UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

Simulated Annealing

Alan Reyes-Figueroa

Métodos Numéricos II

(Aula 36) 07.noviembre.2024

Simulated Annealing

El **recocido simulado** o **enfriamiento simulado** (*simulated annealing*) es un algoritmo estocástico de optimización de búsqueda global.

Utiliza la aleatoriedad como parte del proceso de búsqueda. Esto hace que el algoritmo sea apropiado para funciones objetivas no lineales donde otros algoritmos de búsqueda local no funcionan bien.

Al igual que el algoritmo de búsqueda local estocástico de *hill-climbing*, modifica una única solución y busca en el área relativamente local del espacio de búsqueda hasta que se localiza el óptimo local. A diferencia del algoritmo de escalada, puede aceptar peores soluciones como la solución de trabajo actual.

La probabilidad de aceptar peores soluciones comienza alta al comienzo de la búsqueda y disminuye con el progreso de la búsqueda, lo que le da al algoritmo la oportunidad de ubicar primero la región para los óptimos globales, escapando de los óptimos locales y luego escalar la colina hasta el mismo óptimo.

Simulated Annealing

El algoritmo está inspirado en los procesos de enfriamiento en metalurgia, donde el metal se calienta rápidamente a una temperatura alta y luego se enfría lentamente, lo que aumenta su resistencia y facilita el trabajo.

Simulated Annealing

El material se somete a una temperatura alta, permitiendo que los átomos se muevan mucho, luego disminuyendo su excitación lentamente, permitiendo que los átomos caigan en una nueva configuración más estable.

Cuando está caliente, los átomos del material tienen más libertad para moverse y, mediante un movimiento aleatorio, tienden a asentarse en mejores posiciones. Un enfriamiento lento lleva el material a un estado cristalino ordenado.

El enfriamiento simulado puede considerarse como una versión modificada de la escalada estocástica (*hill-climbing*).

hill-climbing mantiene una única solución candidata *curr* y toma pasos de tamaño aleatorio pero limitado, alrededor de *curr*, en el espacio de búsqueda. Si el nuevo punto es mejor que el punto actual, entonces el punto actual se reemplaza por el nuevo. Este proceso continúa durante un número fijo de iteraciones.

El recocido simulado ejecuta la búsqueda de la misma manera. La principal diferencia es que a veces se aceptan puntos nuevos peores que el punto actual.

Simulated Annealing

Un punto peor se acepta de forma probabilística, donde la probabilidad de aceptar una solución peor que la solución actual es una función de la temperatura de la búsqueda y cuánto peor es la solución que la solución actual.

Típicamente, el método usa analogías de la mecánica estadística. Esta analogía es la siguiente ecuación de aceptación de una solución pero que la actual:

$$\mathbb{P}(\text{aceptación pero solución}) = \exp\left(-\frac{\Delta E}{kt}\right), \tag{1}$$

donde

- ΔE es una función que mide la diferencia de "energía": $\Delta E = f(next) - f(current)$.
- $t = t(i)$ es una función de temperatura. Esta función comúnmente depende del número de iteración, y $t(i)$ describe un itinerario o *schedule* para programar la cómo decae la probabilidad de aceptaciones de soluciones malas.
- $k >$ 0 es una constante, llamada la *constante de* Boltzman.

La distribución de probabilidad en (1), se conoce en física como una distribución de boltzman. (1) se llama el criterio de Metropolis.

Simulated Annealing

El Algoritmo:

- Paso 1: Primero comenzamos con una solución inicial $curr = \mathbf{x}_0$. Puede ser cualquier solución que se ajuste a los criterios de una solución aceptable (factible). También comenzamos con una temperatura inicial $t = t(0) = T_0$.
- Paso 2: Configure una función de reducción de temperatura alfa. Por lo general, se utiliza alguno de los siguientes tipos principales de reglas de reducción de temperatura:

$$\begin{aligned} \text{lineal}: &\quad t(i+1) = t(i) - \alpha,\ 0 < \alpha; \\ \text{geométrica}: &\quad t(i+1) = \alpha\, t(i),\ 0 < \alpha < 1; \\ \textit{slow} &\quad t(i+1) = \frac{1}{1+\beta\, t(i)},\ \beta > 0; \\ \textit{fast} &\quad t(i+1) = \frac{T_0}{i+1}. \end{aligned}$$

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

Simulated Annealing

Cada regla de reducción reduce la temperatura a un ritmo diferente y cada método es mejor para optimizar un tipo diferente de modelo.

La temperatura inicial T_0, (junto con las constantes α, β) para la búsqueda se proporcionan como hiperparámetros. T_0 debe disminuir con el progreso de la búsqueda. Se pueden usar varios esquemas diferentes (programas de enfriamiento) para disminuir la temperatura durante la búsqueda desde el valor inicial a un valor muy bajo. Es común calcular la temperatura en función del número de iteración.

- Paso 3: comenzando con T_0, recorra n iteraciones del paso 4 y luego disminuya la temperatura de acuerdo con el Paso 2. Este ciclo se detiene hasta cumplir alguna condición de paro. Las condiciones de paro pueden incluir:
  - número de iteraciones máximo,
  - tiempo de ejecución máximo,
  - alcanzar cierta temperatura final T_f,
  - alcanzar un umbral aceptable de rendimiento para un conjunto de parámetros

Simulated Annealing

function SIMULATED-ANNEALING(*problem, schedule*) **return** a solution state

input: *problem*, a problem

schedule, a mapping from time to temperature

local variables: *current*, a node.

next, a node.

T, a "temperature" controlling the probability of downward steps

current ← MAKE-NODE(INITIAL-STATE[*problem*])

for t ← 1 to ∞ do

T ← *schedule*[*t*]

if *T* = *0* **then return** *current*

next ← a randomly selected successor of *current*

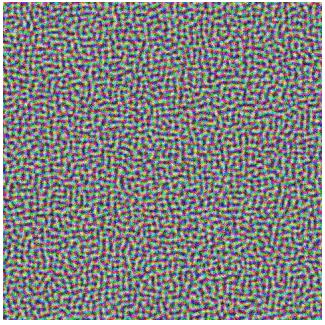
$\Delta E \leftarrow$ VALUE[*next*] - VALUE[*current*]

if $\Delta E > 0$ **then** *current* ← *next*

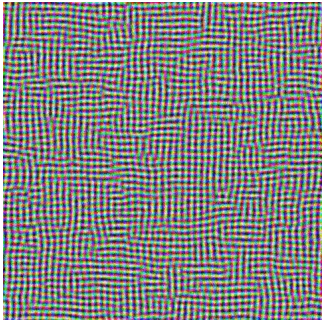
else *current* ← *next* only with probability $e^{\Delta E / T}$

Simulated Annealing

Importante: el itinerario de reducción de temperatura (o tiempo del algoritmos) es importantísimo. Muchas veces, decaimientos rápidos conducen a soluciones no aceptables. Entre mejor calidad de las soluciones deseadas, mayor tiempo de búsqueda (balance entre tiempo de ejecución y calidad).



(a) Enfriamiento rápido.



(b) Enfriamiento lento.

Simulated Annealing

Algunos ejemplos de problemas para optimizar con SA:

- Problema del vendedor ambulante (TSP)
- Problemas de scheduling
- Asignación de tareas
- Coloración y partición de grafos
- Optimización de funciones no lineales

Ventajas:

- Fácil de implementar y usar
- Proporciona soluciones óptimas a una amplia gama de problemas.

Desventajas:

- Puede tardar mucho en ejecutarse si el programa de recocido es muy largo.
- Hay muchos parámetros ajustables en este algoritmo.