

Capítulo 6

Recocido Simulado

6.1 Introducción

El resolver un problema combinatorio se resume a encontrar la “mejor” / “óptima” solución dentro de un conjunto finito o contablemente infinito de alternativas.

Asumimos que la calidad de la solución es cuantificable y comparable con cualquier otra solución y asumimos que las soluciones son finitas.

Muchos problemas combinatorios son NP, por lo que normalmente se buscan soluciones “rápidas” sub-óptimas.

Una instancia de un problema combinatorio se puede formalizar por un par (S, f) donde el *espacio solución* S denota el conjunto finito de todas las posibles soluciones y f denota la *función de costo*:

$$f : S \rightarrow \mathcal{R}$$

Lo que se trata (en minimización) es encontrar una solución $i_{opt} \in S$ tal que $f(i_{opt}) \leq f(i)$ para toda $i \in S$.

Los algoritmos de búsqueda local constituyen una clase de algoritmos aproximados basados en la exploración de vecinos. Estos presuponen que existe

una función de costo y una estructura de vecindad.

La estructura de vecindad define para cada $i \in S$ un conjunto $S_i \subset S$ que son cercanos a i en algún sentido. Se asume que $j \in S_i \Leftrightarrow i \in S_j$.

El *mecanismo de generación* se define como la forma de seleccionar una solución j en la vecindad S_i de la solución i .

Los algoritmos de búsqueda local normalmente iteran partiendo de una solución aleatoria, generando un vecino que sea mejor hasta que se llegue a una solución que no tiene mejores vecinos.

Ya hemos visto varias alternativas que se han propuesto para tratar de evitar quedarse en mínimos locales:

- Ejecutar el algoritmo de búsqueda local a partir de muchos puntos iniciales.
- Introducir restricciones adicionales a la función objetivo.
- Introducir una estructura de vecinos más compleja para buscar en regiones más grandes del espacio.
- Alterar soluciones buscando diversidad.
- Introducir mecanismos de memoria para evitar regresar a estados visitados.

Recocido simulado se basa en aceptar en forma limitada transiciones que no mejoren la función de costo usando un mecanismo no determinista.

Recocido simulado se llama así por su analogía con el proceso físico de recocido de sólidos, en el cual:

- Se incrementa la temperatura de un sólido cristalino a una temperatura alta.
- Se decrementa la temperatura muy lentamente hasta alcanzar un estado base de mínima energía, con una estructura cristalina lo más regular posible.

Tabla 6.1: Algoritmo de Metropolis.

Dado un estado i con energía E_i ,
genera un nuevo estado j mediante un mecanismo de perturbación
(pequeña distorsión del estado i).
calcula la energía del nuevo estado E_j .
si $(E_j - E_i) \leq 0$
entonces acepta el estado j como estado nuevo
sino, acepta el estado con probabilidad: $\exp(\frac{E_i - E_j}{k_B \cdot T})$

Lo que hace recocido simulado es establecer una conexión entre este tipo de proceso termodinámico y la búsqueda de un mínimo global.

Otro proceso relacionado es el de *quenching* en el cual la temperatura se baja inmediatamente.

El núcleo de recocido simulado lo constituye lo que se conoce como el algoritmo de Metropolis.

6.2 Algoritmo de Metropolis

La idea principal puede verse en la tabla 6.1 (Metropolis et al. 1953) donde T denota la temperatura y k_B la constante de Boltzmann.

Lo que hace el algoritmo de Metropolis es generar un vecino, calcularle su energía (i.e., función de costo en problemas de optimización) y aceptar ese vecino si tiene menor energía o aceptarlo con mayor energía pero con cierta probabilidad que depende de la temperatura (T).

Se puede probar que si se realiza este proceso durante muchas transiciones se puede llegar a lo que se conoce como un equilibrio térmico.

El equilibrio térmico está caracterizado por la distribución de Boltzmann.

La distribución da la probabilidad de que el sólido esté en el estado i con

energía E_i a temperatura T y está dada por:

$$P_T\{X = i\} = \frac{1}{Z(T)} \exp\left(\frac{-E_i}{k_B T}\right)$$

donde X es la variable estocástica denotando el estado actual del sólido, y $Z(T)$ es la función de partición definida como:

$$Z(T) = \sum_j \exp\left(\frac{-E_j}{k_B T}\right)$$

6.3 Recocido Simulado

La analogía es la siguiente:

- Las soluciones corresponden a estados del sistema físico.
- El costo de la solución corresponde a la energía del estado.
- Se introduce un parámetro de control que corresponde a la temperatura.

El algoritmo de recocido simulado se puede ver como una iteración de algoritmos de Metrópolis (ver tabla 6.2).

Si se baja la temperatura suficientemente lento se puede alcanzar el equilibrio térmico en cada temperatura. Esto se hace mediante la generación de varias transiciones en cada temperatura.

Se puede demostrar que bajando suficientemente lento el parámetro asociado a la temperatura y generando suficientes transiciones en cada temperatura se puede alcanzar la configuración óptima.

La probabilidad de aceptación se expresa como sigue:

$$P_c\{\text{acepta } j\} = \begin{cases} 1 & \text{si } f(j) \leq f(i) \\ \exp\left(\frac{f(i)-f(j)}{c}\right) & \text{si } f(j) > f(i) \end{cases}$$

Tabla 6.2: Algoritmo de Recocido Simulado.

```

k := 0, i := iinic
repite
  for l := 1 to Lk do
    genera j ∈ Sj
    si  $f(j) \leq f(i)$  entonces i := j
    sino, si  $\exp(\frac{f(i)-f(j)}{c_k}) > \text{random}[0,1)$ 
      entonces i := j
  k := k + 1
  actualiza Lk y ck
hasta criterio de terminación
  
```

donde $c \in \mathcal{R}^+$ denota el parámetro de control.

Una transición consiste en (i) aplicación del mecanismo de generación, y (ii) aplicación del mecanismo de aceptación.

Inicialmente valores grandes de c aceptan cualquier estado. Al tender c a cero, se dejan de aceptar estados.

La velocidad de convergencia del algoritmo está determinada por L_k y c_k .

Búsqueda local se puede ver como *quenching*.

Un sistema físico de muchas partículas puede verse como un “ensamble” estadístico.

Si el ensamble es estacionario, el cual se logra en equilibrio térmico, su densidad es función de la energía del sistema y la probabilidad de estar en un estado i con energía E_i está dado por la distribución de Boltzmann.

Dada una instancia de un problema combinatorio y una estructura de vecinos adecuada, después de un número de transiciones suficientemente largo para un valor fijo de c , aplicando el criterio de aceptación, el algoritmo de recocido simulado encuentra una solución $i \in S$ con probabilidad igual a:

$$P_c\{X = i\} \equiv q_i(c) = \frac{1}{N_0(c)} \exp\left(-\frac{f(i)}{c}\right)$$

donde X es una variable estocástica denotando la solución actual y

$$N_0(c) = \sum_{j \in S} \exp\left(-\frac{f(j)}{c}\right)$$

denota una constante de normalización.

Dada una instancia de un problema de optimización combinatoria, una estructura de vecindad adecuada y una distribución estacionaria equivalente a la distribución de Boltzmann, entonces:

$$\lim_{c \rightarrow 0} q_i(c) \equiv q_i^* = \frac{1}{|\mathcal{S}_{opt}|} \chi_{(\mathcal{S}_{opt})}(i)$$

donde \mathcal{S}_{opt} denota el conjunto de soluciones óptimas globales, y $\chi_{A'}(a)$ es la función característica definida como:

$$\chi_{(A')}(a) = \begin{cases} 1 & \text{si } a \in A' \\ 0 & \text{de otra forma} \end{cases}$$

El resultado es importante porque garantiza una convergencia asintótica hacia el conjunto de soluciones óptimas del algoritmo de recocido simulado bajo la condición que la distribución estacionaria se obtenga para cada valor de c .

La probabilidad de encontrar la solución óptima incrementa monótonicamente al decrementar c .

6.4 Convergencia Asintótica y Cadenas de Markov

El algoritmo de recocido simulado puede ser modelado matemáticamente usando cadenas de Markov.

Una cadena de Markov es una secuencia de eventos, donde la probabilidad del resultado de un evento depende sólo de los resultados del evento anterior.

Sea $X(k)$ una variable estocástica que denota el resultado del k -ésimo evento. Entonces la transición de probabilidad en el k -ésimo evento para cada par i, j de resultados se define como:

$$P_{ij}(k) = \mathbf{P}\{X(k) = j \mid X(k-1) = i\}$$

La matriz $P(x)$ cuyos elementos están dados por la fórmula de arriba se llama la *matriz de transición*.

Denotemos a $a_i(k)$ como la probabilidad del resultado i en el k -ésimo evento, o sea: $a_i(k) = \mathbf{P}\{X(k) = i\}$.

Entonces $a_i(k)$ se define como:

$$a_i(k) = \sum_l a_l(k-1)P_{li}(k)$$

Mencionemos algunas propiedades de cadenas de Markov. Una cadena de Markov se dice:

- *Finita* si tiene un conjunto finito de resultados.
- *No homogénea* si las probabilidades de transición dependen del número del evento k .
- *Homogénea* si las probabilidades de transición son independientes del número del evento.
- *Irreducible* si para cada par de soluciones $i, j \in S$ existe una probabilidad positiva de alcanzar j a partir de i en un número finito de eventos.

En recocido simulado, un evento corresponde a una transición y los resultados posibles corresponden al conjunto de soluciones posibles.

Las probabilidades de transición del algoritmo de recocido simulado están definidas como:

$$\forall i, j \in S : P_{ij}(k) = P_{ij}(c_k) = \begin{cases} G_{ij}(c_k)A_{ij}(c_k) & \text{si } i \neq j \\ 1 - \sum_{l \in S, l \neq i} P_{il}(c_k) & \text{si } i = j \end{cases}$$

donde:

$G_{ij}(c_k)$ denota la probabilidad de generación (la probabilidad de generar una solución j a partir de una solución i).

$$\forall i, j \in S : G_{ij}(c_k) = G_{ij} = \frac{1}{\Theta} \chi_{(S_i)}(j)$$

donde $\Theta = |S_i|$, para toda $i \in S$.

$A_{ij}(c_k)$ denota la probabilidad de aceptación (la probabilidad de aceptar la solución j una vez que fué generada a partir la solución i).

$$\forall i, j \in S : A_{ij}(c_k) = \exp\left(-\frac{(f(j) - f(i))^+}{c_k}\right)$$

donde, $\forall a \in \mathcal{R}, a^+ = a$ si $a > 0$ y $a^+ = 0$ de otra forma.

Lo que queremos probar es que, bajo ciertas condiciones, el algoritmo de recocido simulado converge asintóticamente al conjunto de soluciones óptimas, i.e.,:

$$\lim_{k \rightarrow \infty} P\{X(k) \in S_{opt}\} = 1$$

Una parte esencial para la prueba de convergencia es la existencia de una distribución estacionaria única. Tal distribución existe sólo si se cumplen ciertas condiciones en la cadena de Markov asociada al algoritmo.

Una distribución estacionaria de una cadena de Markov finita con matriz de transición P se define como un vector \mathbf{q} cuyo i -ésimo componente está dado por:

$$q_i = \lim_{k \rightarrow \infty} \mathbf{P}\{X(k) = i \mid X(0) = j\}, \text{ para toda } j$$

Si existe esa distribución estacionaria, entonces,

$$\begin{aligned} \lim_{k \rightarrow \infty} a_i(k) &= \lim_{k \rightarrow \infty} P(X(k) = i) \\ &= \lim_{k \rightarrow \infty} \sum_j P(X(k) = i | X(0) = j) P(X(0) = j) \\ &= q_i \sum_j P(X(0) = j) = q_i \end{aligned}$$

Teniamos que:

$$\lim_{c \rightarrow 0} q_i(c) \equiv q_i^* = \frac{1}{|S_{opt}|} \chi_{(S_{opt})}(i)$$

Por lo que:

$$\begin{aligned} \lim_{c \rightarrow 0} \lim_{k \rightarrow \infty} P(X(k) = i) &= \lim_{c \rightarrow 0} q_i(c) = q_i^* \\ \lim_{c \rightarrow 0} \lim_{k \rightarrow \infty} P(X(k) \in S_{opt}) &= 1 \end{aligned}$$

Aunque no es propiamente una prueba, lo de arriba indica algunas de las ideas principales que se usan en la prueba de convergencia.

Todas las pruebas usan la condición de reversibilidad o también llamada de balance detallado. Lo que dice es que si tenemos una matriz de transición P asociada a una cadena de Markov finita, irreducible, aperiódica y homogénea. Entonces una distribución es estacionaria para la cadena de Markov si satisface:

$$q_i P_{ij} = q_j P_{ji}, \text{ para toda } i, j \in S$$

Las pruebas de convergencia nos dicen que el algoritmo de recocido simulado requiere un número infinito de transiciones para aproximarse lo suficiente a una distribución estacionaria en cada temperatura.

Esto involucraría la generación de secuencias infinitas para valores descendientes del parámetro de control c .

Sin embargo, podemos formular el algoritmo de recocido simulado como una secuencia de cadenas de Markov de longitud finita que converge al conjunto de soluciones óptimas si el enfriamiento se hace lo suficientemente lento.

El proceso se puede describir mediante la combinación de cadenas de Markov homogéneas en una sola cadena de Markov no-homogénea. Dicho de otra

forma, la secuencia de cadenas de Markov homogenas infinitamente largas se reducen a una sola cadena de Markov no homogenas infinita.

6.5 Aproximaciones

Como en la práctica no se puede garantizar llegar a la solución óptima se hacen aproximaciones con longitud de transiciones finitas y número de descensos del parámetro de control finito, que arrojan soluciones sub-óptimas.

Se requiere definir un mecanismo de enfriamiento que especifique:

- Una secuencia finita de valores para el parámetro de control: (i) un valor inicial c_0 , (ii) una función de decremento, y (iii) un valor final (condición de paro).
- Un número finito de transiciones para cada valor del parámetro de control (longitud finita de cada cadena de Markov).

Una idea clave en las aproximaciones es llegar a un cuasi-equilibrio (esto es, si la distribución de probabilidad de las soluciones después de un número finito de eventos es “suficientemente cerca” con la distribución estacionaria).

Existe un balance entre la longitud de las cadenas de Markov y los decrementos realizados en el parámetro de control.

Decrementos largos en c_k requieren muchas transiciones para restablecer el cuasi-equilibrio y viceversa.

Mecanismo de enfriamiento propuesto por Kirkpatrick:

- Valor inicial del parámetro de control: empezar con un entero positivo pequeño e irlo multiplicando por un factor mayor a 1 hasta que las transiciones generadas sean casi todas aceptadas.
- Decremento del parámetro de control: $c_{k+1} = \alpha \cdot c_k$, donde α es una constante cercana a 1 (0.8 – 0.99).

- Valor final del parámetro de control: terminar cuando la solución obtenida permanece igual en un número determinado de cadenas consecutivas.
- Longitud de las cadenas de Markov: hacer una longitud (L) fija (de otra forma $L \rightarrow \infty$ cuando $c_k \rightarrow 0$).

Se han propuesto varios esquemas de enfriamiento en la literatura.

6.6 Implementación

Para aplicar el algoritmo se requieren especificar 3 componentes:

- La representación del problema:
 - Representar el espacio de solución.
 - Expresar la función de costo que represente adecuadamente el costo de las soluciones.
- El mecanismo de transición:
 - Generar una nueva solución (simétrica).
 - Calcular diferencia de costo (a veces se calcula tomando en cuenta las diferencias con la solución anterior).
 - Tomar decisión de aceptación

$$P(\text{aceptación}) = \begin{cases} 1 & \text{si } \Delta f < 0 \\ \exp(-\frac{\Delta f}{c}) & \text{si } \Delta f \geq 0 \end{cases}$$

- El mecanismo de enfriamiento:
 - Valor inicial de c .
 - Función de decremento.
 - Criterio de paro.
 - Longitud de las cadenas.

Experiencia:

- El algoritmo es simple y fácil de implementar.
- Aplicable a una gran cantidad de problemas.
- Su adaptación no es siempre trivial y a veces hay que reformular el problema.
- Su eficiencia depende del esfuerzo de la implementación (e.g., estructura de vecindad y mecanismo de enfriamiento).

6.7 Paralelización

La parte que consume más recursos es la generación de la secuencia de eventos que consiste en 4 partes:

1. Seleccionar una nueva solución.
2. Calcular las diferencias de costo.
3. Decidir si se acepta.
4. Reemplazar la solución nueva.

Los 3 primeros son independientes. El número de veces que se hace el 4º varía durante la ejecución del algoritmo (de muchas veces con c_k grande a prácticamente 0 con c_k pequeña).

Una idea es dividir el esfuerzo de generar la cadena de Markov en varios procesadores.

Posibilidades:

- Cada procesador genera su propia cadena de Markov independientemente y se escoge la mejor de todas al final.

- Cada procesador genera pequeñas subcadenas y la mejor solución se comunica a los demás cada determinado tiempo.
- Cada procesador genera una muestra nueva y se escoge la mejor de todas.
- Cada procesador está asociado con una vecindad de estados y calcula el nuevo estado independientemente de los demás manteniendo comunicación para que exista compatibilidad en las fronteras.
- Cada estado del espacio de búsqueda está asociado con un procesador y se actualiza su nuevo estado independientemente de los demás (autómata celular).

6.8 Variantes

Se han propuesto varias variantes al algoritmo de recocido simulado. Una de las más conocidos es *threshold accepting* propuesta independientemente por Dueck y Scheuer (1990) y por Moscato y Fontanari (1990).

Threshold accepting o aceptación por umbral, propone aceptar un movimiento siempre y cuando rebasa un cierto umbral ($Thres_k$) determinístico y decreciente con respecto a la iteración k .

$$P(\text{aceptación}) = \begin{cases} 1 & \text{si } \Delta f < Thres_k \\ 0 & \text{de otra forma} \end{cases}$$

Aunque mucho más rápido, se pierden las propiedades de convergencia hacia el mínimo global.

Una variante de esto es cambiar el valor del umbral al hacer la búsqueda. Si se dejan de aceptar soluciones en el esquema de vecindad, entonces se aumenta el umbral, en caso contrario se reduce, buscando escapar de mínimos locales (*dwindling expectation*).

En general, en todos los algoritmos que hemos visto es donde se tiene un criterio de aceptación basado en la mejora sobre la función objetivo, se puede

reemplazar por un criterio basado en recocido simulado, que permita aceptar peores soluciones al principio de la búsqueda y gradualmente quedarse solo con mejoras.

Algunos consejos de implementación incluyen:

- En lugar de calcular la diferencia de funciones hacer una aproximación a esa diferencia.
- Seleccionar vecinos en forma inteligente (no aleatoria).
- Usar otra distribución de probabilidad (e.g., decaimiento exponencial).
- Utilizar un esquema de enfriamiento adecuado.
- Pensar cuidadosamente en el esquema de vecindad.