CAPÍTULO III

3. RECOCIDO SIMULADO (SIMULATED ANNEALING)

3.1 INTRODUCCION

En los primeros años de la década de los 80 quedó definido el Recocido Simulado, una heurística (Algoritmo de Aproximación) para la resolución de problemas de alta complejidad en el campo de la Combinatoria. Lo que puede sorprender es el hecho de que es producto de una analogía entre este tipo de problemas y resultados teóricos obtenidos en un campo tan distinto como lo es la Termodinámica.

Aunque recientemente ha aparecido en la Literatura científica en Español el término “Temple Simulado” para ésta heurística, debe aclararse que el proceso físico de “Annealing” es “Recocido”, la traducción de “Temple” al Inglés es “Hardening” o “Quenching” que

dentro del proceso de formación molecular tiene unos objetivos totalmente distintos a los del recocido.

3.2 RESUMEN

El algoritmo heurístico denominado Recocido Simulado, (figura 3.1) es un método global de optimización combinatoria establecido a través de una analogía al proceso físico de recocido de sólidos. Cuando a un material fundido se le baja la temperatura muy lentamente sus partículas se agrupan en un arreglo donde la energía interna del sistema es mínima. Metrópolis (1953) desarrolló un algoritmo para simular la evolución de un sólido en un baño de calor, a una temperatura específica, a través de una caminata estocástica entre configuraciones de átomos. Esto es, el sistema pasa sucesivamente de una configuración a otra mediante el siguiente criterio: Estando en una configuración, *i*, con energía E_i, el sistema pasa a una nueva configuración, *j, si* $\Delta E \leq 0$ *donde* $\Delta E = Ej - Ei$*; o si* $P(\Delta E) > r$*, donde* $P(\Delta E) = \exp\{(Ei - Ej) / kB\ t\}$ es la probabilidad de aceptar el incremento ΔE en la energía del sistema *(kB = 1.38 x 10-16 y t = temperatura) y* $r \in \Re$ *con* $0 \leq r \leq 1$. La caminata se efectúa hasta el equilibrio térmico.

Usando la función de costo Z(xi) como la energía del sistema y definiendo la configuración de átomos mediante los valores del conjunto de variables {xi} se establece la analogía al sistema termodinámico. Mediante el algoritmo de Metrópolis se puede generar una sucesión de configuraciones en alguna temperatura efectiva, c = kB t, la cual no es otra cosa que un parámetro de control. Si c es descendido entre ejecuciones del algoritmo de Metrópolis se simula un proceso de recocido y las configuraciones generadas a lo largo del procedimiento tienden a la solución de costo mínimo.

3.3 ALGORITMO DE METRÓPOLIS

El algoritmo de Metrópolis (1953) el cual es muy conocido en la Física Química es la base del Recocido Simulado. Existen dos métodos de solución numérica en Mecánica-Estadística para estudiar el comportamiento microscópico de los cuerpos el método de “Dinámica Molecular” y el método de “Montecarlo”, pero es este último con el que se estudian usualmente las propiedades de equilibrio, y fue éste el utilizado por Metrópolis para su algoritmo.

El origen Químico-Físico de este procedimiento es el siguiente. Dada una sustancia cualquiera, no todas sus moléculas tienen la misma energía, sino que éstas se encuentran distribuidas en distintos niveles y el menor de los cuales se denomina “Estado Fundamental” ε_0. Si la sustancia está a $0^o K$ todas las moléculas están en su estado fundamental, y de acuerdo al incremento de temperatura, las moléculas van ocupando niveles superiores. Cada una de las maneras en que las moléculas pueden estar distribuidas en los distintos niveles se denomina “Microestado”. Al conjunto de todos los posibles microestados se llama Ω y al número de partículas en el nivel i se denomina “número e ocupación n_i.

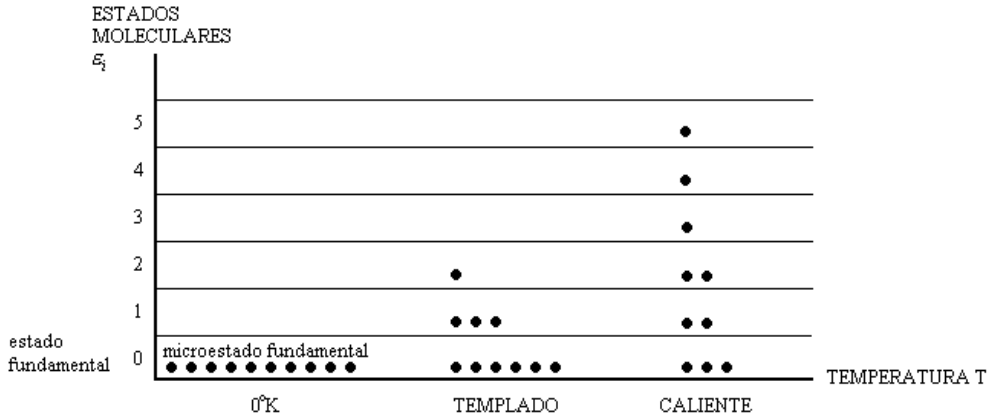

Figura 3.1 *Ejemplo de diferentes Microestados según la temperatura, para* $N = 10$. *En el Microestado correspondiente a TEMPLADO, es* $n_0 = 6, n_1 = 3, n_2 = 1, E = \sum n_i i = 5$. *El número de moléculas decrece exponencialmente para una* T *fija.*

El físico austriaco L. Boltzmann observó que cuando un sistema aislado evoluciona espontáneamente, además de aumentar la entropía, se produce un mayor desorden desde el punto de vista mecánico.

El problema era calcular dado un número de moléculas N y la energía E de la sustancia, de cuántas maneras pueden estar ubicadas las partículas en los distintos niveles, es decir cuantos estados diferentes $card(\Omega)$ pueden existir. Según la Estadística de Maxwell-Boltzmann (la cual considera las partículas totalmente discernibles), $card(\Omega)$ es igual al número de microestados posibles considerando fijos unos números de ocupación $\langle n'_0, n'_1, ..., n'_i \rangle$ (los cuales representan la complexión más probable en el equilibrio, dados N y E), siendo $n'_j = P \exp(-\beta \varepsilon_j)$ con un P factor de proporcionalidad, $\beta = 1/kT$, y $k = 1{,}38054\, x\, 10^{-23}\, J/^0 K$ una constante.

Estrictamente hablando, todo lo anterior deja de ser aplicable en el momento en que se introducen las iteraciones entre partículas ya que las energías ε_j no tienen significado físico alguno. W. Gibbs resolvió este inconveniente introduciendo el concepto de

“Colectivo”, una entidad formada por un número suficientemente grande de “Configuraciones”, cada una de ellas constituida por N partículas.

Metrópolis quería definir un algoritmo que permitiera calcular valores medios de la energía interna y de la presión dentro del formalismo de los colectivos de Gibbs, sin necesidad de recurrir a la solución analítica de las correspondientes integrales. Entonces fue cuando utilizó el conocido método de Montecarlo a la hora de elegir las n configuraciones que serían elegidas para calcular los valores medios buscados. En la elección de esas n configuraciones tienen en cuenta la probabilidad exponencial de que se presente una configuración determinada con energía potencial E_i.

Algoritmo 3.1 : ALGORITMO DE METROPOLIS

---

1. Se colocan las N partículas de la configuración en un cuadrado de arista L, cada una con coordenadas $\langle x_i, y_i \rangle$. Se define una longitud máxima de desplazamiento en un movimiento, β

2. Se calculan dos números aleatorios U_1, U_2 uniformes en el intervalo $[-1,1]$

3. Se calcula la nueva posición de la partícula: $x_{i+1} = x_i + \beta U_1$, $y_{i+1} = y_i + \beta U_2$ (en caso de que se excedan los límites del cuadrado, se entraría por el otro lado)

4. Se calcula la diferencia de energía δ de $\langle x_{i+1}, y_{i+1} \rangle$ respecto a $\langle x_i, y_i \rangle$

5. Se elige la nueva configuración según lo siguiente:

5.1. Si $\delta < 0$, se acepta directamente $\langle x_{i+1}, y_{i+1} \rangle$ como complexión por considerar

5.2. Si no, se acepta $\langle x_{i+1}, y_{i+1} \rangle$ con una probabilidad $p = e^{-\delta/(kT)}$ (para lo cual se genera un $U' \in [0,1]$ y se acepta $\langle x_{i+1}, y_{i+1} \rangle$ si $U' < p$, caso contrario nos quedamos con $\langle x_i, y_i \rangle$

6. Se vuelve al paso 2 para calcular la siguiente configuración a partir de la actual

---

3.4 ANALOGIA ENTRE TERMODINAMICA Y OPTIMIZACION

A principios de los 80 Kirkpatrick (1983) en el diseño de circuitos electrónicos y de modo independiente Cerny (1985) investigando el "Traveling Salesman Problem", aplicaron el algoritmo de metrópolis en algunos problemas de minimización en optimización combinatoria. Es evidente que si lo hicieron tuvieron que crear analogías entre el proceso de simulación termodinámica y los procesos de Optimización, como se muestra en la tabla 3.1 .

Termodinámica		**Optimización**
Configuración	**Analogía**	Solución Factible
Configuración Fundamental	**Analogía**	Solución Optima
Energía de la Configuración	**Analogía**	Costo de la Solución
Temperatura	**Analogía**	?

Tabla 3.1: *Analogía entre Termodinámica y Optimización*

Al concepto físico de temperatura no le corresponde un significado real en el campo de optimización, es entonces un parámetro que tiene que ser ajustado. Se pueden imaginar entonces de manera similar los procesos que ocurren cuando las moléculas de una sustancia van colocándose en los diferentes niveles energéticos buscando el equilibrio a una determinada temperatura, y los que ocurren en optimización. Si se fija la temperatura, las partículas en los distintos niveles siguen una distribución de Boltzmann, entonces cuando una molécula se mueve, ese movimiento será aceptado en la simulación si la energía disminuye, o con una probabilidad proporcional al factor de Boltzmann en caso contrario.

Análogamente en Optimización, si fijamos el parámetro T, se produce una perturbación, aceptando directamente la nueva solución cuando su costo disminuye, o bien con una probabilidad proporcional al factor de Boltzmann en caso contrario.

Como la elección de una solución vecina se selecciona aleatoriamente, este tipo de estrategias presenta el inconveniente de que si durante el proceso de búsqueda se cae en un óptimo local, la técnica no sería capaz de salir de él, para poder evitar esto el recocido simulado nos permite el paso a soluciones peores pero

cada vez con menor probabilidad de acuerdo nos vayamos acercando a la solución óptima.

Si analizamos el factor de Boltzmann en función de la temperatura, vemos según la figura 3.2, que de acuerdo disminuye ésta, también disminuye la probabilidad de un movimiento a una nueva solución peor que la anterior. Recordemos que δ es la diferencia entre la energía de $\langle x_{i+1}, y_{i+1} \rangle$ respecto a $\langle x_i, y_i \rangle$.

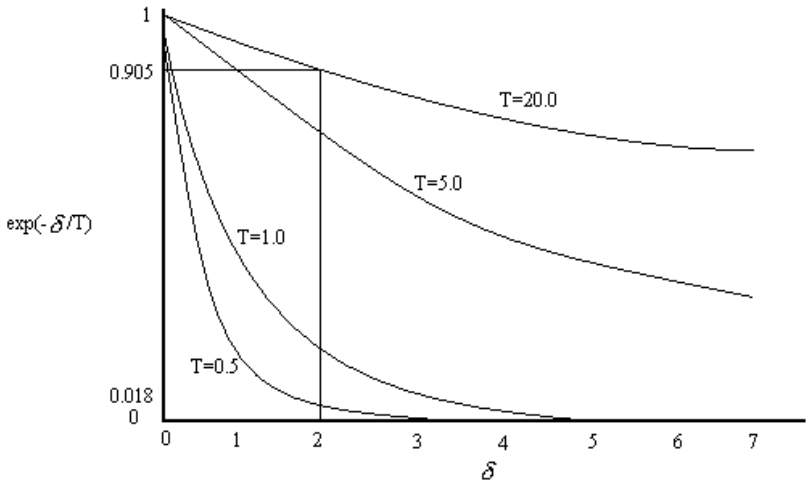

Figura 3.2 *Valor del Factor de Boltzmann en función de la temperatura* T *y de* δ *. Para* $\delta = 2$*, es 50 veces menos probable un movimiento si* $T = 0{,}5$*, que si* $T = 20$

3.5 ESTRATEGIAS PARA EL RECOCIDO SIMULADO

Lo que haremos será lo siguiente:

Primero partir de una “temperatura” alta, con lo cual se permitirá pasos a soluciones peores con mayor probabilidad en los primeros pasos cuando estemos aún lejos de la solución óptima. Luego se irá reduciendo la temperatura, disminuyendo así la probabilidad de pasar soluciones peores cuando estemos cerca del óptimo.

Algoritmo 3.2: ALGORITMO RECOCIDO SIMULADO

$INPUT(T_0, \alpha, L, T_f)$

$T \leftarrow T_0$

$S_{act} \leftarrow Genera_solución_inicial$

$WHILE\ T \geq T_f\ DO$

$BEGIN$

$FOR\ \ cont \leftarrow 1\ TO\ L(T)\ DO$

$BEGIN$

$S_{cand} \leftarrow Selecciona_solución_N(Sact)$

$\delta \leftarrow \cos to(S_{cand}) - \cos to(S_{act})$

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

$\delta < 0\ THEN\ S_{act} \leftarrow Scand$

END

$T \leftarrow \alpha(T)$

END

$\{Escribe_como_soluión, _la_mejor_de_las_S_{act}_visitadas\}$

De acuerdo al algoritmo descrito arriba, se selecciona una temperatura inicial T_0, la velocidad de enfriamiento, es decir, cuanto va a disminuir T_{i+1} con respecto a T_i respecto al parámetro α, luego de estar $L(T)$ iteraciones en esa temperatura T, y una temperatura final T_f. Se genera una solución inicial que pertenece al espacio de soluciones Ω y mientras no se llegue al fin del proceso, para cada T se calcula un numero $L(T)$ de veces, antes de disminuir la temperatura, una solución que esté en la vecindad $N(S_{act})$de la actual, la cual sustituirá a ésta si tiene por su puesto menor costo, o con una probabilidad $e^{-\delta/T}$ (para está probabilidad se debe generar un número aleatorio uniforme $U(0,1)$). Y finalmente la mejor de todas las S_{act} visitadas será nuestra solución.

En el artículo original de Kirkpatrick (1983) la condición de parada (estado frío) consistía en que en tres temperaturas sucesivas no se produjese un número mínimo aceptable de mejoras de la solución.

3.6 PROGRAMA DE ENFRIAMIENTO

3.6.1 Introducción

Un programa de enfriamiento no tiene reglas generales, la elección de los parámetros correspondientes depende de las ideas generales asociadas a éstos.

Muchos han trabajado en el programa óptimo de enfriamiento como por ejemplo: S. Geman y D. Geman, B. Hajek [1988], B. Gidas y Hawng, mientras que al mismo tiempo D. Mitra y A. Sangiovanni-Vicentelli encontraron cotas (para tiempos finitos) a la distancia entre una distribución estacionaria a esa misma temperatura.

Uno de los primeros intentos de automatizar el programa de enfriamiento fue el de S.R. White [1984], quien eligió como la

máxima temperatura que se debía utilizar aquella que es proporcional a la desviación típica del costo de la función objetivo, muestreada en configuraciones tomada al azar.

3.6.2 Temperatura Inicial, T_0

Una de las características que debe cumplir una heurística es que no dependa de la solución de inicio o de partida. Como hemos visto esto lo consigue el Recocido Simulado por medio de elegir una temperatura inicial alta, con lo cual esta claro que al principio recorrerá soluciones lejanas a la óptima.

No es conveniente considerar para T_0 valores fijos independientes del problema, esto parece claro en el siguiente sentido, por ejemplo si la solución inicial S_0 tiene costo $C(S_0)$, y la seleccionada aleatoriamente en su vecindad o entorno un costo de $C(S_{cand}) = C(S_0) + 100$ ¿es entonces aceptable aceptar el candidato?, está claro que la respuesta depende del orden de magnitud de los costos: si por ejemplo $C(S_0)$ vale 0.01, no parece conveniente el movimiento, pero si es del orden de 10^6, podríamos entonces elegir al candidato ya que el valor de 100 no hace una gran diferencia en este caso, incluso para valores pequeños de T.

Podemos, entonces determinar T_0 en función del porcentaje de aceptación deseable. Consideremos que pudiera ser aceptable con un ϕ de probabilidad una solución que sea un μ por uno peor que la inicial S_0. Entonces, si el $C(S_{cand}) = (1+\mu)C(S_{act})$, por lo tanto $C(S_{cand}) - C(S_{act}) = \mu C(S_{act})$, y tenemos luego:

$$\phi = e^{-\delta/T_0} = e^{(-\mu/T_0)C(S_{act})} \qquad \Rightarrow \qquad T_0 = \frac{\mu}{-\ln(\phi)} C(S_{act})$$

Para nuestro problema hemos considerado razonable dentro de la lógica aceptar un $\phi = 13\%$ de las veces (en la primera iteración) una solución que sea un $\mu = 1\%$ peor que la actual, tendríamos entonces un:

$$T_0 = \frac{0.01}{-\ln(0.13)} C(S_{act}) \approx 0.005 C(S_{act})$$

3.6.3 Velocidad de Enfriamiento

$L(T)$ debería ser lo suficientemente grande para que el sistema llegue a alcanzar su estado estacionario para esa temperatura T, lo cual significaría un número de iteraciones al menos igual a $card(\Omega)^2$ [Eglese, 1990] . Como esto no es factible, al definir el algoritmo hay que lograr un equilibrio entre los parámetros α y L, ya que si se realiza una búsqueda durante pocas iteraciones para cada temperatura, son recomendables rápidos decrecimientos de temperatura y viceversa.

Lundy [1986], realiza una sola iteración para cada temperatura, es decir, $L(T)=1\ \forall T$. Define un límite superior $U=\max\{C(S_j)-C(S)\}\quad \forall S_j,\ S\in\Omega$, y para cada iteración considerar un cambio de temperatura $\alpha(T)=\dfrac{T}{1+\beta T}$, para un $\beta<U^{-1}$, de este modo se evitan grandes desviaciones respecto al equilibrio en cada paso. Se realiza la elección de T_0 de tal forma que en las primeras iteraciones se aceptan la mayoría de los movimientos, tomando un $T_0>U$, con lo cual se consigue que el factor de Boltzmann sea próximo a uno.

Otro investigador Connolly [1990] para un problema de asignación considera $\alpha T \approx T$, es evidente que la pregunta es cuál es la T más adecuada para éste problema concreto.

La práctica más común y la recomendada como primera aproximación a un nuevo problema [Dowsland, 1993] consiste en considerar una velocidad geométrica de decrecimiento, es decir, $\alpha(T) = kT, 0 < k < 1$. Se ha visto que los valores más convenientes para el parámetro k son los correspondientes al intervalo $[0.8,0.99]$, el cuál es un conjunto de valores que nos aseguran un enfriamiento lento, [Kuik, Salomon, 1990]. Con el propósito de que se haga una exploración más exhaustiva cuando la temperatura es baja, puede considerarse un $L(T)$ creciente según decrece T.

Es claro que si esto no da buenos resultados debe seleccionarse un programa de enfriamiento específico para cada problema particular.

Otros autores sugieren el uso de un recalentamiento cuando la búsqueda según el enfriamiento sigue sin dar frutos [Dowsland, 1993]

3.6.4 Temperatura Final, T_F

En teoría la temperatura que corresponde al sistema frío debe de ser 0. Sin embargo, bastante antes de llegar a este valor es prácticamente improbable que se acepte un movimiento hacia una solución peor, entonces se puede finalizar con valores $T_f > 0$, sin pérdida en la calidad de la solución.

Según el criterio de Lundy-Mees, el cual consiste en finalizar en cuanto la temperatura sobrepase un umbral mínimo. Estos investigadores han demostrado que si se considera que es θ de que finalmente se obtenga una solución para la cual su costo menos el de la óptima global sea cuando mucho ε, se cumple que:

$$\frac{\theta}{1-\theta} > (n-1)e^{-\frac{\varepsilon}{T}}$$

siendo $n = card(\Omega)$ el número de elementos del espacio de soluciones. Despejando la temperatura, obtenemos que el valor límite T_f es:

$$T_f < -\frac{\varepsilon}{\ln(n-1) - \ln(\theta) + \ln(1-\theta)} \approx \frac{\varepsilon}{\ln(n)}$$