

Metaheurísticas

CONFERENCIA #3. RECOCIDO SIMULADO

Sumario

- ▶ Algoritmos de búsqueda local. Vecindades
- ▶ Elementos básicos del Recocido Simulado
- ▶ Esquemas de enfriado
- ▶ Aplicaciones

Algoritmos de búsqueda local

- ▶ Los algoritmos de búsqueda como primero en amplitud, primero en profundidad o A^* exploran todo el espacio de búsqueda de forma sistemática manteniendo una o más rutas en la memoria y registrando qué alternativas se han explorado.
- ▶ Cuando se encuentra una meta, el camino hacia esa meta constituye una solución.
- ▶ Los algoritmos de búsqueda local pueden ser muy útiles si estamos interesados en el estado de la solución, pero no en el camino hacia ese objetivo. Operan solo en el estado actual y se mueven a estados vecinos.

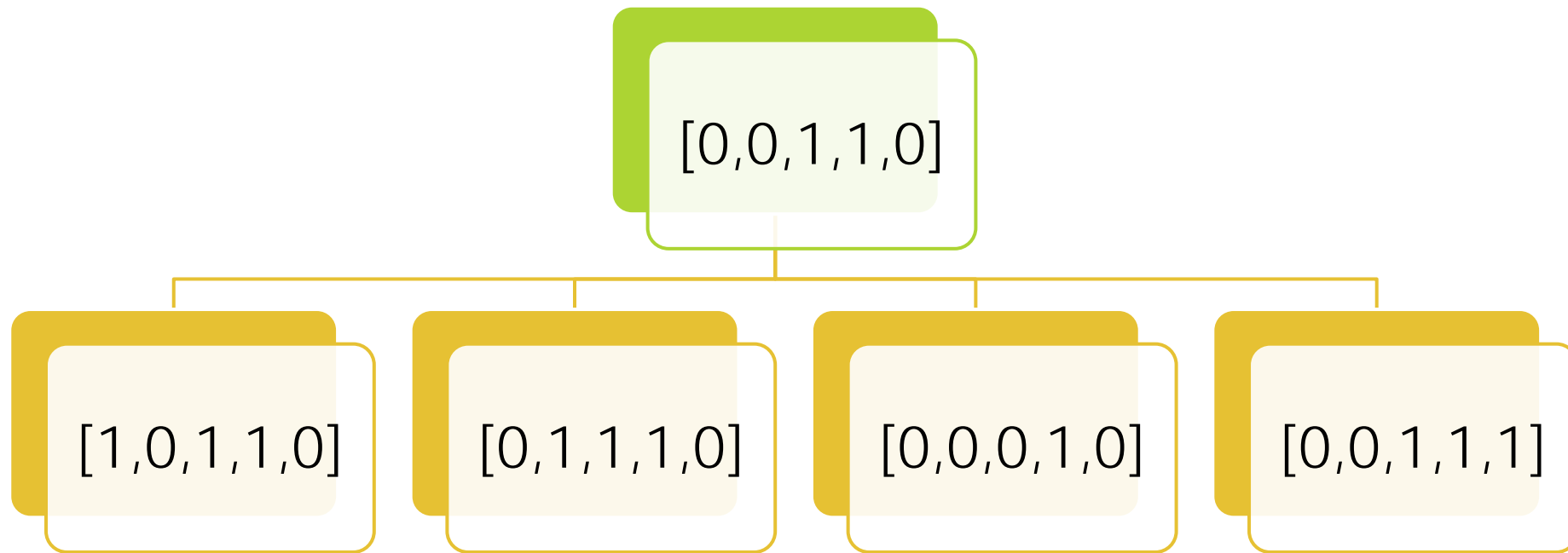
Algoritmos de búsqueda local

- ▶ Los algoritmos de búsqueda local tienen dos ventajas fundamentales:
 - ▶ Usan muy poca memoria
 - ▶ Pueden encontrar soluciones razonables en espacios de estados grandes o infinitos (continuos).
- ▶ Algunos ejemplos de algoritmos de búsqueda local son:
 - ▶ Ascenso de Colinas con Mutación Aleatoria
 - ▶ Caminatas Aleatorias
 - ▶ Recocido Simulado

Vecindades

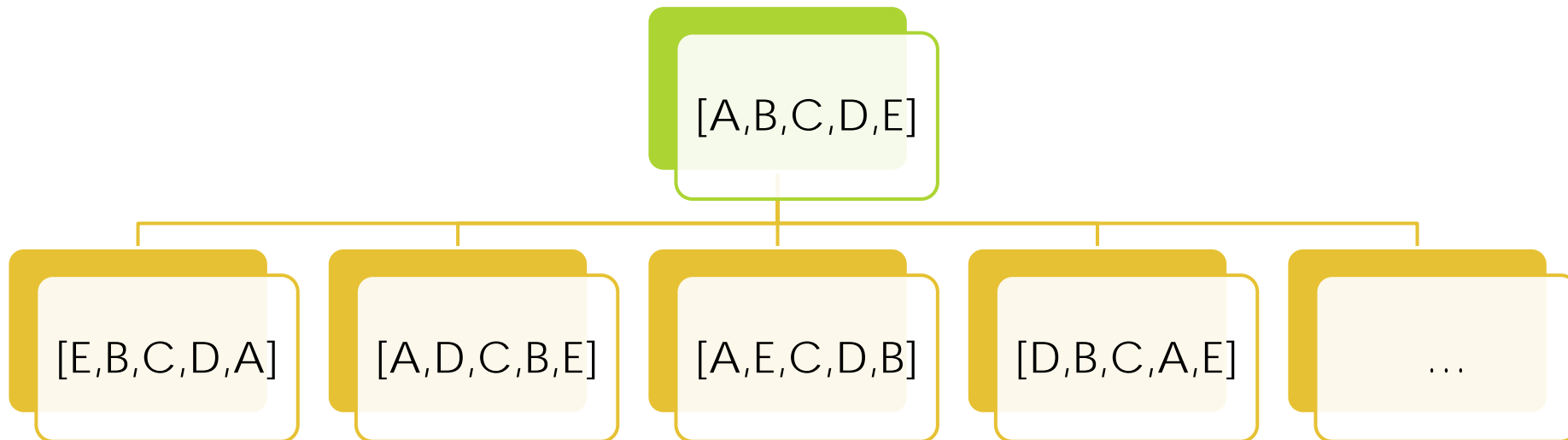
- ▶ Una instancia de un problema de optimización combinatoria consta de un conjunto S de soluciones factibles y una función de costo no negativa f . El problema es encontrar una solución globalmente óptima $i^* \in S$, es decir, una solución con un costo óptimo f^*
- ▶ Una función de vecindad es un mapeo $N:S \rightarrow 2^S$, que define para cada solución $i \in S$ un conjunto $N(i) \in S$ de soluciones que son, de alguna manera, cercanas a i
- ▶ El conjunto $N(i) \in S$ es denominado vecindad de la solución i , y cada elemento $j \in N(i)$ se denomina vecino de i .

Vecindades. Problema de la mochila

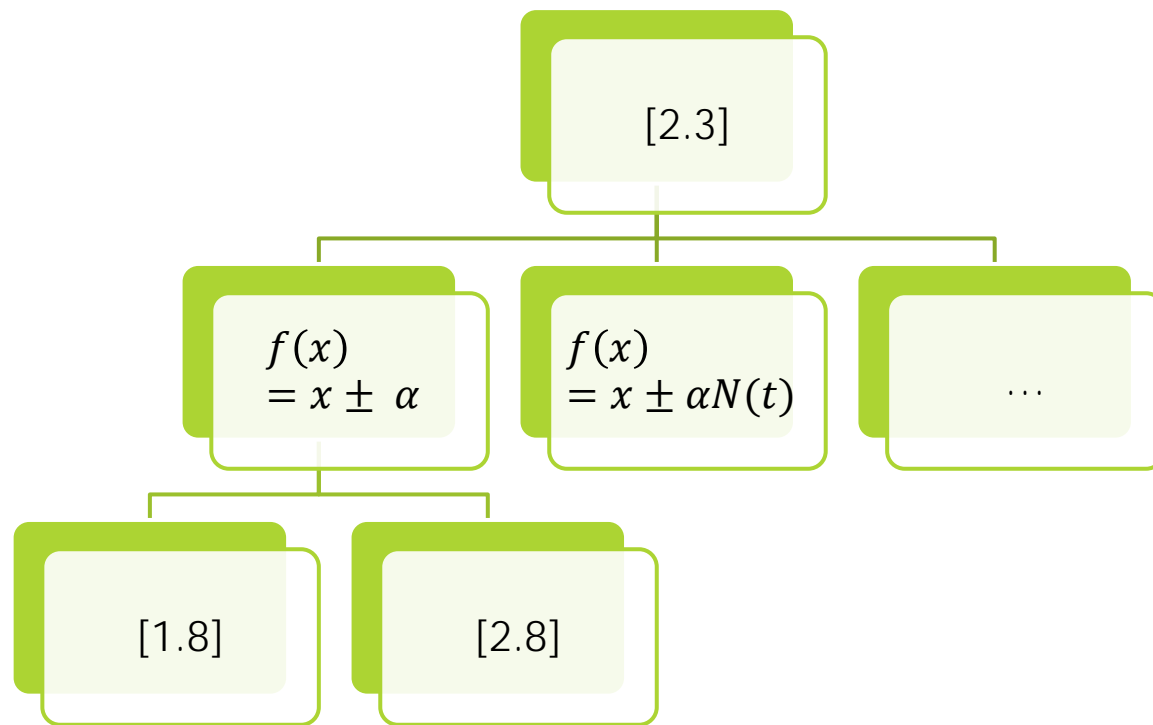


Vecindades. Problema del viajero vendedor

► Vecindad swap



Vecindades. Espacios continuos



Bases del Recocido Simulado

- ▶ El recocido es un proceso térmico para obtener estados de baja energía de un sólido en un baño de calor.
- ▶ El proceso contiene dos pasos:
 - ▶ Aumentar la temperatura del baño de calor a un valor máximo en el que el sólido se derrite.
 - ▶ Disminuir con cuidado la temperatura del baño de calor hasta que las partículas se dispongan en el estado fundamental del sólido.
- ▶ El estado fundamental es un estado de energía mínima del sólido.
- ▶ El estado fundamental del sólido se obtiene solo si la temperatura máxima es lo suficientemente alta y el enfriamiento se realiza lentamente

Recocido Simulado

- ▶ El proceso de recocido se puede simular con el algoritmo Metropolis, que se basa en técnicas de Monte Carlo.
- ▶ Podemos aplicar este algoritmo para generar una solución a los problemas de optimización combinatoria asumiendo una analogía entre ellos y los sistemas físicos de muchas partículas con las siguientes equivalencias:
 - ▶ Las soluciones del problema son equivalentes a los estados de un sistema físico.
 - ▶ El costo de una solución es equivalente a la "energía" de un estado

Kirkpatrick, S., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1), 41-51.

Recocido Simulado

- ▶ Para aplicar el recocido simulado con fines de optimización, necesitamos lo siguiente:
 - ▶ Una función sucesora que devuelve una solución vecina "cercana" dada la actual. Esto funcionará como la "perturbación" de las partículas del sistema.
 - ▶ Una función de destino para optimizar que depende del estado actual del sistema. Esta función funcionará como la energía del sistema.
- ▶ La búsqueda se inicia con un estado aleatorio. En un ciclo de sondeo, nos moveremos a los estados vecinos siempre aceptando los movimientos que disminuyen la energía mientras que solo aceptamos los movimientos malos de acuerdo con una distribución de probabilidad dependiente de la "temperatura" del sistema.

Recocido Simulado

- ▶ Disminuir la temperatura lentamente, aceptando movimientos menos malos en cada nivel de temperatura hasta que a temperaturas muy bajas el algoritmo se convierta en un codicioso algoritmo de escalada de colinas.

- ▶ La distribución utilizada para decidir si aceptamos un mal movimiento se conoce como distribución de Boltzman

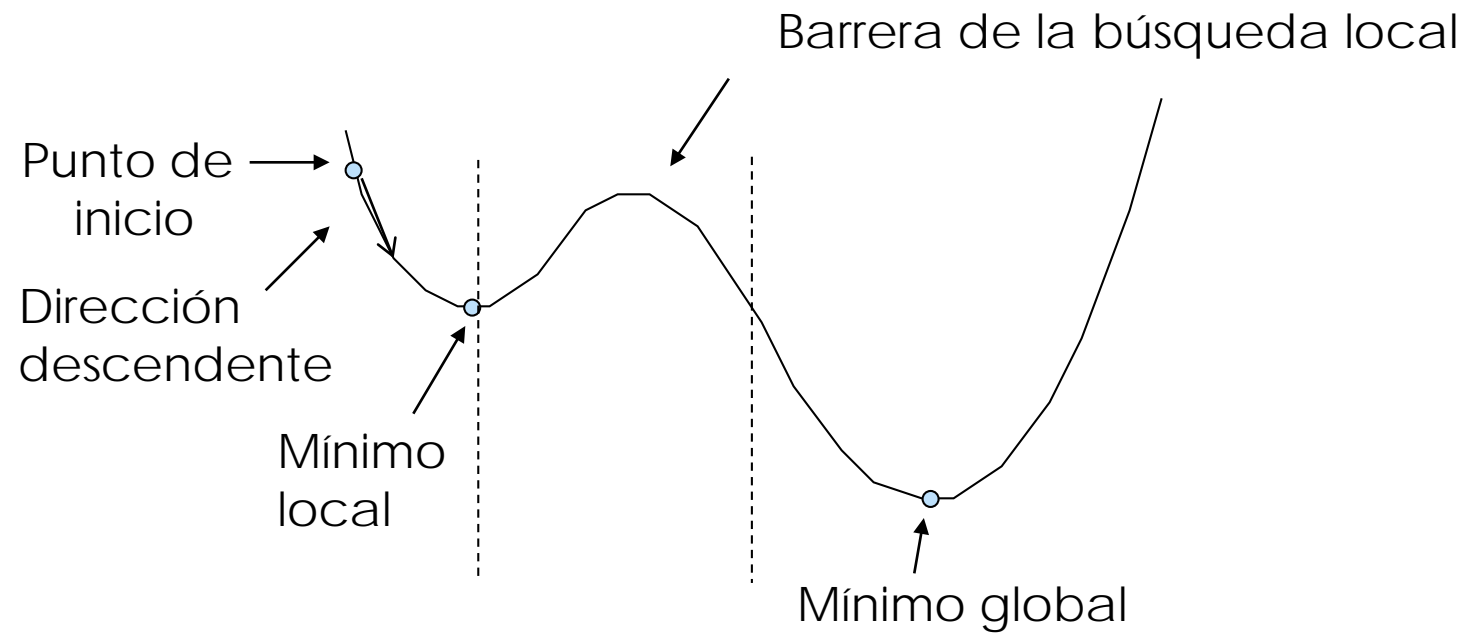
$$P(\gamma) = \frac{e^{\frac{-E_\gamma}{k_B T}}}{\sum_{\gamma'} e^{\frac{-E_{\gamma'}}{k_B T}}}$$

- ▶ Esta distribución es muy conocida en física sólida y juega un papel central en el recocido simulado. Cuando γ es el estado actual, E_γ es la energía relacionada con dicho estado, y γ' son los estados sucesores

Pseudocódigo

```
1. Crear una solución inicial aleatoria  $\gamma$ 
2.  $E_{old} = \text{cost}(\gamma)$ ;
3. for( $\text{temp} = \text{tempmax}$ ;  $\text{temp} \geq \text{tempmin}$ ;  $\text{temp} = \text{next\_temp}(\text{temp})$ ) {
4.     for( $i = 0$ ;  $i < \text{imax}$ ;  $i++$ ) {
5.          $\text{sucesor\_func}(\gamma)$ ; //la función de vecindad
6.          $E_{new} = \text{cost}(\gamma)$ ;
7.          $\text{delta} = E_{new} - E_{old}$ ;
8.         if( $\text{delta} > 0$ )
9.             if( $\text{random}() \geq \exp(-\text{delta}/K * \text{temp})$ );
10.             $\text{undo\_func}(\gamma)$ ; //rechazar un mal movimiento
11.            else  $E_{old} = E_{new}$  //aceptar un mal movimiento
13.        else  $E_{old} = E_{new}$ ; //siempre aceptar buenos movimientos}}
```

Dificultad para buscar un óptimo global

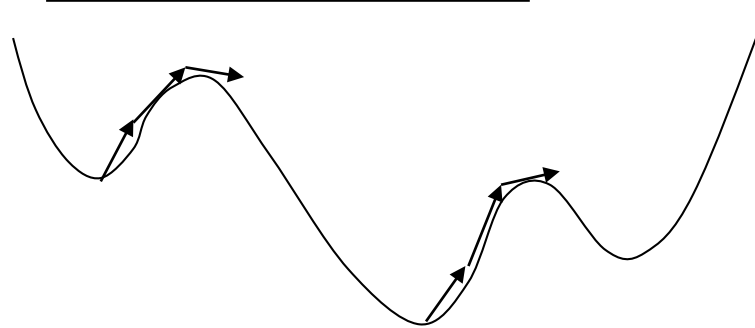


Intuición en el Recocido Simulado

- Al permitir un ascenso ocasional en el proceso de búsqueda, podríamos escapar de la trampa de los mínimos locales.

Efecto deseado

Ayuda a escapar de óptimos locales.



Efecto adverso

Podría pasar óptimos globales después de alcanzarlo

Esquema de recocido

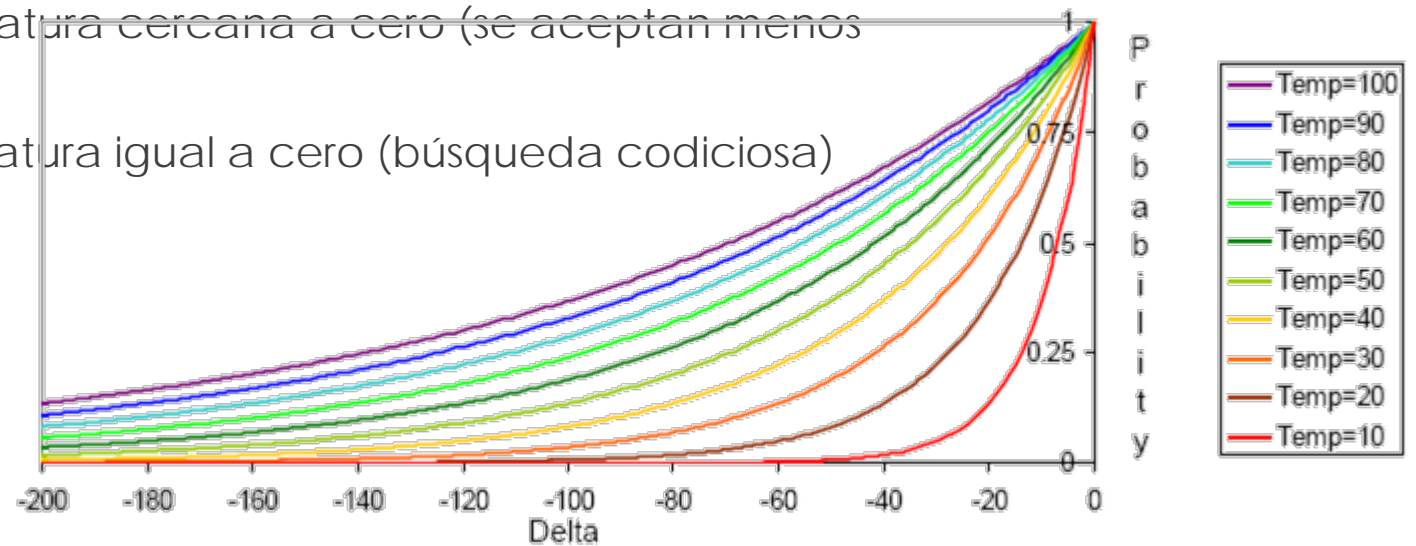
- ▶ T , la temperatura de recocido, es el parámetro que controla la frecuencia de aceptación de los pasos ascendentes.
- ▶ Reducimos gradualmente la temperatura $T(k)$ (temp en el código)
A cada temperatura, se permite que la búsqueda continúe durante un cierto número de pasos, $L(k)$ (imax en código)

```
for(temp=tempmax; temp>=tempmin; temp=next_temp(temp) ) {  
    for(i=0; i<imax; i++ )
```

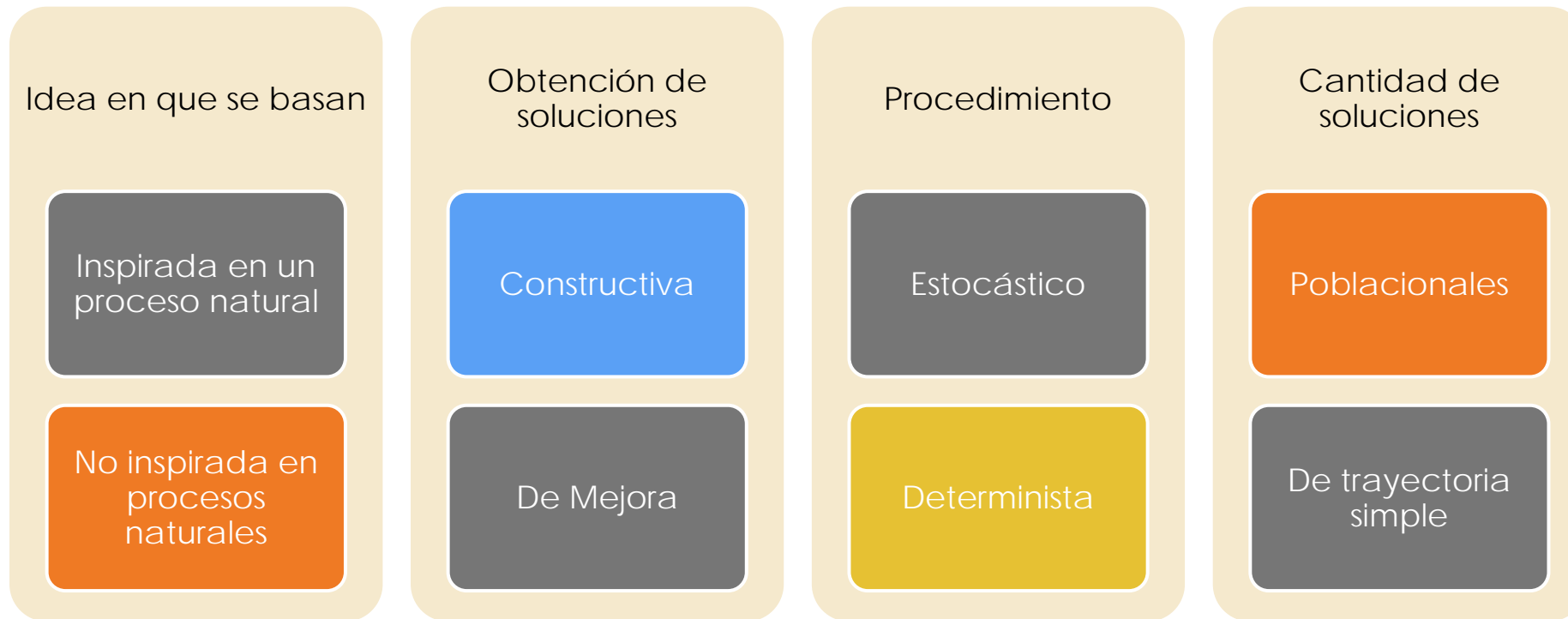
- ▶ La elección de los parámetros $\{T(k), L(k)\}$ se denomina programa de enfriamiento o esquema de recocido.

Esquema de recocido

- If ($\Delta > 0$) **Aceptar**
- Else if($\text{random}() \geq \exp(-\Delta / K * \text{temp})$) **Aceptar**, else **Rechazar**
 - Inicialmente la temperatura es muy alta (se aceptan la mayoría de los malos movimientos)
 - Iteraciones medias con temperatura cercana a cero (se aceptan menos movimientos malos)
 - Ejecuciones finales con temperatura igual a cero (búsqueda codiciosa)

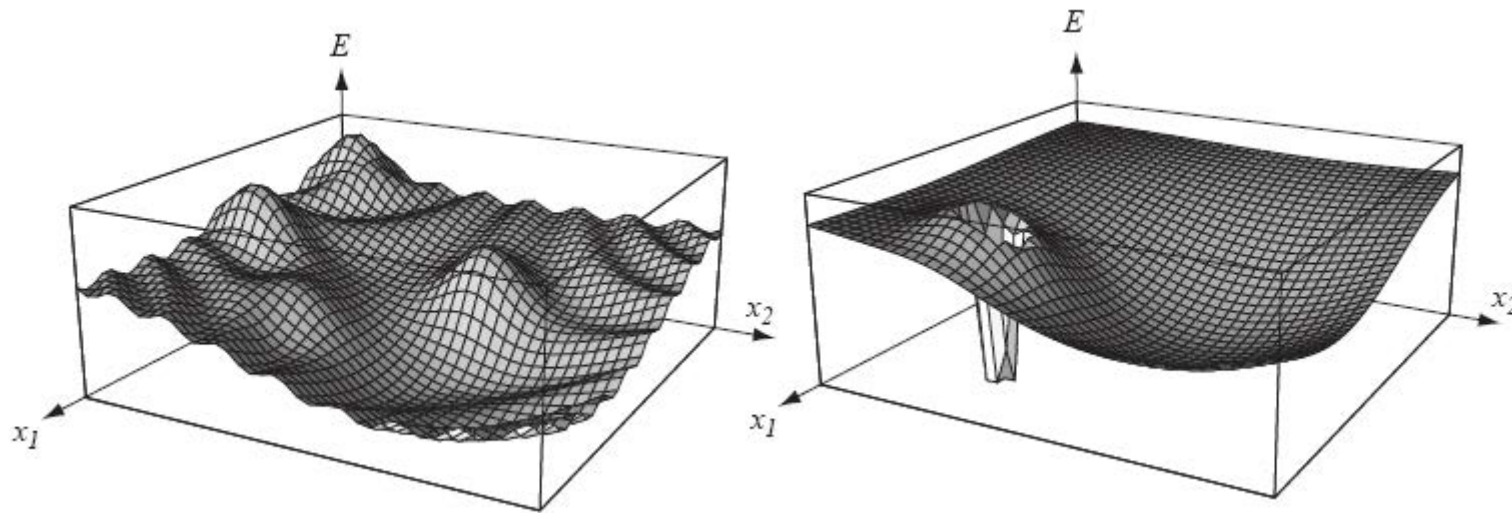


Clasificación



Problemas prácticos con el recocido simulado

- ▶ La función de costo debe desarrollarse cuidadosamente, debe ser "fractal y suave"
- ▶ La función energética de la izquierda funcionaría con SA mientras que la de la derecha fallaría



Problemas prácticos con el recocido simulado

- ▶ La función de costo debe ser rápida, se llamará "millones" de veces.
- ▶ Lo mejor es si solo tenemos que calcular los deltas producidos por la modificación en lugar de atravesar todo el estado.
- ▶ Esto depende de la aplicación.

Problemas prácticos con el recocido simulado

- ▶ En la convergencia asintótica, el recocido simulado converge a soluciones óptimas a nivel mundial.
- ▶ En la práctica, la convergencia del algoritmo depende del programa de enfriamiento.
- ▶ Hay algunas sugerencias sobre el programa de enfriamiento, pero aún requiere muchas pruebas y generalmente depende de la aplicación.

Problemas prácticos con el recocido simulado

- ▶ Comience a una temperatura en la que se acepte el 50% de los malos movimientos.
- ▶ Cada paso de enfriamiento reduce la temperatura en un 10%
El número de iteraciones en cada temperatura debe intentar moverse entre 1 y 10 veces cada "elemento" del estado.
- ▶ La temperatura final no debe aceptar malos movimientos; este paso se conoce como el paso de extinción

Esquema de recocido estático

- ▶ Temp_max
 - ▶ Es la diferencia máxima de costo entre dos soluciones vecinas
En la práctica, algún valor grande
- ▶ Next_temp
 - ▶ $T = \alpha T, 0.8 \leq \alpha \leq 0.99$
- ▶ Temp_min
 - ▶ Generalmente cero, o un número pequeño (Ex. 0.001)
- ▶ Imax
 - ▶ Depende del problema

Aplicaciones



Problemas Básico

- Problema del viajero vendedor
- Particionado de grafos
- Problemas de macheo
- Coloreado de grafos
- Problemas de planeación
- ...

Estudio Independiente

- ▶ Burke & Kendall. Search Metodologies – 2005
 - ▶ Capítulo 7
- ▶ Guerra, M. Simulated Annealing - Advances, Applications and Hybridizations - 2012
 - ▶ Capítulos 1 y 9
- ▶ Guía de la Clase Práctica #2
- ▶ Modelado de los problemas bajo estudio
 - ▶ Problema de la mochila
 - ▶ Problema del viajero vendedor
 - ▶ Mínimos de funciones



¿Preguntas?