Informe Tarea 1

Luis Miguel Fros De Castro

15 de Octubre de 2018

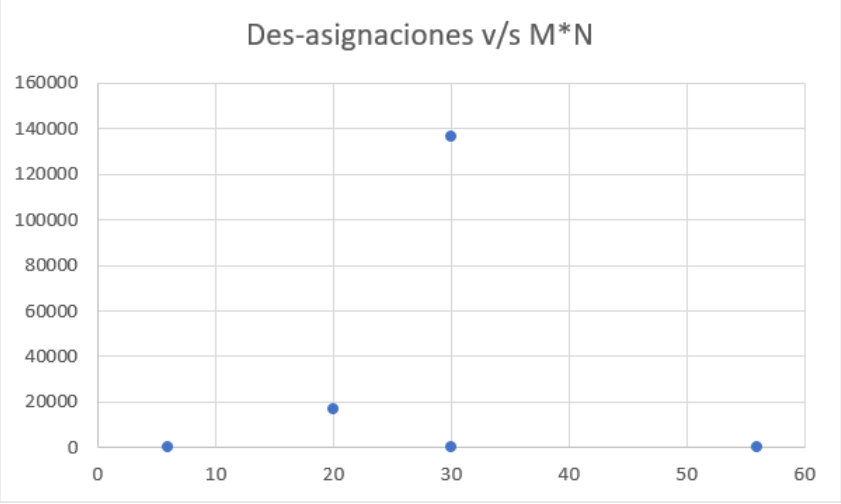
1. Análisis

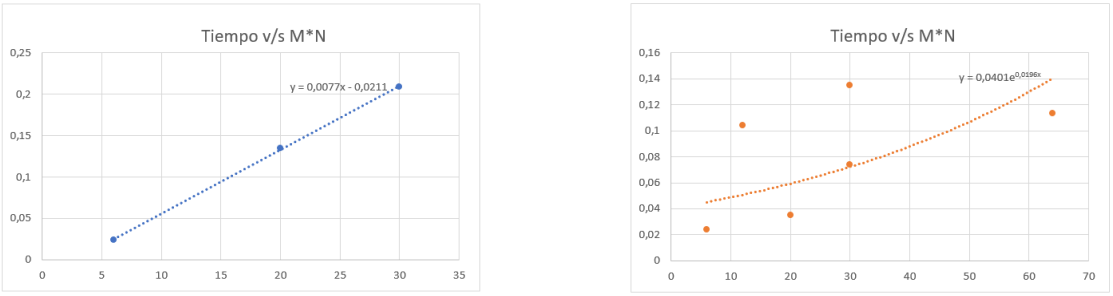
Gráficos

Figura 1: Tabla de tiempo en segundos para tests 0,1,2,3,5

M*N	test	muestra1	muestra2	muestra3	muestra4	media
6	0	0,023	0,019	0,026	0,029	0,02425
30	1	0,124	0,118	0,136	0,161	0,13475
20	2	0,034	0,033	0,027	0,046	0,035
30	3	0,22	0,026	0,023	0,026	0,07375
56	5	0,018	0,019	0,019	0,019	0,01875

Figura 2: Grafico para el numero de des-asignaciones dado un M*





(a) (b)

Figura 3: Graficos de tiempo v/s M*N

- Para analizar los datos, se tomaron 4 muestras para los tests 0,1,2,3,5 . El 4 no se toma en cuenta porque no tiene solución. Como se puede ver en la Figura 1.
- En la Figura 3 se ven graficados los resultados, donde en la figura 3.a se utilizaron los mismos tests de la tabla. Mientras que en la figura 3.b, se tomaron tiempos con otros tests creados con el propósito de tener mas datos.
- En la Figura 2, se puede ver la diferencia de des-asignaciones en el test 1 en comparación al test 3. El cual tiene un efecto claro con respecto al tiempo en la figura 3.b.
- Por lo tanto, queda claro que la complejidad de el algoritmo depende de las veces que se da un paso para atrás o se hace "backtrack".
- Para poder llegar a una estimación, podemos ver el arbol de asignaciones que se crea en este Puzzle. El factor de ramificación es un factor que va a determinar cuantos pasos avanza el algoritmo, este representa el numero de posibles jugadas dado un estado.
  - Si tomamos en cuenta una jugada como un imán positivo (+-) o uno negativo (-+), entonces al agregar un imán al tablero, se disminuye el factor de ramificación por 2.
  - Al principio, el tablero esta vació entonces existen M*N posibles movidas. Por lo tanto después de una jugada, van a haber M*N-2 posibles movidas.
  - M*N siempre es par según las condiciones del Puzzle original.
  - Si seguimos con esta lógica, cada nivel tiene respectivamente este factor de ramificación:

$$M * N, M * N - 2, M * N - 4, M * N - 6, ..., 2$$

  - Si $M * N = 12$, tenemos que los los factores son;

$$12, 10, 8, 6, 4, 2$$

  - Por lo tanto, si en el peor de los casos se des-asignan todos los nodos posibles, entonces se realizan $12 * 10 * 8 * 6 * 4 * 2$ movidas.
  - Generalizando,

$$(2 * 1) * (2 * 2) * (2 * 3) * (2 * 4) * (2 * 5) * (2 * 6) = (2^6) * 6!$$

  - Pero 6 es la altura del árbol, entonces obtenemos:

$$O(2^H * H!)$$

- Donde H es la altura del árbol, que es siempre $M * N/2$
- Es la complejidad de tiempo para el caso donde se realizan todas las designaciones posibles.
- No tenemos suficientes datos para poder confirmar que es efectivamente la complejidad, pero teóricamente no seria una subestación y se puede ver un comportamiento exponencial en la figura 3.b.

2. Optimizaciones

Poda

- Una poda muy simple que se puede realizar es tener una identificación por estado que sea fácil de calcular y no hacer una movida sobre una estado cuyo id este en tabla de hash. Esto es porque el orden de asignación con el algoritmo que use, **si importa**, entonces al estandarizar una representación de los estados del tablero podemos evitar que se repitan los estados. En la practica para hacer esto, lo ideal sera tener una función de hash que calcula según las posiciones y asignaciones un valor que es único y minimice las colisiones. Después guardar en una tabla de hash los estados ya calculados y cada vez que se quiere asignar un imán se debe revisar que este estado no exista en la tabla de hash (colisión). Si este ya existe en la tabla hash, entonces confirmar que efectivamente son iguales y dejar ignorar esa movida porque es un camino ineficiente.

Heurística

- Una heurística que sirva como estimación de cuantas movidas faltan para terminar el puzzle, seria al principio del puzzle, revisar el numero de restricciones que se deben cumplir y encontrar el **numero par** mas cercano (pero siempre menor) que representa el numero de imanes ue se podrían insertar. Usando esta estimación, se ordenan las posiciones del tablero, tal que en cada movida se maximice la cantidad de restricciones satisfechas. Esta heurística es admisible porque toma en cuenta el mejor caso que seria insertar directamente el numero necesario de imanes y nunca sobre estima el numero óptimo real. Por lo tanto nos garantiza que si el costo de calcularla es bajo, nos puede mejorar mucho la solución.

2.1. Propagacion

- Una propagación simple es que después de insertar un imán, revisar las filas y columnas afectadas por ese imán y ver si existe un espacio donde una imán se puede insertar cumpliendo las restricciones. Si ese es el caso, insertar el imán nuevo y actualizar el tablero. Esto aumenta la eficiencia porque dado que se cumple una restricción podemos actualizar los estados sin tener que iterar sobre todo el tablero de vuelta.