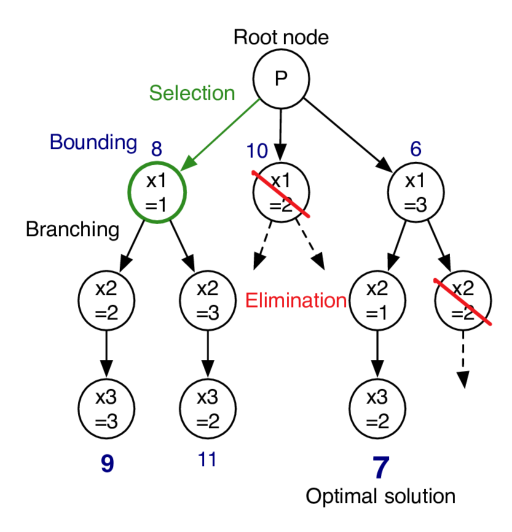


Inteligencia Artificial

Unidad 1: Representación Avanzada del Conocimiento

TEMA 2: Algoritmos de la IA Clásica

Módulo 6: Algoritmo de Ramificación y Poda

Profesora

Ing. Patricia Reyes Silva

Unidad 1

Representación Avanzada del Conocimiento

TEMA 2: Algoritmos de la IA Clásica

Sesión 9

MÓDULO 6: Algoritmo de Ramificación y Poda

Contenido

1. Descripción general
2. Estimación de cotas
3. Estrategia de poda
4. Estrategias de ramificación
5. Ejemplos de aplicación

Preguntas

1. Descripción general

Este algoritmo **Ramificación y Poda**, llamado también **“Branch and Bound” (B&B)** por sus siglas en ingles, es una mejora de la técnica de backtracking.

Se aplica sobre todo a **problemas de optimización**: búsqueda de la mejor solución a un problema, aunque también para buscar una o todas las soluciones a un problema.

Algoritmo Ramificación y Poda vs Algoritmos de Backtraking

SIMILITUDES

- Se basan en el recorrido del árbol de expansión en busca de soluciones.

DIFERENCIAS

- **Estrategias de ramificación:** La generación de los nodos de expansión se puede realizar aplicando distintas estrategias (no solo en profundidad).
- Se establece el orden de ramificación (de modo que comenzaremos explorando las ramas más prometedores del árbol).
- **Estrategias de poda:** Se utilizan **cotas (limite superior y limite inferior)** que permiten podar ramas que no conducen a una solución optima (se evita ramificar nodos).

1. Descripción general

Estrategias de Ramificación

La generación de los nodos del árbol de expansión puede seguir varias estrategias:

1. En anchura **(FIFO)**
2. En profundidad **(LIFO)**
3. Aquella que selecciona el nodo mas prometedor.

Objetivo: Utilizar la estrategia que permita encontrar la solución mas rápidamente.

Estrategia de Poda

En cada nodo se calcula una **cota** del posible valor de aquellas soluciones que pudieran encontrarse más adelante en el árbol.

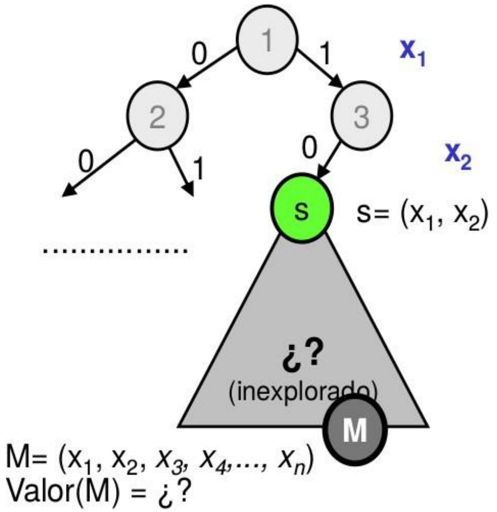
- La **cota** es un valor limite inferior y superior que nos permite delimitar la búsqueda dentro del árbol (espacio de búsqueda)

Objetivo: No explorar aquellas ramas que no conducen a una solución válida u óptima.

2. Estimación de cotas

Estimación de cotas a partir de una solución parcial

Problema: antes de explorar **S**, acotar el beneficio de la mejor solución alcanzable, M.



CI (s) **≤ Valor (M)** ≤ CS(s)

Para cada **nodo i** tendremos:

CS(i): Cota superior del beneficio (o coste) óptimo que podemos alcanzar a partir del nodo i.

CI(i): Cota inferior del beneficio (o coste) óptimo que podemos alcanzar a partir del nodo i.

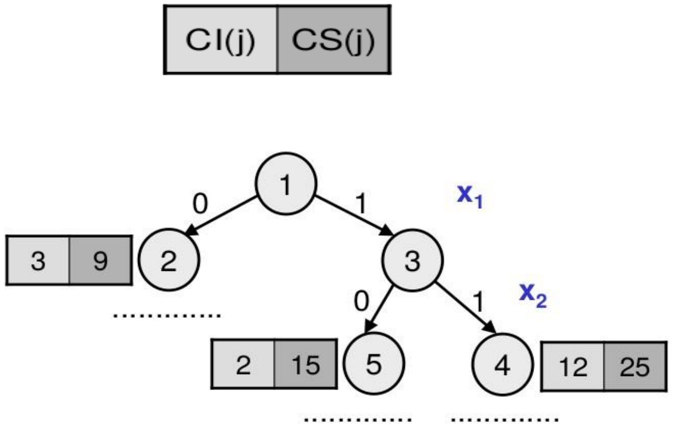
BE(i): Beneficio estimado (o coste) óptimo que se puede encontrar a partir del nodo i.

- Las cotas deben ser **"fiables"**: porque determinan cuándo se puede realizar una poda.
- El beneficio (o coste) estimado nos ayudará a <u>decidir qué parte de árbol se va a evaluar primero</u>.

3. Estrategia de Poda

Supongamos un problema de MAXIMIZACION

Hemos recorrido varios nodos, estimando para cada uno la cota superior CS(j) e inferior CI(j).



- ¿Merece la pena seguir explorando por el nodo 2?
- ¿Y por el nodo 5?

Podar según los valores de **CI** y **CS**

<u>Estrategia de poda (maximización)</u>

Podar un nodo i si se cumple que:

CS(i) ≤ CI(j), para algún nodo **j** generado, o bien
CS(i) ≤ Valor(s), para algún nodo **s** como solución final

Implementación: Usar una variable de poda C
C =max ({CI(j) | ∀j generado}, {Valor(s) | ∀s solución final})
→ Podar i si: CS(i) ≤ C, entonces podar i.

¿Cómo sería para el caso de minimización?

Implementación: Usar una variable de poda C
C =min ({CI(j) | ∀j generado}, {Valor(s) | ∀s solución final})
→ Podar i si: CS(i) ≥ C, entonces podar i.

4. Estrategias de Ramificación

Estrategias de Ramificación

Permiten:

- Determinar distintos **tipos de recorrido** del árbol de expansión: por profundidad, por anchura, por el beneficio estimado, etc.
- Determinar que nodo va a ser expandido, dependiendo de la estrategia de ramificación elegido.

Para hacer el recorrido de un árbol de expansión se utiliza una lista de nodos al que llamaremos **Lista de Nodos Vivos (LNV).**

Un **nodo vivo** del árbol, es aquel que tiene posibilidades de ser ramificado, es decir, aquel que ha sido creado y no ha sido explorado ni podado todavía.

>>> **La lista de nodos vivos (LNV) contiene nodos pendientes de tratar por el algoritmo** <<<

4. Estrategias de Ramificación

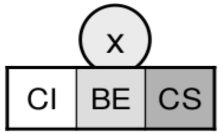
Estrategias de Ramificación:

METODO GENERAL

La idea general del algoritmo es la siguiente:

1. Sacar un elemento N de la lista de nodos vivos (LNV)
2. Generar todos los descendientes de N.
3. Si no se podan y no son solución, se introducen en la LNV.

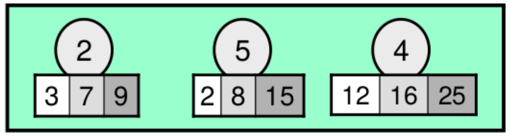
CI=Cota Inferior



CS=Cota Superior

BE=Coste estimado

LNV



METODOS DE RECORRIDO DEL ARBOL

El recorrido del árbol depende de como se maneje la lista

- Sin tener en cuenta los costes/beneficios, realizando una búsqueda “a ciegas”:

1. **Recorrido en anchura ->** La lista se trata como una **cola.**
2. **Recorrido en profundidad ->** La lista se trata como una **pila.**

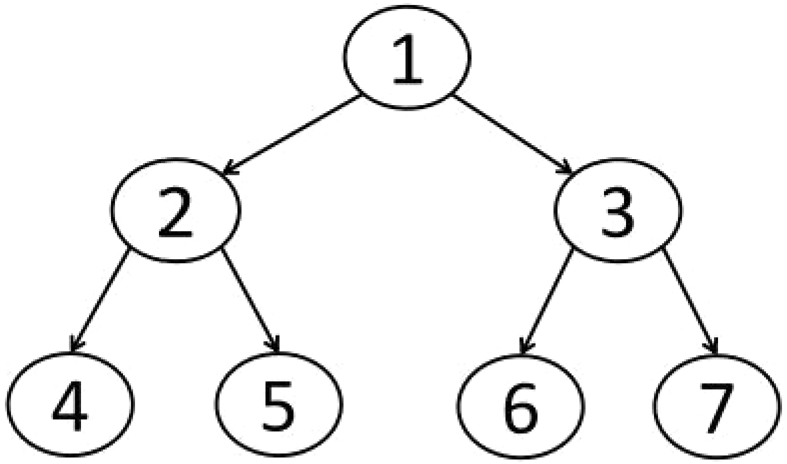
- Teniendo en cuenta los costes/beneficios:

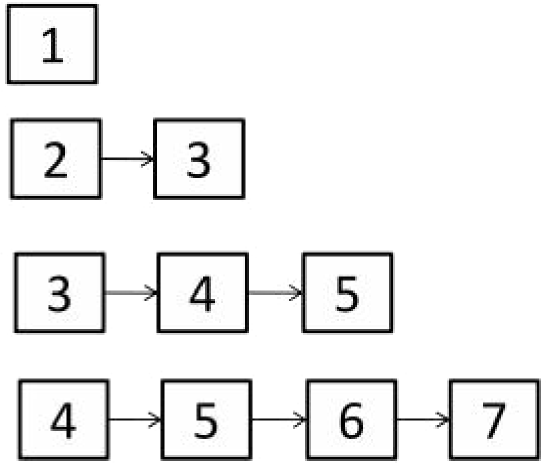
3. **La estrategia del mínimo coste ->** Se utiliza una función de clasificación denominada **mínimo coste** para determinar que elemento de la lista de nodos vivos va a ser explorado en cada momento.
    - Ramificar según los valores de BE

4. Estrategias de Ramificación

1. Recorrido del árbol en anchura (FIFO)

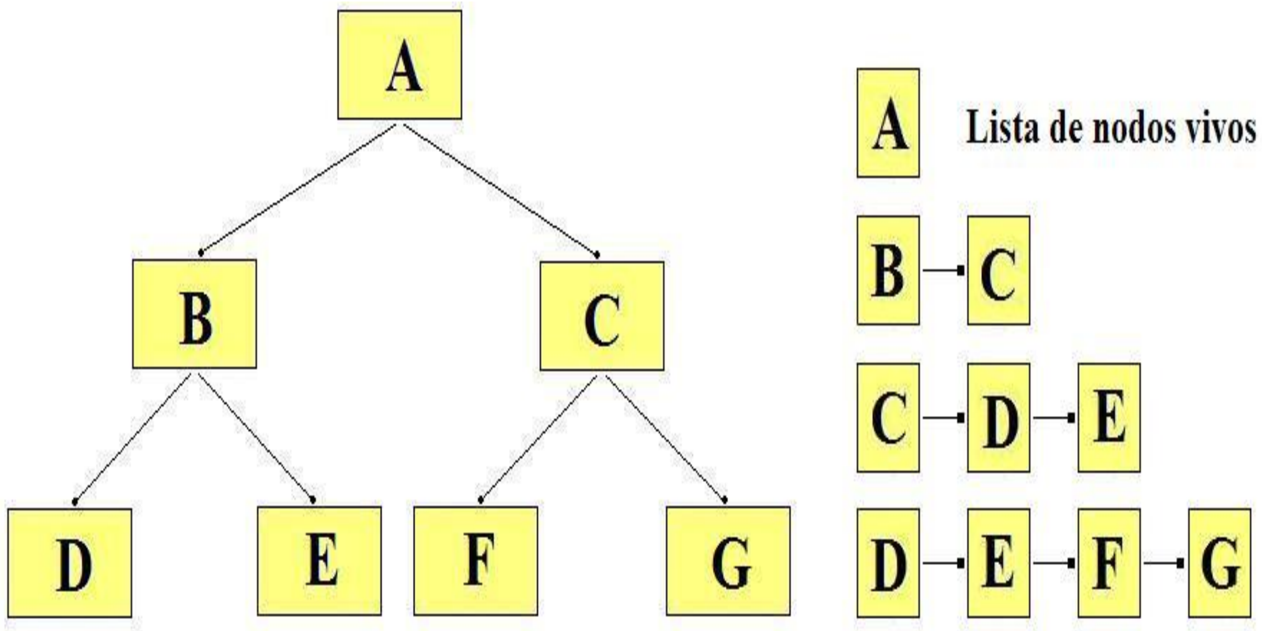
- La estructura de **la lista de nodos vivos (LNV)** se trata como **una cola (FIFO)**, dando lugar a un recorrido en anchura del árbol.
- **FIFO** = **FIRST IN FIRST OUT** = EL PRIMERO EN ENTRAR ES EL PRIMERO EN SALIR.





4. Estrategias de Ramificación

1. Recorrido del árbol en anchura (FIFO)



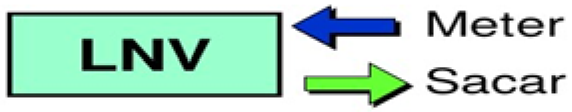
Recorrido FIFO

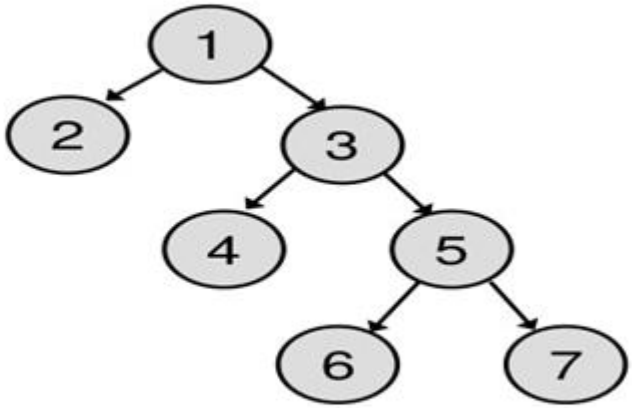
1. Inicia introduciendo en la LNV el nodo A.
2. Sacamos el nodo A de la cola y se expande generando los nodos B y C que son introducidos en la LNV.
3. Seguidamente se saca el primer nodo que es el B y se vuelve a expandir generando los nodos D y E que se introducen en la LNV.
4. Este proceso se repite mientras que quede algún elemento en la cola.

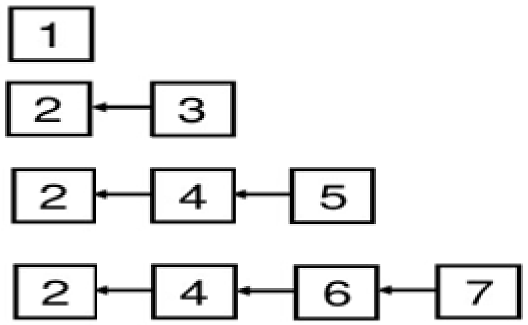
4. Estrategias de Ramificación

2. Recorrido del árbol en profundidad (LIFO)

- La estructura de **<u>la lista de nodos vivos (LNV)</u>** se trata como **<u>una pila (LIFO)</u>**
- **LIFO** = **LAST IN FIRST OUT** = EL ULTIMO EN ENTRAR ES EL PRIMERO EN SALIR.

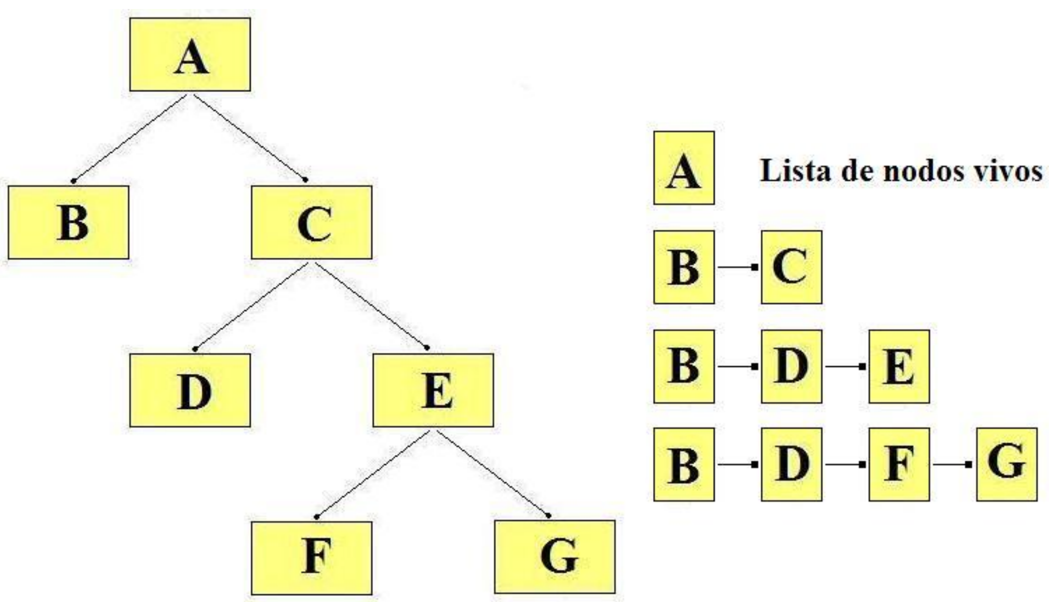






4. Estrategias de Ramificación

2. Recorrido del árbol en profundidad (LIFO)



Recorrido LIFO

C
B

1. Generar hijos del nodo A y colocar estos nodos activos en una pila (B, C).

E
D
B

2. Quitar el elemento el ultimo ingresado a la pila y generar sus hijos, coloque esos nodos en la pila. C se quita de la pila. Los hijos de C son D, E.

G
F
D
B

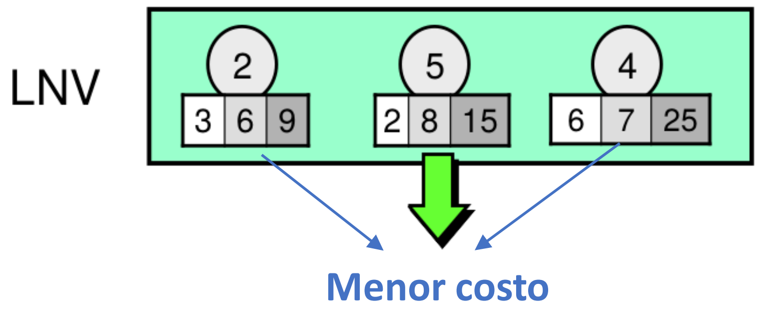
4. Nuevamente, elimine un elemento de la pila, es decir, se elimina el nodo E y los nodos generados por E son F, G que se introducen a la pila.

4. Estrategias de Ramificación

3. La estrategia del mínimo coste (LC)

- Entre todos los nodos de la **lista de nodos vivos**, elegir el que tenga **mayor beneficio (o menor coste)** para explorar a continuación.

- **LC** = **Least Cost** = MENOR COSTO.



¿Y si sucede un empate?
(más de un nodo de menor coste con el mismo valor)

Estrategia LC-FIFO: seleccionar de la LNV el nodo que tenga mayor beneficio y en caso de empate escoger **el primero** que se introdujo (de los que empatan).

Estrategia LC-LIFO: seleccionar de la LNV el nodo que tenga mayor beneficio y en caso de empate escoger **el último** que se introdujo (de los que empatan).

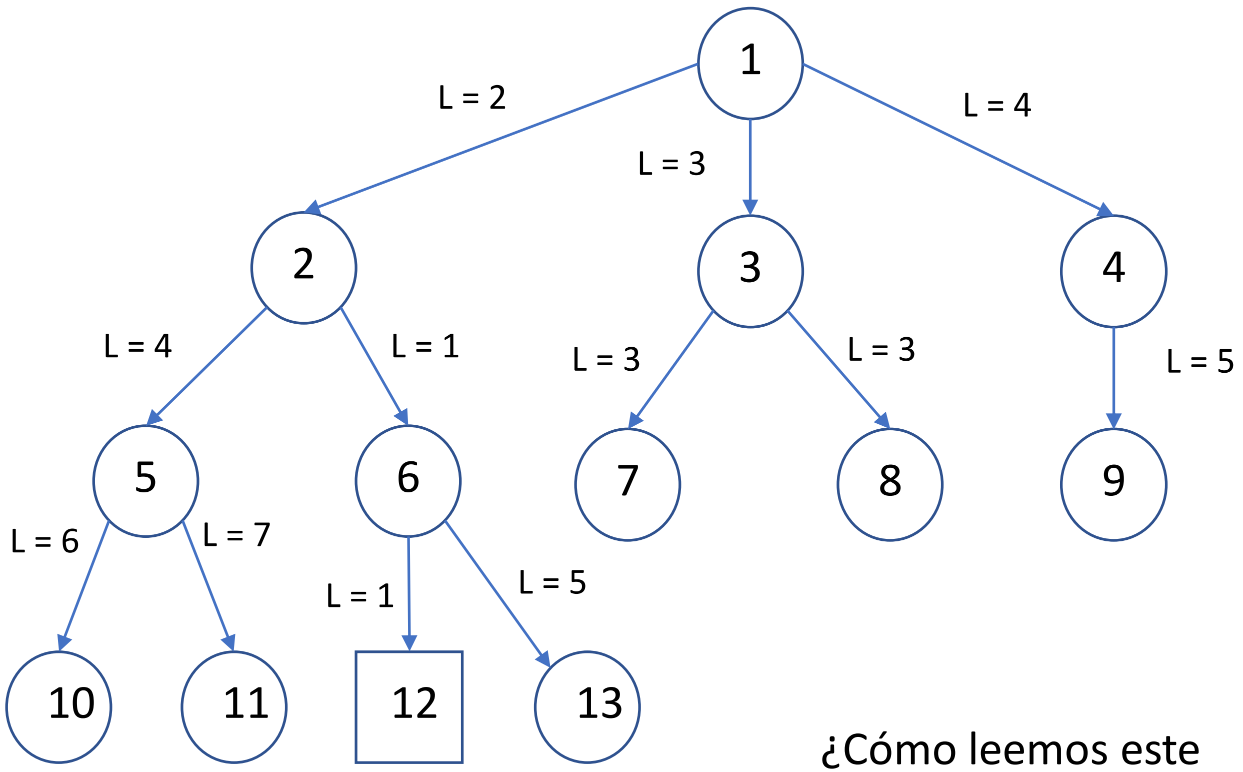
4. Estrategias de Ramificación

3. La estrategia del mínimo o menor coste (LC)

- Tanto en **FIFO** como en **LIFO**, para el algoritmo de Ramificación y Poda, las reglas de selección para el siguiente nodo-E (nodo vivo cuyos hijos se están explorando actualmente) son rígidas y ciegas.

- Usaremos la **función de clasificación** o la **función de costo**, seleccionando un nodo que tiene un costo mínimo.

- Al usar la función de clasificación, calcularemos el **costo de cada nodo**.



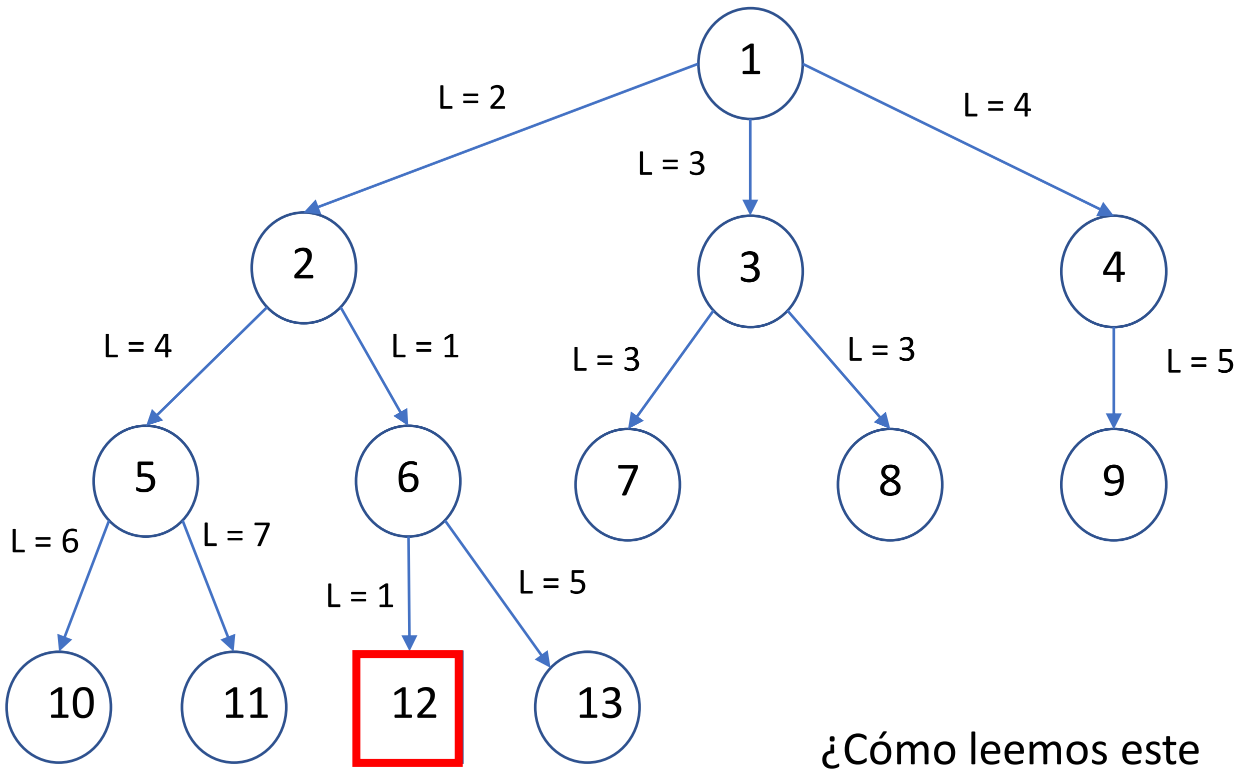
¿Cómo leemos este árbol ponderado?

4. Estrategias de Ramificación

3. La estrategia del mínimo o menor coste (LC)

Recorrido LC-FIFO

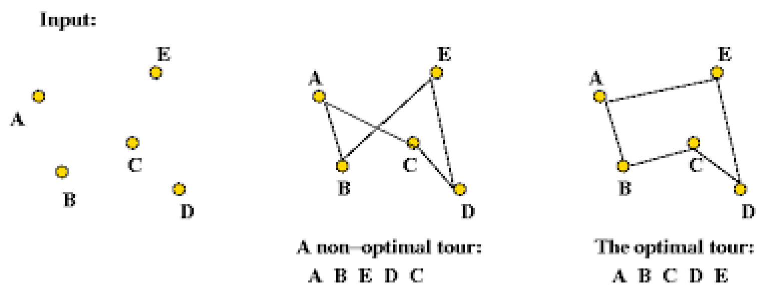
- Inicialmente, tomaremos el nodo 1 como nodo-E.
- Generar hijos del nodo 1, los hijos son 2, 3, 4.
- Usando la función de clasificación calcularemos el costo de 2, 3, 4 nodos es ĉ = 2, ĉ = 3, ĉ = 4 respectivamente.
- Ahora seleccionaremos un nodo que tenga costo mínimo i., nodo-E 2.
- Para el nodo 2, los hijos son 5, 6.
- Entre 5 y 6 seleccionaremos el nodo 6 ya que su costo mínimo es 1.
- Generar hijos del nodo 6, nodo-E 12 y 13.
- Seleccionaremos el nodo 12 ya que su costo (ĉ = 1) es mínimo.
- El nodo 12 es el nodo de respuesta. Entonces, terminamos el proceso de búsqueda.



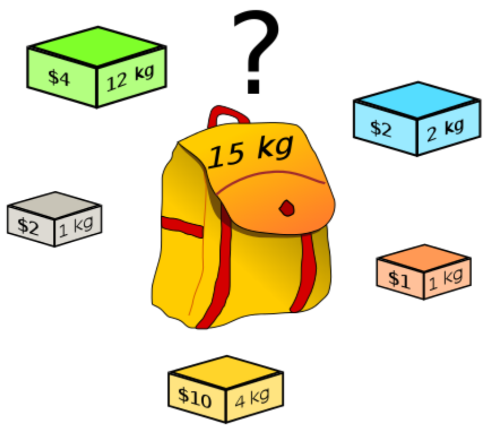
¿Cómo leemos este árbol ponderado?

5. Ejemplos aplicando Ramificación y Poda

Ejemplo #1: El problema del Viajero



Ejemplo #2: El problema de la mochila



5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

Solución al problema del vendedor ambulante (TSP) – Algoritmo Ramificación y Poda

El vendedor debe visitar todas las ciudades sin parar dos veces en la misma ciudad utilizando la menor distancia posible.

Ciudades a visitar: n = 5 -> [a, b, c, d, e] las identificamos como [1,2,3,4,5]

<u>Distancias entre ciudades:</u>

	1	2	3	4	5
1	--	10	8	9	7
2	10	--	10	5	6
3	8	10	--	8	9
4	9	5	8	--	6
5	7	6	9	6	--

Pasos a seguir:

1. Calcular el limite inferior Sumar las distancias mas cortas partiendo desde cada ciudad
2. Ramificar desde el limite inferior.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #1: Calcular CI o el limite inferior

Sumar cada una de las distancias mas cortas (la menor) partiendo desde cada ciudad, es decir, seleccionamos la primera distancia menor de cada fila y totalizamos los valores obtenidos.

Distancias entre ciudades:

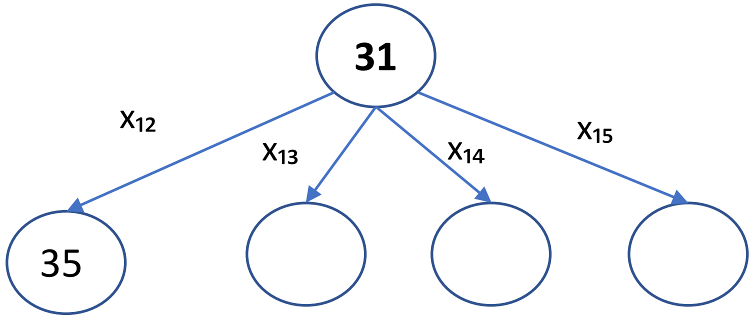
	1	2	3	4	5
1	--	10	8	9	7
2	10	--	10	5	6
3	8	10	--	8	9
4	9	5	8	--	6
5	7	6	9	6	--

7+5+8+5+6 = 31 = CI = Cota o limite inferior

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el CI o el limite inferior



Distancias entre ciudades:

	1	2	3	4	5
1	--	$\boxed{10}$	8	9	7
2	~~10~~	--	10	$\boxed{5}$	6
3	$\boxed{8}$	10	--	8	9
4	9	5	8	--	$\boxed{6}$
5	7	6	9	$\boxed{6}$	--

X_1->2 **No podemos ir de 2->1** (no se puede retroceder o visitar la ciudad anterior ya visitada)

Seleccionamos las distancias menores por fila

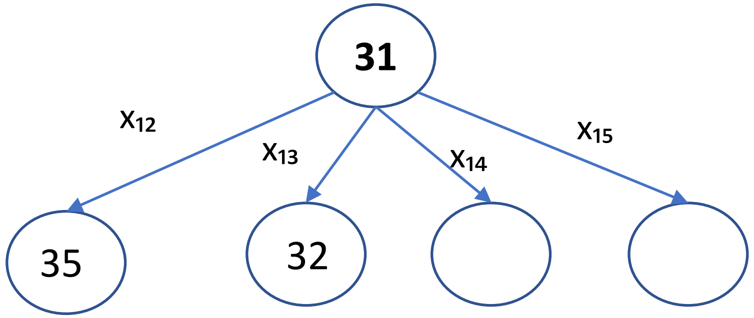
Totalizamos las distancias seleccionadas
10+5+8+6+6 = 35

35 = nuevo cota o limite inferior a alcanzar.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el CI o el limite inferior



Distancias entre ciudades:

	1	2	3	4	5
1	--	10	8	9	7
2	10	--	10	5	6
3	~~8~~	10	--	8	9
4	9	5	8	--	6
5	7	6	9	6	--

X_1->3 **No podemos ir de 3->1** (no se puede retroceder o visitar la ciudad anterior ya visitada)

Seleccionamos las distancias menores por fila

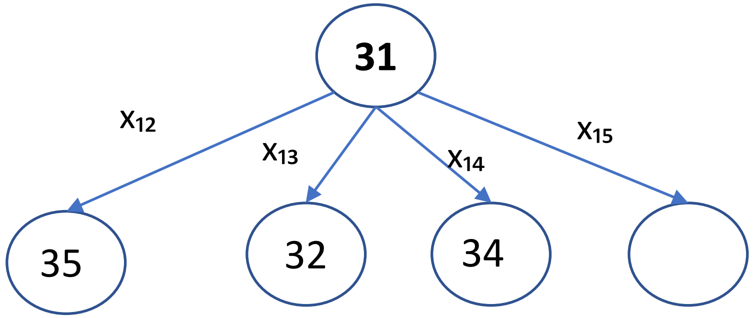
Totalizamos las distancias seleccionadas
8+5+8+5+6 = 32

32 = nuevo cota o limite inferior a alcanzar.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el <u>CI o el limite inferior</u>



<u>Distancias entre ciudades:</u>

	1	2	3	4	5
1	--	10	8	$\boxed{9}$	7
2	10	--	10	5	$\boxed{6}$
3	$\boxed{8}$	10	--	8	9
4	~~9~~	$\boxed{5}$	8	--	6
5	7	$\boxed{6}$	9	6	--

X_1->4 **No podemos ir de 4->1** (no se puede retroceder o visitar la ciudad anterior ya visitada)

Seleccionamos las distancias menores por fila

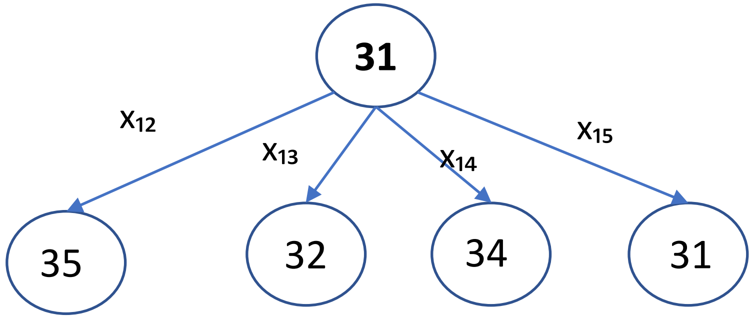
Totalizamos las distancias seleccionadas
9+6+8+5+6 = 34

34 = nuevo cota o limite inferior a alcanzar.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el CI o el limite inferior



Distancias entre ciudades:

	1	2	3	4	5
1	--	10	8	9	$\boxed{7}$
2	10	--	10	$\boxed{5}$	6
3	$\boxed{8}$	10	--	8	9
4	9	$\boxed{5}$	8	--	6
5	~~7~~	$\boxed{6}$	9	6	--

- Evaluando el primer nivel, encontramos que la distancia de 31 minimiza el recorrido.
- La ramificación continua desde dicho nodo.

X_1->5 **No podemos ir de 5->1** (no se puede retroceder o visitar la ciudad anterior ya visitada)

Seleccionamos las distancias menores por fila

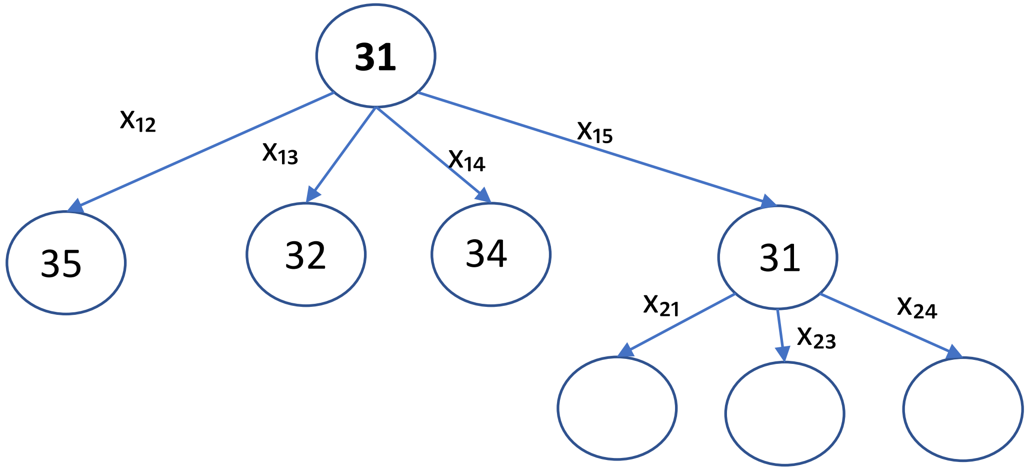
Totalizamos las distancias seleccionadas
7+5+8+5+6= 31

31 = nuevo cota o limite inferior a alcanzar.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el CI o el limite inferior



Distancias entre ciudades:

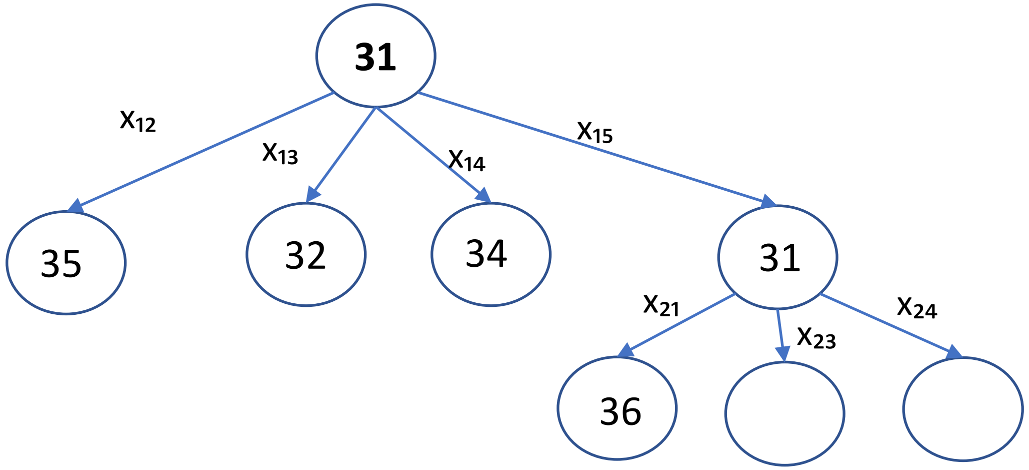
	1	2	3	4	5
1	--	10	8	9	7
2	10	--	10	5	6
3	8	10	--	8	9
4	9	5	8	--	6
5	7	6	9	6	--

- Ahora, en el segundo nivel del árbol, evaluaremos ir desde la ciudad 2 a las otras 3 ciudades que no sean la 5 porque ya la visitamos en el primer nivel ni la 2 (porque de 2->2 no podemos ir porque ya estamos allí).

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el CI o el limite inferior



Distancias entre ciudades:

	1	2	3	4	5
1	--	10	8	9	$\boxed{7}$
2	$\boxed{10}$	--	10	5	6
3	8	10	--	$\boxed{8}$	9
4	9	$\boxed{5}$	8	--	6
5	7	6	9	$\boxed{6}$	--

X_1->5 de 1 fuimos a 5

X_2->1 de 2 fuimos a 1

No podemos ir de 5->1 ni de 1->2

Seleccionamos las distancias menores por fila

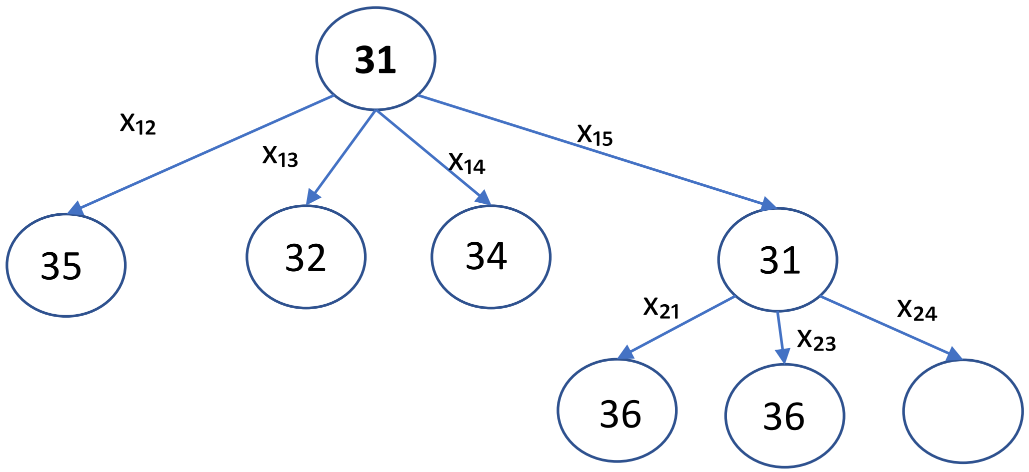
Totalizamos las distancias seleccionadas
7+10+8+5+6= 36

36 = nuevo cota o limite inferior a alcanzar.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el CI o el limite inferior



Distancias entre ciudades:

	1	2	3	4	5
1	--	10	8	9	$\boxed{7}$
2	10	--	$\boxed{10}$	5	6
3	8	~~10~~	--	$\boxed{8}$	9
4	9	$\boxed{5}$	8	--	6
5	~~7~~	6	9	$\boxed{6}$	--

X_1->5 de 1 ya fuimos a 5

X_2->3 de 2 vamos a 3

No podemos ir de 5->1 ni de 3->2

Seleccionamos las distancias menores por fila

Totalizamos las distancias seleccionadas

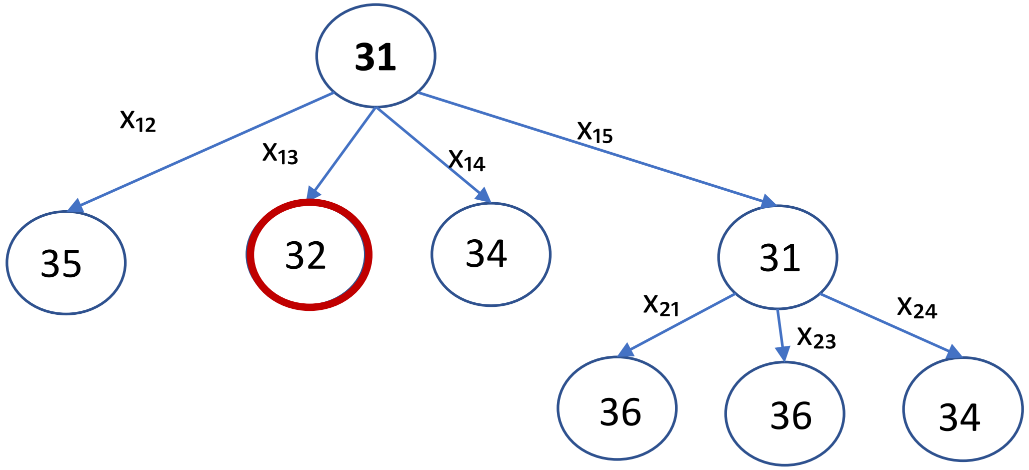
7+10+8+5+6= 36

36 = nuevo cota o limite inferior a alcanzar.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el CI o el limite inferior



Distancias entre ciudades:

	1	2	3	4	5
1	--	10	8	9	$\boxed{7}$
2	10	--	10	$\boxed{5}$	6
3	$\boxed{8}$	10	--	8	9
4	9	~~5~~	$\boxed{8}$	--	6
5	~~7~~	$\boxed{6}$	9	6	--

X_1->5 de 1 ya fuimos a 5

X_2->4 de 2 vamos a 4

No podemos ir de 5->1 ni de 4->2

Seleccionamos las distancias menores por fila

Totalizamos las distancias seleccionadas
7+5+8+8+6= 34

34 = nuevo cota o limite inferior a alcanzar.

Cuál rama seguimos expandiendo?

Respuesta: La que menor valor tenga….

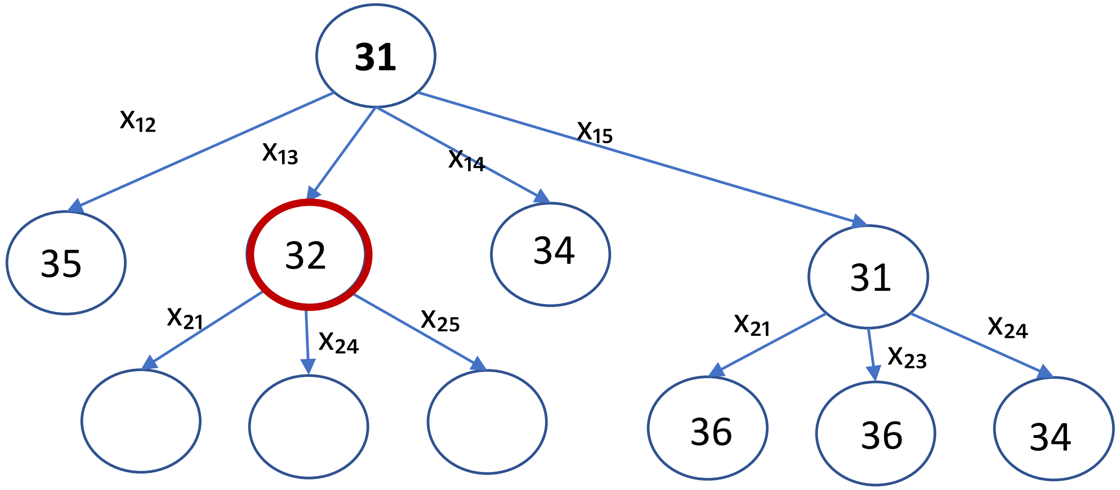
Entre el primer y segundo nivel….

Y esa es la de valor de **nodo 32**

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el <u>CI o el limite inferior</u>



Distancias entre ciudades:

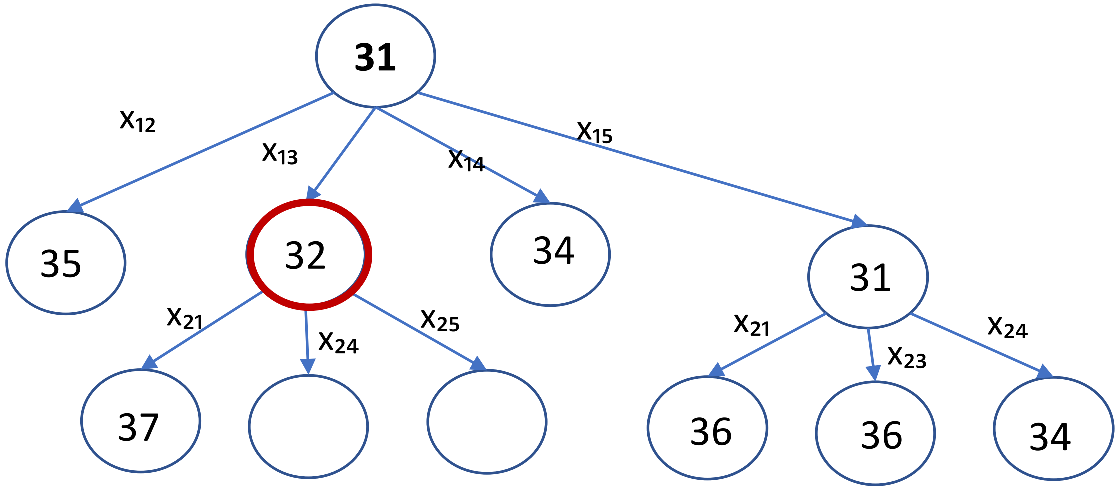
	1	2	3	4	5
1	--	10	8	9	7
2	10	--	10	5	6
3	8	10	--	8	9
4	9	5	8	--	6
5	7	6	9	6	--

- **<u>Seguimos en el segundo nivel del árbol</u>**, evaluaremos ir desde la ciudad 2 a las otras 3 ciudades que no sean la 3 porque ya la visitamos en el primer nivel ni la 2 (porque de 2->2 no podemos ir).
- Queremos encontrar una ruta que minimice la distancia 32.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el CI o el limite inferior



Distancias entre ciudades:

	1	2	3	4	5
1	--	10	8	9	7
2	10	--	10	5	6
3	8	10	--	8	9
4	9	5	8	--	6
5	7	6	9	6	--

X_1->3 (de 1 ya fuimos a 3 en el nivel 1)

X_2->1 de 2 vamos a 1

No podemos ir de 3->1 ni de 1->2

Seleccionamos las distancias menores por fila

Totalizamos las distancias seleccionadas

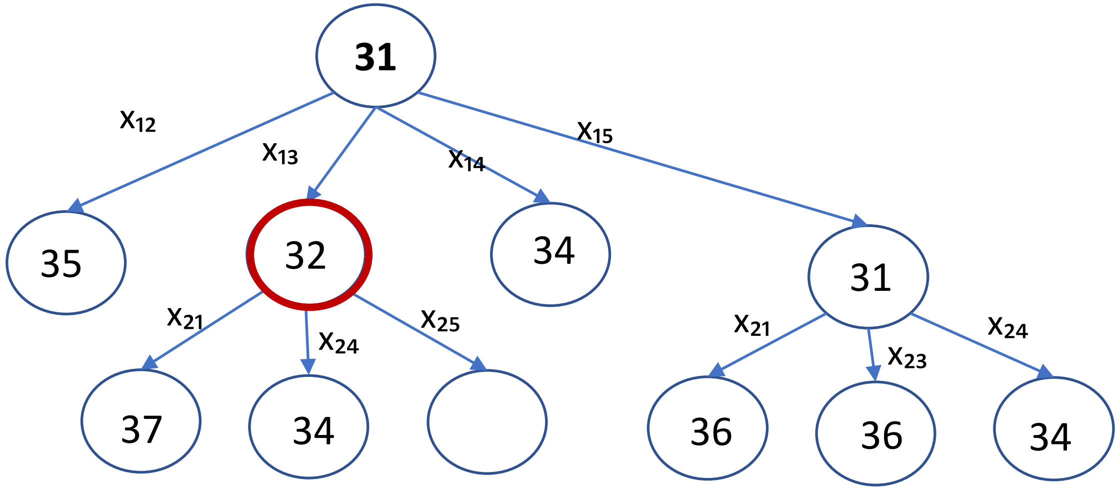
8+10+8+5+6= 37

37 = nuevo cota o limite inferior a alcanzar.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el <u>CI o el limite inferior</u>



<u>Distancias entre ciudades:</u>

	1	2	3	4	5
1	--	10	$\boxed{8}$	9	7
2	10	--	10	$\boxed{5}$	6
3	~~8~~	10	--	8	$\boxed{9}$
4	9	~~5~~	8	--	$\boxed{6}$
5	7	$\boxed{6}$	9	6	--

X_1->3 (de 1 ya fuimos a 3 en el nivel 1)

X_2->4 de 2 vamos a 4

No podemos ir de 3->1 ni de 4->2

Seleccionamos las distancias menores por fila

Totalizamos las distancias seleccionadas

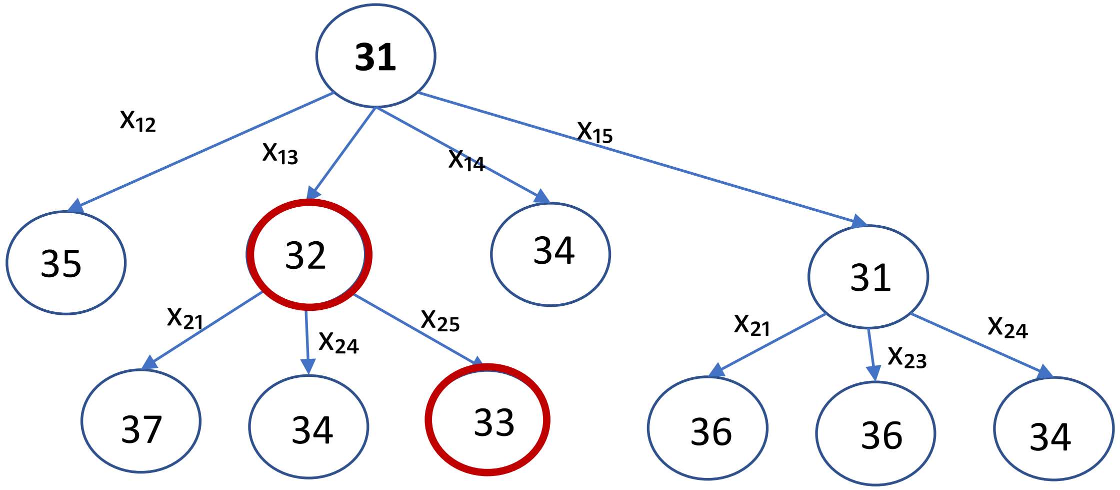
8+5+9+6+6= 34

34 = nuevo cota o limite inferior a alcanzar.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el <u>CI o el limite inferior</u>



Cuál rama seguimos expandiendo?

Respuesta: La que menor valor tenga…. Y esa es la de valor 33

<u>Distancias entre ciudades:</u>

	1	2	3	4	5
1	--	10	$\boxed{8}$	9	7
2	10	--	10	5	$\boxed{6}$
3	~~8~~	10	--	$\boxed{8}$	9
4	9	$\boxed{5}$	8	--	6
5	7	~~6~~	9	$\boxed{6}$	--

X_1->3 (de 1 ya fuimos a 3 en el nivel 1)

X_2->5 de 2 vamos a 5

No podemos ir de 3->1 ni de 5->2

Seleccionamos las distancias menores por fila

Totalizamos las distancias seleccionadas

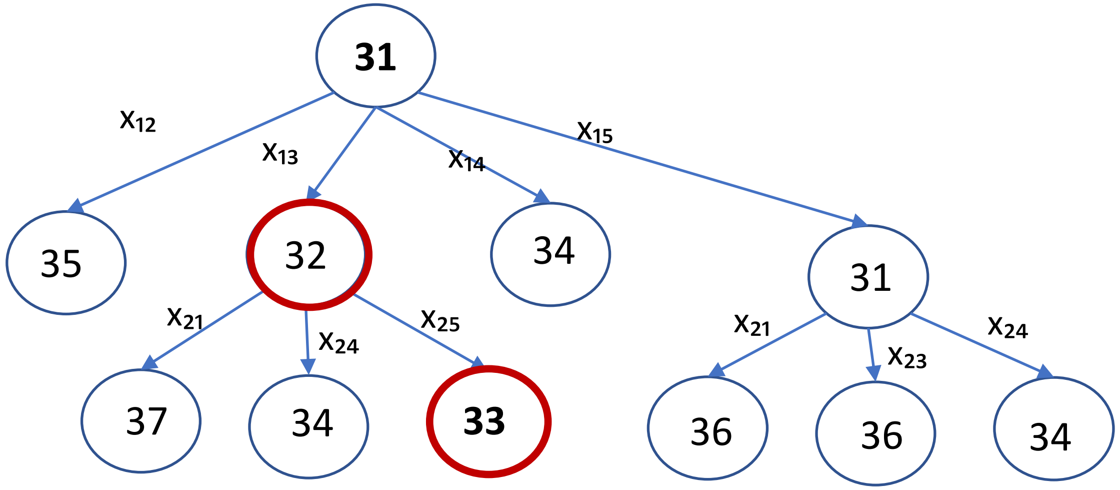
8+6+8+5+6= 33

33 = nuevo cota o limite inferior a alcanzar.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el CI o el limite inferior



Distancias entre ciudades:

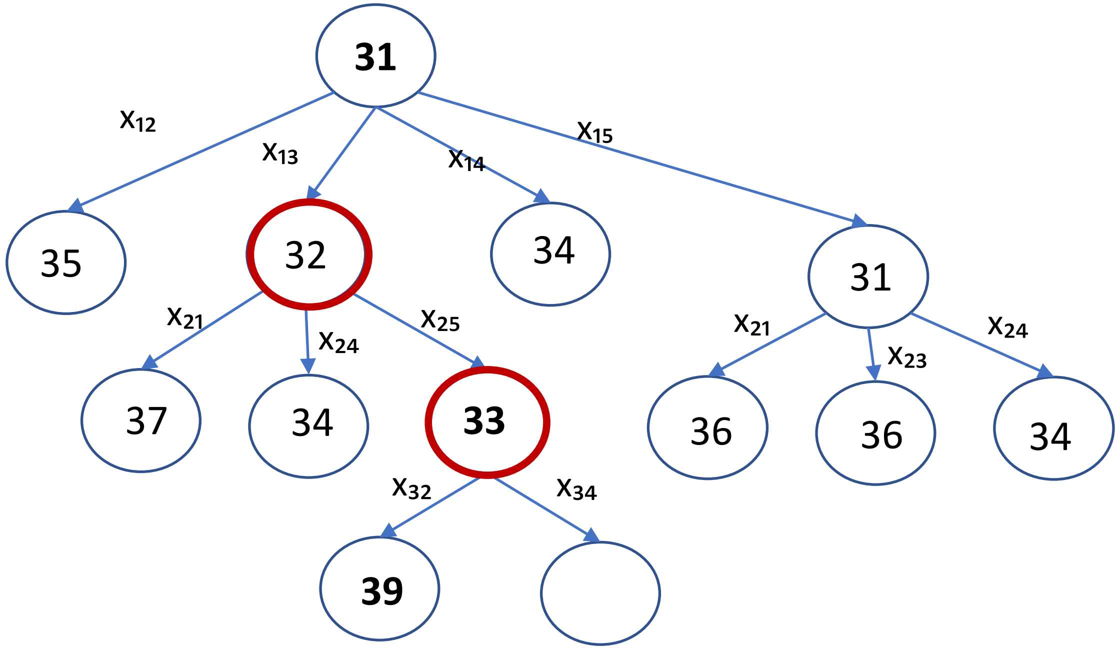
	1	2	3	4	5
1	--	10	8	9	7
2	10	--	10	5	6
3	8	10	--	8	9
4	9	5	8	--	6
5	7	6	9	6	--

- Pasamos al tercer nivel del árbol, evaluaremos ir desde la ciudad 3 a las otras 2 ciudades que no sean la 5 porque ya la visitamos en el segundo nivel, ni la 3 (porque de 3->3 no podemos ir). Nos queda por visitar las ciudades: 2 y 4.
- Queremos encontrar una ruta que minimice la distancia 33.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el CI o el limite inferior



1-3-2-5-4-1
(una posible solución)

Distancias entre ciudades:

	1	2	3	4	5
1	--	10	$\boxed{8}$	9	7
2	10	--	10	5	$\boxed{6}$
3	8	$\boxed{10}$	--	8	9
4	$\boxed{9}$	5	8	--	6
5	7	6	9	$\boxed{6}$	--

X_1->3 de 1 ya fuimos a 3
X_2->5 de 2 ya fuimos a 5
X_3->2 de 3 vamos a 2

No podemos ir de 3->1 ni de 5->2, ni 2->3

Seleccionamos las distancias menores por fila

Totalizamos las distancias seleccionadas

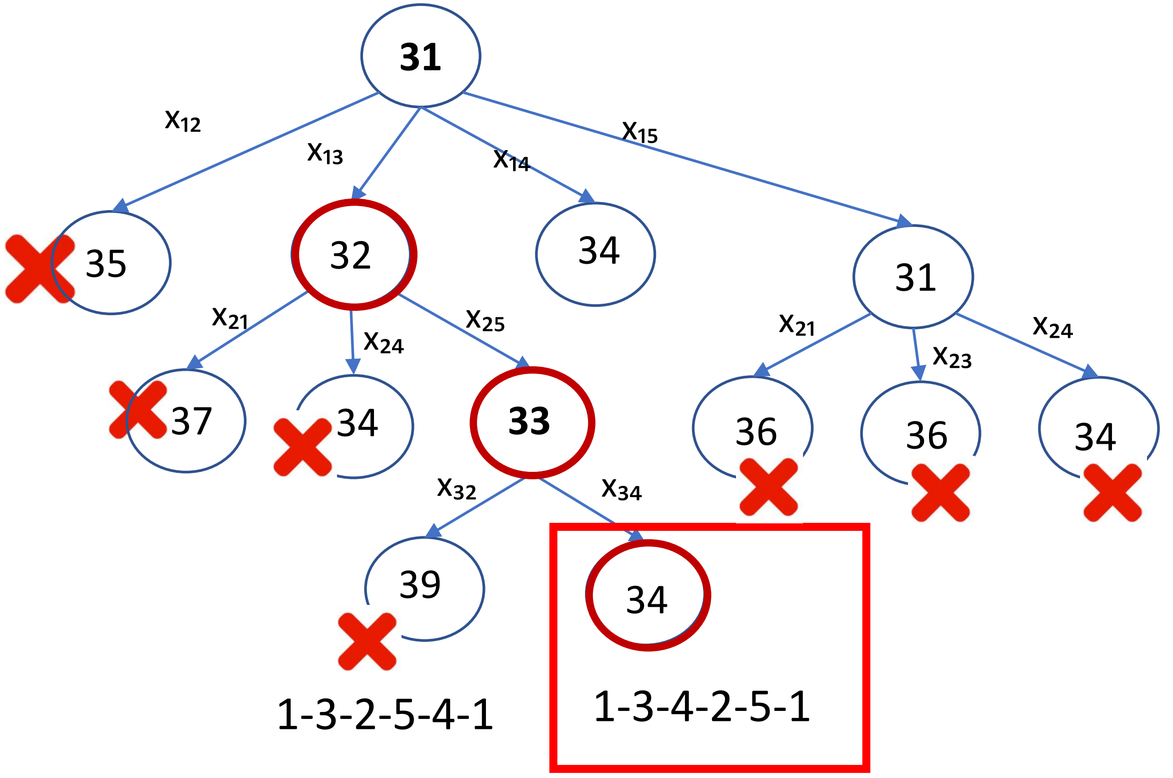
8+6+10+9+6= 39

39 = nuevo cota o limite inferior a alcanzar.

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO# 1:

PASO #2: Ramificar desde el <u>CI o el limite inferior</u>



<u>Distancias entre ciudades:</u>

	1	2	3	4	5
1	--	10	$\boxed{8}$	9	7
2	10	--	10	5	$\boxed{6}$
3	8	10	--	$\boxed{8}$	9
4	9	$\boxed{5}$	8	--	6
5	$\boxed{7}$	6	9	6	--

X1->3 de 1 ya fuimos a 3
X2->5 de 2 ya fuimos a 5
X3->4 de 3 vamos a 4

No podemos ir de 3->1 ni de 5->2, ni 4->3

Seleccionamos las distancias menores por fila

Totalizamos las distancias seleccionadas
8+6+8+5+7= 34

34 = nuevo cota o limite inferior a alcanzar.

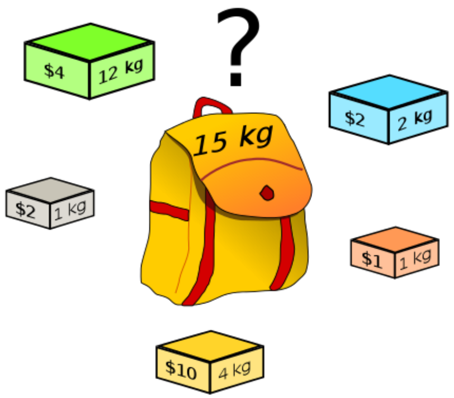
Esta solución permite podar ramas cuyos valores son mayores a los encontrados.
Para las ramas con el mismo valor (34) no se ramifica porque no se encontrará una cota menor a 34

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

Pasos a seguir:

- Hay **n** objetos dados y la capacidad de la mochila es **M.**
- Seleccione algunos objetos para llenar la mochila de tal manera que no exceda la capacidad de la mochila y se pueda obtener el máximo beneficio.
- El problema de la mochila es un problema de maximización. Significa que siempre buscaremos para **v máximo1X1** (donde v1 representa el beneficio del objeto x1).



Considere la instancia **M** = 15 (máxima capacidad de la mochila)
cantidad de productos -> **n** = 4
valor de cada producto -> **v** = (v1, v2, v3, v4) = 10, 10, 12, 18
peso de cada producto -> **w** = (w1, w2, w3,w4) = (2, 4, 6, 9)

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

Pasos a seguir:

Paso# 1: Calcular el limite inferior y el limite superior de cada nodo

- Ordenar las ganancias y pesos del artículo con respecto a la <u>proporción de ganancia por peso</u>.
- Después de eso, colocar el primer artículo en la mochila.

Producto	Valor (v)	Peso (w)	LS = CS (*)	LI = CI
1	10	2	10	10
2	10	4	10	10
3	12	6	12	12
4	18	9	0	6

12kg

CS= 32

CI = 38

(*) **CS** solo permite valores enteros
CI si permite fracciones

CS

Beneficio de artículos en M = 10+10+12 = 32
(para los 3 primeros artículos)

CS = Limite superior = 32 (introducimos solo los 3 primeros artículos y sobraron 3kg de espacio libre)

CI

Beneficio de artículos en M = 10+10+12 + 6 = 38

Límite inferior = 10 + 10 + 12 + ($\underbrace{3/9 * 18}_{\text{(proporción de ganancia x 3kg)}}$) = 32 + 6 = 38

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

Pasos a seguir:

Paso# 1: Calcular el limite inferior y el limite superior de cada nodo (continuación)

- <u>La mochila es un problema de maximización</u>, pero la técnica de **ramificación** es aplicable solo para problemas de minimización. Para <u>convertir un problema de maximización en un problema de minimización</u>, debemos tomar un **signo negativo** para el límite superior y el límite inferior.

Limite superior (LS o CS) = - 32

Límite inferior (LI o CI) = -38

1

-38	CI
-32	CS

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

Paso# 2: Elegimos la ruta, que ha minimizado la diferencia entre el límite superior y el límite inferior entre el **nodo 2 y 3**.

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

- Ahora calcularemos el límite superior y el límite inferior para los **nodos 2, 3**

Producto	Valor (v)	Peso (w)	LS = CS (*)	LI = CI
1	10	2	10	10
2	10	4	10	10
3	12	6	12	12
4	18	9	0	6

12kg

-------- CS= 32

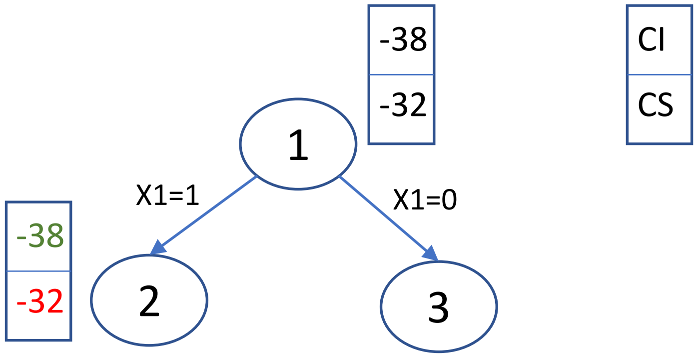
-------- CI = 38

(*) **CS** solo permite valores enteros
CI si permite fracciones

Para el Nodo 2
x1=1, significa que debemos colocar el artículo en la mochila.

CS = 10 + 10 + 12 = 32, hazlo como -32

CI = 10 + 10 + 12 + (**3/9** * 18) = 32 + 6 = 38
lo hacemos como -38



5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

Paso# 2: Continuación…

- Ahora calcularemos el límite superior y el límite inferior para el **nodo 3**

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

Producto	Valor (v)	Peso (w)	LS = CS (*)	LI = CI
~~1~~	~~10~~	~~2~~	~~10~~	~~10~~
2	10	4	10	10
3	12	6	12	12
4	18	9	0	10
			--------	--------
			CS= 22	CI = 32

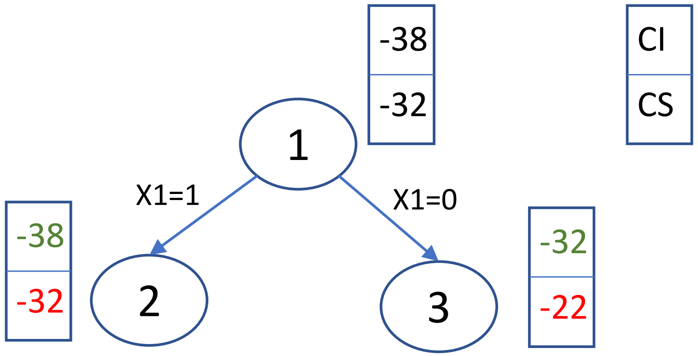
10kg

(*) **CS** solo permite valores enteros
CI si permite fracciones

Para el Nodo 3
x1=0, significa que **NO debemos colocar** el primer artículo en la mochila.

CS = 10 + 12 = 22, hazlo como -22

CI = 10 + 12 + (**5/9** * 18) = 10 + 12 + 10 = 32
lo hacemos como -32



5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

Paso# 2: Continuación…

- Ahora calcularemos el límite superior y el límite inferior para el **nodo 3**

Producto	Valor (v)	Peso (w)	LS = CS (*)	LI = CI
~~1~~	~~10~~	~~2~~	~~10~~	~~10~~
2	10	4	10	10
3	12	6	12	12
4	18	9	0	10

10kg

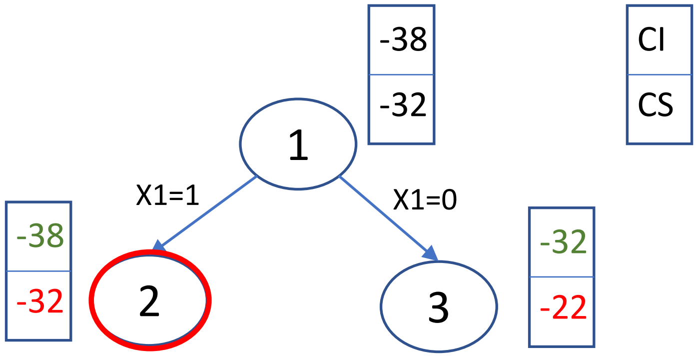
CS= 22   CI = 32

Para el Nodo 3

x=0, significa que **<u>NO debemos colocar</u>** el artículo en la mochila.

CS = 10 + 12 = 22, hazlo como -22

CI = 10 + 12 + (5/9 * 18) = 10 + 12 + 10 = 32
lo hacemos como -32



- Calcularemos la diferencia del límite superior y el límite inferior para los **nodos 2, 3**

Para el nodo 2, CS-CI = -32 + 38 = 6

Para el nodo 3, CS-CI = -22 + 32 = 10

Elegimos seguir por el **nodo 2**, ya que tiene un valor de diferencia mínima de **6**

(*) **CS** solo permite valores enteros
CI si permite fracciones

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

Paso# 3: Calcularemos el límite inferior y el límite superior de los **nodos 4 y 5.**

- Ahora calcularemos el límite superior y el límite inferior para el **nodo 4**

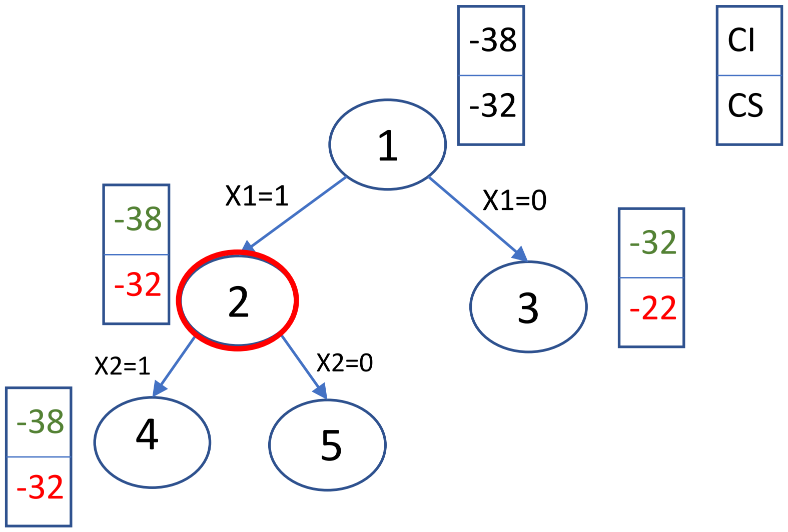
Producto	Valor (v)	Peso (w)	LS = CS (*)	LI = CI
1	10	2	10	10
2	10	4 (12kg)	10	10
3	12	6	12	12
4	18	9	0	6
			--------	--------
			CS= 32	CI = 38

(*) **CS** solo permite valores enteros
CI si permite fracciones

Para el Nodo 4
x2=1, significa que debemos colocar el artículo en la mochila.

CS = 10 + 10 + 12 = 32, hazlo como -32

CI = 10 + 10 + 12 + (**3/9** * 18) = 32 + 6 = 38
lo hacemos como -38



5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

Paso# 3: Continuación…

- Ahora calcularemos el límite superior y el límite inferior para el **nodo 5**

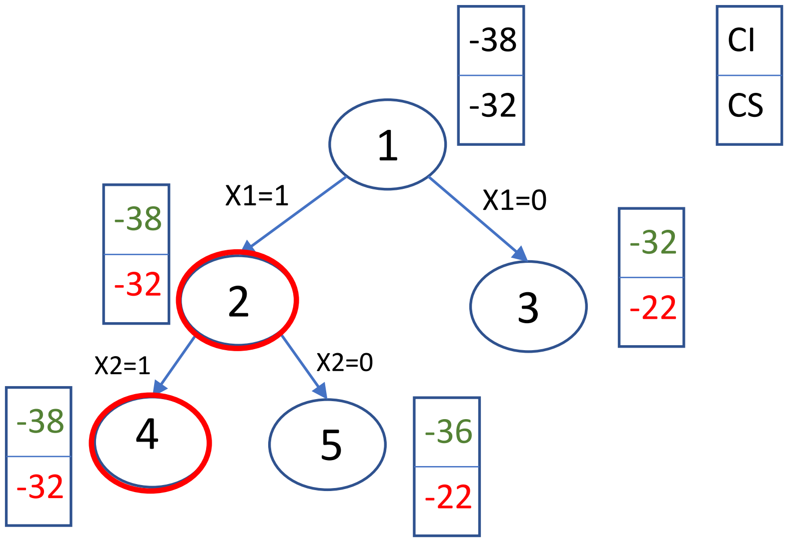
8kg

Producto	Valor (v)	Peso (w)	LS = CS (*)	LI = CI
1	10	2	10	10
~~2~~	~~10~~	~~4~~	~~10~~	~~10~~
3	12	6	12	12
4	18	9	0	14
			--------	--------
			CS= 22	CI = 36

Para el Nodo 5
x2=0, significa que **NO debemos colocar** el artículo en la mochila.

CS = 10 + 12 = 22, hazlo como -22

CI = 10 + 12 + ((7/9 * 18) = 10 + 12 + 14 = 36 lo hacemos como -36



- Calcularemos la diferencia del límite superior y el límite inferior para los **nodos 4, 5**

Para el nodo 4, CS-CI = -32 + 38 = 6

Para el nodo 5, CS-CI = -22 + 36 = 14

Elegimos seguir por el **nodo 4**, ya que tiene un valor de diferencia mínima de **6**

(*) **CS** solo permite valores enteros
CI si permite fracciones

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

Paso# 4: Calcularemos el límite inferior y el límite superior de los **nodos 6 y 7.**

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

- Ahora calcularemos el límite superior y el límite inferior para el **nodo 6**

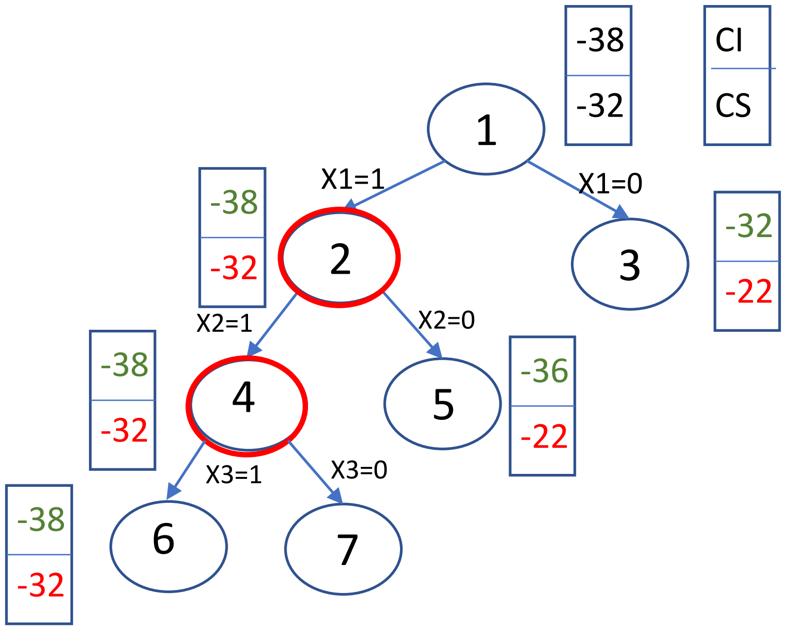
Producto	Valor (v)	Peso (w)	LS = CS (*)	LI = CI
1	10	2	10	10
2	10	4 (12kg)	10	10
3	12	6	12	12
4	18	9	0	6
			--------	--------
			CS= 32	CI = 38

(*) **CS** solo permite valores enteros
CI si permite fracciones

Para el Nodo 6
x3=1, significa que debemos colocar el artículo en la mochila.

CS = 10 + 10 + 12 = 32, hazlo como -32

CI = 10 + 10 + 12 + (3/9 * 18) = 32 + 6 = 38
lo hacemos como -38



5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

Paso# 4: Continuación…

- Ahora calcularemos el límite superior y el límite inferior para el **nodo 7**

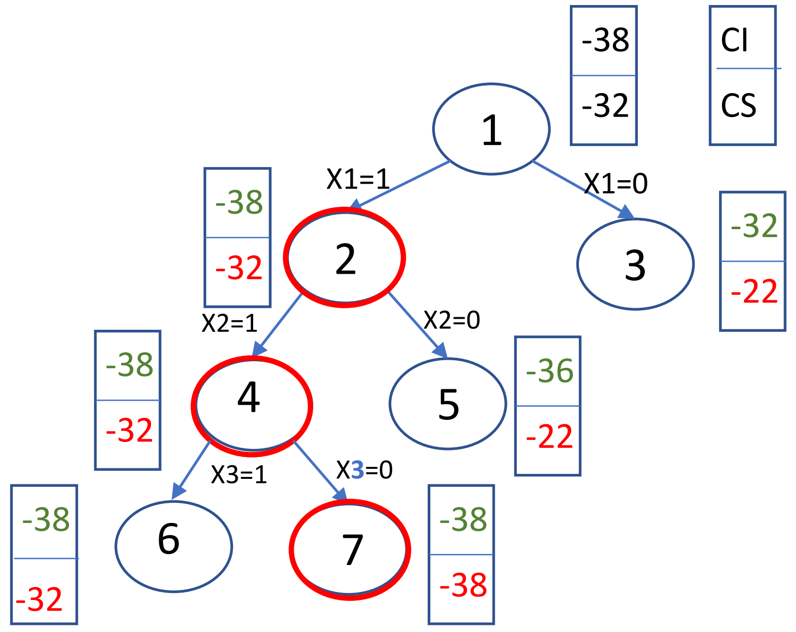
15kg

Producto	Valor (v)	Peso (w)	LS = CS (*)	LI = CI
1	10	2	10	10
2	10	4	10	10
3	12	6	12	12
4	18	9	18	18
			---------	--------
			CS= 38	CI = 38

Para el Nodo 7
x3=0, significa que **NO debemos colocar** el artículo en la mochila.

CS = 10 + 10 + 18 = 38, hazlo como -38

CI = 10 + 10 + 18 = 38 lo hacemos como -38



- Calcularemos la diferencia del límite superior y el límite inferior para los **nodos 6, 7**

Para el nodo 6, CS-CI = -32 + 38 = 6
Para el nodo 7, CS-CI = -38 + 38 = 0

Elegimos seguir por el **nodo 7**, ya que tiene un valor de diferencia mínima de **0**

(*) **CS** solo permite valores enteros
CI si permite fracciones

5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

Paso# 5: Calcularemos el límite inferior y el límite superior de los **nodos 8 y 9.**

- Ahora calcularemos el límite superior y el límite inferior para el **nodo 8**

Producto	Valor (v)	Peso (w)	LS = CS (*)	LI = CI
1	10	2	10	10
2	10	4	10	10
3	12	6	12	12
4	18	9	18	6
			-------- CS= 32	-------- CI = 38

6kg

9kg

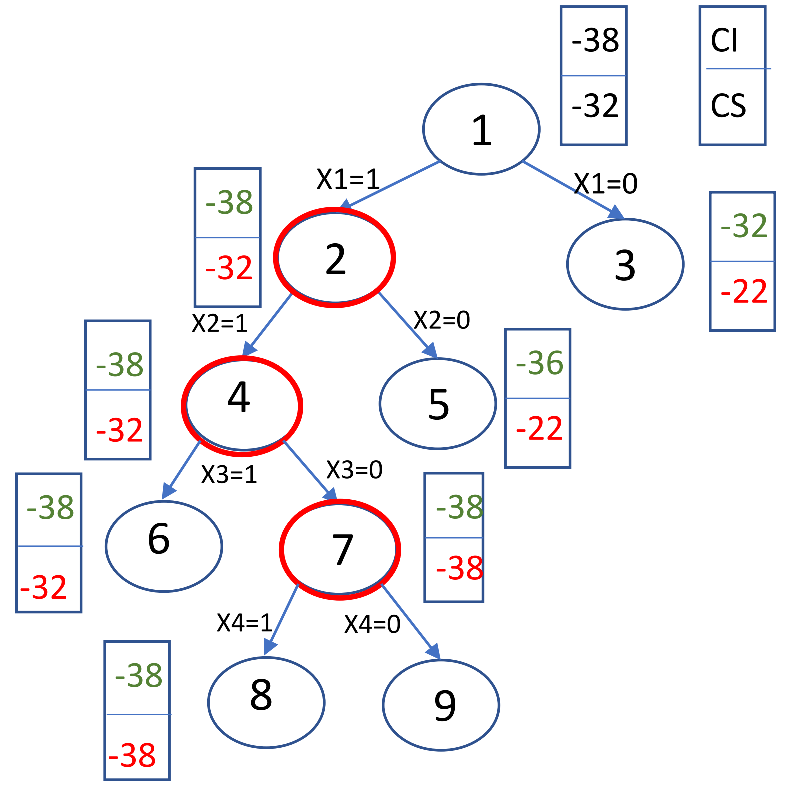
15kg

(*) **CS** solo permite valores enteros
CI si permite fracciones

Para el Nodo 8
x4=1, significa que debemos colocar el artículo en la mochila.

CS = 10 + 10 + 18 = 38, hazlo como -38

CI = 10 + 10 + 12 + (3/9 * 18) = 32 + 6 = 38
lo hacemos como -38



5. Ejemplos aplicando Ramificación y Poda

EJEMPLO #2: El problema de la mochila

M = 15 (máxima capacidad de la mochila)
n = 4
v = (v1, v2, v3, v4) = 10, 10, 12, 18
w = (w1, w2, w3,w4) = (2, 4, 6, 9)

Paso# 5: Continuación…

- Ahora calcularemos el límite superior y el límite inferior para el **nodo 9**

Producto	Valor (v)	Peso (w)	LS = CS (*)	LI = CI
1	10	2	10	10
2	10	4	10	10
3	12	6	12	12
~~4~~	~~18~~	~~9~~	~~18~~	~~14~~

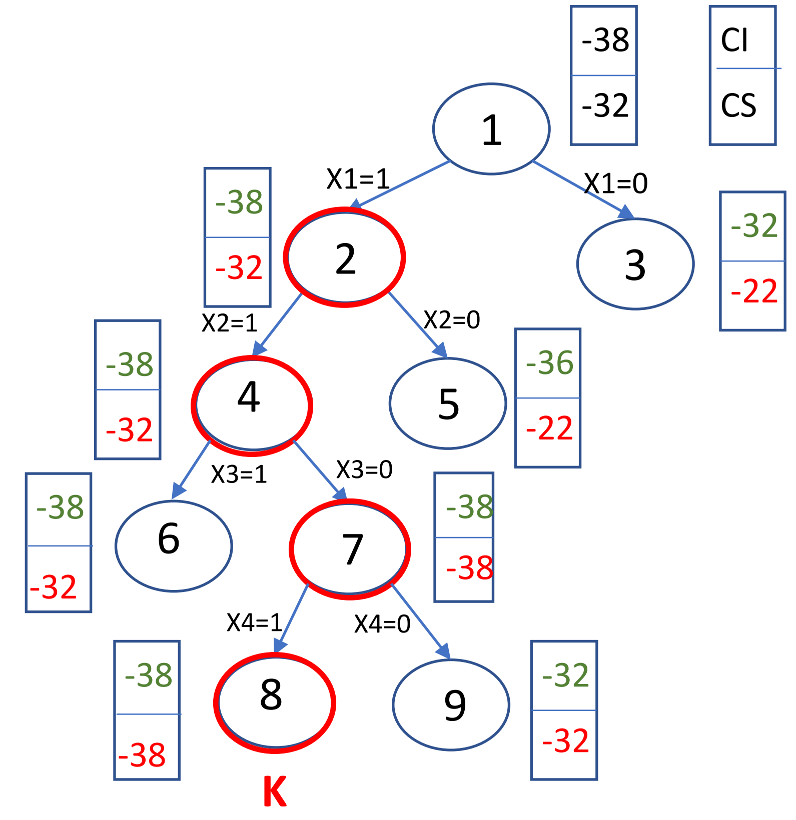
12kg

CS= 32 CI = 32

Para el Nodo 9
x4=0, significa que **NO debemos colocar** el artículo en la mochila.

CS = 10 + 10 + 12 = 32, hazlo como -32

CI = 10 + 10 + 12 = 32, lo hacemos como -32



- Calcularemos la diferencia del límite superior y el límite inferior para los **nodos 8, 9**

Para el nodo 8, CS-CI = -38 + 38 = 0
Para el nodo 9, CS-CI = -32 + 32 = 0

La diferencia es la misma = 0, debemos comparar los límites superiores de los nodos 8 y 9. Descartamos el nodo 9, que tiene el límite superior máximo (-32)

(*) **CS** solo permite valores enteros
CI si permite fracciones

PREGUNTAS

Dudas y opiniones