



Padrón: 109447

Fecha de entrega: 17/06/2024

Índice

1	Tercer problema	3
1.1	Paso 1	3
1.2	Paso 2	4
1.3	Paso 3	5
1.4	Paso 4	6
1.5	Paso 5	7
1.6	Paso 6	8
1.7	Paso 7	9
1.8	Conclusiones de entregas.....	10

1. Tercer problema

1.1. Paso 1

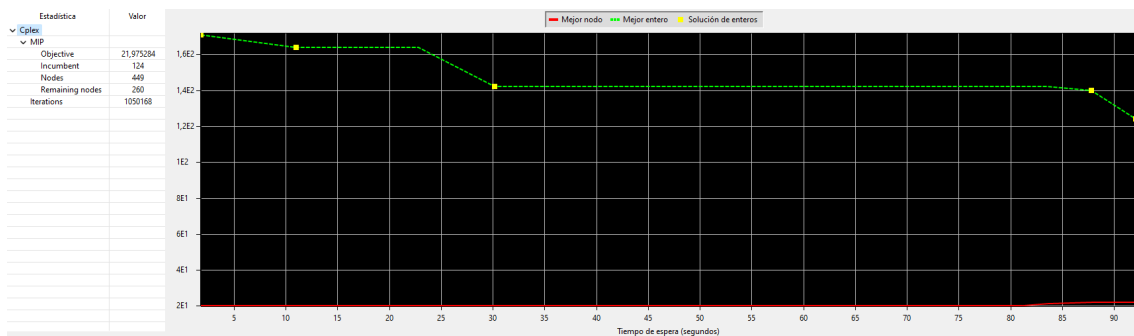
Para esta parte, decidí correr mis tres heurísticas sobre la instancia. Los resultados fueron:

- Heurística `coloreo_sin_pl`: 3.5 segundos - Tiempo total de lavado: 167.
- Heurística `solucion_sin_pl`: 0.002 segundos - Tiempo total de lavado: 171.
- Heurística `otra_solucion_sin_pl`: 0.15 segundos - Tiempo total de lavado: 213.

Las tres heurísticas utilizan 11 lavados (colores). A modo de observación, me gustaría comentar que la heurística que peor se comportó para esta instancia del problema (es decir, `otra_solucion_sin_pl`) fue la que mejor resultado obtuvo en la instancia de la segunda parte.

1.2. Paso 2

Para la corrida de este modelo, el mejor valor del funcional obtenido luego de 13 minutos fue de 117. En la parte de Statistics, podemos ver el valor del Best Node (la mejor solución continua) es de 103, y que el valor del Best Integer (la mejor solución entera) es 117. Viendo el gráfico, esto se puede visualizar en la línea verde, la cual representa justamente la mejor solución entera encontrada, y la línea roja, que es la mejor solución continua:



Como vemos en el gráfico, el modelo no llegó a alcanzar la solución actual en los primeros 90 segundos (se queda en 124). De todas formas, arranca desde 2760 y converge relativamente rápido a un valor aproximado a la solución actual, obteniendo 4 soluciones enteras en el camino. Luego, buscando en el Engine log el momento en el que se alcanza el valor de 117 para la solución, podemos ver que el modelo tardó entre 600 y 650 segundos (≈ 10 minutos) en llegar a la mejor solución encontrada, lo cual es bastante considerando que alcanzó un valor relativamente cercano en los primeros 90 segundos.

La alta cantidad de nodos explorados (18.183) y los nodos restantes (8.809) indican que el problema es bastante complejo y todavía hay muchas posibles soluciones por explorar. Luego, viendo el Engine log:

Nodes		Cuts/					
Node	Left	Objective	IInf	Best Integer	Best Bound	ItCnt	Gap
Elapsed time = 749,19 sec. (615883,53 ticks, tree = 106,40 MB, solutions = 20)							
12621	2879	116,0000	135	117,0000	103,0000	17544799	11,97%
13483	3710	107,0000	211	117,0000	103,0000	17903667	11,97%
14277	4426	103,0000	266	117,0000	103,0000	18224680	11,97%
14740	5016	105,0000	221	117,0000	103,0000	18496920	11,97%
15603	5705	105,0000	177	117,0000	103,0000	18816887	11,97%
16511	6458	105,0000	286	117,0000	103,0000	19136759	11,97%
17198	7612	108,0000	146	117,0000	103,0000	19605059	11,97%
17992	8287	104,1667	221	117,0000	103,0000	19972457	11,97%

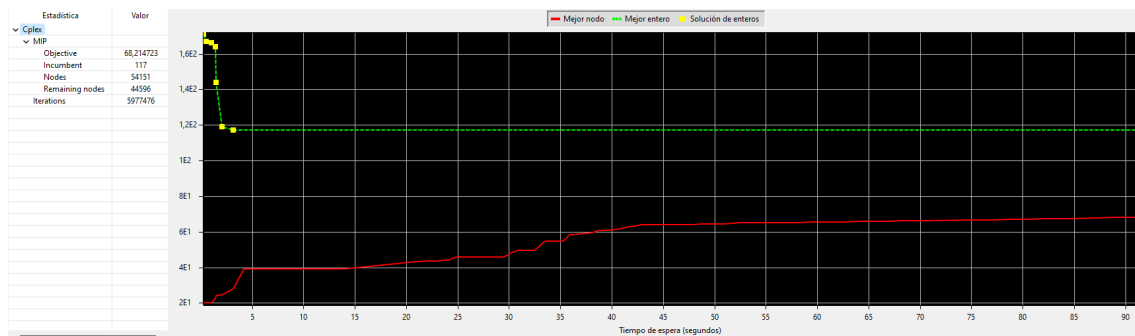
La Best Bound es una cota inferior para la solución óptima, y representa el óptimo del mismo problema sin forzar a las variables a ser enteras. Por lo tanto, sabemos que la solución óptima debe ser al menos 103, pero el modelo no encontró (al menos en el tiempo que le dimos) una solución factible que sea mejor que 117.

Teniendo en cuenta la cota inferior (Best Bound) y la mejor solución entera obtenida (Best Integer), podemos definir algo llamado gap. El gap o brecha es una medida de cuánto se acerca la solución actual al óptimo real, y nos permite justamente saber a qué distancia máxima estamos de dicho óptimo. En nuestra corrida, nos termina quedando en 11,97 % al cortar la ejecución. Esto indica que la solución encontrada es cercana a la óptima. Concretamente, indica que la solución actual es, en el peor de los casos, un 11,97 % superior al óptimo. Por lo tanto, podemos concluir que la solución encontrada nos sirve como una estimación bastante cercana a lo que sería la solución óptima real.

1.3. Paso 3

Saber que existe una solución que usa 15 lavados nos permite limitar la cantidad de colores que usamos en el modelo, haciendo que el número de combinaciones posibles de colores para cada par de nodos sea menor. Esto hace que haya una menor cantidad de soluciones posibles (dado que se descartan todas las soluciones válidas con más de 15 colores) y que el modelo requiera de menos tiempo para recorrer todas las soluciones. Para eso, simplemente ponemos en el modelo el valor de `limiteColores` en 15.

Nuevamente, el mejor valor del funcional obtenido luego de 12 minutos fue de 117. El Best Node ahora es de 106.63, ligeramente superior al caso anterior, y el valor del Best Integer, por supuesto, es el mismo. Gráficamente:



Viendo la estabilidad del gráfico (la mejor solución entera permanece constante a lo largo de casi todo el tiempo mostrado), podemos concluir que el modelo no encontró mejores soluciones a partir del segundo 4 aproximadamente. Esto podría indicar que está cerca de encontrar la solución óptima o que la solución actual lo es. Luego, viendo el Engine log,

```

      0      0      25,8736  1044      119,0000      Cuts: 490      24423      77,94%
*    0+      0      25,9381   996      117,0000      26,2563      77,56%
      0      0      26,0113  1037      117,0000      Cuts: 518      25444      76,15%
      0      0      26,0693   954      117,0000      Cuts: 502      26842      76,15%
      0      0      26,0946   959      117,0000      Cuts: 363      27469      66,67%
      0      0      26,0946   959      117,0000      Cuts: 584      28238      66,67%
      0      2      26,0946   913      117,0000      Cuts: 584      28238      66,67%
      Cuts: 584      28238
Elapsed time = 4,41 sec. (6438,90 ticks, tree = 0,02 MB, solutions = 10)

```

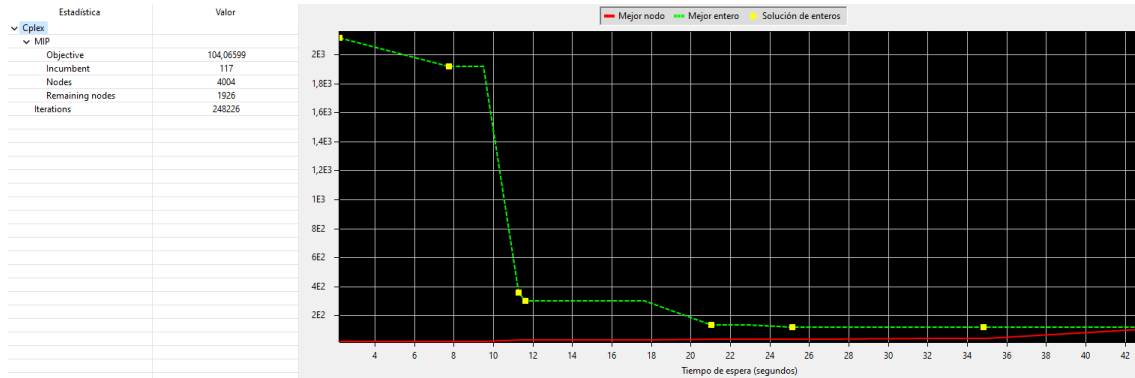
podemos observar que la solución actual de 117 ahora fue obtenida en menos de 4 segundos (lo cual puede notarse en el gráfico), una diferencia muy grande frente al caso anterior (que se obtuvo en aproximadamente 10 minutos). Además, podemos notar que, casi en el mismo tiempo que la corrida anterior, logró recorrer 205.000 nodos frente a los 18.000 del caso anterior, pero así y todo, no logra la convergencia. Ahora, observando la parte final del Engine log:

Nodes		Objective	IInf	Best Integer	Cuts/		Gap
Node	Left				Best Bound	ItCnt	
190655	92915	116,0000	215	117,0000	106,2777	18934197	9,16%
193066	94328	111,1429	355	117,0000	106,3178	19081200	9,13%
194972	96325	112,0000	334	117,0000	106,3333	19307774	9,12%
196901	97762	112,0000	293	117,0000	106,3864	19470456	9,07%
198660	99488	112,0000	335	117,0000	106,4275	19685352	9,04%
200366	101407	108,6154	391	117,0000	106,4656	19943785	9,00%
202410	102466	110,8889	410	117,0000	106,4930	20082664	8,98%
203900	104267	111,1818	408	117,0000	106,5635	20302637	8,92%
205581	105835	112,0000	239	117,0000	106,6188	20541845	8,87%

podemos notar que se redujo el gap con respecto a la corrida anterior. En este caso, al cortar la ejecución, nos marca que el gap es de 8,87% frente al 11,97% de la corrida anterior. No es una diferencia muy sustancial, pero nos indica que nuestra solución actual es cada vez más cercana a la solución óptima real.

1.4. Paso 4

Descomentando la restricción de simetría, vemos el siguiente gráfico



El modelo arranca desde un valor muy alto (2760 para ser exactos) y converge en 45 segundos a la solución óptima (que justamente es igual a las soluciones que habíamos obtenido cortando las ejecuciones de los modelos anteriores). En el recorrido, se encuentra con 7 soluciones enteras posibles (sin contar la óptima), las cuales están representadas por los puntos amarillos. En el Scripting log podemos ver que se usan 11 lavados para todas las prendas.

En este caso, al eliminar todas las simetrías, bastó con recorrer 4004 nodos para llegar a la solución óptima. Lo que sucedía antes de activar esta restricción es que había muchas soluciones alternativas para cada combinación posible de colores. Es decir, el modelo encontraba todas las soluciones posibles distintas que tienen esos colores, pero no nos sirven todas las soluciones que tienen igual valor de función objetivo, nos basta con una. Entonces, activar esta restricción nos permite deshacernos de todas esas soluciones (nodos) y tener un menor espacio de soluciones para recorrer, permitiéndole al modelo realizarlo en un menor tiempo (y terminarlo, cosa que no podía hacer antes ni en +10 minutos).

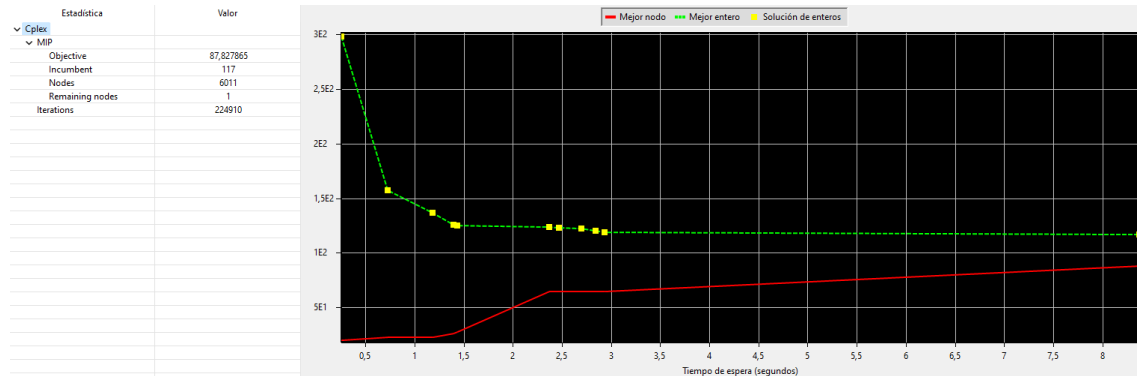
Viendo el Engine log:

Nodes		Cuts/					
Node	Left	Objective	IInf	Best Integer	Best Bound	ItCnt	Gap
Elapsed time = 40,55 sec. (57518,28 ticks, tree = 9,49 MB, solutions = 22)							
3996	1942	117,0000	99	118,0000	103,9624	236801	11,90%
* 4003	1935	integral	0	117,0000	103,9624	243270	11,14%
5387	1615	infeasible		117,0000	108,6405	297345	7,14%
7204	843	115,0000	108	117,0000	113,1297	336454	3,31%

Vemos que la Best Bound ahora es 113.13, superior a todas las demás cotas inferiores halladas en las corridas anteriores. Esto es porque, mientras CPLEX progresa en la búsqueda de soluciones enteras, va encontrando nuevas cotas inferiores para el óptimo, y, en este caso, logramos recorrer casi todo el espacio de soluciones (faltaron recorrer al menos 1926 nodos que, entiendo yo, no fueron expandidos por tener un valor peor en el funcional).

1.5. Paso 5

En este caso, el modelo comienza a partir de una solución con un valor mucho menor (300 para ser exactos) dado que no es necesario utilizar más de 15 colores, y en tan sólo 11 segundos converge a la solución óptima, siendo incluso más rápido que el caso anterior. Gráficamente:



Esto tiene sentido dado que estamos utilizando las ventajas de acotar la cantidad de colores a usar y de eliminar las soluciones alternativas, haciendo una gran reducción del espacio de soluciones. También es importante notar que, para esta solución, se recorren una mayor cantidad de soluciones y se deja simplemente una sin recorrer. Nuevamente, en el Scripting log vemos que se usan 11 lavados para la solución.

Viendo el Engine log podemos notar que la mejor cota se redujo con respecto al caso anterior, haciendo que el gap sea mayor (aunque sigue siendo el segundo más bajo).

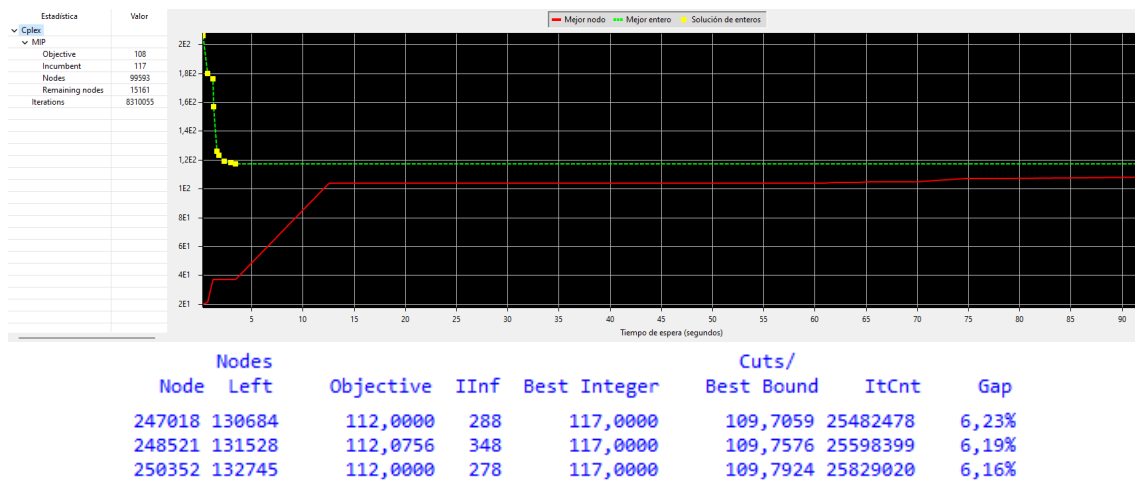
Nodes				Cuts/			
Node	Left	Objective	IInf	Best Integer	Best Bound	ItCnt	Gap
Elapsed time = 8,31 sec. (9787,65 ticks, tree = 0,02 MB, solutions = 19)							
6031	7	88,6023	333	117,0000	88,2773	230887	24,55%
6864	335	99,6157	314	117,0000	89,1219	296625	23,83%
8280	527	114,0000	109	117,0000	109,8111	381184	6,14%

1.6. Paso 6

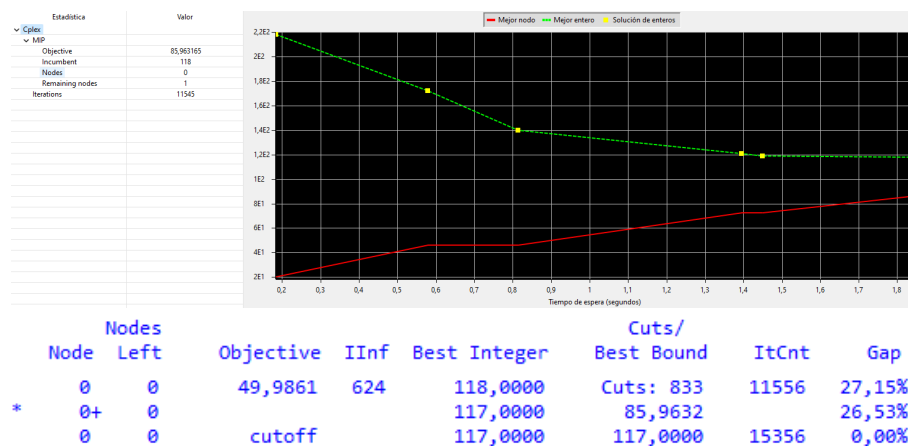
La primera diferencia que vemos entre el paso 3 y el paso 5 es que, al no tener la restricción que quita las simetrías, la cantidad de nodos recorridos (y los que faltan por recorrer) se incrementan en una proporción muy grande. Esto es por lo que dijimos antes (se incluye una cantidad importante de soluciones alternativas para cada combinación de colores válida) y hace que el modelo del paso 3 no pueda finalizar aún incluso dejándolo 12 minutos (el paso 5 terminó en 11 segundos).

Más allá de eso, el paso 3 logra llegar a la solución óptima del paso 5 en menos de 4 segundos, y se pasa el resto del tiempo recorriendo el espacio de soluciones en busca de una solución superadora (cosa que no logra hallar). Esto se puede ver en el gap del paso 3 y del paso 5, los cuales son similares (no hay grandes diferencias). La cantidad de colores usada también es la misma (11).

Para el caso sin usar las restricciones de simetría, los resultados son (lo corté a los 12 minutos):



Para el caso de usar las restricciones de simetría, los resultados obtenidos son:



Es importante remarcar que, para este último caso, logramos llegar a un gap del 0%, lo cual asegura que la solución encontrada es la óptima. Esto se da porque la mejor solución encontrada (117) y la mejor cota (bound) son iguales, lo que implica que no hay ninguna mejora posible sobre la solución actual y, por lo tanto, se ha alcanzado la solución óptima.

Para el caso del modelo sin las restricciones de simetría, la solución óptima se alcanza entre los 2 y 3 segundos, y el resto del tiempo simplemente continúa recorriendo el espacio de soluciones. Para el caso del modelo con dichas restricciones, se alcanza el óptimo en 2 segundos y medio.

1.7. Paso 7

En comparación con las heurísticas que corrí al principio del informe, podemos notar que coinciden en la cantidad de lavados usados (todas las soluciones utilizan 11 lavados). El problema se encuentra en cómo se forman esos lavados, lo cual incrementa en gran medida el tiempo de lavado total. Tomando la heurística que mejor desempeño mostró (tiempo de lavado de 167), notamos que la diferencia de tiempo con respecto a la solución óptima del modelo de PLE es de 50 unidades de tiempo. Y no solo eso, sino que dicha heurística se corre en un tiempo superior al modelo de PLE (aunque la diferencia es tan solo de un segundo).

En conclusión, ninguna de las heurísticas que utilicé para las partes anteriores logra tener un comportamiento cercano al modelo de PLE, siendo importante la diferencia entre los resultados. Dado el poco tiempo en el que corre el último modelo de PLE usado, es totalmente conveniente utilizarlo antes que usar alguna de dichas heurísticas.

Ahora, para el caso de los modelos de PLE que no llegaban a terminar, correr la instancia del problema en las heurísticas nos puede ayudar a concluir en si la solución obtenida con el modelo de PLE cortado a los 10 minutos es buena o no (más allá de basarnos en el gap). Si al cortar el modelo de PLE obtenemos una solución de 170 de tiempo total de lavado, es lógico pensar (mirando los resultados de las heurísticas) que la solución obtenida no es fiel a la solución óptima.

1.8. Conclusiones de entregas

En la primera entrega, me pareció un problema relativamente sencillo. De hecho, lo pensé inicialmente de forma parecida a un problema que vimos en clase. Tuve una confusión con respecto a cuál era el objetivo del problema, pero no hubo muchos más inconvenientes en esta parte.

En la segunda parte, como consecuencia del aumento importante del tamaño de la instancia del problema, lo que creía sencillo dejó de serlo. Este problema me llevó a tener que iterar bastante en las heurísticas e ir puliendo los enfoques realizados, porque sino se obtenían soluciones malas (con programación lineal directamente el problema no terminaba, aunque igual hice un par de modelos para probar). Me gustó el hecho de poder visualizar como se puede complicar un mismo problema al escalar en la instancia del mismo. Estuvo bueno el desafío de intentar obtener una heurística que devuelva buenos resultados para esa instancia.

En esta última entrega, si bien no hubo que realizar ningún modelo ni heurística, me permitió comprender la información que brinda el CPLEX en las diferentes solapas. También sirvió para ver visualmente la importancia de las restricciones de simetría en un problema de coloreo, algo que habíamos visto en clase pero que ahora queda mucho más grabado.