
INSTITUTO POLITÉCNICO NACIONAL

Centro de Investigación en Computación



ASIGNATURA:

Metaheurísticas

Actividad #13:

Guía Clase Práctica No.2

"Solución de problemas mediante Recocido Simulado"

PROFESORA:

Dra. Yenny Villuendas Rey

PRESENTA:

Juan René Hernández Sánchez

Adriana Montserrat García Carrillo



Centro de Investigación
en Computación
Instituto Politécnico Nacional

1. Introducción

Muchos problemas de ingeniería, planificación y fabricación pueden ser modelados como minimizar o maximizar una función de coste sobre un conjunto finito de variables discretas. Esta clase de problemas, llamados de optimización combinatoria, ha recibido mucha atención en las dos últimas décadas y se han conseguido importantes logros en su análisis. Uno de estos logros es la separación de esta clase en dos subclases. La primera contiene los problemas que pueden resolverse de forma eficiente, es decir, los problemas para los que se conocen algoritmos que resuelven cada instancia de forma óptima en tiempo polinómico. La segunda subclase contiene los problemas que son notoriamente difíciles, denominados formalmente NP-duros.

Para un algoritmo NP-duro se cree que no existe ningún algoritmo que resuelva cada instancia en tiempo polinómico. En consecuencia, hay instancias que requieren un tiempo súper polinomial o exponencial para ser resueltas de forma óptima. Está claro que los problemas difíciles deben tratarse en la práctica, lo cual puede hacerse mediante dos tipos de algoritmos: de optimización, que encuentran soluciones óptimas posiblemente utilizando grandes cantidades de tiempo de cómputo, o algoritmos heurísticos que encuentran soluciones aproximadas en pequeñas cantidades de tiempo de cálculo.

Los algoritmos de búsqueda local son del tipo heurísticos. Constituyen un enfoque general muy utilizado para los problemas de optimización combinatoria. Suelen ser instancias de varios esquemas de búsqueda general, pero todas tienen la misma característica de una función de vecindad subyacente, que se utiliza para guiar la búsqueda de una buena solución.

El recocido simulado, es uno de los algoritmos de búsqueda local más conocidos, ya que tiene un rendimiento bastante bueno y es ampliamente aplicable.

En la física de la materia condensada, el recocido simulado se conoce como un proceso térmico para obtener estados de baja energía de un sólido en un baño de calor. El proceso consiste en los siguientes dos pasos [1]:

- aumentar la temperatura del baño de calor hasta un valor máximo en el que el sólido se funde
- disminuir cuidadosamente la temperatura del baño de calor hasta que las partículas se organicen en el estado básico del sólido.

A continuación, se presentará una explicación del recocido simulado, sus características, aplicaciones e implementación

2. Desarrollo

Asignatura: Metaheurísticas

Actividad No.13

Guía Clase Práctica No.2

Título: Solución de problemas mediante Recocido Simulado

Contenido:

- Métodos heurísticos de solución de problemas.
- Recocido Simulado.

Objetivo: Modelar problemas clásicos de búsqueda mediante el uso de algoritmos de Recocido Simulado, para la solución de problemas de la profesión.

1. Enuncie las ventajas y desventajas del Recocido Simulado (SA) [2]

Ventajas:

- El recocido simulado puede tratar con modelos altamente no lineales, datos ruidosos y muchas restricciones.
- Es una técnica robusta y general.
- Sus principales ventajas sobre otros métodos de búsqueda local son su flexibilidad, su capacidad para acercarse a la optimización global y su capacidad de escapar de los mínimos locales.
- El algoritmo es bastante versátil, ya que no depende de ninguna propiedad restrictiva del modelo.

Desventajas:

- Existe una relación entre la calidad de las soluciones y el tiempo necesario para calcularlas.
- La precisión de los números utilizados en la implementación del SA puede tener un efecto significativo sobre la calidad del resultado.
- El trabajo de adaptación necesario para tener en cuenta las diferentes clases de restricciones y para afinar los parámetros del algoritmo puede ser bastante delicado.

2. Detalle el pseudocódigo del algoritmo de Recocido Simulado [1]

```
x = solución inicial aleatoria
edo_anterior = costo(x)
for temp = tempMax, temp ≥ tempMin, temp=next_temp(temp)
  repeat
    sucesor = función_vecindad(x)
    edo_nuevo = costo(sucesor)
    delta = edo_nuevo – edo_anterior
    if delta > 0 then
      if random() ≥ exp(-delta/K*temp) then
        rechazar(x)
      else
        edo_anterior = edo_nuevo
        x = sucesor
    else
      edo_anterior = edo_nuevo
      x = sucesor
    iteracion ++
  until iteracion < iteracionMax
end for
```

3. Compárelo con el algoritmo de RMHC

Algorithm 1 Heurística Random mutation hill-climbing (RMHC)

Input: Un Estado Aleatorio del problema

Output: Un óptimo local

```
1: iterations ← S                                ▷ Siendo S ∈ ℕ Un número determinado de Evaluaciones
2: bestEvaluated ← random string
3: bestFitness ← ComputeFitness(bestEvaluated)
4: length ← bestEvaluated.length()
5: repeat
6:   locus ← Rand(0, length)
7:   mutatedHilltop ← Mutate(bestEvaluated, locus)
8:   mutatedFitness ← ComputeFitness(mutatedHilltop)
9:   if mutatedFitness ≥ bestFitness then
10:    bestEvaluated := mutatedHilltop
11:    bestFitness := mutatedFitness
12:   end if
13:   iterations := iterations – 1
14: until iterations ≠ 0
15: return bestEvaluated
```

El SA posee mayor flexibilidad en cuanto a las soluciones que va evaluando, a diferencia del RMHC que siempre conservaba la mejor solución encontrada. Por lo tanto, el SA puede caer en errores o peores soluciones o lograr salir de óptimos locales.

Otra diferencia es que los ciclos del RMHC dependen del número de evaluaciones determinadas, mientras que en el SA se realiza un proceso completo, que se cumplirá en su totalidad hasta alcanzar cierta temperatura mínima.

4. Mencione aplicaciones recientes (últimos 3 años) de los algoritmos de Recocido Simulado

1. La aplicación del método de recocido simulado para la detección óptima de rutas entre objetos: En este documento se demuestra la aplicación del método de recocido simulado y el problema del agente viajero y se realiza un experimento destinado a encontrar la ruta más corta entre las empresas de procesamiento de Bielorrusia. [3]
2. Aplicación del algoritmo de recocido simulado al modelo de radiación WMP correlacionado para las llamas: Se han propuesto varias alternativas para modelar el perfil de radiación emitido por las llamas. Entre ellas se encuentra el modelo de fuente multipunto ponderado (WMP), que es capaz de proporcionar buenas predicciones para distancias medias y lejanas. A pesar de ser un modelo sencillo, la determinación de sus parámetros puede ser difícil, lo que crea la oportunidad de aplicar métodos de optimización para encontrar la mejor respuesta. El objetivo del estudio es profundizar en el modelo WMP y aplicar el método de optimización "Recocido Simulado" para evaluar si se pueden mejorar las respuestas obtenidas con otros algoritmos. [4]
3. Recocido simulado aplicado al escalonamiento de los telescopios: Una aplicación interdisciplinar de la Física Estadística es el mapeo de las transiciones de desbaste (y su solución mediante Metrópolis Monte Carlo/reconocimiento simulado) al problema del desfase de los telescopios. [5]

5. Realice la modelación matemática necesaria para la solución, mediante SA, del Problema de la mochila (Knapsack problem).

Recuerde que la modelación matemática incluye: definición de los estados inicial y final, definición del test objetivo, y definición de las acciones posibles (operadores).

- Estado inicial: n productos con valor C_i y peso P_i que se deben poner en una mochila. Se tiene como límite un peso P_m a cargar por la mochila. $n \in \mathbb{N}$. Los valores $C_i, P_i \in \mathbb{R}_*^+$. El estado de la mochila se representa mediante una cadena binaria x de longitud n . El estado inicial es una cadena x con n 0's.
- Estado final: Una cadena x' tal que $x' = \{x_i | \text{Max}(f(x_i), f(x_i')), 0 \leq i \leq n-1\}$ y donde $f(x) = \sum_{i=1}^n C_i x_i$ de manera que $f(x) \leq f(x')$.
- Test objetivo:
 $f(x) = \sum_{i=1}^n C_i x_i$ s.a. $\sum_{i=1}^n P_i x_i \leq P_m, x_i \in \{0, 1\}$
- Acciones posibles:
 $f : x \mapsto x'$ s.a. $\exists x_i \neq x_i', i \in \{0, \dots, n\}$ donde $x_i = 1$ si el producto se empaqueta o está dentro de la mochila, de lo contrario $x_i = 0$ si el producto no está empaquetado o se sacó de la mochila. Resultando de ahí la condición binaria de la variable x_i .

6. Realice una corrida manual del algoritmo de Recocido Simulado sobre el problema anterior. Defina para ello un esquema de recocido de su preferencia.

Descripción 5 productos:

Productos	Peso	Valor
P1	2	6
P2	1	9
P3	3	2
P4	5	5
P5	1	4

#Definición de parámetros

pesoMax = 10 kg #peso máximo de la mochila

iteracionMax = 3 #número de vecinos

a = 0.5 #valor de alfa

#Se selecciona un estado inicial al azar

X = 10010

#Se calcula la función de costo

edo_anterior = 13

#Se designan las temperaturas

temp = 30° C

tempMax = temp

tempMin = 10° C

#Evaluación vecino 1 de 3

sucesor = 11010

edo_nuevo = 18

#Calculo delta = edo_nuevo – edo_anterior

delta = 5 #Como delta es positivo y se está maximizando entonces se acepta el edo_nuevo (edo_anterior = edo_nuevo)

edo_anterior = 18

x = 11010

#Evaluación vecino 2 de 3

sucesor = 11011

edo_nuevo = 24

delta = 6

edo_anterior = 24

x = 11011

#Evaluación vecino 3 de 3

sucesor = 11111

```
edo_nuevo = 0 #Se asigna el valor de 0, ya que el peso máximo de la mochila se excedió  
delta = -24
```

```
#Se entra al if y se evalúa si es aceptado el estado que empeora  
random = 0.5  
var = exp(-24/30) = 0.44 #exp(delta/K*temp)  
#como 0.5 > 0.44 se rechaza la solución
```

```
#Disminución de la temp  
temp = a * temp = 0.5 * 30 = 15
```

```
#Se vuelven a realizar las evaluaciones
```

```
#Evaluación vecino 1 de 3  
sucesor = 01011  
edo_nuevo = 20  
delta = -4  
random = 0.3  
var = exp(-4/15) = 0.76 #como 0.3 < 0.76 se acepta la solución  
edo_anterior = 20  
x = 01011
```

```
#Evaluación vecino 2 de 3  
sucesor = 00011  
edo_nuevo = 15  
delta = -5  
random = 0.6  
var = exp(-5/15) = 0.71 #como 0.6 < 0.71 se acepta la solución  
edo_anterior = 15  
x = 00011
```

```
#Evaluación vecino 3 de 3  
sucesor = 00111  
edo_nuevo = 17  
delta = 2  
edo_anterior = 17  
x = 00111
```

```
#Disminución de la temp  
temp = a * temp = 0.5 * 15 = 7.5 #Como temp es menor a 10° C se termina el algoritmo
```

3. Conclusiones

Al realizar la corrida manual del algoritmo SA, se pudo apreciar que los algoritmos de búsqueda local pueden ser muy útiles si estamos interesados en el estado de la solución, pero no en el camino hacia ese objetivo. Estos algoritmos operan sólo en el estado actual y se mueven a estados vecinos. Al permitir un ascenso ocasional en el proceso de búsqueda, se puede escapar de la trampa de los mínimos locales, pero también existe la posibilidad de pasar óptimos globales después de alcanzarlos.

Se puede aplicar el SA para generar una solución a los problemas de optimización combinatoria asumiendo una analogía entre ellos y los sistemas físicos de muchas partículas con las siguientes equivalencias:

- Las soluciones del problema son equivalentes a los estados de un sistema físico.
- El costo de una solución es equivalente a la "energía" de un estado

Por último, es importante mencionar que los algoritmos de búsqueda local tienen dos ventajas fundamentales: usan muy poca memoria y pueden encontrar soluciones razonables en espacios de estados grandes o infinitos (continuos).

4. Referencias

- [1] Burke & Kendall. Search Metodologies – 2005. Capítulo 7
- [2] Buseti, F. (2020). Simulated annealing overview. 16/03/2022, de aiinfinance Sitio web: <http://www.aiinfinance.com/saweb.pdf>
- [3] Grabustsa, P., Musatovsa, J., Golenkovb, V. (2019). The application of simulated annealing method for optimal route detection between objects. Elsevier, 149, 95-101.
- [4] Silva B.H. (2020). Application of the simulated annealing algorithm to the correlated WMP radiation model for flames. Inverse Problems in Science and Engineering, 28, 1345-1360.
- [5] Adler, J., Ribak E. (2021). Simulated annealing in application to telescope phasing. Physica A: Statistical Mechanics and its Applications, 572, 125900.