ALGORÍTMICA
2006 - 2007

ALGORÍTMICA

TEMA 3. Algoritmos de Enfriamiento Simulado

1. **FUNDAMENTOS**
2. **ALGORITMO BÁSICO DE ES**
3. **PARÁMETROS Y COMPONENTES**
4. **APLICACIONES**

- *A. Díaz y otros. Optimización Heurística y Redes Neuronales. Paraninfo, 1996*
- *K.A. Dowsland, A. Díaz. Diseño de heurísticas y fundamentos del recocido simulado. Inteligencia Artificial VII:2 (2003) 93-101.*

1. FUNDAMENTOS

1.1. Introducción

El Enfriamiento o Recocido Simulado es un algoritmo de búsqueda por entornos con un criterio probabilístico de aceptación de soluciones basado en Termodinámica

1. FUNDAMENTOS

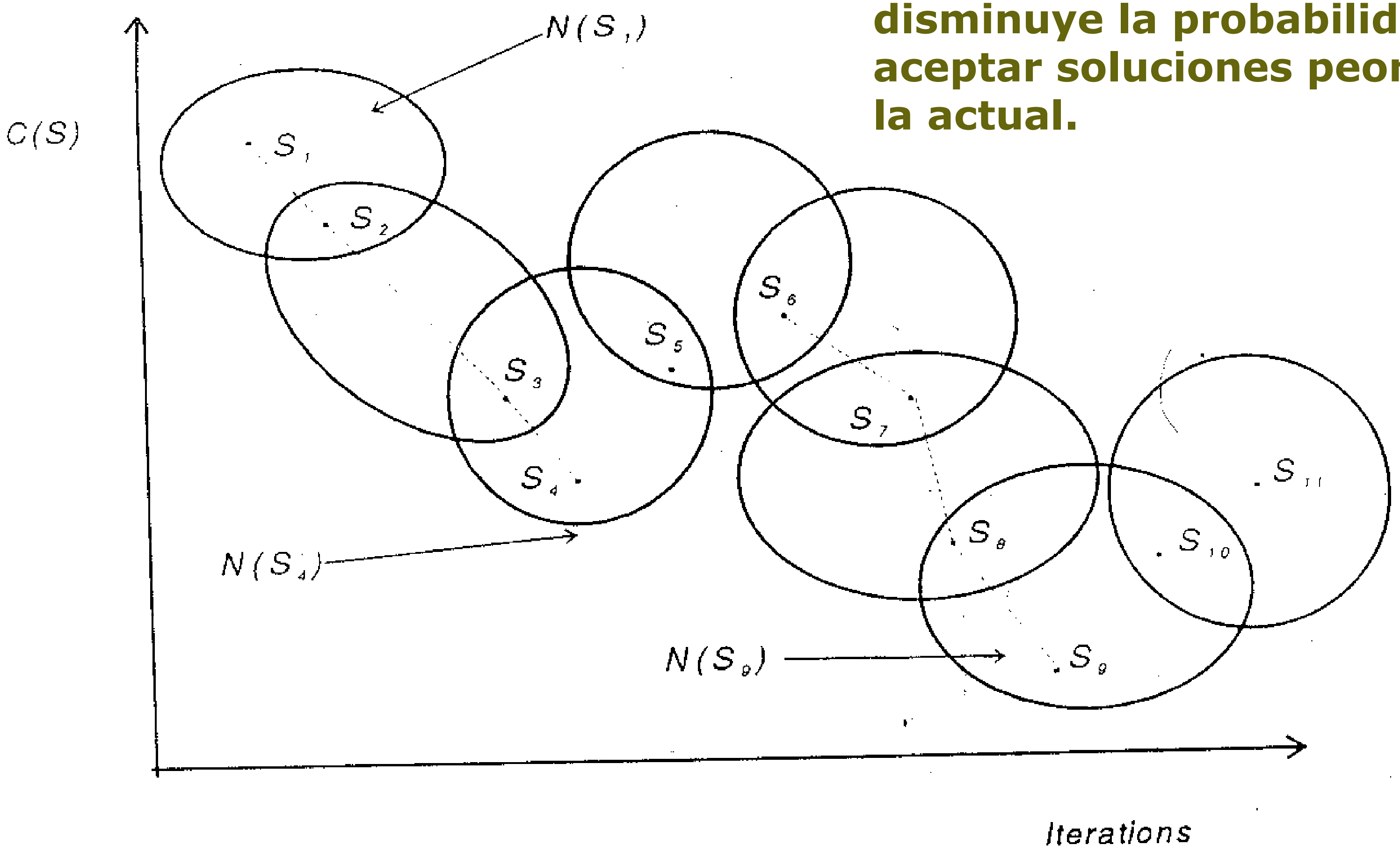
- **Un modo de evitar que la búsqueda local finalice en óptimos locales, hecho que suele ocurrir con los algoritmos tradicionales de búsqueda local, es permitir que algunos movimientos sean hacia soluciones peores**

- **Pero si la búsqueda está avanzando realmente hacia una buena solución, estos movimientos "de escape de óptimos locales" deben realizarse de un modo controlado**

- **En el caso del Enfriamiento Simulado (ES), esto se realiza controlando la frecuencia de los movimientos de escape mediante una función de probabilidad que hará disminuir la probabilidad de estos movimientos hacia soluciones peores conforme avanza la búsqueda (y por tanto estamos más cerca, previsiblemente, del óptimo local)**

- **Así, se aplica la filosofía habitual de búsqueda de diversificar al principio e intensificar al final**

1. FUNDAMENTOS



Con el aumento de iteraciones disminuye la probabilidad de aceptar soluciones peores que la actual.

1. FUNDAMENTOS

1.2. Modelo de Metrópolis

- **El fundamento de este control se basa en el trabajo de Metrópolis (1953) en el campo de la termodinámica estadística**

- **Básicamente, Metrópolis modeló el proceso de enfriamiento simulando los cambios energéticos en un sistema de partículas conforme decrece la temperatura, hasta que converge a un estado estable (congelado). Las leyes de la termodinámica dicen que a una temperatura t la probabilidad de un incremento energético de magnitud δE se puede aproximar por**

$$P[\delta E] = \exp(- \delta E/kt)$$

siendo k una constante física denominada Boltzmann

- **En el modelo de Metrópolis, se genera una perturbación aleatoria en el sistema y se calculan los cambios de energía resultantes: si hay una caída energética, el cambio se acepta automáticamente; por el contrario, si se produce un incremento energético, el cambio será aceptado con una probabilidad indicada por la anterior expresión**

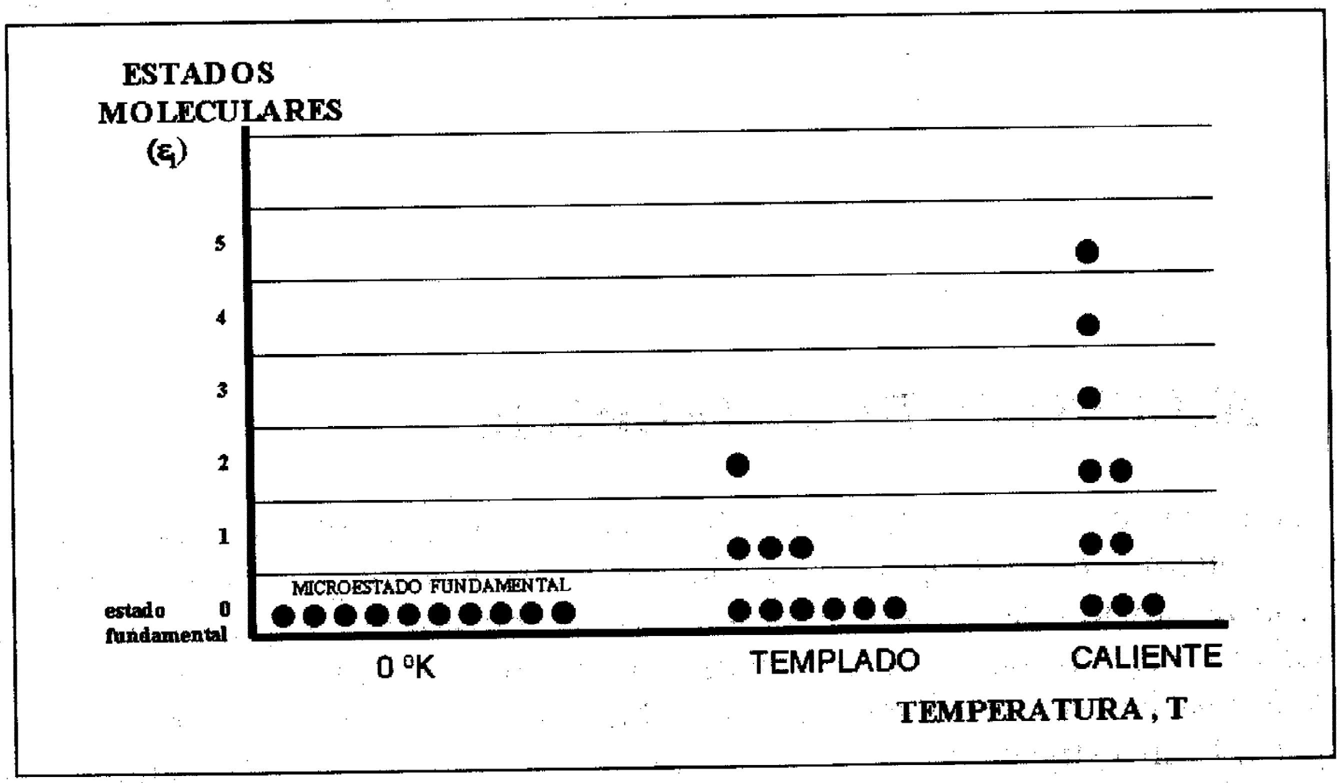


Figura 2.1. *Ejemplo de diferentes microestados según la temperatura, para N=10. En el microestado correspondiente a TEMPLADO, es $n_0=6$, $n_1=3$, $n_2=1$, $E=\Sigma n_i \cdot i=5$. El número de moléculas en los estados superiores decrece exponencialmente para una T fija.*

1. FUNDAMENTOS

1.3. Analogías: Termodinámica - Optimización

Simulación Termodinámica	Optimización Combinatoria
Estados del sistema	Soluciones factibles
Energía	Coste
Cambio de estado	Solución en el entorno
Temperatura	Parámetro de control
Estado congelado	Solución heurística

2. ALGORITMO BÁSICO DE ES

- **El algoritmo de Enfriamiento Simulado es un método de búsqueda por entornos caracterizado por un criterio de aceptación de soluciones vecinas que se adapta a lo largo de su ejecución**

- **Hace uso de una variable llamada Temperatura, T, cuyo valor determina en qué medida pueden ser aceptadas soluciones vecinas peores que la actual**

- **La variable Temperatura se inicializa a un valor alto, denominado Temperatura inicial, T_0, y se va reduciendo cada iteración mediante un mecanismo de enfriamiento de la temperatura, $\alpha(\cdot)$, hasta alcanzar una Temperatura final, T_f**

2. ALGORITMO BÁSICO DE ES

- **En cada iteración se genera un <u>número concreto de vecinos</u>, L(T), que puede ser fijo para toda la ejecución o depender de la iteración concreta**

- **<u>Cada vez que se genera un vecino, se aplica el criterio de aceptación para ver si sustituye a la solución actual</u>**

- **Si la solución vecina es mejor que la actual, se acepta automáticamente, tal como se haría en la búsqueda local clásica**

- **En cambio, <u>si es peor, aún existe la probabilidad de que el vecino sustituya a la solución actual</u>. Esto permite al algoritmo salir de óptimos locales, en los que la BL clásica quedaría atrapada**

2. ALGORITMO BÁSICO DE ES

- **Esta probabilidad depende de la diferencia de costes entre la solución actual y la vecina, δ, y de la temperatura T:**

$$P_{aceptación} = \exp(-\delta/T)$$

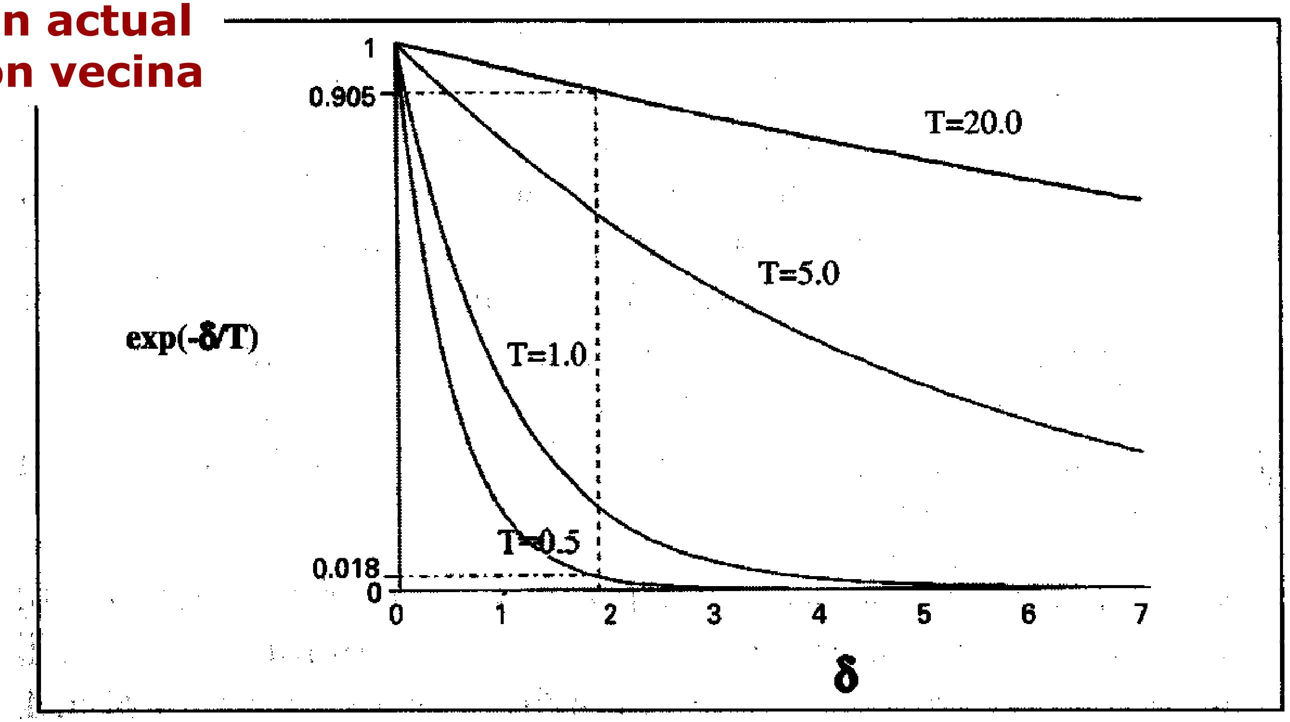
- **A mayor temperatura, mayor probabilidad de aceptación de soluciones peores. Así, el algoritmo acepta soluciones mucho peores que la actual al principio de la ejecución (exploración) pero no al final (explotación)**

- **A menor diferencia de costes, mayor probabilidad de aceptación de soluciones peores**

- **Una vez finalizada la iteración, es decir, tras generar L(T) soluciones vecinas, se enfría la temperatura y se pasa a la siguiente iteración**

2. ALGORITMO BÁSICO DE ES

$\delta = C(s') - C(s)$
s = solución actual
s' = solución vecina



2. Valor del factor de Boltzmann en función de la temperatura T y de δ. Para δ=2, es 50 veces menos probable un movimiento si T=0,5 que si T=20.

2. ALGORITMO BÁSICO DE ES

```
INPUT (T_o,α,L,T_f)
T←T_o
S_act←Genera_solucion_inicial
WHILE T≥T_f DO
      BEGIN
      FOR cont←1 TO L(T) DO
            BEGIN
            S_cand←Selecciona_solucion_N(S_act)
            δ←coste(S_cand)–coste(S_act)
            IF (U(0,1)<e^(-δ/T)) OR
               (δ<0)   THEN S_act←S_cand
            END
      T←α(T)
      END
{Escribe como solución, la mejor de las S_act visitadas}
```

3. PARÁMETROS Y COMPONENTES DE ES

3.1. Representación

3.2. Generación de la Solución Inicial

3.3. Mecanismo de Transición de Soluciones

3.4. Secuencia de Enfriamiento

3.1. Representación

ALGUNOS EJEMPLOS

Vector ordenado de números enteros.
Ejemplo: Problema del Viajante de Comercio
(7 6 3 1 5 4 2 8)

Vector binario. Ejemplo: Problema de la Mochila
(0 1 0 0 1 1 1 0)

Vector de Números Reales.
Ejemplo: Problemas de Optimización con Parámetros Continuos
(2.7 3.5 4 6.27)

....

3.2. Generación de la Solución Inicial

- **Uso de técnicas eficientes para obtenerla**
- **Uso de conocimiento experto**

  Ejemplo: solución generada por un algoritmo *greedy*

3. PARÁMETROS Y COMPONENTES DE ES

3.3. Mecanismo de Transición de Soluciones

1. Generación de una nueva solución

- **Definición del conjunto de vecinos**
- **Selección de un elemento de dicho conjunto**

2. Cálculo de la diferencia de costos entre la solución actual y la vecina

3. Aplicación del criterio de aceptación

3. PARÁMETROS Y COMPONENTES DE ES

3.3. Mecanismo de Transición de Soluciones

INPUT (T_o, α, L, T_f)
$T \leftarrow T_o$
$S_{act} \leftarrow$ Genera_solucion_inicial
WHILE $T \geq T_f$ DO
    BEGIN
    FOR cont $\leftarrow$ 1 TO $L(T)$ DO
        BEGIN
        $S_{cand} \leftarrow$ Selecciona_solucion_N(S_{act})
        $\delta \leftarrow \text{coste}(S_{cand}) - \text{coste}(S_{act})$
        IF $(U(0,1) < e^{(-\delta/T)})$ OR
            $(\delta < 0)$ THEN $S_{act} \leftarrow S_{cand}$
        END
    $T \leftarrow \alpha(T)$
    END
{Escribe como solución, la mejor de las S_{act} visitadas}

Generación de una nueva solución

Cálculo de la diferencia de costos

Aplicación del criterio de aceptación

3. PARÁMETROS Y COMPONENTES DE ES

3.4. Secuencia de Enfriamiento

INPUT (T_o, α, L, T_f)

$T \leftarrow T_o$ **Valor inicial del parámetro de control**

$S_{act} \leftarrow$ Genera_solucion_inicial

WHILE $T \geq T_f$ DO **Condición de Parada**

  BEGIN

  FOR cont $\leftarrow$ 1 TO $L(T)$ DO **Velocidad de enfriamiento L(T)**

    BEGIN

    $S_{cand} \leftarrow$ Selecciona_solucion_N(S_{act})

    $\delta \leftarrow$ coste(S_{cand}) – coste(S_{act})

    IF $(U(0,1) < e^{(-\delta/T)})$ OR

      $(\delta < 0)$ THEN $S_{act} \leftarrow S_{cand}$

    END

  $T \leftarrow \alpha(T)$ **Mecanismo de enfriamiento**

  END

{Escribe como solución, la mejor de las S_{act} visitadas}

3. PARÁMETROS Y COMPONENTES DE ES

3.4. Secuencia de Enfriamiento

1. **Valor Inicial del Parámetro de Control**

No parece conveniente considerar valores fijos independientes del problema.

PROPUESTA: $T_0 = (\mu / -\ln(\phi)) \cdot C(S_0)$

Tanto por uno ϕ de probabilidad de que una solución sea un μ por uno peor que la solución inicial S_0

3. PARÁMETROS Y COMPONENTES DE ES

3.4. Secuencia de Enfriamiento

2. Mecanismo de Enfriamiento

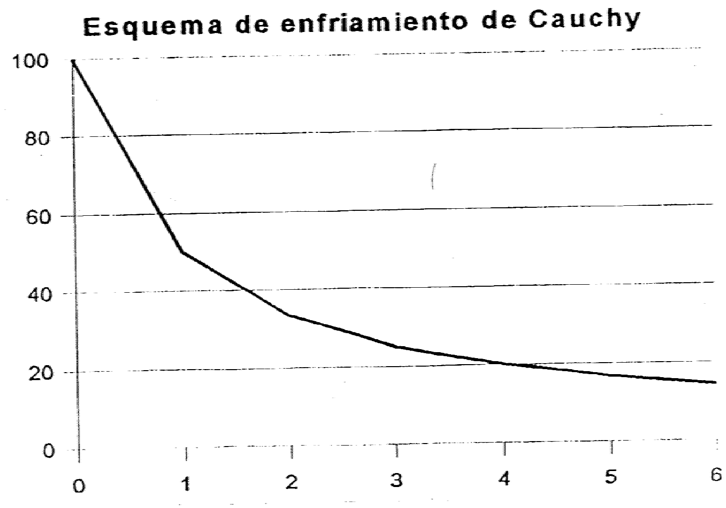
Existen varios mecanismos de enfriamiento:

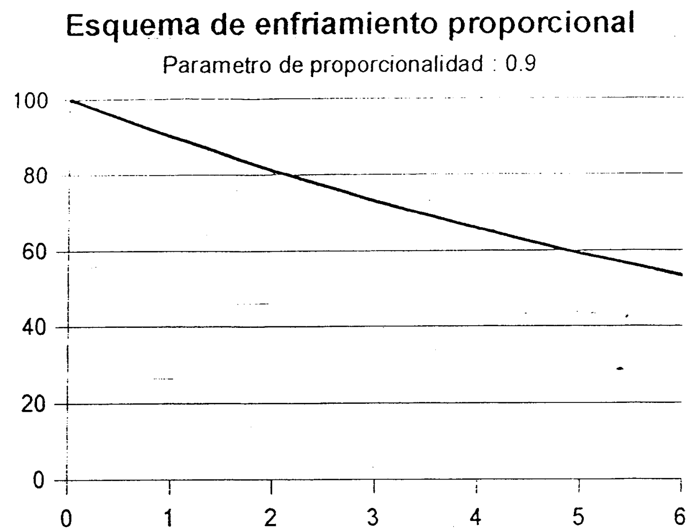
- **Enfriamiento basado en sucesivas temperaturas descendentes fijadas por el usuario**
- **Enfriamiento con descenso constante de temperatura**
- **Descenso exponencial: $T_{k+1} = \alpha \cdot T_k$**
  k = nº iteración actual, α constante cercana a 1 (usualmente, $\alpha \in [0.8, 0.99]$)
- **Criterio de Boltzmann:** $T_k = T_0 / (1 + \log(k))$
- **Esquema de Cauchy:** $T_k = T_0 / (1 + k)$
- **Para controlar el número de iteraciones (Cauchy modificado): $T_{k+1} = T_k / (1 + \beta \cdot T_k)$**
  Para ejecutar exactamente M iteraciones $\Rightarrow$ $\beta = (T_o - T_f) / (M \cdot T_o \cdot T_f)$

3. PARÁMETROS Y COMPONENTES DE ES

2. Mecanismo de Enfriamiento

3.4. Secuencia de Enfriamiento





3. PARÁMETROS Y COMPONENTES DE ES

3.4. Secuencia de Enfriamiento

3. Velocidad de Enfriamiento

- **L(T) debe ser suficientemente grande como para que el sistema llegue a alcanzar su estado estacionario para esa temperatura**

- **Lo habitual es que sea un valor fijo, pero hay una variante que permite decidir mejor cuando finalizar la iteración actual y enfriar**

- **Consiste en enfriar cuando se dé una de las dos situaciones siguientes:**
  - **Se ha generado un número máximo de vecinos (*máx_vecinos*).**
  - **Se han aceptado un número máximo de vecinos (*máx_éxitos*).**

- **Lógicamente, *máx_vecinos* tiene que ser mayor que *máx_éxitos*. Un buena proporción puede ser *máx_éxitos=0.1*máx_vecinos***

3. PARÁMETROS Y COMPONENTES DE ES

3.4. Secuencia de Enfriamiento

4. Condición de Parada

- **En teoría, el algoritmo debería finalizar cuando T=0. En la práctica, se para:**
  - **cuando T alcanza o está por debajo de un valor final T_f, fijado previamente, o**
  - **después de un número fijo de iteraciones.**

- **Como es difícil dar valor de T_f, se suele usar un número fijo de iteraciones**

- **Una buena opción es parar cuando no se haya aceptado ningún vecino de los L(T) generados en la iteración actual (*num_éxitos=0*)**

  En ese caso, es muy probable que el algoritmo se haya estancado y no vaya a mejorar la solución obtenida

  Combinando este criterio de parada y la condición de enfriamiento de los *máx_vecinos* y *máx_éxitos* se obtiene un equilibrio en la búsqueda que evita malgastar recursos

3. PARÁMETROS Y COMPONENTES DE ES

Mecanismo de Transición/Secuencia de Enfriamiento

```
INPUT (T_o, α, L, T_f)
T ← T_o
S_act ← Genera_solucion_inicial
WHILE T ≥ T_f DO
      BEGIN
      FOR cont ← 1 TO L(T) DO
            BEGIN
            S_cand ← Selecciona_solucion_N(S_act)
            δ ← coste(S_cand) – coste(S_act)
            IF (U(0,1) < e^(-δ/T)) OR
                (δ < 0)   THEN S_act ← S_cand
            END
      T ← α(T)
      END
{Escribe como solución, la mejor de las S_act visitadas}
```

Valor inicial del parámetro de control

Condición de Parada

Velocidad de enfriamiento L(T)

Generación de una nueva solución

Cálculo de la diferencia de costos

Aplicación del criterio de aceptación

Mecanismo de enfriamiento

4. APLICACIONES

Aplicación de ES a Max-Cut, QAP y SI:

1. **Inicialización: aleatoria o greedy**
2. **Representación: igual que en la Búsqueda Local**
3. **Operador de generación de vecinos: igual que en la Búsqueda Local**
4. **Esquema de enfriamiento: esquema de Cauchy**
5. **Condición de enfriamiento: cuando se genera un número máximo de soluciones vecinas, L(T) constante**
6. **Condición de parada: cuando se alcance un número máximo de iteraciones**

ALGORÍTMICA
2006 - 2007

- Parte I. Introducción a las Metaheurísticas
  - Tema 1. Metaheurísticas: Introducción y Clasificación
- Parte II. Métodos Basados en Trayectorias y Entornos
  - Tema 2. Algoritmos de Búsqueda Local Básicos
  - Tema 3. Algoritmos de Enfriamiento Simulado
  - Tema 4. Algoritmos de Búsqueda Tabú
  - Tema 5. Métodos Basados en Trayectorias Múltiples I: Métodos Multiarranque Básicos y GRASP
  - Tema 6. Métodos Basados en Trayectorias Múltiples II: ILS y VNS
- Parte III. Métodos Basados en Poblaciones
  - Tema 7. Algoritmos Genéticos
- Parte IV. Intensificación y Diversificación
  - Tema 8. Estudio del Equilibrio entre Intensificación y Diversificación
- Parte V. Metaheurísticas Híbridas: Poblaciones y Trayectorias
  - Tema 9. Algoritmos Meméticos
  - Tema 10. Modelos Híbridos II: *Scatter Search*
- Parte VI. Paralelización de Metaheurísticas
  - Tema 11. Metaheurísticas en Sistemas Descentralizados
- Parte VII. Conclusiones
  - Tema 12. Algunas Consideraciones sobre la Adaptación de Metaheurísticas a la Resolución de Problemas