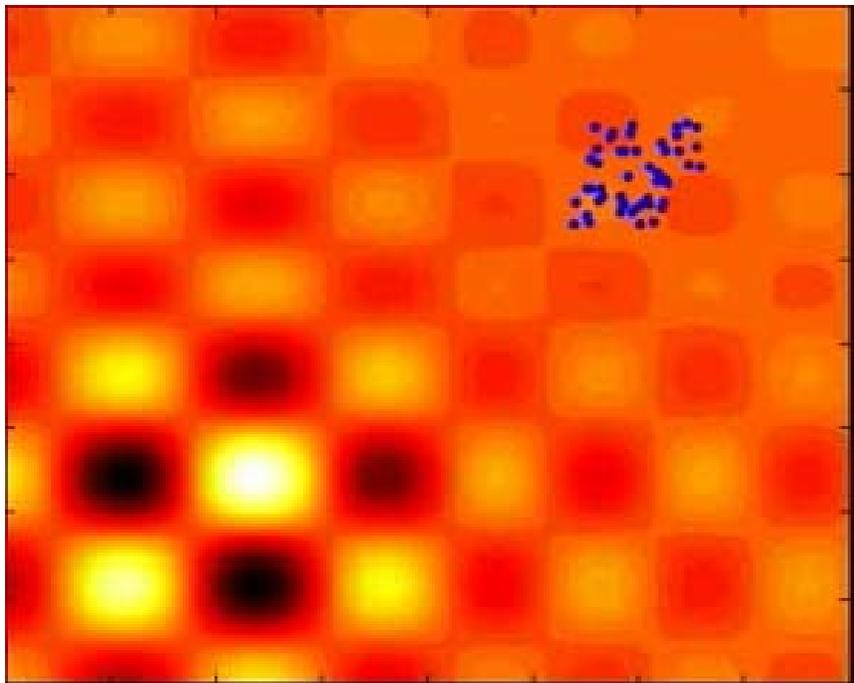
Ant Colony Optimization

Experimento con Hormigas reales. Como encuentran el camino mínimo (159 segundos)



Particle Swarm Optimization

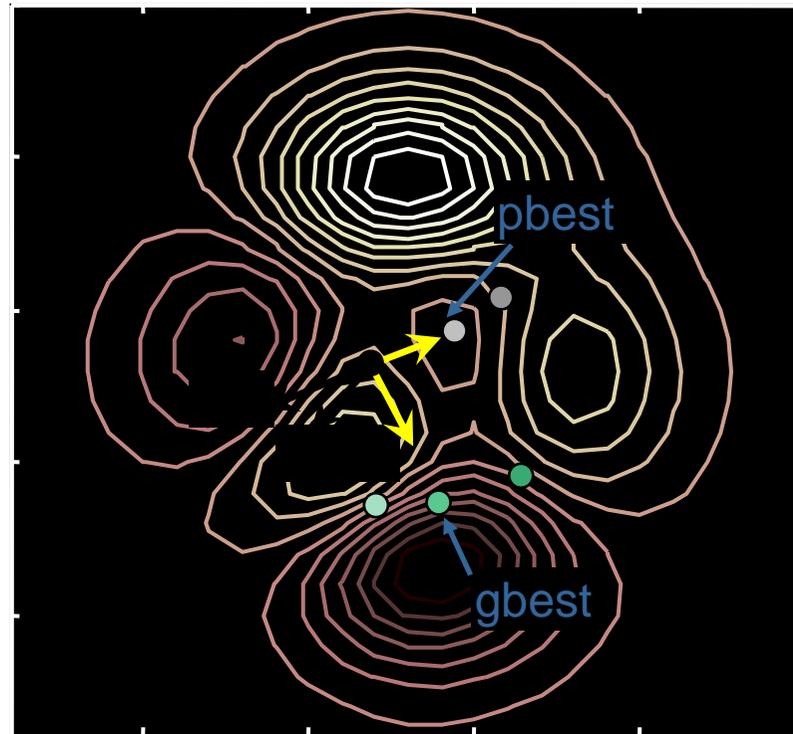
- PSO emula este escenario para resolver problemas de optimización. Cada solución (**partícula**) es un *“ave”* en el espacio de búsqueda que está siempre en continuo movimiento y que nunca muere.



Particle Swarm Optimization

Particle Swarm Optimization (PSO) aplica conceptos de interacción social a la resolución de problemas de búsqueda/optimización.

En PSO, un enjambre de n individuos se comunica directa o indirectamente con otros vía las direcciones de búsqueda.

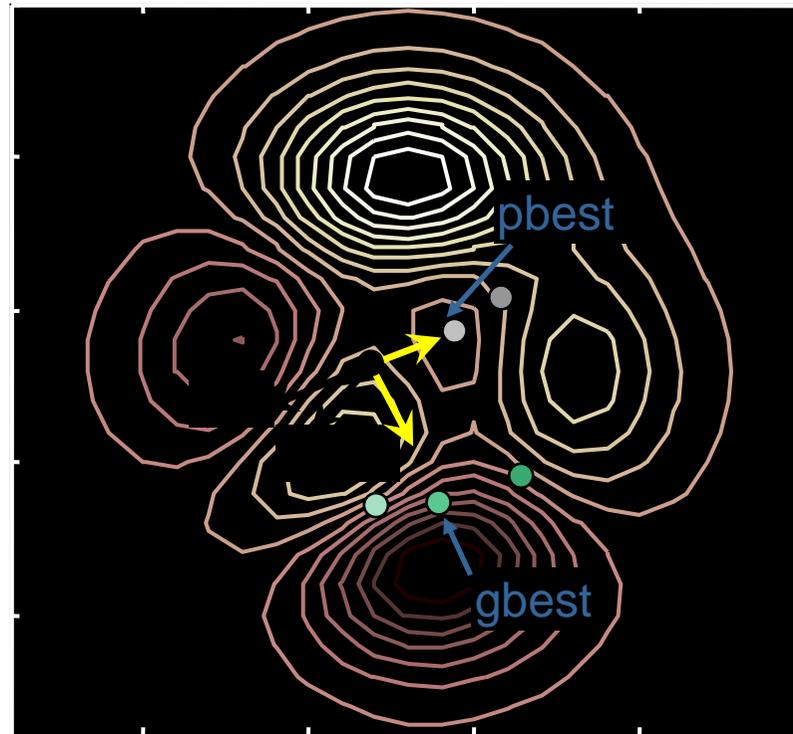


Particle Swarm Optimization

Las Partículas pueden ser simples agentes que vuelan a través del espacio de búsqueda y almacenan (y posiblemente comunican) la mejor solución que han descubierto.

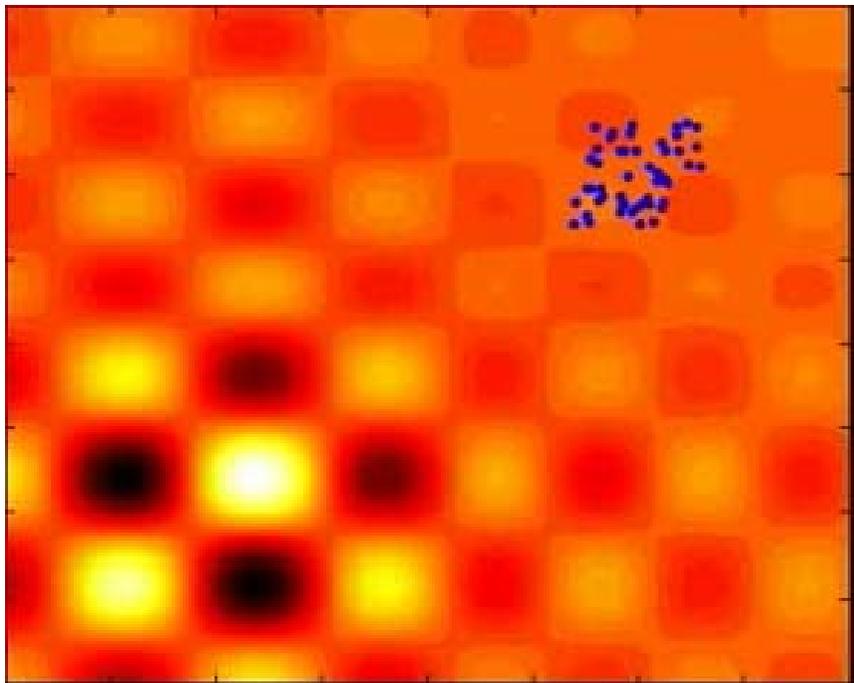
En PSO, las partículas nunca mueren.

La pregunta es, "¿Cómo se mueve una partícula desde una localización a otro del espacio de búsqueda?"

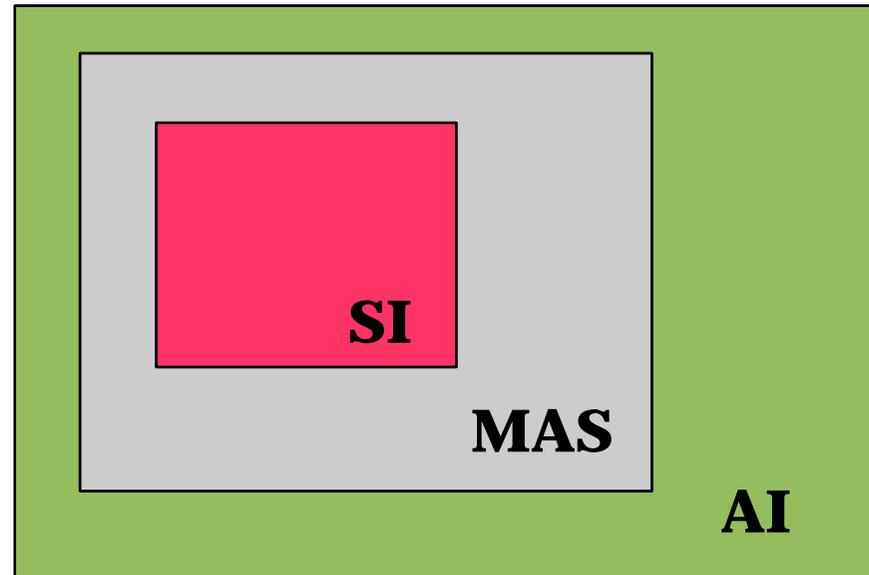


Particle Swarm Optimization

- PSO emula este escenario para resolver problemas de optimización. Cada solución (**partícula**) es un *“ave”* en el espacio de búsqueda que está siempre en continuo movimiento y que nunca muere.



Comentarios Finales



- AI** - **Artificial Intelligence**
- MAS** - **Multi Agent Systems**
- SI** - **Swarm Intelligence**

**Una definición de contexto de Swarm Intelligence
en el ámbito de la Inteligencia Artificial**

Comentarios Finales

Swarm Intelligence (SI) es la propiedad de un sistema por la cual el comportamiento colectivo de agentes (no sofisticados) interactúa localmente con el entorno proporcionando un patrón global de funcionamiento coherente como “emergente”.

SI proporciona una base con la cual es posible explorar la resolución de problemas colectivamente (o de forma distribuida) sin un control centralizado ni un modelo global de comportamiento.

Comentarios Finales

“Dumb parts, properly connected into a swarm, yield smart results”.

“Partes tontas/mudas, conectadas adecuadamente en un enjambre, producen resultados elegantes/inteligentes”.

Bibliografía

E. Bonabeau, M. Dorigo, G. Theraulaz. *Swarm Intelligence. From Nature to Artificial Systems.* Oxford Univ. Press, 1999.

M. Dorigo, T. Stuetzle. *Ant Colony Optimization.* MIT Press, 2004.

Kennedy, J., Eberhart, R. C., and Shi, Y.. *Swarm intelligence.* Morgan Kaufmann Publishers, 2001.

Lecturas

- S. Garnier, J. Gautrais, G. Theraulaz. The biological principles of swarm intelligence. *Swarm Intelligence* 1 (2007) 3-31.
- O. Cordón, F. Herrera and T. Stützle. A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends. *Mathware and Soft Computing* 9:2-3, 2002, pp. 141-175.
- M. Dorigo, T. Stützle. Chapter 9: The ant colony optimization metaheuristic: Algorithms, applications and advances. In: F. Glover, G.A. Kochenberber, (Eds.). *Handbook of Metaheuristics*. Kluwer Academics (2003) 251-285.
- A. Banks, J. Vincent, C. Anyakoha. A review of particle swarm optimization. Part I: background and development. *Natural Computing* 6 (2007) 467-484.
- A. Banks, J. Vincent, C. Anyakoha. A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Natural Computing* 7 (2008) 109-124.
- W.B. Langdon, R. Poli. Evolving Problems to Learn About Particle Swarm Optimizers and Other Search Algorithms. *IEEE Transactions on Evolutionary Computation* 11:5 (2007) 561-578.
- Y. del Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, R.G. Harley. Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation* 12:2 (2008) 171-195.

Introducción a los Algoritmos Metaheurísticos



1. Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda
2. MH I: Búsqueda Basada Trayectorias
3. MH II: Swarm Intelligence
4. **MH III: Algoritmos Evolutivos**
5. Limitaciones de los Algoritmos MH: Teorema NFL
6. MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias
7. Consideraciones Sobre el Uso de las Metaheurísticas
8. Algunos Estudios/Extensiones de Interés
9. Comentarios Finales

Algoritmos Evolutivos

Introducción

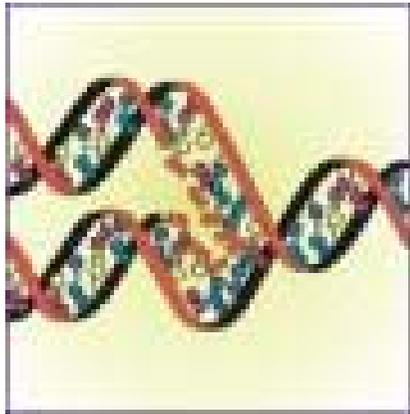
Evolución Natural

Evolución Artificial

Contexto

Comentarios Finales

INTRODUCCIÓN



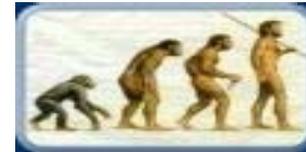
COMPUTACION EVOLUTIVA

Está compuesta por modelos de evolución basados en poblaciones cuyos elementos representan soluciones a problemas.

La simulación de este proceso en un ordenador resulta ser una técnica de optimización probabilística, que con frecuencia mejora a otros métodos clásicos en problemas difíciles.

EVOLUCIÓN NATURAL

En la naturaleza, los procesos evolutivos ocurren cuando se satisfacen las siguientes condiciones:



Una entidad o individuo tiene la habilidad de reproducirse.

Hay una población de tales individuos que son capaces de reproducirse.

Existe alguna variedad, diferencia, entre los individuos que se reproducen.

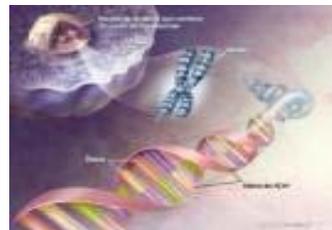
Algunas diferencias en la habilidad para sobrevivir en el entorno están asociadas con esa variedad.



EVOLUCIÓN NATURAL

Los mecanismos que conducen esta evolución no son totalmente conocidos, pero sí algunas de sus características, que son ampliamente aceptadas:

La evolución es un proceso que opera sobre los cromosomas más que sobre las estructuras de la vida que están codificadas en ellos.

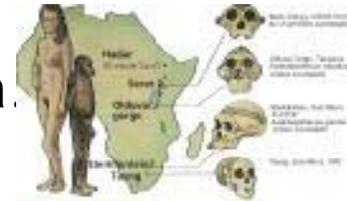


EVOLUCIÓN NATURAL

La selección natural es el enlace entre los cromosomas y la actuación de sus estructuras decodificadas.

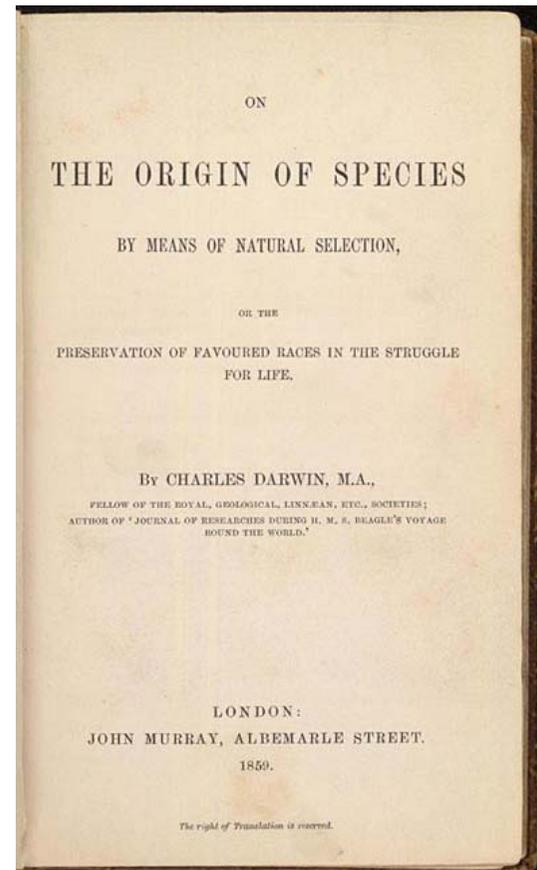
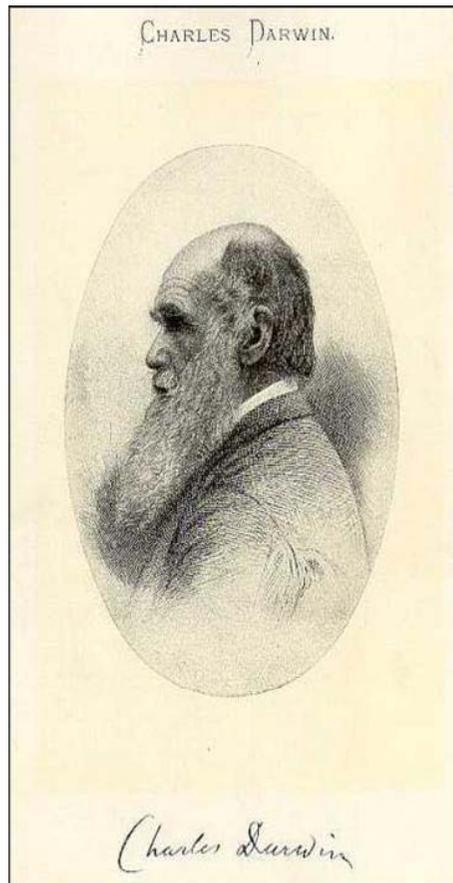
El proceso de reproducción es el punto en el cual la evolución toma parte, actúa.

La evolución biológica no tiene memoria.



Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection or the Preservations of Favored Races in the Struggle for Life*. London: John Murray.

EVOLUCIÓN NATURAL



EVOLUCIÓN ARTIFICIAL

LA METÁFORA

EVOLUCIÓN

RESOLUCIÓN
DE PROBLEMAS

Individuo



Solución Candidata

Adaptación



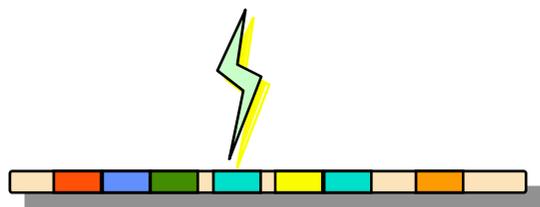
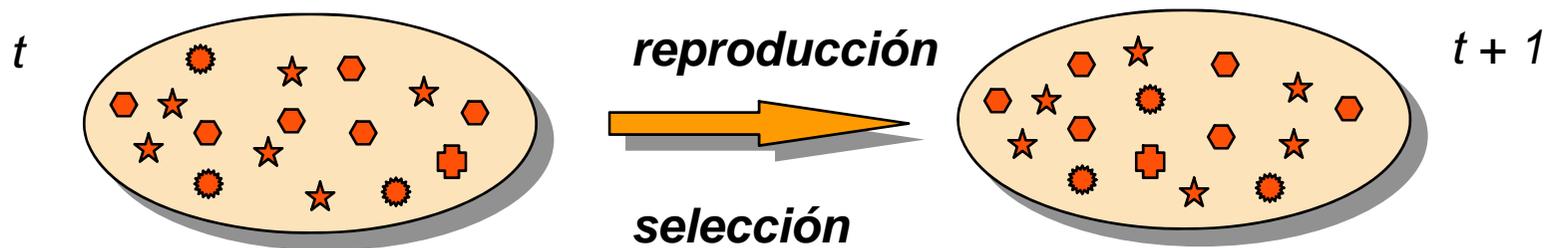
Calidad

Entorno

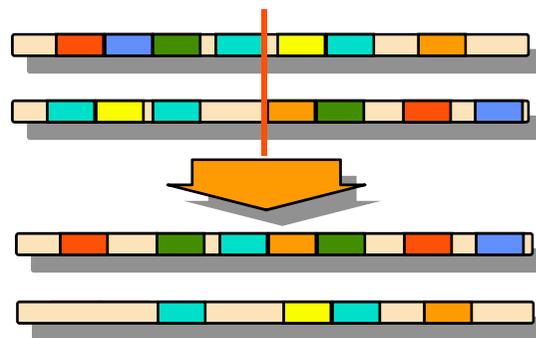


Problema

EVOLUCIÓN ARTIFICIAL



mutación



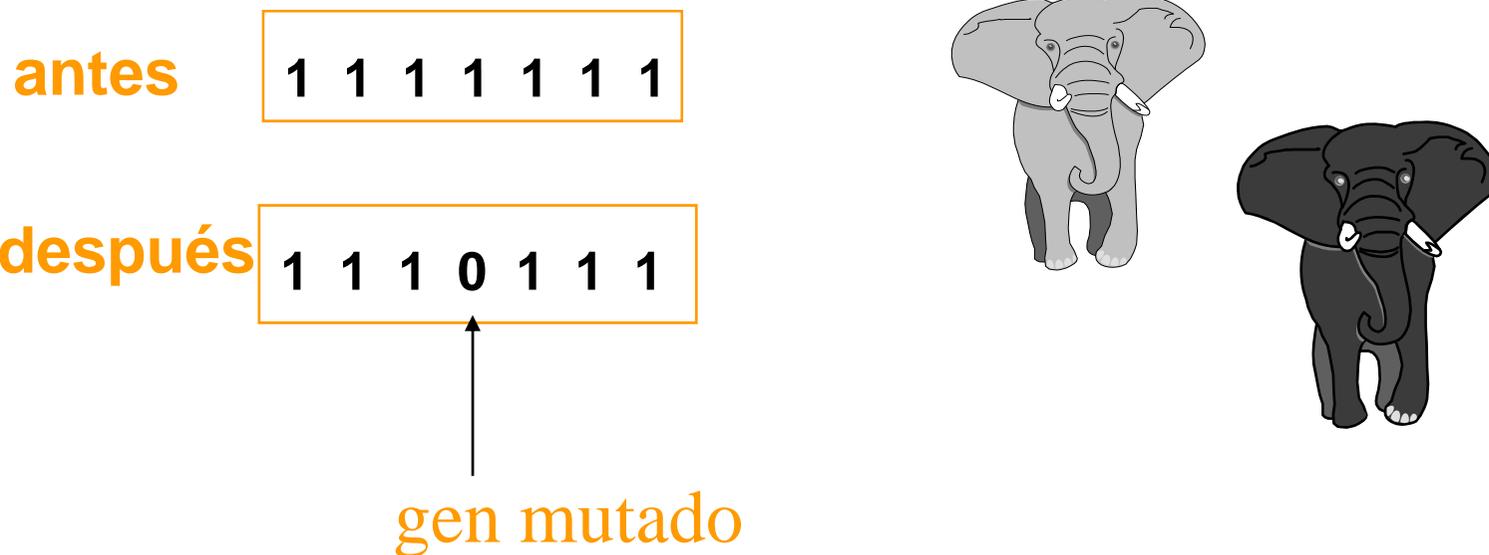
recombinación

**RECOMBINACIÓN
OPTATIVA**

LOS INGREDIENTES

EVOLUCIÓN ARTIFICIAL

Ejemplo: Mutación para representación binaria

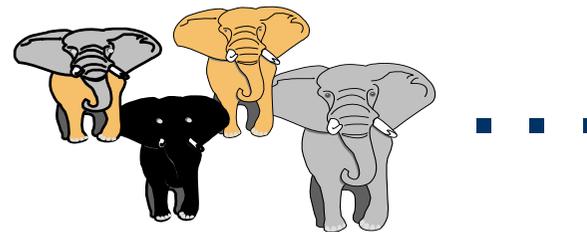


La mutación suele ocurrir con probabilidad p_m para cada gen

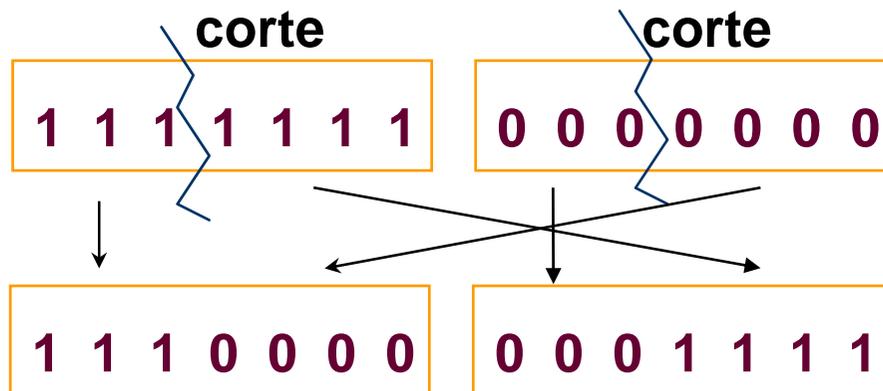
EVOLUCIÓN ARTIFICIAL

Ejemplo: Recombinación para representación binaria

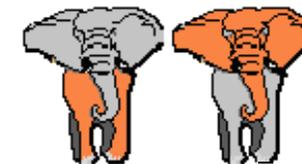
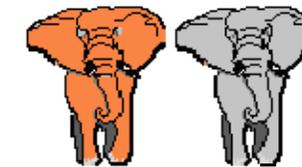
Población:



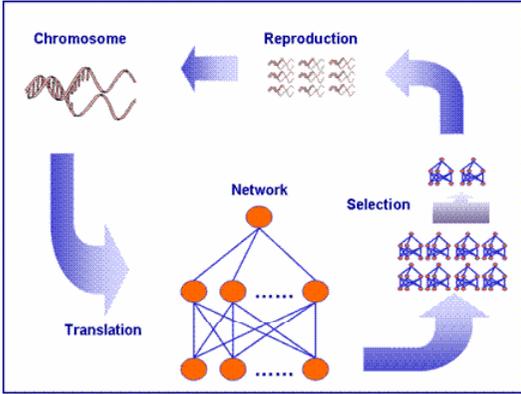
Cada cromosoma se trocea en n partes las cuales son recombinadas. (Ejemplo para $n=1$)



padres



descendientes



EVOLUCIÓN ARTIFICIAL

Selección

PADRES

Recombinación

Mutación

*El ciclo de
la
Evolución*

POBLACIÓN

Reemplazamiento

DESCENDIENTES

EVOLUCIÓN ARTIFICIAL

Existen cuatro paradigmas básicos:

Algoritmos Genéticos que utilizan operadores genéticos sobre cromosomas. 1975, Michigan University



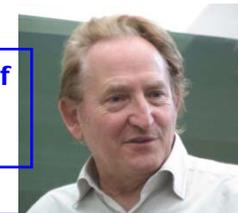
John Holland
Inventor of genetic algorithms
Professor of CS and Psychology at the U. of Michigan.

Estrategias de Evolución que enfatizan los cambios de comportamiento al nivel de los individuos. 1964, Technische Universität Berlin



Hans-Paul Schwefel
Universität Dortmund

Inventors of Evolution Strategies



Ing. Ingo Rechenberg
Bionics & Evolutionstechnik
Technical University Berlin
<http://www.bionik.tu-berlin.de/>

Programación Evolutiva que enfatizan los cambios de comportamiento al nivel de las especies. 1960-1966, Florida



Lawrence J. Fogel,
Natural Selection, Inc.
Inventor of Evolutionary Programming

Programación Genética que evoluciona expresiones representadas como árboles. 1989, Stanford University



John Koza
Stanford University.
Inventor of Genetic Programming

EVOLUCIÓN ARTIFICIAL

Existen otros múltiples Modelos de Evolución de Poblaciones:

EDA: Estimation Distribution Algorithms (Algoritmos basados en Estimación de Distribuciones)

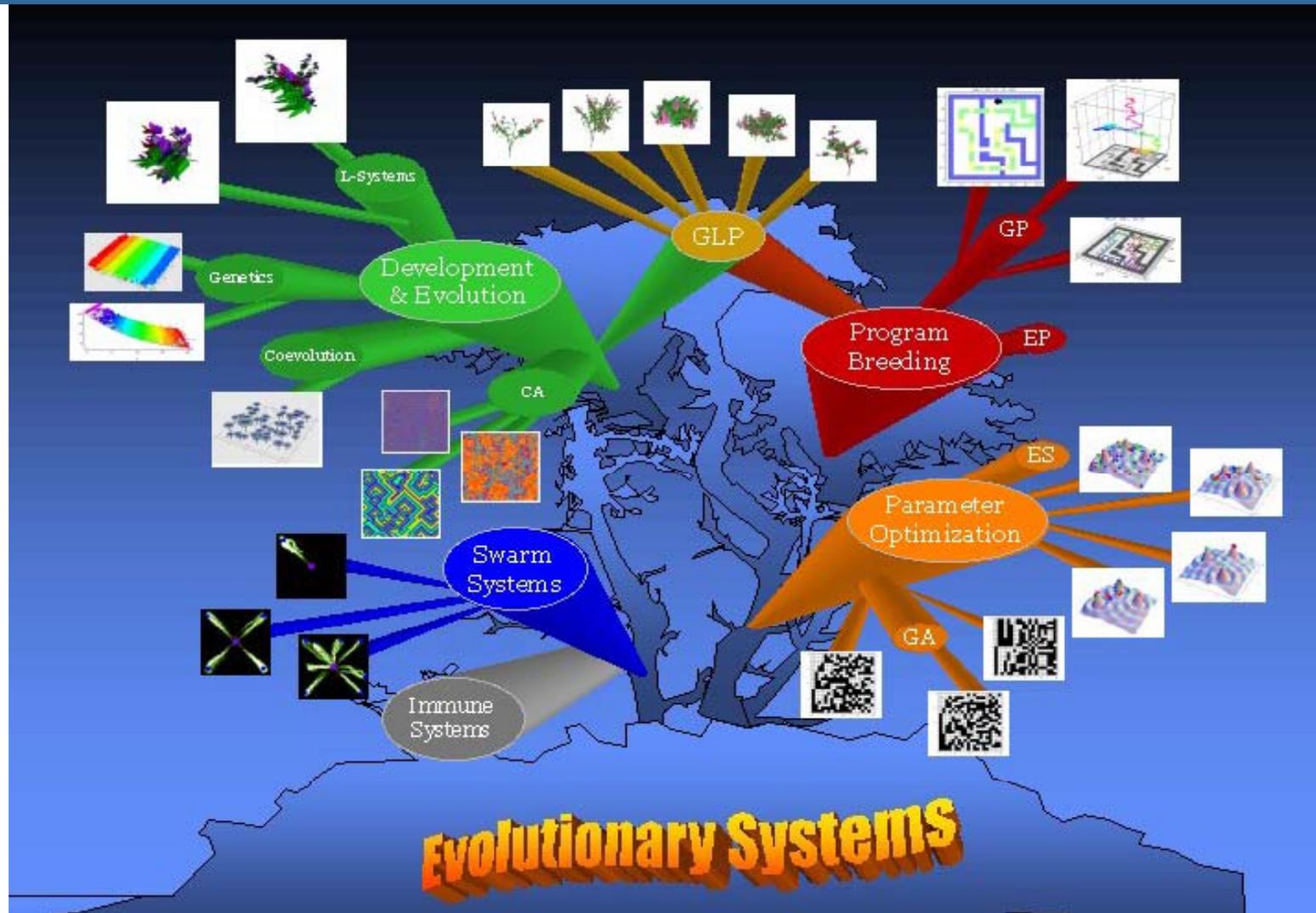
DE: Differential Evolution (Evolución Diferencial)

Algoritmos Evolutivos Culturales

Algoritmos Meméticos

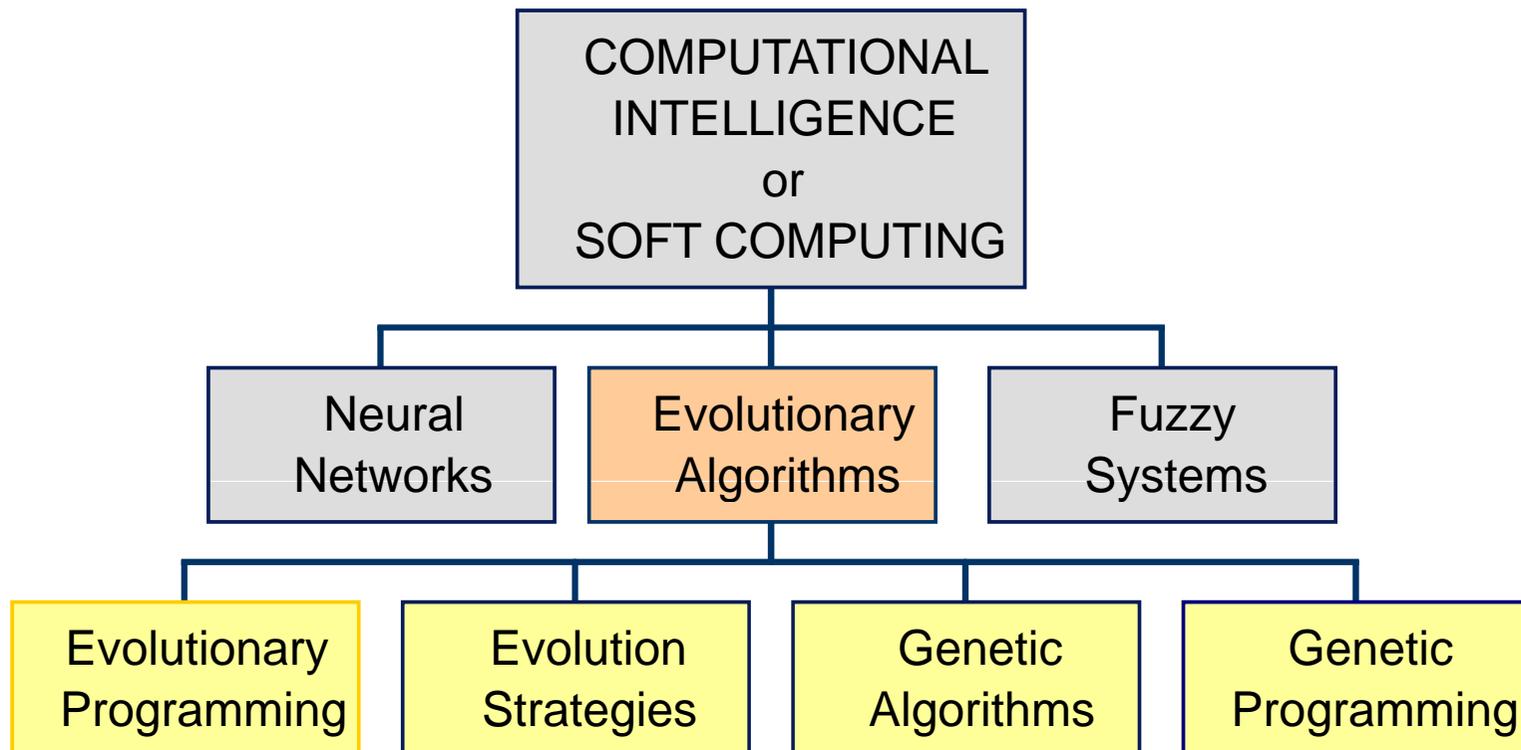
Scatter Search – Búsqueda Dispersa

EVOLUCIÓN ARTIFICIAL



CONTEXTUALIZACIÓN

COMPUTACIÓN INTELIGENTE TAXONOMÍA



COMENTARIOS FINALES

COMPORTAMIENTO

- Buena actuación a un costo aceptable en una amplia variedad de problemas
- Paralelismo intrínseco
- Superioridad con respecto a otras técnicas en problemas complejos:
 - con muchos parámetros
 - relación compleja entre parámetros
 - muchos óptimos (locales)

COMENTARIOS FINALES

VENTAJAS

- **Sin restricciones sobre el espacio de soluciones**
- **Amplia aplicabilidad**
- **Bajo coste en desarrollo**
- **Fáciles de hibridar con otras técnicas**
- **Soluciones interpretables**
- **Se pueden ejecutar interactivamente**
- **Proporcionan un conjunto de soluciones**

COMENTARIOS FINALES

DESVENTAJAS

- **No garantizan una solución óptima en un tiempo finito**
- **Débil base teórica**
- **Tienen muchos parámetros a ajustar**
- **Computacionalmente costosos (lentos)**

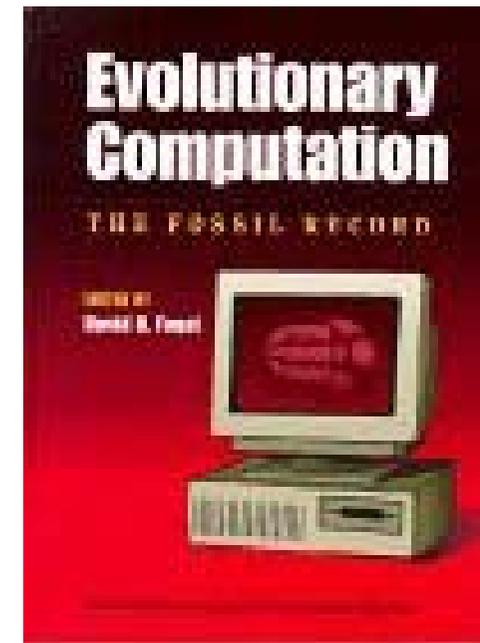
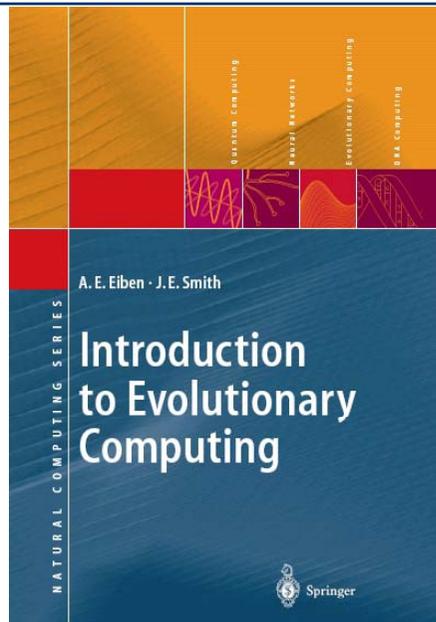
COMENTARIOS FINALES

RESUMEN

- **basados en una metáfora biológica:
la evolución**
 - **gran potencialidad de aplicación**
 - **muy popular en muchos campos**
 - **muy potente en diversas aplicaciones**
 - **altas prestaciones a bajo costo**
-
- **SON ATRACTIVOS DESDE UN PUNTO
DE VISTA COMPUTACIONAL**

COMPUTACIÓN EVOLUTIVA

A.E. Eiben, J.E. Smith
Introduction to Evolutionary Computation.
Springer Verlag 2003.
(Natural Computing Series)



D.B. Fogel (Ed.)
Evolutionary Computation. The Fossil Record.
(Selected Readings on the
History of Evolutionary Computation).
IEEE Press, 1998.

Lecturas

- Thomas Bäck, Ulrich Hammel, Hans-Paul Schwefel. Evolutionary Computation: Comments on the History and Current State. IEEE Transactions on Evolutionary Computation 1:1 (1997) 3-17.
- C. Reeves. Chapter 3: Genetic Algorithms. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 55-82.
- F. Herrera, M. Lozano, A.M. Sánchez. A Taxonomy for the Crossover Operator for Real-Coded Genetic Algorithms: An Experimental Study. International Journal of Intelligent Systems 18 (2003) 309-338R.
- Storn, K. Price. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11 (1997) 341-359.

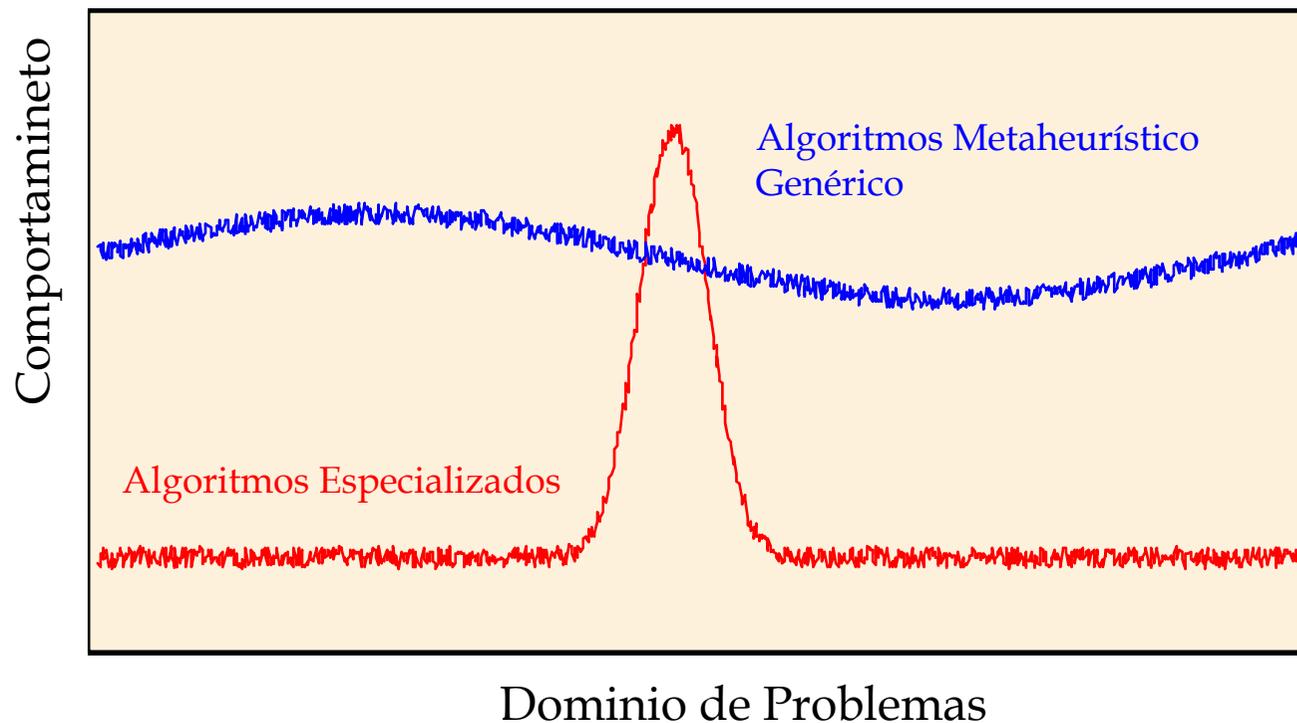
Introducción a los Algoritmos Metaheurísticos



1. Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda
2. MH I: Búsqueda Basada Trayectorias
3. MH II: Swarm Intelligence
4. MH III: Algoritmos Evolutivos
5. **Limitaciones de los Algoritmos MH: Teorema NFL**
6. **MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias**
7. **Consideraciones Sobre el Uso de las Metaheurísticas**
8. **Algunos Estudios/Extensiones de Interés**
9. **Comentarios Finales**

Limitaciones los Algoritmos de Metaheurísticos

Sobre el Comportamiento de los Algoritmos Metaheurísticos



Limitaciones de los Algoritmos de Metaheurísticos

No Free Lunch Theorem (1995):

"...for any algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class." **Wolpert and Macready (1997)**

$$\sum_f E(\vec{c} / f, m, a) = \sum_f E(\vec{c} / f, m, b)$$



David. H. Wolpert



William G. Macready

No free lunch theorems for optimization

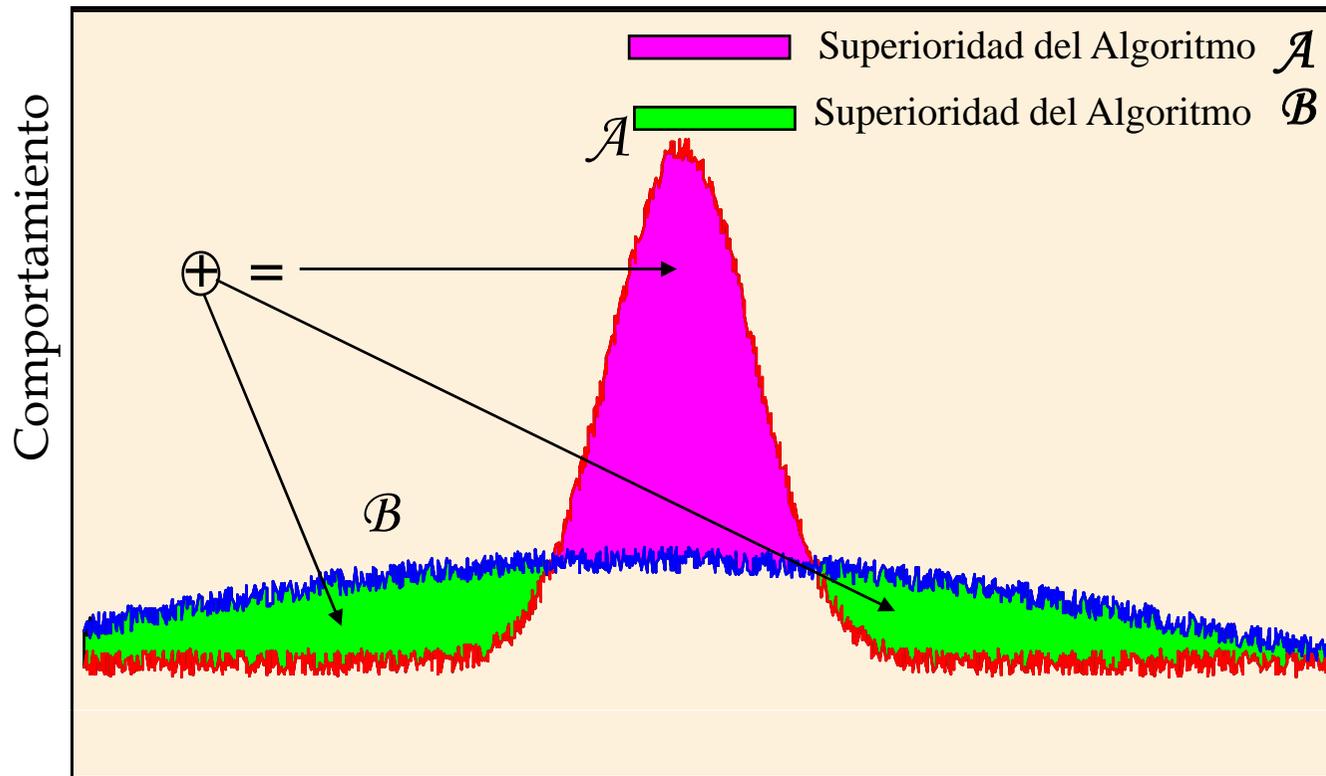
Wolpert, D.H.; Macready, W.G.;

Evolutionary Computation, IEEE Transactions on 1:1, April 1997, 67 – 82

Limitaciones los Algoritmos de Metaheurísticos

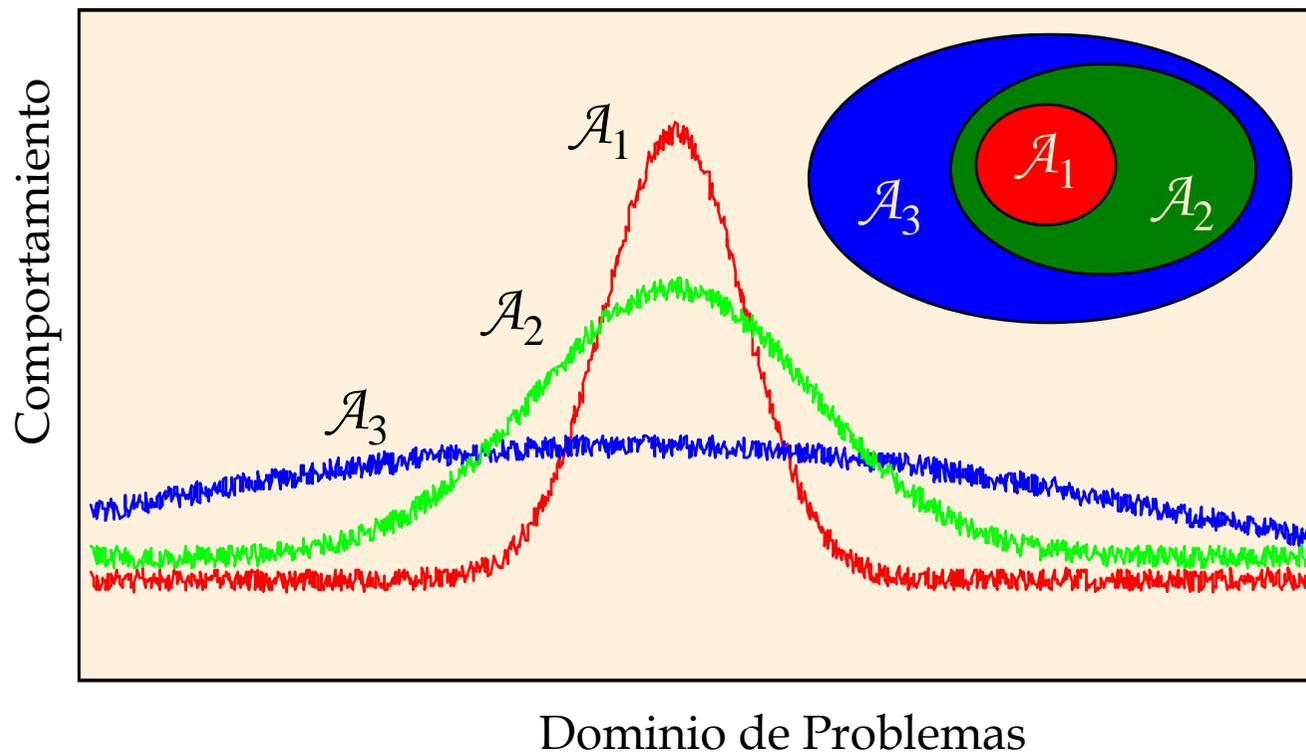
Implicaciones de NFL (I)

$$\sum_f E(\vec{c} / f, m, a) = \sum_f E(\vec{c} / f, m, b)$$



Limitaciones los Algoritmos de Metaheurísticos

Implicaciones de NFL (II): Ganar en un cierto dominio implica perder en los restantes

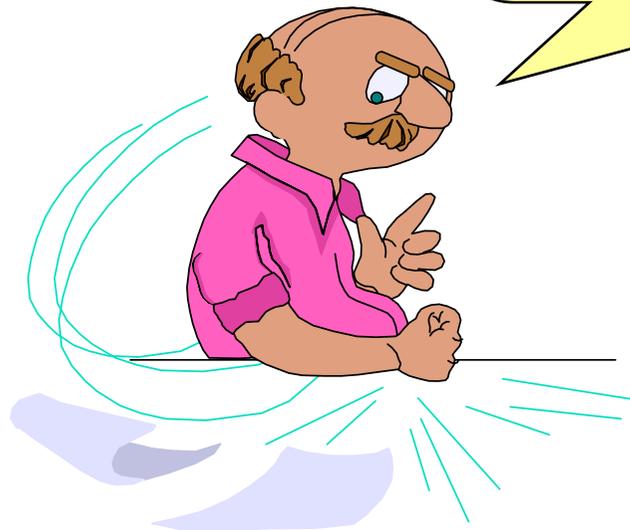


Limitaciones los Algoritmos de Metaheurísticos

Discusión

¿La búsqueda aleatoria se comporta como nuestro algoritmo?

¡Hay que discutir algunas cuestiones en el teorema de NFL!



Limitaciones los Algoritmos de Metaheurísticos

Consideremos los diferentes grados de conocimiento del problema:

1. Conocimiento perfecto
2. **Conocimiento parcial**
3. Poco conocimiento
4. Muy poco conocimiento
5. **Ningún conocimiento (NFL)**

Los resultados del teorema NFL son críticos tanto en cuanto (2) es comparado con (5).

El conocimiento sobre el problema debe ser incluido en el algoritmos de búsqueda.

Introducción a los Algoritmos Metaheurísticos

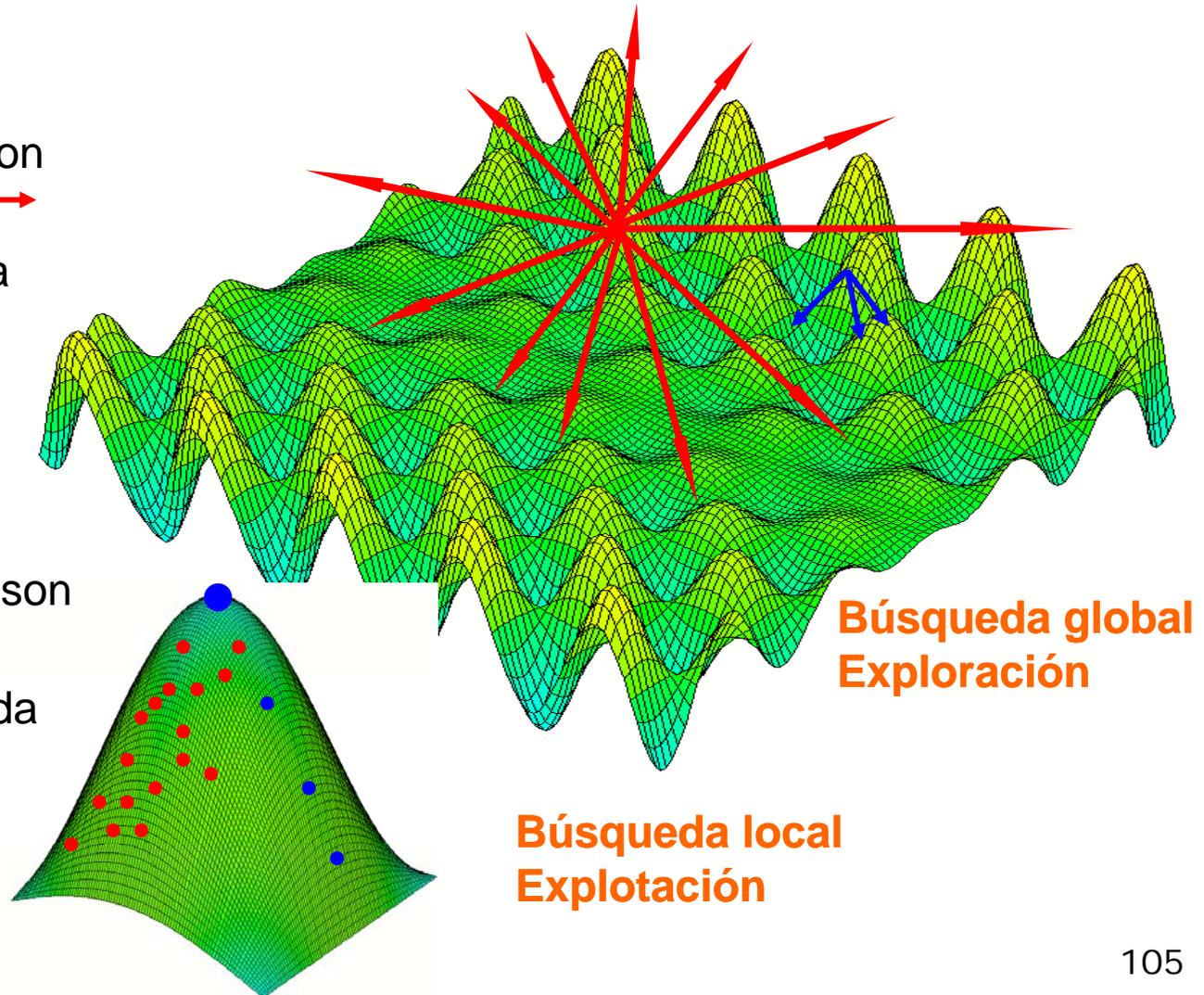


1. Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda
2. MH I: Búsqueda Basada Trayectorias
3. MH II: Swarm Intelligence
4. MH III: Algoritmos Evolutivos
5. Limitaciones de los Algoritmos MH: Teorema NFL
6. **MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias**
7. **Consideraciones Sobre el Uso de las Metaheurísticas**
8. **Algunos Estudios/Extensiones de Interés**
9. **Comentarios Finales**

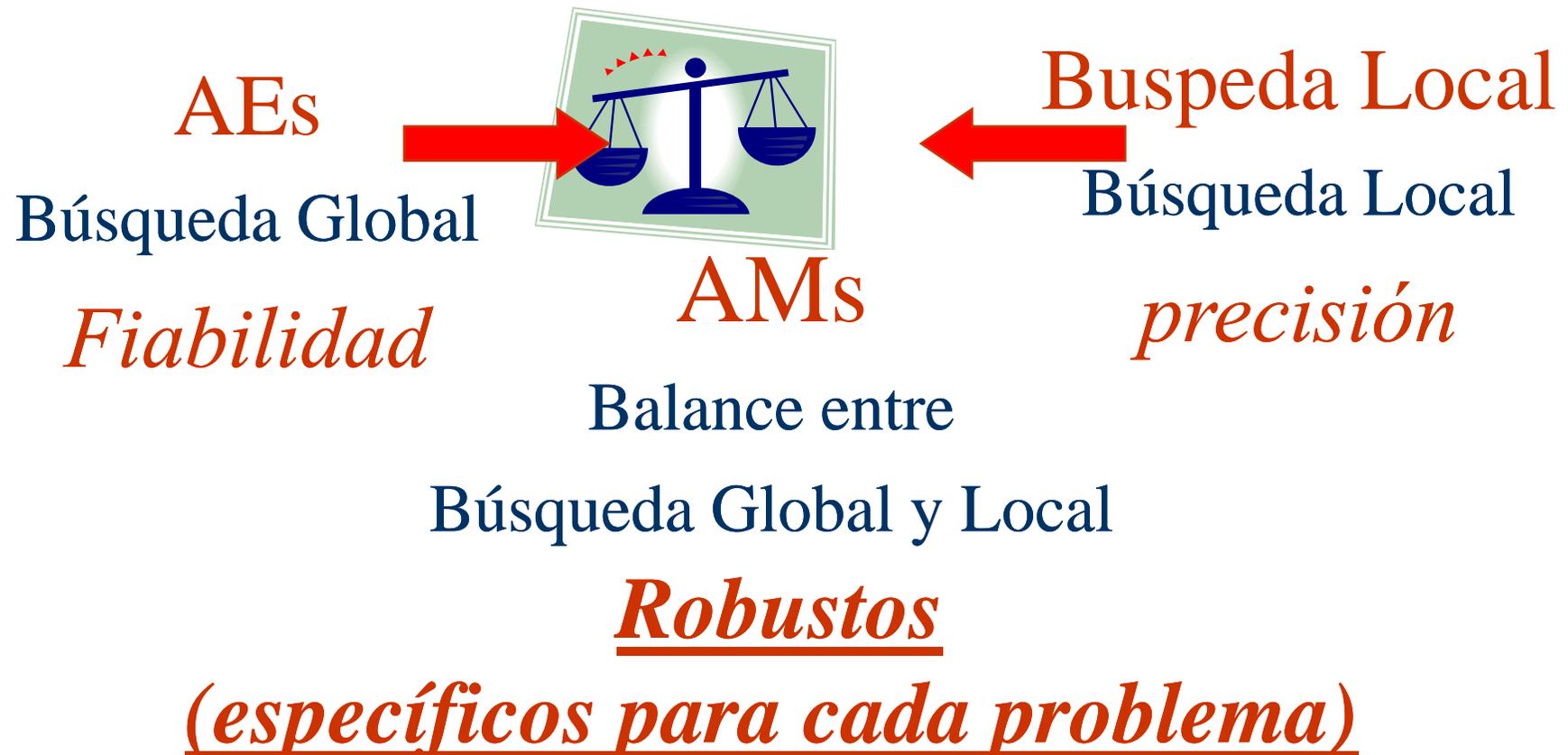
Algoritmos Híbridos: Poblaciones vs Trayectorias

- Algoritmos evolutivos son buenos exploradores →
- Algoritmos de búsqueda local son malos exploradores →

- Algoritmos evolutivos son malos explotadores ●
- Algoritmos de búsqueda local son buenos explotadores ●

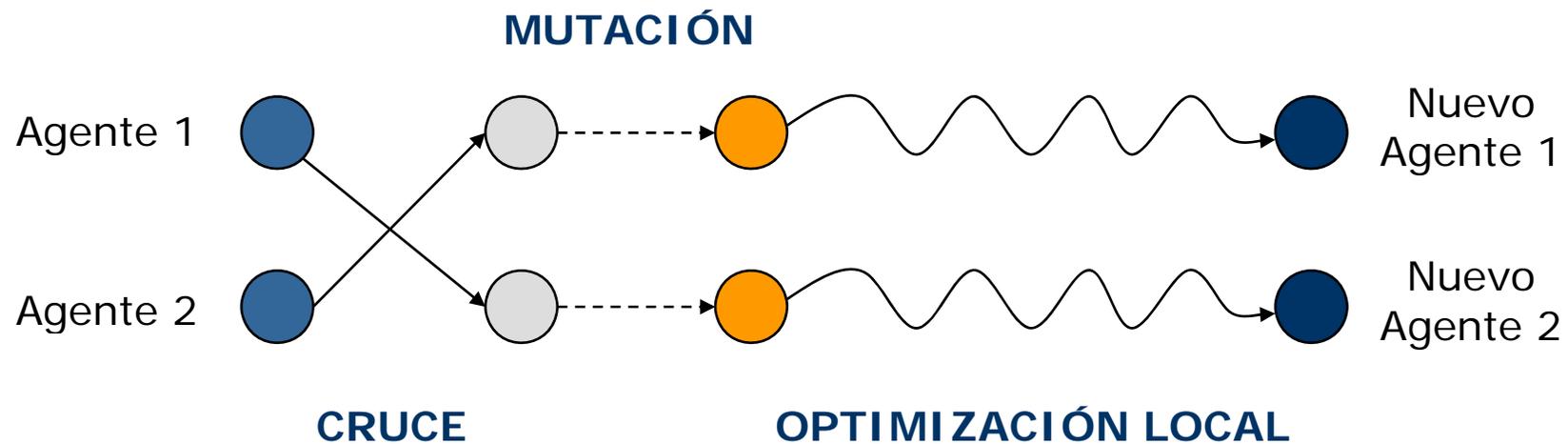


Algoritmos Híbridos: Poblaciones vs Trayectorias



Algoritmos Híbridos: Poblaciones vs Trayectorias

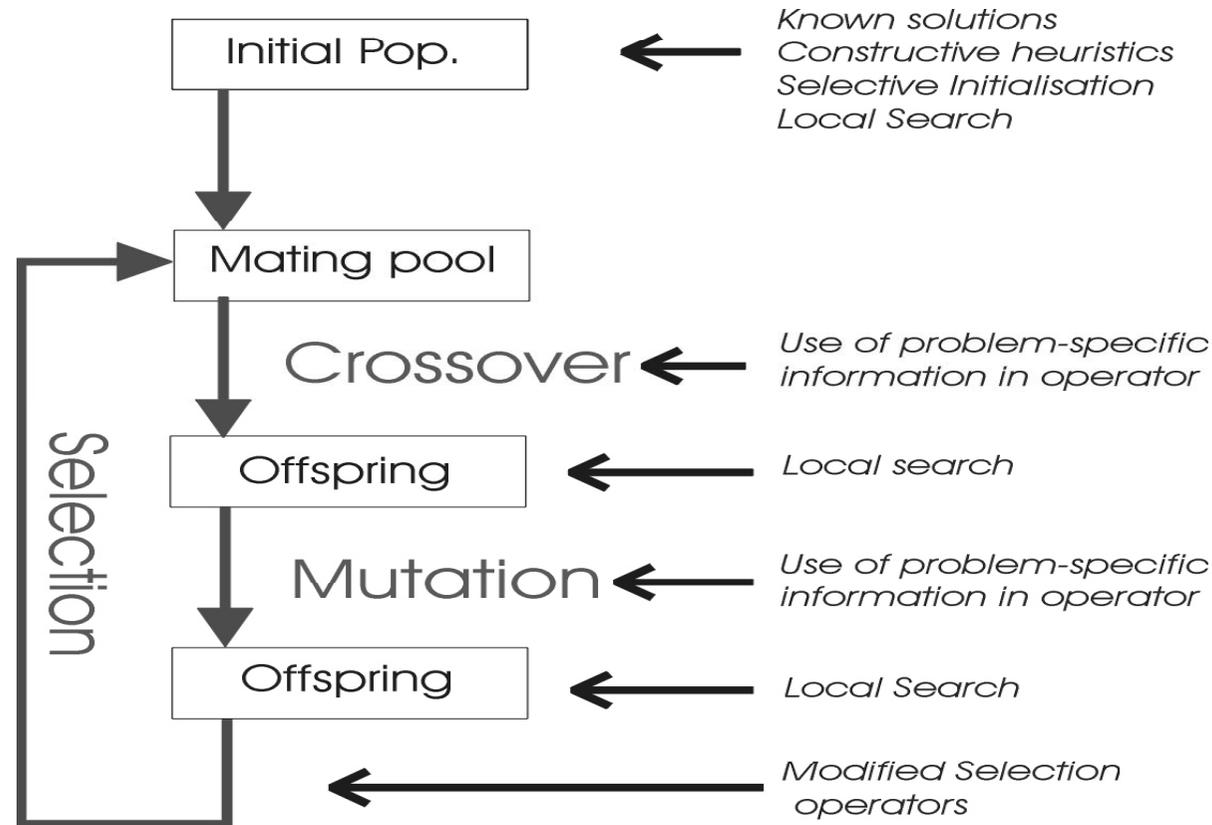
Algoritmos Meméticos



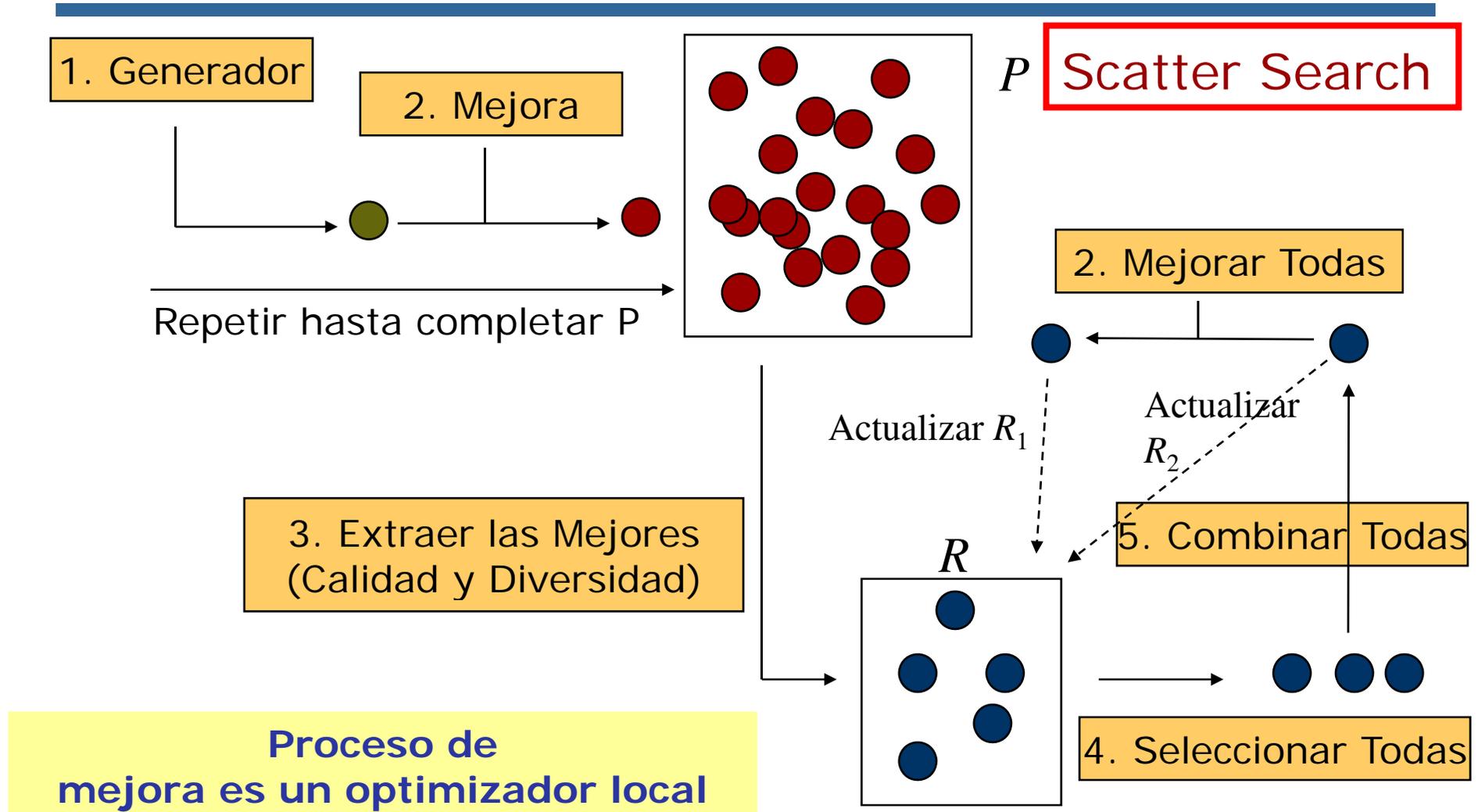
Algoritmos Híbridos: Poblaciones vs Trayectorias

Uso de Conocimiento:
Otras Hibridaciones

Algoritmos Meméticos



Algoritmos Híbridos: Poblaciones vs Trayectorias



Lecturas

- P. Moscato, C. Cotta. Una Introducción a los Algoritmos Meméticos. Revista Iberoamericana de Inteligencia Artificial 19 (2003) 131-148.
- P. Moscato, C. Cotta. Chapter 5: A Gentle Introduction to Memetic Algorithms. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 105-144.
- N. Krasnogor and J.E. Smith. A tutorial for competent memetic algorithms: model, taxonomy and design issues. IEEE Transactions on Evolutionary Computation 9(5):474-488, 2005.

Introducción a los Algoritmos Metaheurísticos



1. Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda
2. MH I: Búsqueda Basada Trayectorias
3. MH II: Swarm Intelligence
4. MH III: Algoritmos Evolutivos
5. Limitaciones de los Algoritmos MH: Teorema NFL
6. MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias
7. **Consideraciones Sobre el Uso de las Metaheurísticas**
8. **Algunos Estudios/Extensiones de Interés**
9. **Comentarios Finales**

Consideraciones sobre el uso de las MH

Búsquedas basadas en trayectorias/entornos

Búsquedas basadas en poblaciones

Lectura: A. Hertz, M. Widmer. Guidelines for the use of meta-heuristics in combinatorial optimization. European Journal of Operational Research 151 (2003) 247-252.

Búsquedas Basadas en Trayectorias/Entornos

- Debería ser fácil generar una solución al problema
 - En algunos problemas, generar una solución factible resulta en sí mismo un problema NP-duro
 - Igualmente, en estos casos resulta complejo aplicar un operador de vecino que garantice obtener una solución factible
 - En estos casos, puede ser más útil relajar las restricciones del problema permitiendo generar soluciones no válidas, y añadir una penalización en la función objetivo que evite aceptar estas soluciones

Búsquedas Basadas en Trayectorias/Entornos

- A partir de cada solución, debería poder trazarse un camino que lleve esta solución a una solución óptima
 - El operador de vecino empleado debe permitir que aplicándolo sucesivas veces se alcance un óptimo
- Cualquier solución del entorno generado a partir de una solución actual debe estar relativamente cercana a ésta
 - El operador de vecino debe hacer cambios “suaves”

Búsquedas Basadas en Trayectorias/Entornos

- La topología generada a partir de la función de coste sobre el conjunto de soluciones factibles no debería ser demasiado plana
 - En algunos problemas, la función de coste es, por ejemplo, del tipo “minimizar la máxima distancia entre dos puntos”
 - Este tipo de funciones genera planicies haciendo que distintas soluciones tengan el mismo valor objetivo
 - Sin embargo, unas soluciones estarán más cerca (desde el punto de vista de la codificación) que otras de la solución óptima
 - En estos casos, se puede añadir una segunda componente a la función objetivo que discrimine entre estas soluciones

Búsquedas Basadas en Poblaciones

- Debe transmitirse la información relevante en la fase de cooperación (combinación)

En algunos problemas, no basta con asegurar que los descendientes contengan parte de la codificación de los padres, sino que sea la información contenida en los padres la que realmente se transmita

- La combinación de dos padres equivalentes no debería producir un descendiente diferente a los padres
- Debe mantenerse diversidad en la población para evitar que el algoritmo se estanque (convergencia prematura)

Introducción a los Algoritmos Metaheurísticos



1. Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda
2. MH I: Búsqueda Basada Trayectorias
3. MH II: Swarm Intelligence
4. MH III: Algoritmos Evolutivos
5. Limitaciones de los Algoritmos MH: Teorema NFL
6. MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias
7. Consideraciones Sobre el Uso de las Metaheurísticas
8. **Algunos Estudios/Extensiones de Interés**
9. **Comentarios Finales**

Algunos Estudios/Extensiones de Interés

Diversidad versus Convergencia

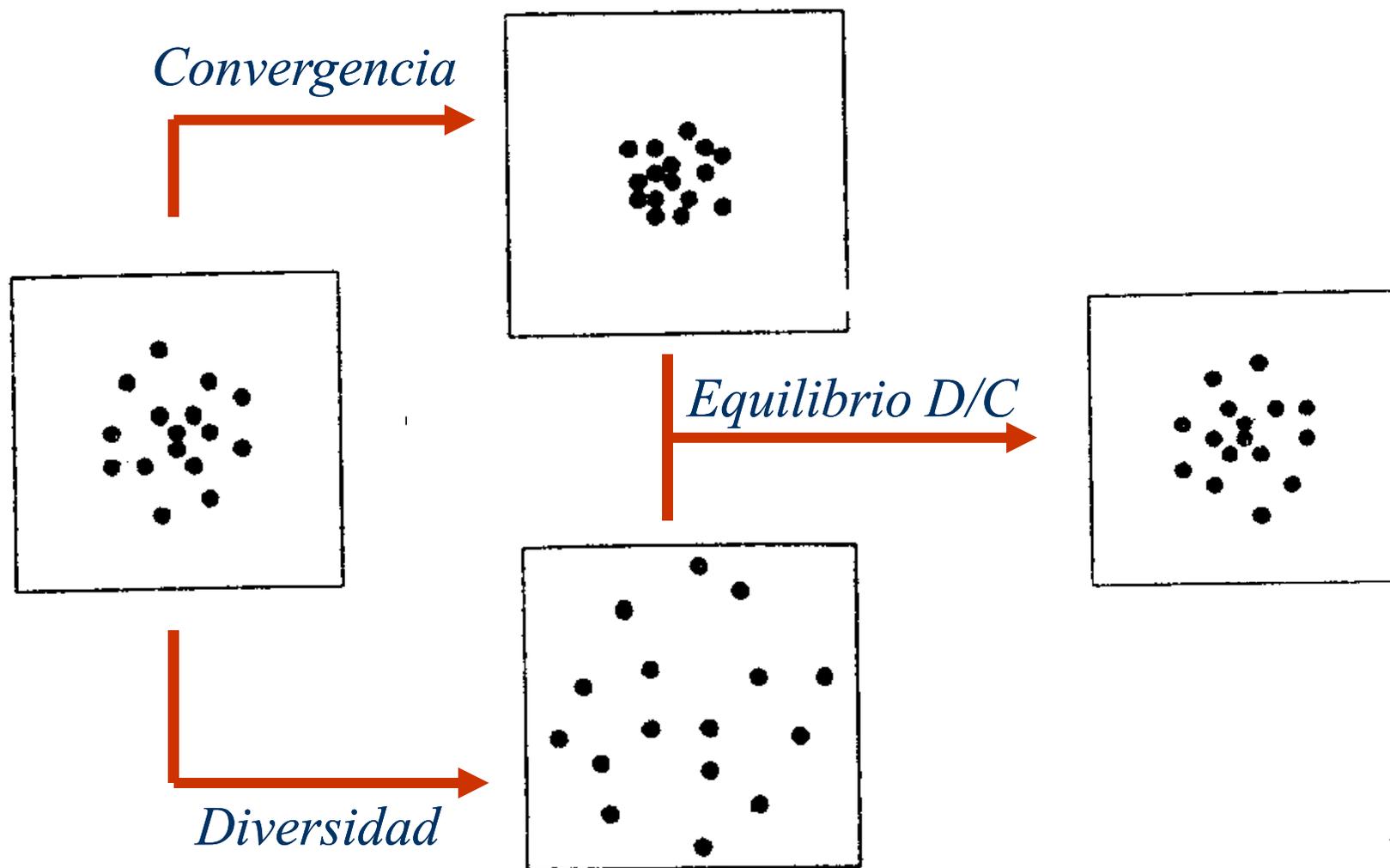
Problemas Multimodales

Problemas Multiobjetivo

Algoritmos Evolutivos Paralelos

Coevolución

Diversidad vs. Convergencia



Diversidad vs. Convergencia

Algunas propuestas en el ámbito de los Algoritmos Evolutivos:

- **Diversidad con la Mutación**
- **Diversidad con el Cruce**
- **Separación Espacial**
- **Adaptación, Auto-adaptación, Metaevolución**
- **Estrategias de Reemplazamiento**

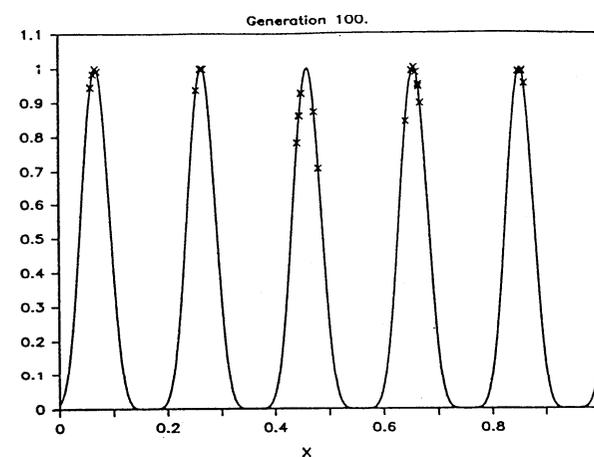
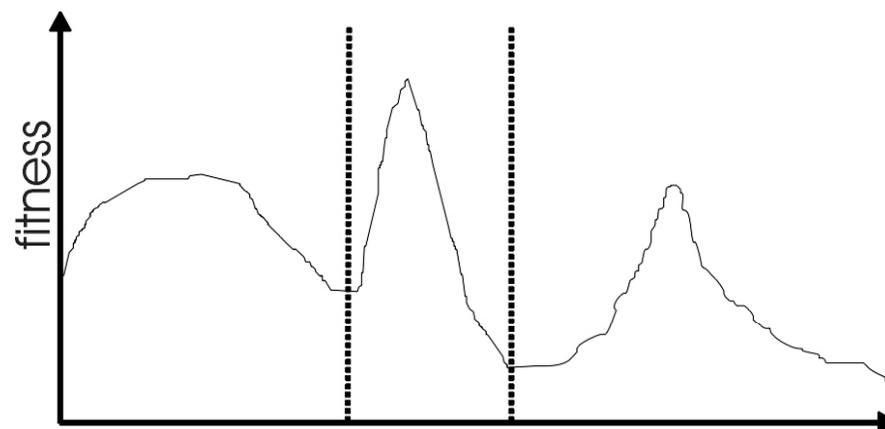
Problemas Multimodales

Muchos problemas interesantes tienen más de una solución localmente óptima.

Son problemas que tienen múltiples óptimos locales o múltiples óptimos globales (múltiples soluciones al problema).

En la resolución del problema se desea obtener varios de esos óptimos locales o globales.

Técnica: Niching Genetic Algorithms



Problemas Multiobjetivo

Un **Problema Multiobjetivo** consiste en:

$$\text{Max o Min } z = f(x) = (f_1(x), f_2(x), \dots, f_n(x))$$

Soluciones pareto-optimales o no-dominadas: Se dice que un vector **a** domina a otro **b** (se nota como $\mathbf{a} \prec = \mathbf{b}$) si, y sólo si:

$$\forall i \in \{1, 2, \dots, n\} \mid f_i(\mathbf{a}) \geq f_i(\mathbf{b}) \wedge \exists j \in \{1, 2, \dots, n\} \mid f_j(\mathbf{a}) > f_j(\mathbf{b})$$

Es decir, una solución domina a otra si es mejor o igual en todos los objetivos y al menos mejor en uno de ellos.

Todos los vectores de decisión que no son dominados por ningún otro vector se llaman **pareto-optimales o no-dominados**.

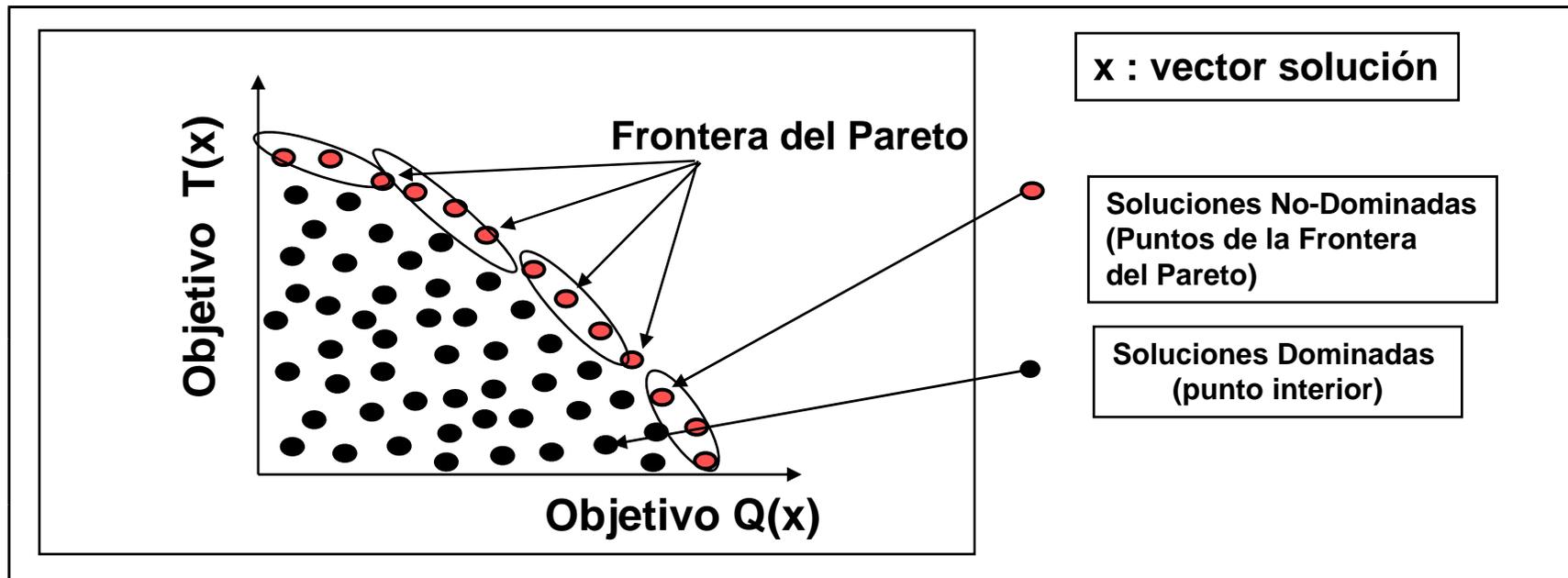
Bibliografía

K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.

C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Pub., 2002.

Problemas Multiobjetivo

- No suele existir una única solución óptima, existe un conjunto (a veces infinito) de soluciones No-Dominadas que forma la Frontera del Pareto
 - Ejemplo:
Identificar la frontera del Pareto para $[Max Q(x), Max T(x)]$



Algoritmos Metaheurísticos Paralelos

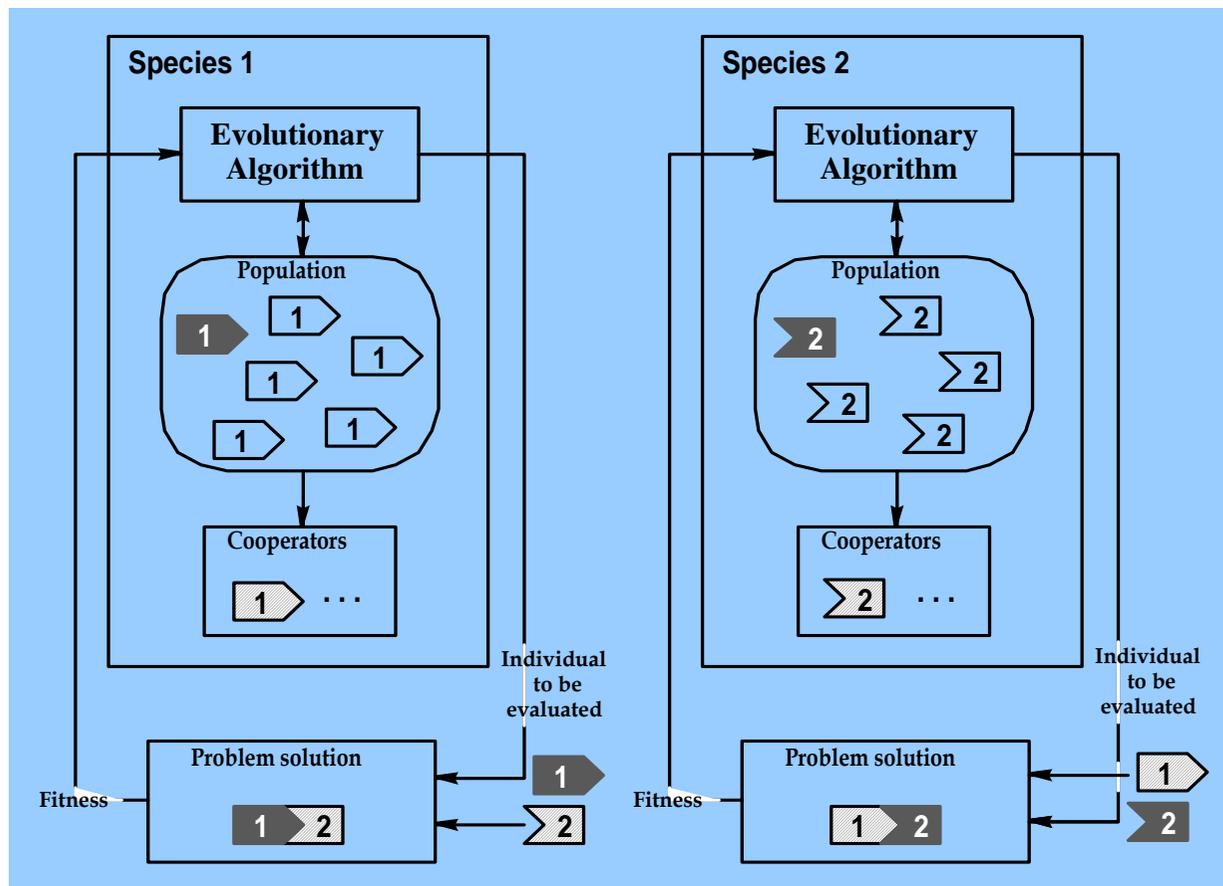
- Preservan la calidad de las soluciones reduciendo el tiempo de ejecución
- Incrementan la calidad de las soluciones sin aumentar el tiempo de cálculo:
 - Aumentando las iteraciones con una paralelización efectiva
 - Ventajas de un diseño paralelo ejecutado secuencialmente, que permita introducir mayor diversidad en el proceso de búsqueda y que evite la convergencia prematura

Bibliografía

E. Alba (Ed.), *Parallel Metaheuristics. A New Class of Algorithms*. Wiley-Interscience, 2005.

Coevolución

Algoritmo de Coevolución Cooperativa



Lecturas

- C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2007.
- E. Alba (Ed.), *Parallel Metaheuristics. A New Class of Algorithms*. Wiley-Interscience, 2005.
- B. Sareni and L. Krähenbühl. Fitness Sharing and Niching Methods Revisited. *IEEE Transactions on Evolutionary Computation*, vol. 2, No. 3, September 1998, pp. 97-106.
- E. Pérez, F. Herrera and C. Hernández. Finding multiple solutions in job shop scheduling by niching genetic algorithms. *Journal of Intelligent Manufacturing* 14, 2003, pp. 323-339.
- K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6:2 (2002) 182-197.
- E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V. Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation* 7:2, April, 2003, pp. 117 - 132.
- M. Potter and K. D. Jong. Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1): 1–29, 2000.

Introducción a los Algoritmos Metaheurísticos



1. Introducción a las Metaheurísticas. Resolución de Problemas Mediante Algoritmos de Búsqueda
2. MH I: Búsqueda Basada Trayectorias
3. MH II: Swarm Intelligence
4. MH III: Algoritmos Evolutivos
5. Limitaciones de los Algoritmos MH: Teorema NFL
6. MH IV: Algoritmos Híbridos: Poblaciones vs. Trayectorias
7. Consideraciones Sobre el Uso de las Metaheurísticas
8. Algunos Estudios/Extensiones de Interés
9. **Comentarios Finales**

Comentarios Finales

En la actualidad, los algoritmos metaheurísticos constituyen una de las áreas de investigación más activas en el ámbito de los problemas de optimización y búsqueda. Su progreso está siendo rápido, surgiendo continuamente nuevas ideas en un intento de alcanzar una mayor eficiencia en el proceso de resolución de problemas.

Desde una perspectiva aplicada, el conjunto de áreas de aplicación es muy amplio, encontrándonos múltiples aplicaciones en los diferentes ámbitos de la ingeniería, hibridaciones con técnicas de Inteligencia Computacional e Inteligencia Artificial, ...

Introducción a los Algoritmos Metaheurísticos



Gracias!!!

¿Preguntas?

IEA-AIE 2010 CALL FOR PAPERS

The 23rd International Conference on Industrial, Engineering & Other
Applications of Applied Intelligent Systems

(<http://www.iea-aie2010.org>)

June 1-4, 2010 Córdoba (Spain)



Extended Deadline: December 10, 2009

<http://www.iea-aie2010.org>