

Práctica 6, Backtracking

Eduardo Blanco Bielsa

U0285176

3.1 Programación

(Mirar código adjuntado)

3.2 Medidas

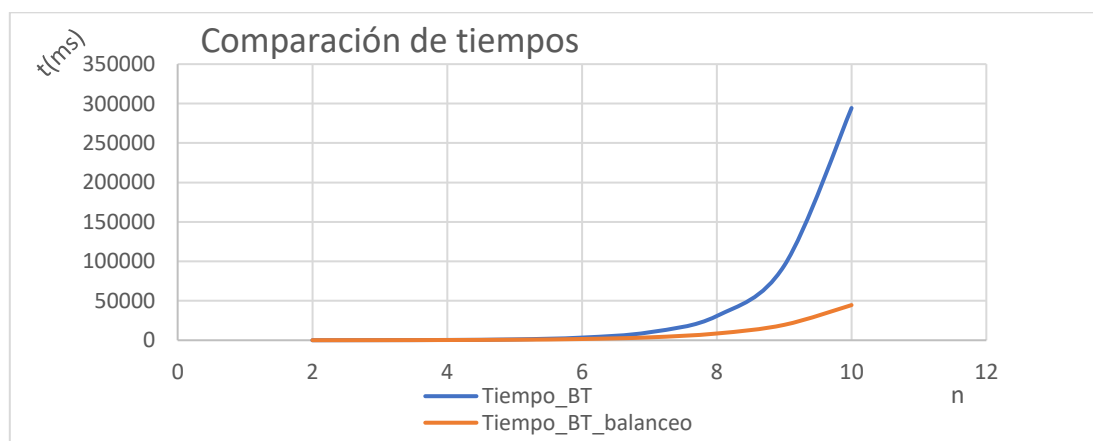
Aclaraciones

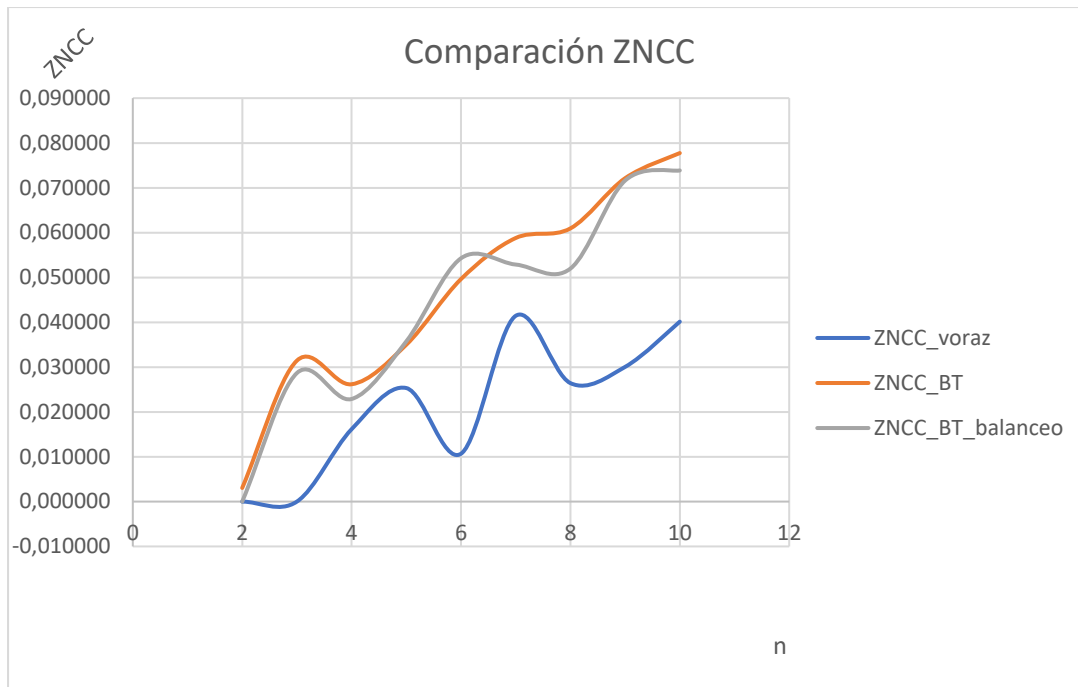
n: número de imágenes

Para backtracking, se empieza siempre en el nivel 0, es decir, en el nodo padre.

Puesto que el conjunto de datos es generado mediante un proceso aleatorio, los valores de ZNCC pueden variar para diferentes ejecuciones.

n	Tiempo_BT	Tiempo_BT_balanceo	ZNCC_voraz	ZNCC_BT	ZNCC_BT_balanceo
2	41	65	0,000000	0,003045	0,000000
3	127	149	0,000000	0,031446	0,028679
4	362	302	0,016147	0,026174	0,022887
5	1098	675	0,025330	0,034991	0,035745
6	3363	1586	0,010680	0,049632	0,054273
7	10095	3666	0,041438	0,058798	0,052879
8	30716	8568	0,026431	0,060937	0,051969
9	94463	19526	0,030043	0,072133	0,071607
10	294388	44560	0,040141	0,077747	0,073873





3.3 Preguntas

a) ¿Qué algoritmo proporciona mejores resultados y por qué?

En cuanto al **rendimiento**, el algoritmo que proporciona el mejor rendimiento es el voraz, sin embargo, es el que posee una peor calidad en el resultado ya que obtendrá un ZNCC muy bajo.

En cuanto a **resultados del ZNCC** (calidad del resultado), el que menos errores proporciona es el backtracking sin balanceo, pero a cambio, sacrifica el rendimiento, ya que como se deduce tanto de las gráficas como de la tabla, posee una complejidad exponencial.

b) ¿Qué algoritmo usarías para procesar un conjunto de datos con un millón de imágenes? Explica por qué.

Utilizaría el backtracking con balanceo, que aunque comete un ligero error (peor ZNCC) respecto al backtracking sin balanceo (ver gráficas), posee un rendimiento considerablemente bueno con respecto al backtracking sin balanceo. Es decir, el algoritmo voraz es bastante más rápido, pero proporciona un resultado mucho peor (ZNCC_voraz) y el backtracking sin balanceo proporciona un resultado prácticamente perfecto (ZNCC_BT), pero tiene un rendimiento pésimo. Por tanto, escogería **backtracking con balanceo** (ZNCC_BT_balanceo).

c) Determina la complejidad temporal del algoritmo backtracking sin considerar la condición de balanceo. Valida este análisis utilizando las medidas experimentales.

Posee una complejidad exponencial ($O(3^n)$), véase gráfica y código: *línea 268 PromediadoImagen*), ya que a mayor número de imágenes, mayor tiempo requerido para realizar todos los cálculos, puesto que este algoritmo ha de tener en cuenta todas las posibles combinaciones.

d) En términos de tiempo, ¿es ventajoso incluir la condición de balanceo? ¿afecta esta condición a la calidad de los resultados?

Respecto al tiempo, es ciertamente ventajoso con respecto al backtracking sin dicha condición, ya que se impone un criterio para ignorar las soluciones que no nos convengan. Sin embargo, esta condición afecta reducidamente a la calidad de los resultados puesto que es posible omitir soluciones mejores.