Metaheurísticas

Unidad 2
Metaheurísticas basadas en Trayectorias

Tema 1: Enfriamiento Simulado

Objetivos

- Conocer los fundamentos de las búsquedas basadas en trayectorias: enfriamiento simulado
- Entender los fundamentos termodinámicos que sirven de base al enfriamiento simulado
- Conocer las diferentes fases que componen un algoritmo de enfriamiento simulado, así como los parámetros y componentes que lo condicionan
- Saber aplicar el enfriamiento simulado a problemas reales

Bibliografía

[Kirk83] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by Simulated Annealing, Science 220:4598 (1983) 671-680

[Diaz96] Díaz y otros. Optimización Heurística y Redes Neuronales. Paraninfo, 1996

[Dows03] K. A. Dowsland, A. Díaz. Diseño de heurísticas y fundamentos del recocido simulado. Inteligencia Artificial VII:2 (2003) 93-101.

[Sum06] 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): 1143-1160 OCT 2006

[Hen03] D. Henderson, S.H. Jacobson, A.W. Johnson. Chapter 10: The Theory and Practice of Simmulated Annealing. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 287-319

Motivación

- Recordamos que la búsqueda local tiende hacia óptimos locales que pueden estar muy alejados del óptimo global

- Necesitamos movimientos, en algunos casos de empeoramiento, que permitan modificar el movimiento de la trayectoria hacia el óptimo local/global

Índice

Fundamentos

introducción

El enfriamiento simulado es un algoritmo de búsqueda por entornos con un criterio probabilístico de aceptación de soluciones basados en la Termodinámica

Fundamentos

introducción

El enfriamiento simulado es un algoritmo de búsqueda por entornos con un criterio probabilístico de aceptación de soluciones basados en la Termodinámica

S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi, Optimization by Simulated Annealing, Science, Vol 220, Number 4598, pages 671-680, 1983

Fundamentos

introducción

S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi, Optimization by Simulated Annealing, Science, Vol 220, Number 4598, pages 671-680, 1983

Fundamentos

introducción

S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi, Optimization by Simulated Annealing, Science, Vol 220, Number 4598, pages 671-680, 1983

There is a deep and useful connection between statistical mechanics (the behavior of systems with many degrees of freedom in thermal equilibrium at a finite temperature) and multivariate or combinatorial optimization (finding the minimum of a given function depending on many parameters). A detailed analogy with annealing in solids provides a framework for optimization of the properties of very large and complex systems. This connection to statistical mechanics exposes new information and provides an unfamiliar perspective on traditional optimization problems and methods.

Fundamentos

introducción

- 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

Fundamentos

introducción

- En concreto se basa en el proceso físico del templado de metales
- Para conseguir que la estructura molecular del metal tenga las propiedades deseadas de resistencia o flexibilidad es necesaria controlar la velocidad del proceso de templado
- Si se hace adecuadamente, el estado final del metal es un estado de mínima energía

Fundamentos

introducción

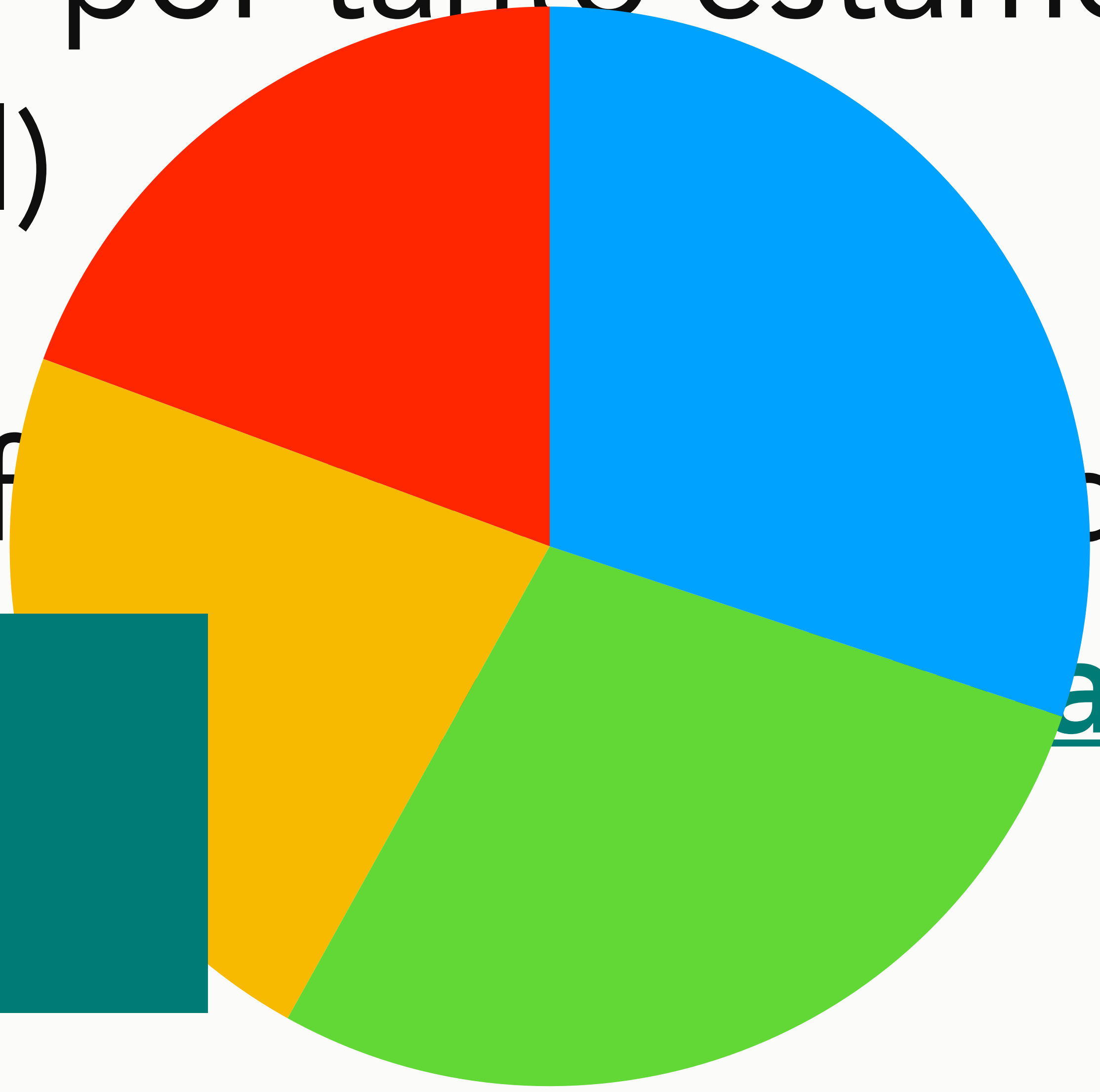
- En el caso del Enfriamiento Simulado (ES), esto se realiza controlando la frecuencia de los movimientos de escape mediante una **función de probabilidad que hará disminuir la probabilidad de estos movimientos hacia soluciones peores conforme avanza la búsqueda** (y por tanto estamos más cerca, previsiblemente,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

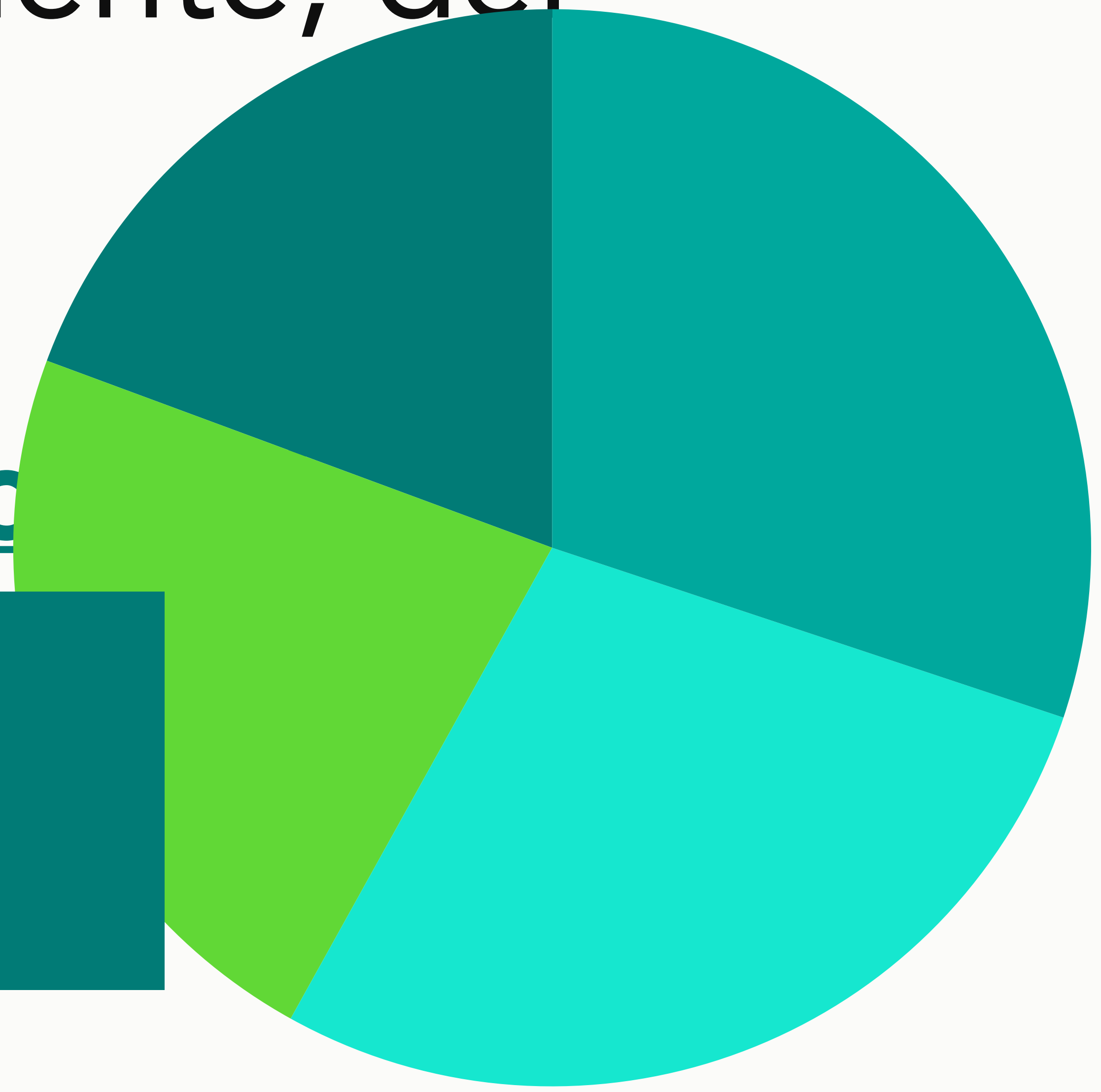
Fundamentos

introducción

- 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)

- Se aplica la f de búsqueda de **diversific** **al**





Fundamentos

algoritmo 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 donde k es un constante física denominada Boltzmann**, y se puede aproximar por:

$$P[\delta E] = exp\left(\frac{-\delta E}{kt}\right)$$

Fundamentos

algoritmo de metrópolis

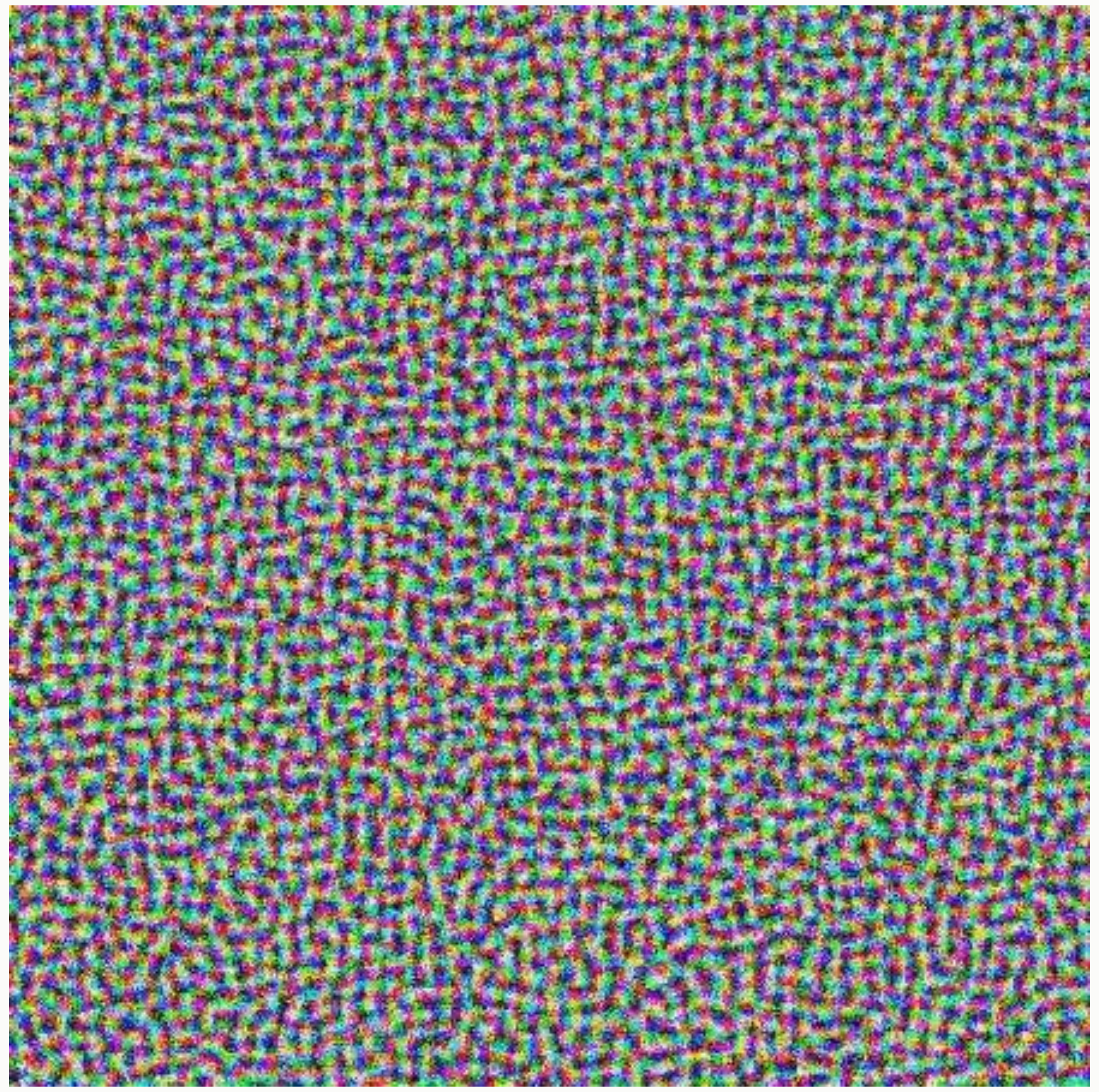
- En el modelo de Metrópolis, se genera una perturbación aleatoria en el sistema y se calculan los cambios de energía resultantes: si hay una caída energética, el cambio se acepta automáticamente; por el contrario, **si se produce un incremento energético**, el cambio será aceptado con una probabilidad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

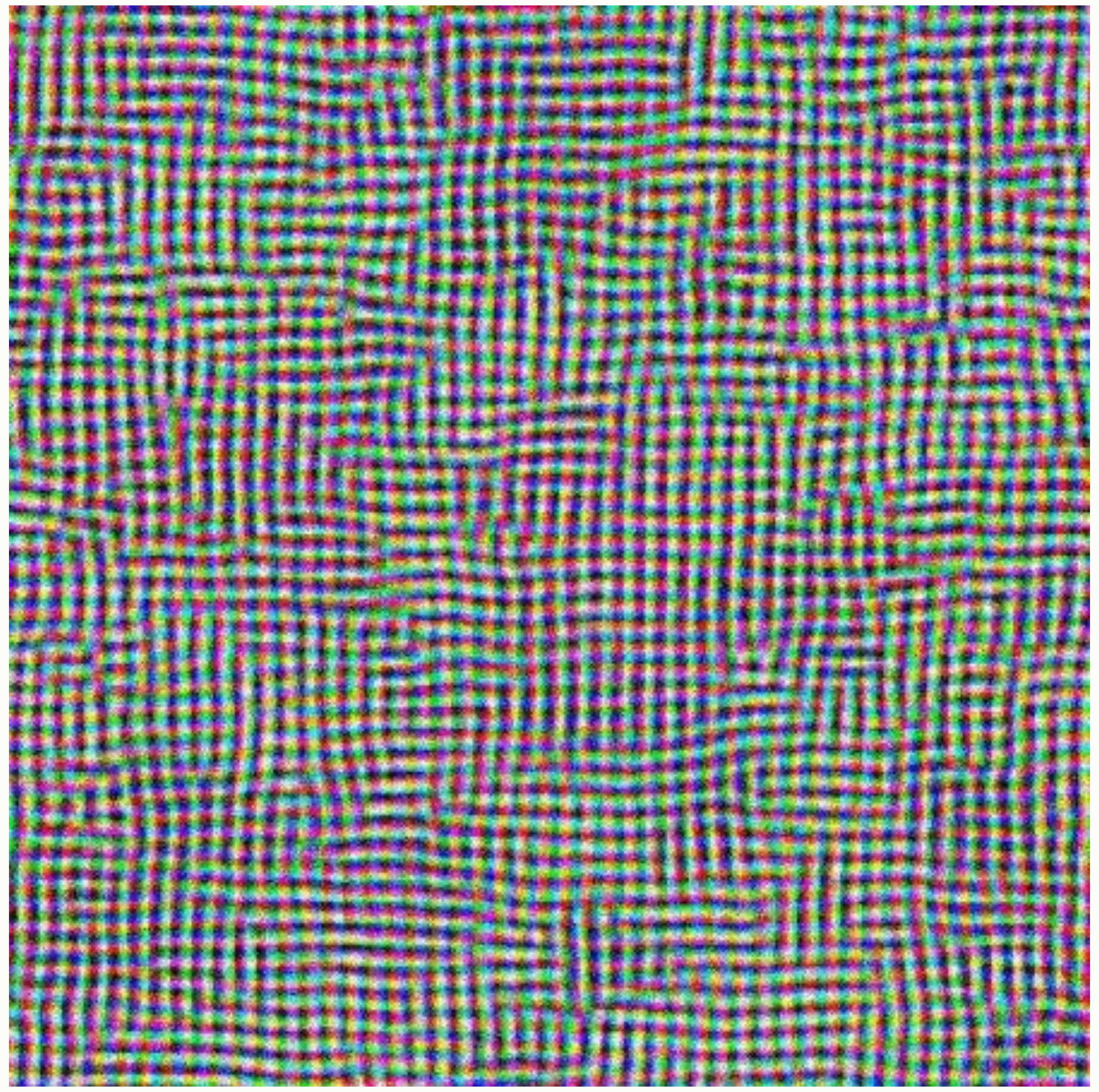
Fundamentos

algoritmo de metrópolis

Wikipedia: Recocido simulado

El problema consiste en disponer los píxeles en la imagen de tal manera que se minimice una función de energía potencial que causa que los colores similares se atraigan a distancias cortas y se repelan a distancias largas. En cada iteración se intercambian las posiciones de dos píxeles adyacentes





La imagen de la izquierda es obtenida con un protocolo de enfriado rápido, en el que la temperatura desciende rápidamente, y la de la derecha, con un protocolo lento, equiparables a los procesos de formación de sólidos amorfos y cristalinos respectivamente

Fundamentos

analogías: termodinámica - optimización

Simulación termodinámica	Optimización combinatoria
Estados del sistema	Soluciones factibles
Energía	Coste
Cambio de estado	Solución del entorno
Temperatura	**Parámetro de control**
Estado congelado	Solución heurística

Algoritmo básico del Enfriamiento Simulado

- 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)**

Algoritmo básico del Enfriamiento Simulado

- En cada iteración se genera **un número concreto de vecinos L(T)** que puede ser fijo para toda la ejecución o depender de la iteración concreta
- **Cada vez que se genera un vecino, se aplica el criterio de aceptación para ver si sustituye a la solución actual**
- Si la solución vecina es mejor que la actual, se acepta automáticamente, tal como se haría en la búsqueda local clásica
- En cambio, **si es peor, aún existe la probabilidad de que el vecino sustituya a la solución actual**. Esto permite al algoritmo salir de óptimos locales, en los que la BL clásica quedaría atrapada

Algoritmo básico del Enfriamiento Simulado

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

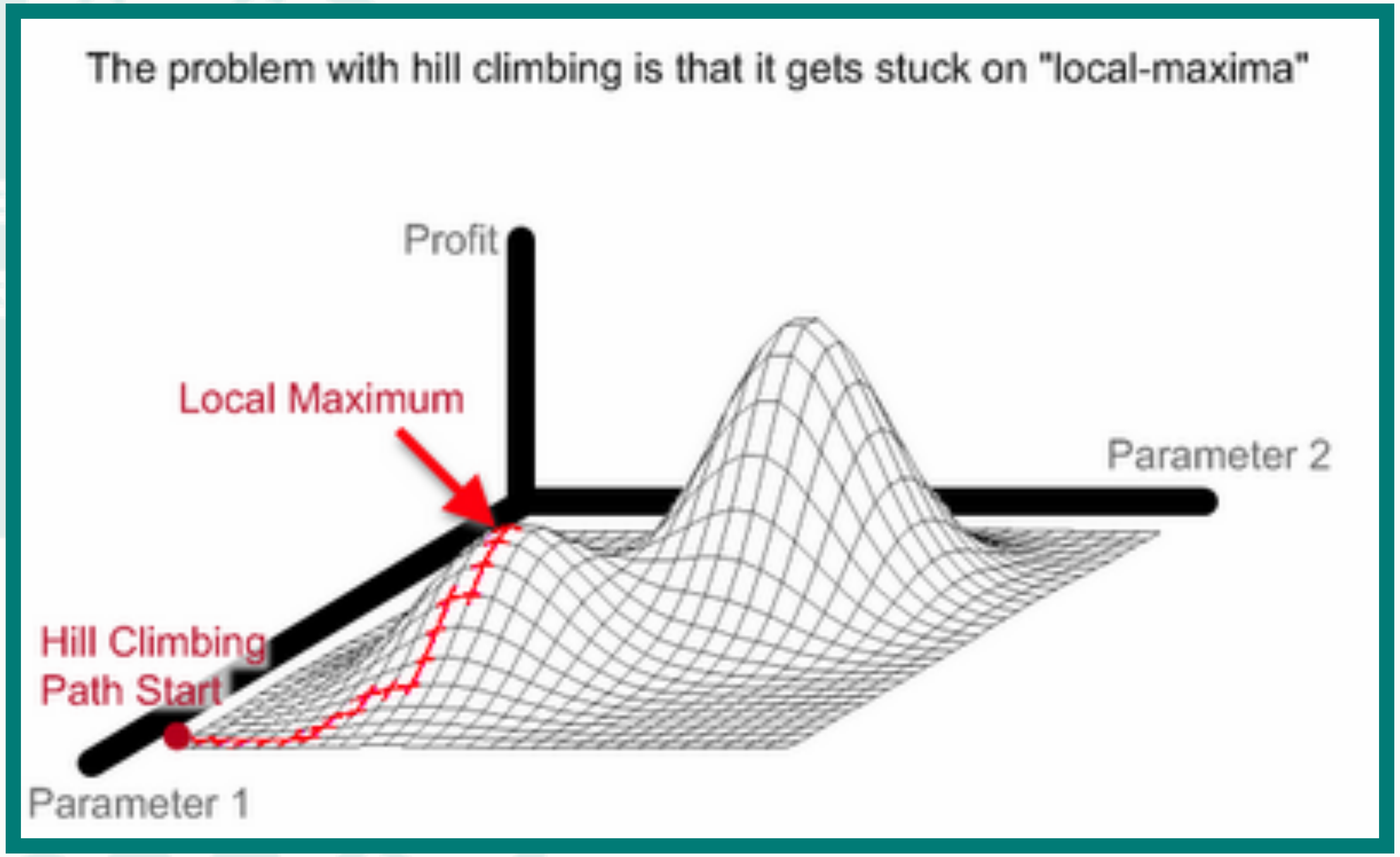
$$P_{aceptacion} = exp\frac{-\delta}{T}$$

- **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**)

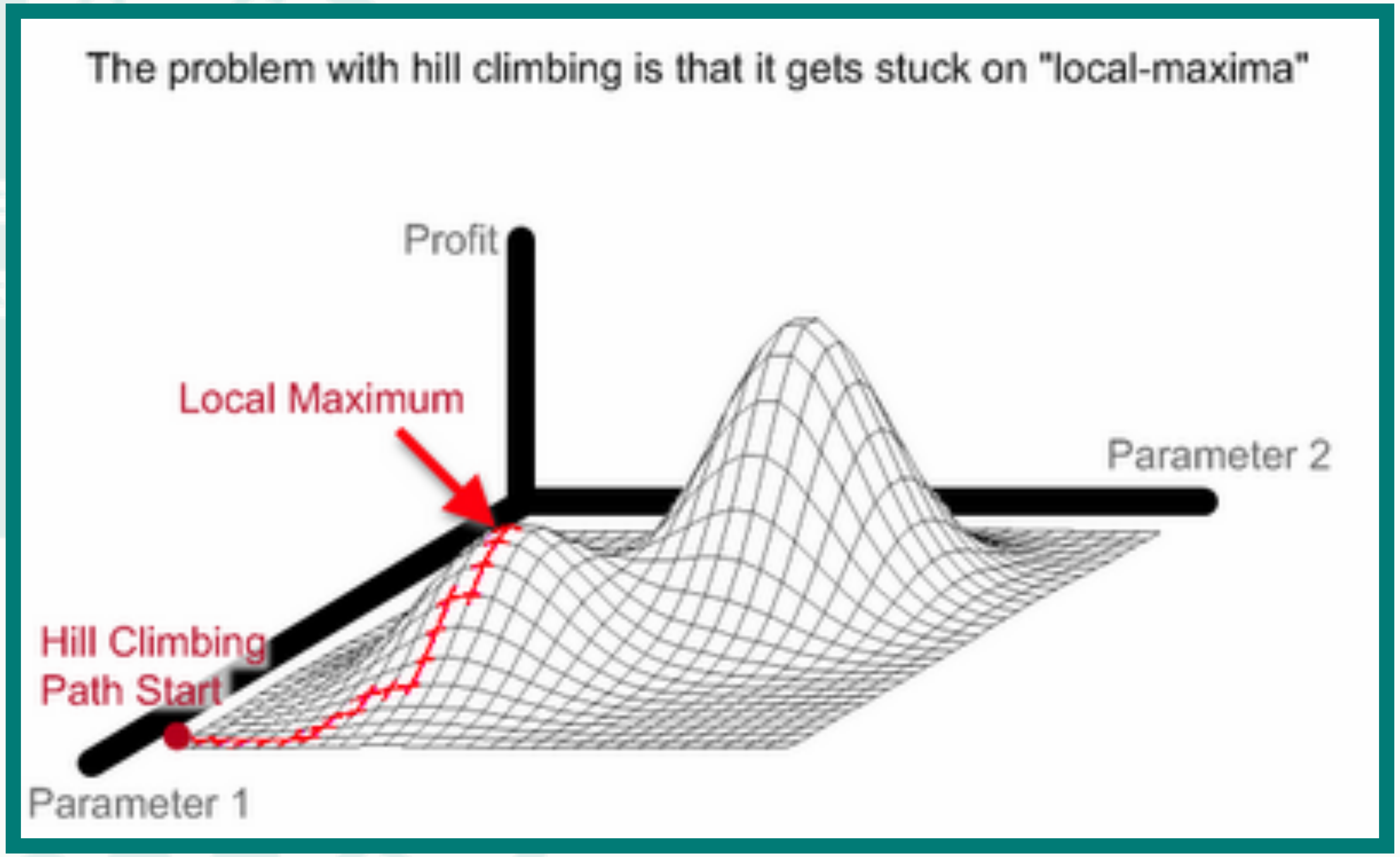
- **A menor diferencia de costes, mayor probabilidad de aceptación de soluciones peores**

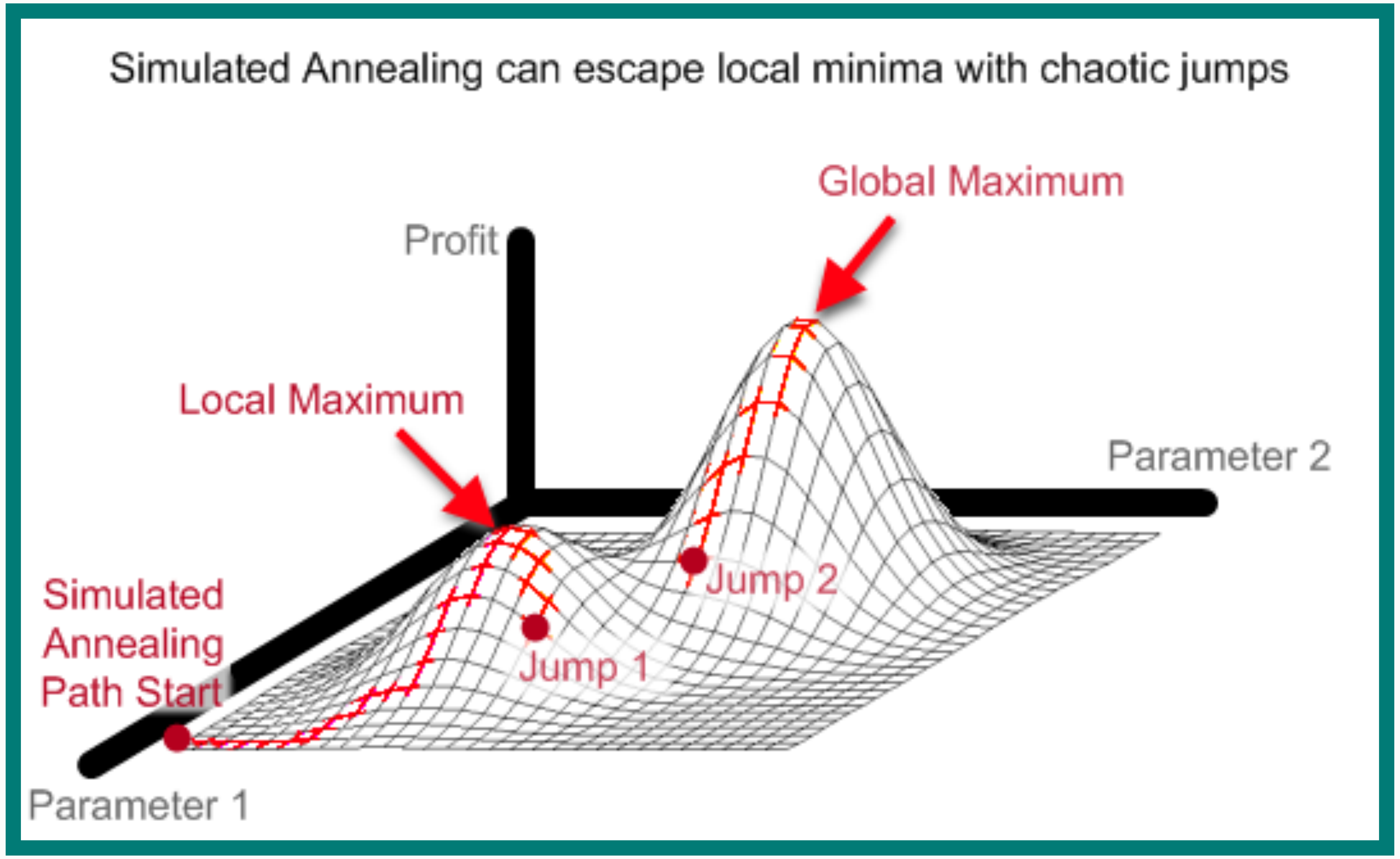
- Una vez finalizada la iteración, es decir, tras generar L(T) soluciones vecinas, **se enfría la temperatura y se pasa a la siguiente iteración**

Algoritmo básico del Enfriamiento Simulado

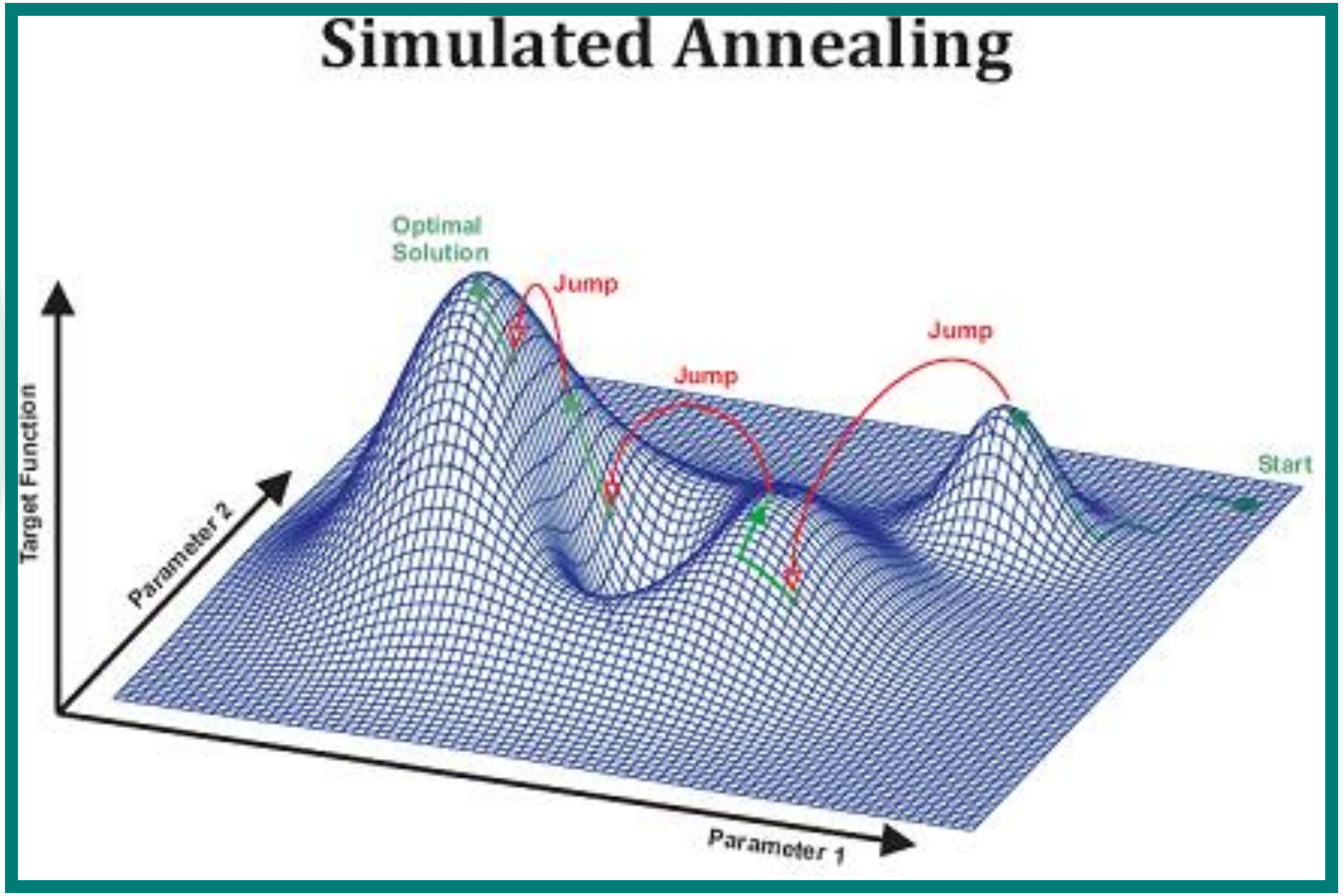


Algoritmo básico del Enfriamiento Simulado





Algoritmo básico del Enfriamiento Simulado

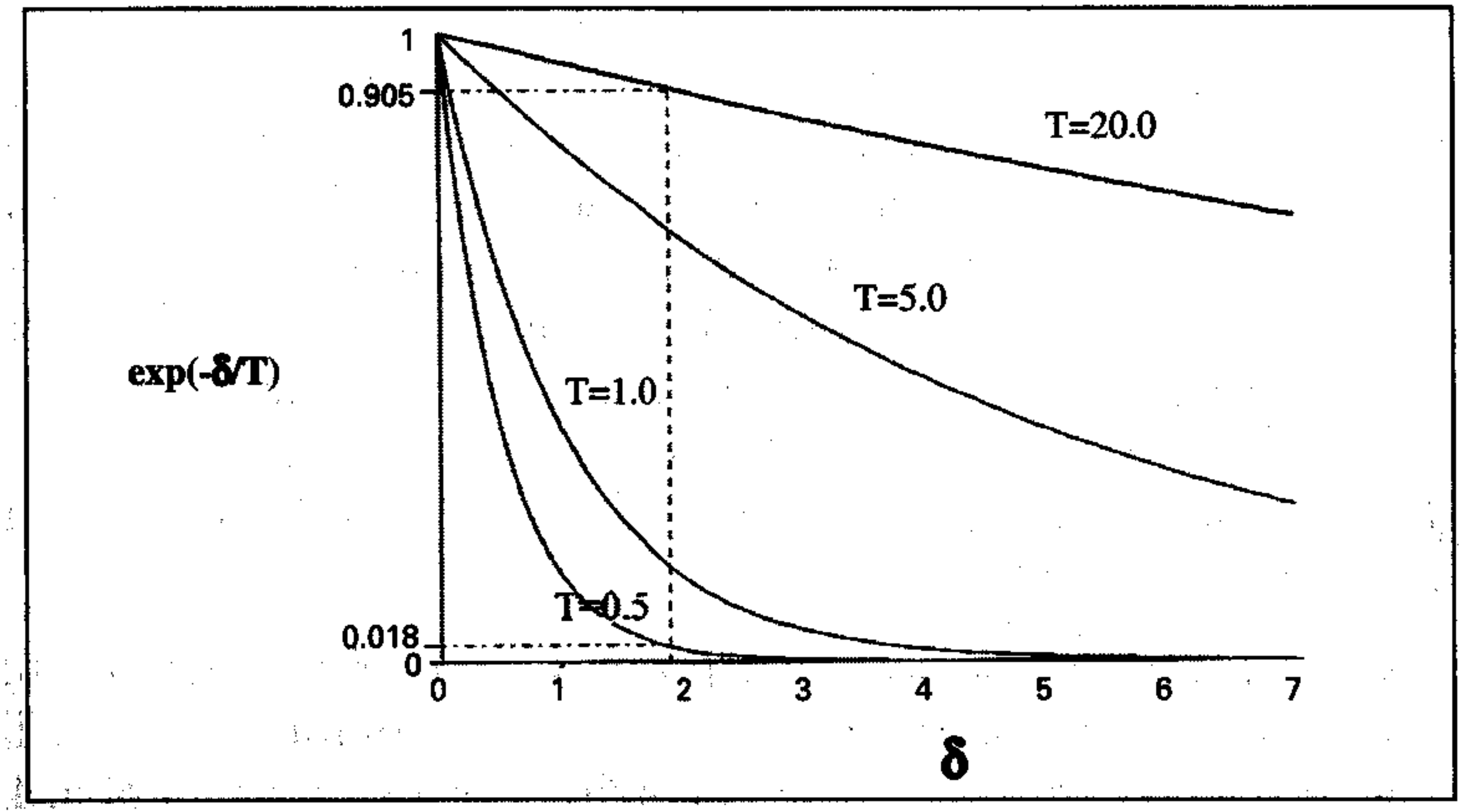


Algoritmo básico del Enfriamiento Simulado

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

```
Procedimiento Simulated Annealing (Δf para minimizar)
START
  T = T0;   s = GENERATE();   Best Solution = s;
  REPEAT
    FOR cont = 1 to L(T)
      s' = NEIGHBORHOOD_OP(s);
      Δf=f(s')–f(s);
      IF ((Δf<0) or (U(0,1) <= exp(-Δf/k·T))) THEN
        s = s';
        IF COST(s) is better than COST(Best Solution) THEN
          Best Solution = s;
    ENDFOR
    T = g(T);  //Esquema de enfriamiento. El clásico es geométrico T = α·T
  UNTIL (T <= Tf); //Normalmente Tf=0
  RETURN (Best Solution);
END
```

Algoritmo básico del Enfriamiento Simulado



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.*

$\Delta f = \delta = C(s') - C(s)$

s = solución actual

s′ = solución vecina

Parámetros y componentes

Solución

Representación de la solución del problema

- Representación
  - Vector ordenado como el viajante de comercio
  - Vector binario como el problema de la mochila
- Solución inicial
  - Aleatoria
  - Conocimiento experto (greedy)

(2 4 5 1 3)

(1 0 0 0 0 1 0)

Cristóbal J. Carmona <ccarmona@ujaen.es>

Parámetros y componentes

Entorno

Soluciones cercanas

- Generación de una nueva solución
  - Definición del conjunto de vecinos
  - Selección de un elemento de dicho conjunto

Cristóbal J. Carmona <ccarmona@ujaen.es>

Parámetros y componentes

Movimiento

Transformación de la solución actual en otra

Procedimiento Simulated Annealing (Δf para minimizar)

START
  T = T0;   s = GENERATE();   Best Solution = s;
  REPEAT
    FOR cont = 1 to L(T)
      s' = NEIGHBORHOOD_OP(s);
      Δf=f(s')–f(s);
      IF ((Δf<0) or (U(0,1) <= exp(-Δf/k·T))) THEN
        s = s';
        IF COST(s) **is better than** COST(Best Solution) THEN
          Best Solution = s;
    ENDFOR
    T = g(T); *//Esquema de enfriamiento. El clásico es geométrico T = α·T*
  UNTIL (T <= Tf); *//Normalmente Tf=0*
  RETURN (Best Solution);
END

Cristóbal J. Carmona <ccarmona@ujaen.es>

Parámetros y componentes

Movimiento

Transformación de la solución actual en otra

Procedimiento Simulated Annealing (Δf para minimizar)

START

T = T0;   s = GENERATE();   Best Solution = s;

REPEAT

FOR cont = 1 to L(T)

s' = NEIGHBORHOOD_OP(s);

Δf=f(s')–f(s);

IF ((Δf<0) or (U(0,1) <= exp(-Δf/k·T))) THEN

s = s';

IF COST(s) **is better than** COST(Best Solution) THEN

Best Solution = s;

ENDFOR

T = g(T); *//Esquema de enfriamiento. El clásico es geométrico T = α·T*

UNTIL (T <= Tf); *//Normalmente Tf=0*

RETURN (Best Solution);

END

Nueva solución

Parámetros y componentes

Movimiento

Transformación de la solución actual en otra

Procedimiento Simulated Annealing (Δf para minimizar)

```
START
 T = T0;   s = GENERATE();   Best Solution = s;
 REPEAT
   FOR cont = 1 to L(T)
     s' = NEIGHBORHOOD_OP(s);
     Δf=f(s')–f(s);
     IF ((Δf<0) or (U(0,1) <= exp(-Δf/k·T))) THEN
      s = s';
      IF COST(s) is better than COST(Best Solution) THEN
       Best Solution = s;
   ENDFOR
   T = g(T);  //Esquema de enfriamiento. El clásico es geométrico T = α·T
 UNTIL (T <= Tf); //Normalmente Tf=0
 RETURN (Best Solution);
END
```

Nueva solución

Diferencia de costes

Parámetros y componentes

Movimiento

Transformación de la solución actual en otra

Procedimiento Simulated Annealing (Δf para minimizar)

START
  T = T0;   s = GENERATE();   Best Solution = s;
  REPEAT
    FOR cont = 1 to L(T)
      s' = NEIGHBORHOOD_OP(s);
      Δf=f(s')–f(s);
      IF ((Δf<0) or (U(0,1) <= exp(-Δf/k·T))) THEN
        s = s';
        IF COST(s) **is better than** COST(Best Solution) THEN
          Best Solution = s;
    ENDFOR
    T = g(T); *//Esquema de enfriamiento. El clásico es geométrico T = α·T*
  UNTIL (T <= Tf); *//Normalmente Tf=0*
  RETURN (Best Solution);
END

Nueva solución

Diferencia de costes

Criterio de aceptación

Parámetros y componentes

Enfriamiento

Mecanismo para controlar el enfriamiento de la temperatura

```
Procedimiento Simulated Annealing (Δf para minimizar)
START
 T = T0;   s = GENERATE();   Best Solution = s;
 REPEAT
   FOR cont = 1 to L(T) //¿Qué es L(T)?
     s' = NEIGHBORHOOD_OP(s);
     Δf=f(s')–f(s);
     IF ((Δf<0) or (U(0,1) <= exp(-Δf/k·T))) THEN
      s = s';
      IF COST(s) is better than COST(Best Solution) THEN
       Best Solution = s;
   ENDFOR
   T = g(T);  //Esquema de enfriamiento. El clásico es geométrico T = α·T
 UNTIL (T <= Tf); //Normalmente Tf=0
 RETURN (Best Solution);
END
```

Parámetros y componentes

Valor inicial del parámetro de control

Procedimiento Simulated Annealing (Δf para minimizar)

```
START
 T = T0;  s = GENERATE();  Best Solution = s;
 REPEAT
   FOR cont = 1 to L(T) //¿Qué es L(T)?
     s' = NEIGHBORHOOD_OP(s);
     Δf=f(s')–f(s);
     IF ((Δf<0) or (U(0,1) <= exp(-Δf/k·T))) THEN
      s = s';
      IF COST(s) is better than COST(Best Solution) THEN
       Best Solution = s;
   ENDFOR
   T = g(T);  //Esquema de enfriamiento. El clásico es geométrico T = α·T
 UNTIL (T <= Tf); //Normalmente Tf=0
 RETURN (Best Solution);
END
```

Enfriamiento

Mecanismo para controlar el enfriamiento de la temperatura

Parámetros y componentes

Procedimiento Simulated Annealing (Δf para minimizar)

```
START
T = T0;   s = GENERATE();   Best Solution = s;
 REPEAT
    FOR cont = 1 to L(T) //¿Qué es L(T)?
       s' = NEIGHBORHOOD_OP(s);
       Δf=f(s')–f(s);
       IF ((Δf<0) or (U(0,1) <= exp(-Δf/k·T))) THEN
        s = s';
        IF COST(s) is better than COST(Best Solution) THEN
          Best Solution = s;
    ENDFOR
    T = g(T); //Esquema de enfriamiento. El clásico es geométrico T = α·T
 UNTIL (T <= Tf); //Normalmente Tf=0
 RETURN (Best Solution);
END
```

Valor inicial del parámetro de control

Mecanismo de enfriamiento

Enfriamiento

Mecanismo para controlar el enfriamiento de la temperatura

Parámetros y componentes

```
Procedimiento Simulated Annealing (Δf para minimizar)
START
T = T0; s = GENERATE(); Best Solution = s;
 REPEAT
   FOR cont = 1 to L(T) //¿Qué es L(T)?
     s' = NEIGHBORHOOD_OP(s);
     Δf=f(s')–f(s);
     IF ((Δf<0) or (U(0,1) <= exp(-Δf/k·T))) THEN
      s = s';
      IF COST(s) is better than COST(Best Solution) THEN
       Best Solution = s;
   ENDFOR
   T = g(T); //Esquema de enfriamiento. El clásico es geométrico T = α·T
 UNTIL (T <= Tf); //Normalmente Tf=0
 RETURN (Best Solution);
END
```

Valor inicial del parámetro de control

Mecanismo de enfriamiento

Condición de parada

Enfriamiento

Mecanismo para controlar el enfriamiento de la temperatura

Parámetros y componentes

Valor inicial del parámetro de control

```
Procedimiento Simulated Annealing (Δf para minimizar)
START
T = T0;  s = GENERATE();  Best Solution = s;
 REPEAT
   FOR cont = 1 to L(T) //¿Qué es L(T)?
     s' = NEIGHBORHOOD_OP(s);
     Δf=f(s')–f(s);
     IF ((Δf<0) or (U(0,1) <= exp(-Δf/k·T))) THEN
      s = s';
      IF COST(s) is better than COST(Best Solution) THEN
       Best Solution = s;
   ENDFOR
   T = g(T); //Esquema de enfriamiento. El clásico es geométrico T = α·T
 UNTIL (T <= Tf); //Normalmente Tf=0
 RETURN (Best Solution);
END
```

Mecanismo de enfriamiento

Condición de parada

Enfriamiento

Mecanismo para controlar el enfriamiento de la temperatura

Parámetros y componentes

valor inicial del parámetro de control (temperatura)

Enfriamiento

Mecanismo para controlar el enfriamiento de la temperatura

- No parece conveniente considerar valores fijos independientes del problema. Por ejemplo:

$$T_0 = \left(\frac{\mu}{-ln(\phi)}\right) \cdot C(S_0)$$

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

Parámetros y componentes

mecanismo de enfriamiento

Enfriamiento

Mecanismo para controlar el enfriamiento de la temperatura

- Existen muchos mecanismo de enfriamiento:
  - Temperaturas descendentes fijadas por el usuario
  - Descenso constante
  - Descenso geométrico
  - ***Criterio de Boltzmann***
  - Esquema de Cauchy
  - Cauchy modificado para ejecutar exactamente M iteraciones

$$T_k = \frac{T_0}{(1 + log(k))}$$

k=nº iteración actual

Parámetros y componentes

condición de parada

Enfriamiento

Mecanismo para controlar el enfriamiento de la temperatura

- 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 Tf, fijado previamente, o después de un número fijo de iteraciones
- Como es difícil dar valor de Tf, 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

EJEMPLO

Maximizar la función:

$$f(x)=x^3-60x^2+900x+100$$

- Representación binaria con 5 bits de x
- Vecindad mediante el cambio de 1 bit

Cristóbal J. Carmona <ccarmona@ujaen.es>

Ejemplo

`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

Ejemplo

maximizar la función f(x)=x3-60x2+900x+100

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					

Ejemplo

```
maximizar la función f(x)=x3-60x2+900x+100
```

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					
1	500	1	"00011"	2287	-112	0.80	0.54	Yes	"00011"

Cristóbal J. Carmona <ccarmona@ujaen.es>

Ejemplo

`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					
1	500	1	"00011"	2287	-112	0.80	0.54	Yes	"00011"
2	500*0.9	3	"00111"	3803	>0			Yes	"00111"

Ejemplo

`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					
1	500	1	"00011"	2287	-112	0.80	0.54	Yes	"00011"
2	500*0.9	3	"00111"	3803	>0			Yes	"00111"
3	405	5	"00110"	3556	-247	0.82	0.78	Yes	"00110"

Ejemplo

maximizar la función f(x)=x3-60x2+900x+100

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol’
0			“10011”	2399					
1	500	1	“00011”	2287	-112	0.80	0.54	Yes	“00011”
2	500*0.9	3	“00111”	3803	>0			Yes	“00111”
3	405	5	“00110”	3556	-247	0.82	0.78	Yes	“00110”
4	364.5	2	“01110”	3684	>0			Yes	“01110”

Cristóbal J. Carmona <ccarmona@ujaen.es>

Ejemplo

`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					
1	500	1	"00011"	2287	-112	0.80	0.54	Yes	"00011"
2	500*0.9	3	"00111"	3803	>0			Yes	"00111"
3	405	5	"00110"	3556	-247	0.82	0.78	Yes	"00110"
4	364.5	2	"01110"	3684	>0			Yes	"01110"
5	328	4	"01100"	3998	>0			Yes	"01100"

Ejemplo

maximizar la función f(x)=x3-60x2+900x+100

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol’
0			“10011”	2399					
1	500	1	“00011”	2287	-112	0.80	0.54	Yes	“00011”
2	500*0.9	3	“00111”	3803	>0			Yes	“00111”
3	405	5	“00110”	3556	-247	0.82	0.78	Yes	“00110”
4	364.5	2	“01110”	3684	>0			Yes	“01110”
5	328	4	“01100”	3998	>0			Yes	“01100”
6	295.2	3	“01000”	3972	-16	1.00	0.23	Yes	“01000”

Ejemplo

`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					
1	500	1	"00011"	2287	-112	0.80	0.54	Yes	"00011"
2	500*0.9	3	"00111"	3803	>0			Yes	"00111"
3	405	5	"00110"	3556	-247	0.82	0.78	Yes	"00110"
4	364.5	2	"01110"	3684	>0			Yes	"01110"
5	328	4	"01100"	3998	>0			Yes	"01100"
6	295.2	3	"01000"	3972	-16	1.00	0.23	Yes	"01000"
7	265.7	4	"01010"	***4100***	>0			Yes	"01010"

Cristóbal J. Carmona <ccarmona@ujaen.es>

Ejemplo

maximizar la función f(x)=x3-60x2+900x+100

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol’
0			“10011”	2399					
1	500	1	“00011”	2287	-112	0.80	0.54	Yes	“00011”
2	500*0.9	3	“00111”	3803	>0			Yes	“00111”
3	405	5	“00110”	3556	-247	0.82	0.78	Yes	“00110”
4	364.5	2	“01110”	3684	>0			Yes	“01110”
5	328	4	“01100”	3998	>0			Yes	“01100”
6	295.2	3	“01000”	3972	-16	1.00	0.23	Yes	“01000”
7	265.7	4	“01010”	***4100***	>0			Yes	“01010”
…	…	…	…	…	…	…		…	…

Ejemplo

```
maximizar la función f(x)=x3-60x2+900x+100
```

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

Ejemplo

```
maximizar la función f(x)=x3-60x2+900x+100
```

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					

Ejemplo

`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					
1	100	1	"00011"	2287	-112	0.32	0.45	No	"10011"

Ejemplo

maximizar la función f(x)=x3-60x2+900x+100

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					
1	100	1	"00011"	2287	-112	0.32	0.45	No	"10011"
2	100*0.9	3	"10111"	1227	-1172	0.00	0.94	No	"10011"

Ejemplo

`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol’
0			“10011”	2399					
1	100	1	“00011”	2287	-112	0.32	0.45	No	“10011”
2	100*0.9	3	“10111”	1227	-1172	0.00	0.94	No	“10011”
3	81	5	“10010”	2692	>0			Yes	“10010”

Ejemplo

`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol’
0			“10011”	2399					
1	100	1	“00011”	2287	-112	0.32	0.45	No	“10011”
2	100*0.9	3	“10111”	1227	-1172	0.00	0.94	No	“10011”
3	81	5	“10010”	2692	>0			Yes	“10010”
4	72.9	2	“11010”	516	-2176	0.00	0.42	No	“10010”

Ejemplo

```
maximizar la función f(x)=x3-60x2+900x+100
```

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					
1	100	1	"00011"	2287	-112	0.32	0.45	No	"10011"
2	100*0.9	3	"10111"	1227	-1172	0.00	0.94	No	"10011"
3	81	5	"10010"	2692	>0			Yes	"10010"
4	72.9	2	"11010"	516	-2176	0.00	0.42	No	"10010"
5	65.6	4	"10000"	***3236***	>0			Yes	"10000"

Ejemplo

`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol’
0			“10011”	2399					
1	100	1	“00011”	2287	-112	0.32	0.45	No	“10011”
2	100*0.9	3	“10111”	1227	-1172	0.00	0.94	No	“10011”
3	81	5	“10010”	2692	>0			Yes	“10010”
4	72.9	2	“11010”	516	-2176	0.00	0.42	No	“10010”
5	65.6	4	“10000”	***3236***	>0			Yes	“10000”
6	59	3	“10100”	2100	-1136	0.04	0.14	No	“10000”

Ejemplo

`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					
1	100	1	"00011"	2287	-112	0.32	0.45	No	"10011"
2	100*0.9	3	"10111"	1227	-1172	0.00	0.94	No	"10011"
3	81	5	"10010"	2692	>0			Yes	"10010"
4	72.9	2	"11010"	516	-2176	0.00	0.42	No	"10010"
5	65.6	4	"10000"	***3236***	>0			Yes	"10000"
6	59	3	"10100"	2100	-1136	0.04	0.14	No	"10000"
7	53.1	4	"10010"	2692	-544	0.23	0.17	Yes	"10010"

Ejemplo

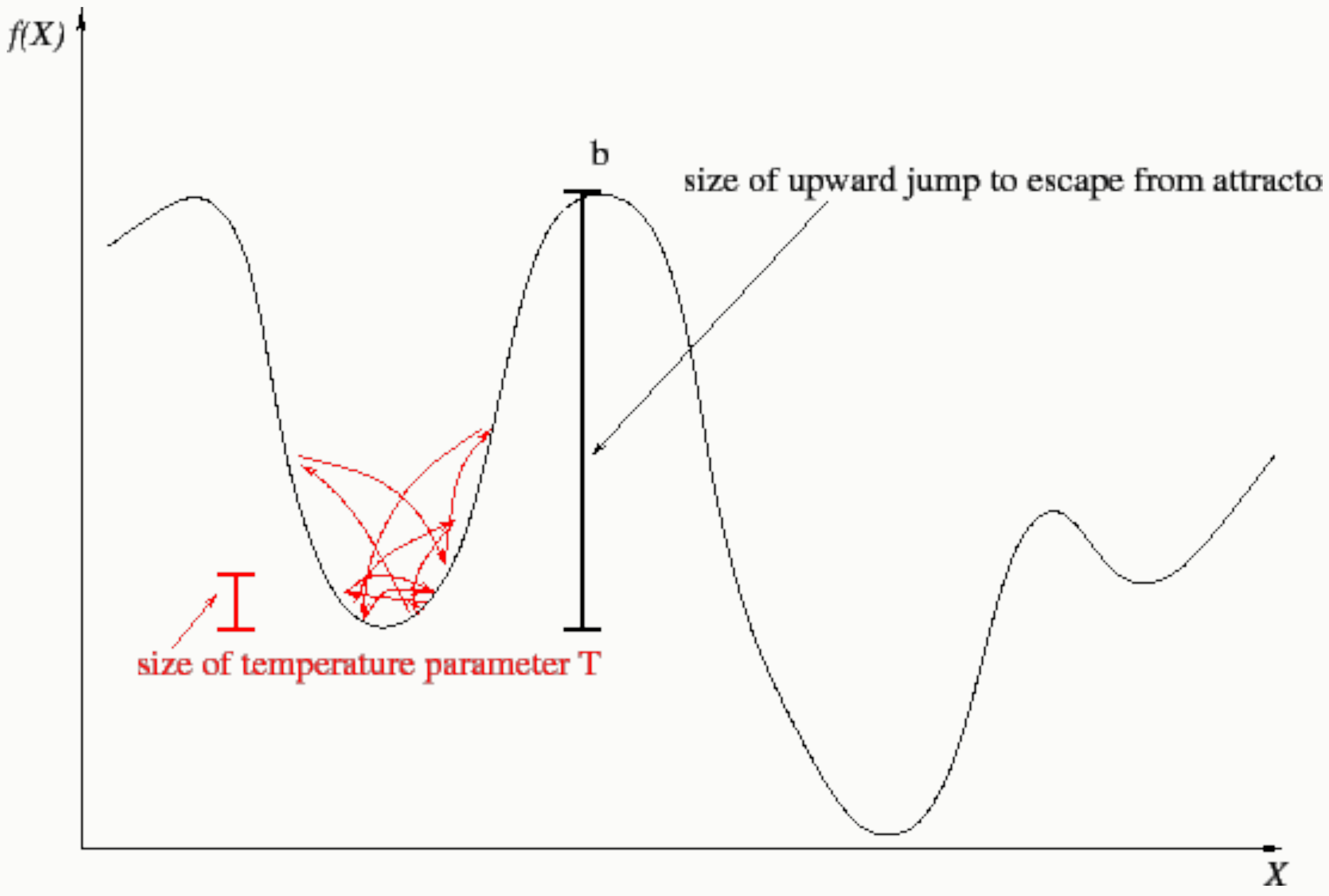
`maximizar la función f(x)=x3-60x2+900x+100`

Solución global con x=10 (0 1 0 1 0) donde f(x)=4100

It	T	Mov	Sol	f(x)	Δf	Pacep	U(0,1)	Mov?	Sol'
0			"10011"	2399					
1	100	1	"00011"	2287	-112	0.32	0.45	No	"10011"
2	100*0.9	3	"10111"	1227	-1172	0.00	0.94	No	"10011"
3	81	5	"10010"	2692	>0			Yes	"10010"
4	72.9	2	"11010"	516	-2176	0.00	0.42	No	"10010"
5	65.6	4	"10000"	***3236***	>0			Yes	"10000"
6	59	3	"10100"	2100	-1136	0.04	0.14	No	"10000"
7	53.1	4	"10010"	2692	-544	0.23	0.17	Yes	"10010"
…	…	…	…	…	…	…		…	…

Comentarios finales

- Técnica de uso muy fácil en muchos problemas pero con riesgos de quedar atrapado en un óptimo local
- Si la temperatura es muy baja respecto al tamaño del salto entonces hay riesgos de quedar atrapado en un óptimo local



Comentarios finales

- **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)

Δf	T	exp(-Δf/T)
0.2	0.95	0.8101
0.4	0.95	0.6563
0.6	0.95	0.5317
0.8	0.95	0.4308

Δf	T	exp(-Δf/T)
0.2	0.1	0.1353
0.4	0.1	0.0183
0.6	0.1	0.0024
0.8	0.1	0.0003

Comentarios finales

- **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)

Δf	T	exp(-Δf/T)
0.2	0.95	0.8101
0.4	[illegible]	[illegible]
0.6	[illegible]	[illegible]
0.8	[illegible]	[illegible]

Δf	T	exp(-Δf/T)
0.2	0.1	0.1353
[illegible]	[illegible]	0.0183
[illegible]	[illegible]	0.0024
[illegible]	[illegible]	0.0003

$$T_0 = \left(\frac{\mu}{-ln(\phi)}\right) \cdot C(S_0)$$

Metaheurísticas
Grado en Ingeniería Informática
Universidad de Jaén
Cristóbal J. Carmona
Curso 2023/2024