

PREGUNTAS SECCIÓN 9

Sesión práctica 9

Jonathan Arias Busto




UO283586

71780982-Y

Escuela de Ingeniería Informática - EII

INFORMACIÓN SOBRE ORDENADOR USADO PARA MEDIR TIEMPOS

Procesador	Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz
RAM instalada	16,0 GB (15,8 GB usable)

- ▼  Adaptadores de pantalla
 -  Intel(R) UHD Graphics
 -  NVIDIA GeForce GTX 1650

Se trata de un portátil con el modo de energía en “High performance”.

Cambiar la configuración del plan: High performance

MEDIDAS

La tabla de tiempos correspondiente a esta práctica:

n	t_bt	t_bt_poda	zncc_greedy	znnc_bt	znnc_bt_poda
2	13	13	0	0.00463	0.00850
4	223	43	0	0.03203	0.03638
6	1547	333	0.00511	0.05228	0.05088
8	4764	2676	0.02866	0.05751	0.04811
10	39557	18039	0.00508	0.07779	0.07355
12	TIEMPOS ALTOS	TIEMPOS ALTOS	0.03373	TIEMPOS ALTOS	TIEMPOS ALTOS

Como se puede ver la parte de Backtracking con poda nos ahorra mucho tiempo de computación al estar "podando"/ignorando soluciones que no cumplen la condición dada.

En cuanto a las soluciones estas son muy similares, por lo que es buena señal. Hay variaciones porque hay que tener en cuenta el factor de la generación aleatoria de ruido.

PREGUNTAS

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

Si hablamos de resultados como tal, la mejor solución viene dada por la versión de Backtracking sin poda, debido a que en la poda nos podemos estar saltando o cortando ramas de generación, que pueden ser una mejor solución, sin querer.

Si hablamos de el equilibrio entre tiempo de computación y resultados el mejor resultado es la versión de Backtracking con poda ya que se ahorra mucho tiempo en encontrar las soluciones más prometedoras.

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

Si tuviese un conjunto de un millón de imágenes no dudaría en descartar la opción de Backtracking sin poda, puesto que nos llevaría mucho más tiempo.

La duda reside en si la versión Greedy es una buena opción. En principio no es lo más óptimo porque el resultado no depende de la cantidad de imágenes que tengamos sino de la suerte que se tenga a la hora de generar soluciones. Por lo tanto mi elección sería la versión de Backtracking con poda siendo necesario un superordenador o un Cluster para ser capaz de computar tanta información.

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

Para determinar la complejidad necesitamos saber que es un algoritmo recursivo, por lo que la complejidad temporal vendrá dada por una factorial o exponencial. En este caso la complejidad es factorial.

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

Es muy ventajoso ya que nos ahorramos muchos pasos en la computación de los datos. Y puede llegar a afectar un poco, muy poco, negativamente dependiendo de lo buena o mala que sea la condición de poda. En este caso la condición de poda elegida ahorra muchos pasos al limitar las posibles soluciones a un rango mucho menor.