ALGORITMIA

PRÁCTICA 7

Héctor Lavandeira Fernández
UO277303 | UNIVERSIDAD DE OVIEDO – CURSO 2021/22

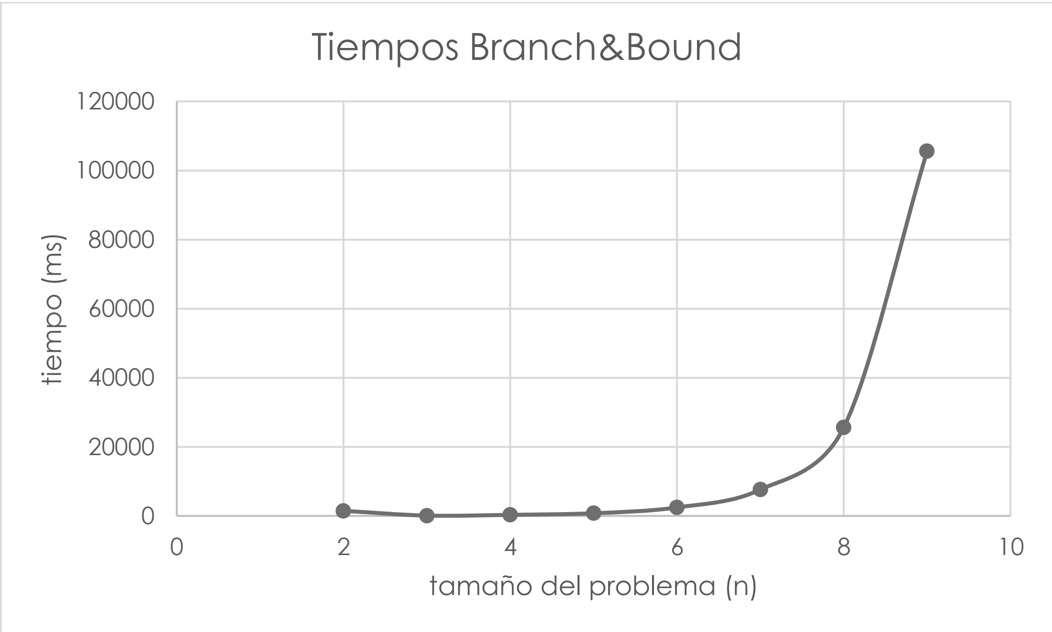
Características principales del ordenador:

Procesador:	i5-8250U
Memoria RAM:	8GB

Medidas de tiempo

n	tiempo_bt_bal (ms)	tiempo_bnb (ms)	nodes_bt_bal	nodes_bnb	zncc_bt_bal	zncc_bnb
2	38	1471	12	12	0.0	0.0
3	70	93	33	39	0.0257	0.0257
4	163	305	84	120	0.034	0.034
5	299	770	207	363	0.0327	0.0452
6	730	2443	504	1092	0.0461	0.0461
7	1761	7627	1221	3279	0.055	0.0607
8	4292	25588	2952	9840	0.0539	0.0547
9	11014	105589	7131	29523	0.0704	0.0711

Gráfica de Branch & Bound:



Justificación de la complejidad:

La implementación realizada de Branch & Bound (en la que ningún nodo es podado) proporciona los mismos resultados que el algoritmo de backtracking sin balanceo, ya que se genera el árbol de expansión completo. Por tanto, su complejidad será también $O(3^n)$.

Para comprobarlo, vamos a utilizar las mediciones realizadas y la fórmula $t_2 = f(n_2)/f(n_1) \cdot t_1$.

Datos: t_1=770ms, n_1=5, n_2=6:

$$t_2 = (3^6)/(3^5) \cdot 770 = 3 \cdot 770 = 2310 \text{ ms}$$

Comprobando con el valor obtenido para n=6 anteriormente:

2443 ms (valor calculado) ≈ 2310 ms (valor obtenido)

Por tanto, podemos asegurar que la complejidad del algoritmo es $O(3^n)$.

Comparación de resultados con los algoritmos de backtracking:

Resultados para n=7 imágenes:

```
TESTING BACKTRACKING SIN BALANCEO:
  -ZNCC: 0,052651
  -Contador: 3279
TESTING BACKTRACKING CON BALANCEO:
  -ZNCC: 0,049592
  -Contador: 1221
TESTING BRANCH AND BOUND:
  -ZNCC: 0,052651
  -Contador: 3279
```

Si representamos las mediciones de los algoritmos de Branch and bound y de backtracking con balanceo, observamos que BnB no mejora los tiempos (ya que dicho algoritmo genera el árbol completo).