

# Sistemas Inteligentes



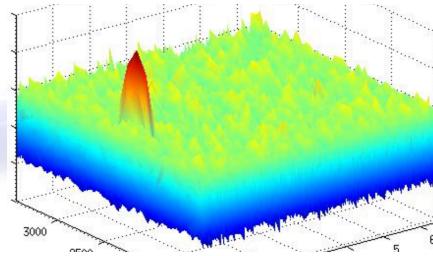
## Tema 2 Metaheurísticas basadas en trayectorias

Dra. Elisa Guerrero Vázquez  
Dpto. Ingeniería Informática  
Universidad de Cádiz

# Algoritmos de búsqueda clásicos

- El coste de encontrar la solución óptima puede ser bastante alto (Backtracking y AC3 por ejemplo)
  - No garantizan encontrar la solución óptima: alta probabilidad de caer en máximos/mínimos locales (Búsquedas Locales)
- ⇒ Alternativa: **Metaheurísticas** que permitan escapar de los óptimos locales

# Metaheurísticas

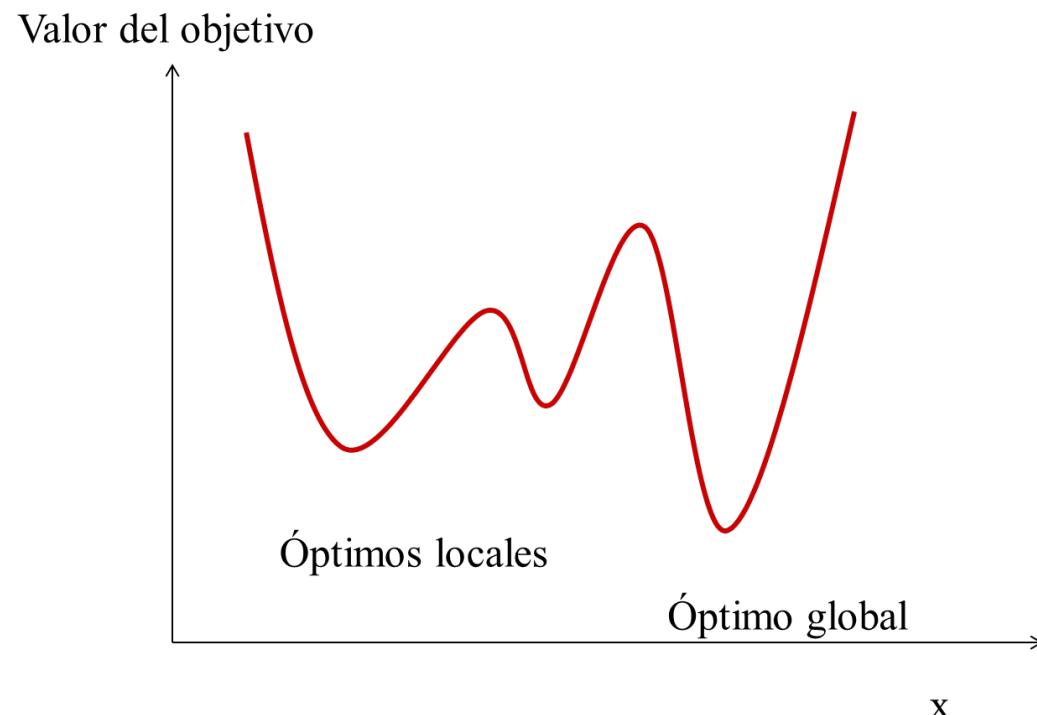


- Son estrategias para **guiar el proceso de búsqueda** en el espacio de estados
- Su objetivo consiste en explorar el espacio de estados con la intención de encontrar soluciones (casi) óptimas
- Metaheurísticas son algoritmos aproximados y normalmente no deterministas (suelen ser probabilísticos)
- Incorporan mecanismos para escapar de los óptimos locales
- No dependen del problema concreto que se esté tratando
- Pero pueden incorporar conocimiento específico del problema

# Métodos basados en Trayectorias

Describen una trayectoria en el espacio de búsqueda a lo largo de la ejecución

- Enfriamiento Simulado (Simulated Annealing)
- Búsqueda Tabú (Tabu Search)



# 1. Simulated Annealing

*Temple simulado (recocido simulado, enfriamiento simulado)*

*en metalurgia se utiliza esta técnica de enfriar de forma controlada un metal para su endurecimiento, cristalización, etc.*



- Se aplica una temperatura muy alta al material y se va gradualmente enfriando
- Se intenta alcanzar el estado de mínima energía
- Se propone una similitud entre una buena estructura cristalina de metales y una buena estructura de soluciones
- Se trata de minimizar una función objetivo que representa la energía del sistema



# Simulated Annealing

En comparación con la búsqueda local:

- Sólo se evaluarán algunos sucesores del estado actual
  - si el nuevo estado es mejor se toma como el estado actual
  - si no es mejor se utiliza una función de probabilidad que decidirá si se acepta o no (en función de la Temperatura)
- La probabilidad de realizar los movimientos anteriores decrece durante la búsqueda:

Al comienzo, la temperatura es alta y se pueden aceptar sucesores peores que el estado actual con mayor probabilidad que después, cuando la temperatura disminuye



# Algoritmo Simulated Annealing (1)

## Algoritmo Simulated Annealing

Actual= estado inicial aleatorio

**mientras** ( $T$  no sea suficientemente baja)

    Nuevo = Sucesor aleatorio de Actual

**si** Nuevo mejora Actual

        Actual=Nuevo

**si\_no**

        Acepta Nuevo con probabilidad  $P$

**fin\_si**

    enfriar( $T$ )

**fin\_mientras**

**Devolver** Actual

# Simulated Annealing: Función de Probabilidad

- La probabilidad de actualizar el estado depende de la medida en que una solución es peor que la otra:
  - $\Delta E$ : diferencia entre las funciones de evaluación del nuevo estado y el estado actual
  - $T$ : temperatura del sistema que va progresivamente disminuyendo

$$p = \exp\left(\frac{-\Delta E}{T}\right)$$

$$\Delta E = feval(Nuevo) - feval(Actual)$$

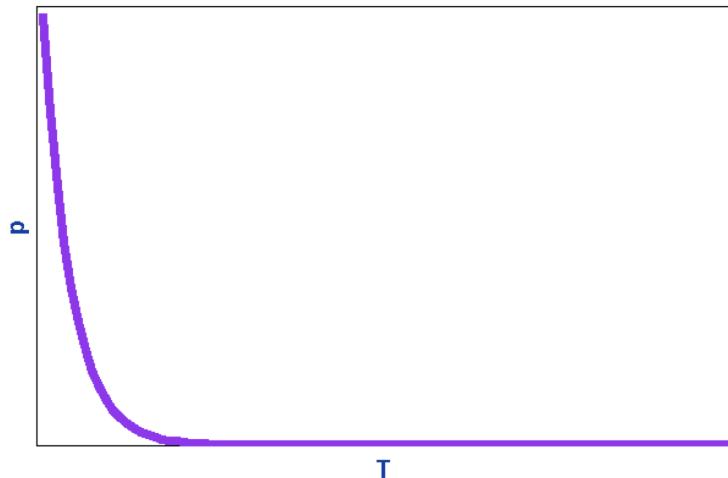
\*\* en problemas donde se intenta minimizar la función de evaluación

# Influencia de T

- Supongamos un problema genérico en el que las funciones de evaluación del estado actual y el nuevo sucesor son las siguientes:  $feval(actual)=15$   $feval(sucesor)=19$

$$p = \exp\left(\frac{-\Delta E}{T}\right)$$

$$\Delta E = feval(Nuevo) - feval(Actual)$$



T	P
1000	
100	
50	
30	
10	
1	

# Influencia de T

- Supongamos un problema genérico en el que las funciones de evaluación del estado actual y el nuevo sucesor son las siguientes:  $feval(actual)=15$   $feval(sucesor)=19$

$$p = \exp\left(\frac{-\Delta E}{T}\right)$$

$$\Delta E = feval(Nuevo) - feval(Actual)$$

Cuando se acerca a cero, el algoritmo solo aceptará cambios a estados con mejor función de evaluación que el estado actual

T	P
1000	0.9960
100	0.9608
50	0.9231
30	0.8752
10	0.6703
1	0.0183

# Algoritmo Simulated Annealing (2)

## Algoritmo Simulated Annealing

**Inicializa T, T\_min**

Actual= estado inicial aleatorio

**mientras (T>T\_min)**

    Nuevo = Sucesor aleatorio de Actual

    deltaE = feval(Nuevo) - feval(Actual)

**si** (deltaE<0)   %% si la solución es mejor se toma directamente

            Actual=Nuevo

**si\_no**

            p=exp(-deltaE/T)

**si** p>rand(0,1)   %% acepta la solución de forma aleatoria

                Actual=Nuevo

**fin\_si**

**fin\_si**

    enfriar( $T$ )

**fin\_mientras**

**devolver** Actual

# Análisis de parámetros

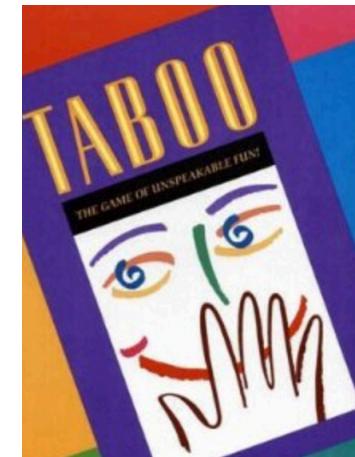
- Un valor muy alto de  $T$  provocaría una convergencia muy lenta
- Un valor muy bajo no permitiría encontrar un estado aceptable
- $T_{\min}$  puede ser sustituida por un número máximo de iteraciones
- La función  $\text{enfriar}(T)$  va disminuyendo el valor de  $T$  en cada iteración.
  - Dependiente del problema
  - Disminución lineal es lo más habitual
  - Otras alternativas

*Descenso exponencial*  $T_{i+1} = \delta * T_i$   $\delta \in [0.8, 0.9]$

$$\text{Cauchy} \quad T_i = \frac{T_o}{1+i}$$

## 2. Búsqueda Tabú

- Los estados *tabú* son estados ya visitados u operaciones realizadas recientemente.
  - Al seleccionar un sucesor de Actual, si es tabú no se seleccionaría
  - Se almacena la permanencia de cada estado en la lista tabú
  - En cada iteración se actualiza para no dejar indefinidamente a ningún estado en la lista tabú



## 2. Búsqueda Tabú

1. Se generan todos los sucesores de Actual y se elige el mejor

### **Lista Tabú**

2.1 Si NO es tabú se selecciona como Actual

Se selecciona como Mejor si supera al Mejor hasta ahora

Vuelve a 1

- Si es tabú se elige el siguiente mejor sucesor de Actual y se vuelve al paso 2.1

### **Criterio de Aspiración**

2.2 Si el elegido Tabú mejora al Mejor estado hasta el momento

- Se selecciona como Mejor
- Se selecciona como Actual y vuelve a 1

## Búsqueda Tabú

- Se ejecuta hasta que una condición externa de finalización se satisface
- El sucesor generado no tiene por qué ser mejor que el estado actual
- Memoria adaptativa: se va actualizando una lista de estados tabú, los estados no deseables
- **Tiempo de persistencia:** número de iteraciones que mantendrán a un estado en la lista tabú

# Condiciones de parada

- Cantidad de iteraciones
- Alcanzar una solución i que sea mejor que un cierto valor fijado al inicio
- No obtener una nueva mejor solución después de un cierto número de iteraciones
- Todos los sucesores del paso actual están incluidos en la lista tabú y no mejoran al mejor

## Criterio de Aspiración

- Puede ocurrir que algún estado sucesor, aún siendo tabú, pueda ofrecer una solución excepcionalmente buena.
- El **criterio de aspiración** permite comprobar si un estado tabú es mejor que el estado mejor, y si es así lo selecciona, actualizando las estructuras de datos correspondientes.

# Algoritmo para la Búsqueda Tabú

## Algorithm Tabu\_Search

Current= random or initial state

Best=Current

**Initial values for tabu list, tenure, iteration counter and max number of iterations**

**While** <stop conditions>

Obtain the list of successor states ordered by cost function

**While** <successor list not empty and not a new current state is generated >

**New=Next(Successor list)**

**if** Cost(New) < Cost(Best) %% Tabu or not tabu but improves the best

        Current=New

        Best=Current

        TabuList + Current

        Sucessor\_list={}

**elsif** New is Not Tabu

        Current=New

        TabuList + Current

        Sucessor\_list={}

**else**

        <New is tabu and worse than Best, take the following successor>

**endif**

**endWhile**

    Update tabu list (decreasing tenure)

    Update rest of variables

**endWhile**

**end**

# Conclusiones

Ambas estrategias:

- Utilizan Metaheurísticas para guiar la búsqueda y evitar caer en un óptimo local
- Están basadas en un individuo (vs. poblaciones)
- Son estrategias basadas en trayectoria (vs. constructiva):
  - Métodos iterativos
  - Hay que definir la vecindad utilizada

# Conclusiones

## ■ Simulated Annealing

- Selecciona un nuevo sucesor de forma aleatoria
- Si el movimiento mejora la situación es siempre aceptado
- Cuando el Nuevo estado es peor lo acepta con una cierta probabilidad  $p$
- $T$  comienza con valores altos e iterativamente se va disminuyendo
- Es un proceso estocástico

## ■ Tabu Search

- Utiliza una memoria que se actualiza dinámicamente y que contiene estados que hay que evitar durante un cierto número de iteraciones
- Es un proceso determinista

## Bibliografía

- Russell, S. y Norvig, P. *Inteligencia Artificial (un enfoque moderno)* (Pearson Educación, 2004). Segunda edición. Cap. 5: “Problemas de Satisfacción de Restricciones”
- García Serrano, A. *Inteligencia artificial. Fundamentos, práctica y aplicaciones* (RC Libros, 2012)
- Michalewicz, Z. y Fogel D. *How to solve it. Modern Heuristics* (Springer, 2004)
- Tim Jones, J. *Artificial Intelligence. A systems approach.* Computer Sciences Series. (Jones & Bartlett publishers, 2008)