

Optimización y metaheurísticas I

Unidad 4: Metaheurísticas - Optimización Discreta

Dr. Jonás Velasco Álvarez

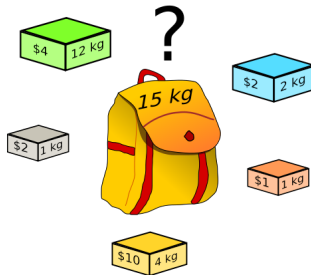
jvelascoa@up.edu.mx

El problema de la mochila

Definición

The knapsack problem. Se tiene un conjunto de n objetos, donde cada objeto j tienen asociado un beneficio p_j y un peso w_j . Por otro lado se tiene una mochila, donde se pueden introducir los objetos, que tiene una capacidad máxima de peso W . Se asume que el peso, el beneficio y la capacidad no son negativos.

El problema consiste en seleccionar un subconjunto de objetos que proporcionen el mayor beneficio, sujeto a la restricción de no exceder la capacidad de la mochila.



Modelo

Variables de decisión

$$x_j = \begin{cases} 1, & \text{si el objeto } j \text{ se introduce en la mochila.} \\ 0, & \text{en otro caso.} \end{cases}$$

donde $j = 1, \dots, n$.

Restricción de capacidad

$$\sum_{j=1}^n w_j x_j \leq W$$

Función objetivo

$$\max z = \sum_{j=1}^n p_j x_j$$

Aplicaciones

Este tipo de problemas pueden representar diversas situaciones en la vida real tales como:

- operaciones de carga
- selección de proyectos de inversión
- control de presupuestos
- corte de materiales
- diseño de métodos criptográficos

Recocido simulado

Input:

$T, \alpha, K, \epsilon :=$ Parámetros del SA

```
1:  $k \leftarrow 0$ 
2: Elegir un vector inicial  $\mathbf{x}_k$  de manera aleatoria o determinista.
3: while  $k \leq K \wedge T > \epsilon$  do
4:   Generar una solución candidata  $\hat{\mathbf{x}}$ 
5:   if  $f(\hat{\mathbf{x}}) > f(\mathbf{x}_k)$  then
6:      $\mathbf{x}_{k+1} \leftarrow \hat{\mathbf{x}}$ 
7:   else
8:      $r \leftarrow \mathcal{U}(0, 1)$ 
9:     if  $r < e^{\left(\frac{f(\mathbf{x}_k) - f(\hat{\mathbf{x}})}{T}\right)}$  then
10:       $\mathbf{x}_{k+1} \leftarrow \hat{\mathbf{x}}$ 
11:     else
12:       $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k$ 
13:     end if
14:   end if
15:    $T \leftarrow \alpha T$ 
16:    $k \leftarrow k + 1$ 
17: end while
```

Recocido simulado

Input:

$T, \alpha, K, \epsilon :=$ Parámetros del SA

```
1:  $k \leftarrow 0$ 
2: Elegir un vector inicial  $\mathbf{x}_k$  de manera aleatoria o determinista.
3: while  $k \leq K \wedge T > \epsilon$  do
4:   Generar una solución candidata  $\hat{\mathbf{x}}$ 
5:   if  $f(\hat{\mathbf{x}}) > f(\mathbf{x}_k)$  then
6:      $\mathbf{x}_{k+1} \leftarrow \hat{\mathbf{x}}$ 
7:   else
8:      $r \leftarrow \mathcal{U}(0, 1)$ 
9:     if  $r < e^{\left(\frac{f(\mathbf{x}_k) - f(\hat{\mathbf{x}})}{T}\right)}$  then
10:       $\mathbf{x}_{k+1} \leftarrow \hat{\mathbf{x}}$ 
11:     else
12:       $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k$ 
13:     end if
14:   end if
15:    $T \leftarrow \alpha T$ 
16:    $k \leftarrow k + 1$ 
17: end while
```

- Una solución candidata $\hat{\mathbf{x}}$ se obtiene por hacer pequeños cambios a una solución \mathbf{x}_k .

- Una solución candidata $\hat{\mathbf{x}}$ se obtiene por hacer pequeños cambios a una solución \mathbf{x}_k .
- **1-flip**. Dado $\mathbf{x}_k = (0, 1, 0, 0)$, las posibles soluciones candidatas $\hat{\mathbf{x}}$ son: $(0, 0, 0, 0)$, $(1, 1, 0, 0)$, $(0, 1, 1, 0)$, $(0, 1, 0, 1)$.

Búsqueda Tabú

Características básicas

- Es un algoritmo de búsqueda local basado en el uso de **memoria**.
- Es posible aceptar soluciones que empeoran la solución actual.
- La memoria se utiliza para no repetir la trayectoria de búsqueda.
- Existen dos tipos de memoria: memoria reciente y memoria a largo plazo.
- En la memoria se guardan atributos de soluciones.

Ideas básicas

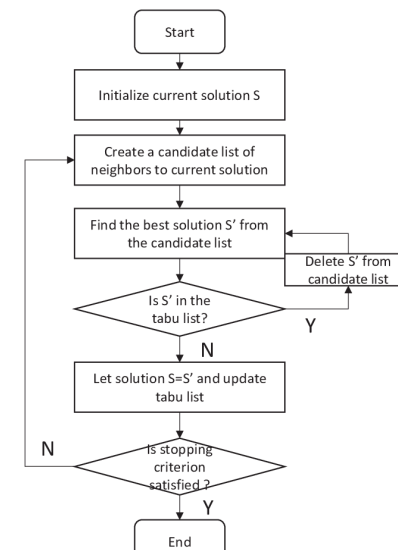
- **Movimiento:** Encontrar el valor de la j -ésima variable activa no tabú, tal que $x_j = 0$ y definir $x_j = 1$, siempre que no se viole la restricción mochila. Si no hay un movimiento disponible de este tipo, busque la j -ésima variable activa no tabu más grande de manera que $x_j = 1$ y establezca $x_j = 0$.
- **Atributo tabú:** Es el índice de la variable cuyo valor se cambió en el movimiento más reciente.
- **Regla de activación tabú:** Una variable tabú activa no puede cambiar su valor por ciertas iteraciones de permanencia tabú.
- **Criterio de aspiración:** Remover la variable tabú activa.

Pseudocódigo

Algorithm 3 (STM First-Improving Tabu Search)

1. $x \leftarrow x^{\text{best}} \leftarrow$ initial solution
2. **do**
3. $f(x^*) = \infty$
4. **for all** $x' \in N^*(x)$
5. **if** $f(x') < f(x^*)$ then $x^* \leftarrow x'$
6. **if** $f(x') < f(x)$ then break
7. **if** $f(x^*) < f(x^{\text{best}})$ then $x^{\text{best}} \leftarrow x^*$
8. $x \leftarrow x^*$
9. update STM
10. **while** termination criterion is not satisfied

Pseudocódigo



Problema de la mochila

Maximize $10x_1 + 14x_2 + 9x_3 + 8x_4 + 7x_5 + 5x_6 + 9x_7 + 3x_8$
subject to $7x_1 + 12x_2 + 8x_3 + 9x_4 + 8x_5 + 6x_6 + 11x_7 + 5x_8 \leq 38,$
 $x_j = \{0,1\}$ for $j = 1, \dots, 8.$

Table 1. Initial solution.

Variable	Profit	Weight	Ratio	Value	Profit	Weight
1	10	7	1.43	1	10	7
2	14	12	1.17	0	0	0
3	9	8	1.13	0	0	0
4	8	9	0.89	1	8	9
5	7	8	0.88	1	7	8
6	5	6	0.83	1	5	6
7	9	11	0.82	0	0	0
8	3	5	0.60	1	3	5
Total	—	—	—	—	33	35

Problema de la mochila

Table 2. Neighborhood of initial solution.

No.	Move	Neighbor	Profit	Weight	Feasible?
1	$x_1 = 0$	(4, 5, 6, 8)	23	28	Yes
2	$x_2 = 1$	(1, 2, 4, 5, 6, 8)	47	47	No
3	$x_3 = 1$	(1, 3, 4, 5, 6, 8)	42	43	No
4	$x_4 = 0$	(1, 5, 6, 8)	25	26	Yes
5	$x_5 = 0$	(1, 4, 6, 8)	26	27	Yes
6	$x_6 = 0$	(1, 4, 5, 8)	28	29	Yes
7	$x_7 = 1$	(1, 4, 5, 6, 7, 8)	42	46	No
8	$x_8 = 0$	(1, 4, 5, 6)	30	30	Yes

Problema de la mochila

Table 3. Iterations of the STM TS with *TabuTenure* = 2.

Iteration	Current solution	Profit	Weight	Tabu active	Move
1	(1, 4, 5, 6, 8)	33	35		8
2	(1, 4, 5, 6)	30	30	8	3
3	(1, 3, 4, 5, 6)	39	38	3 8	6
4	(1, 3, 4, 5)	34	32	6 3	8
5	(1, 3, 4, 5, 8)	37	37	8 6	5
6	(1, 3, 4, 8)	30	29	5 8	6
7	(1, 3, 4, 6, 8)	35	35	6 5	8
8	(1, 3, 4, 6)	32	30	8 6	5
9	(1, 3, 4, 5, 6)	39	38	5 8	6
10	(1, 3, 4, 5)	34	32	6 5	8
11	(1, 3, 4, 5, 8)	37	37	8 6	5
12	(1, 3, 4, 8)	30	29	5 8	6
13	(1, 3, 4, 6, 8)	35	35	6 5	8
14	(1, 3, 4, 6)	32	30	8 6	5
15	(1, 3, 4, 5, 6)	39	38	5 8	6

Problema de la mochila

Table 4. Iterations of the STM TS with *TabuTenure* = 3.

Iteration	Current solution	Profit	Weight	Tabu active	Move
1	(1, 4, 5, 6, 8)	33	35		8
2	(1, 4, 5, 6)	30	30	8	3
3	(1, 3, 4, 5, 6)	39	38	3 8	6
4	(1, 3, 4, 5)	34	32	6 3 8	5
5	(1, 3, 4)	27	24	5 6 3	2
6	(1, 2, 3, 4)	41	36	2 5 6	4
7	(1, 2, 3)	33	27	4 2 5	6
8	(1, 2, 3, 6)	38	33	6 4 2	8
9	(1, 2, 3, 6, 8)	41	38	8 6 4	3
10	(1, 2, 6, 8)	32	30	3 8 6	5
11	(1, 2, 5, 6, 8)	39	38	5 3 8	6
12	(1, 2, 5, 8)	34	32	6 5 3	8
13	(1, 2, 5)	31	27	8 6 5	3
14	(1, 2, 3, 5)	40	35	3 8 6	5
15	(1, 2, 3)	33	27	5 3 8	4

Problema de la mochila

