

TP: Entrega 3

[71.14] Modelos y optimización I

| Apellido, nombre | Padrón | Mail |
|------------------|--------|--------------------|
| Gomez, Joaquin | 103735 | joagomez@fi.uba.ar |

Enunciado

- 1) Corran su heurística sobre la instancia. Registren el resultado obtenido.
- 2) Prueben correr el código sin cambios, pueden detenerlo a los 10 minutos si no termina. Indicar en el informe todo lo que notan de esta corrida
- 3) Sabiendo que existe una solución que usa 15 lavados (se obtuvo mediante una heurística) ver cómo acelerar reduciendo el modelo (cantidad de restricciones), pueden detenerlo a los 10 minutos si no termina. Indicar en el informe todo lo que notan de esta corrida
- 4) Volviendo al modelo original (sin el límite de 15 lavados), descomentar la restricción "simetría". Indicar en el informe todo lo que notan de esta corrida
- 5) Modificar el modelo del punto anterior para que aproveche el límite de 15 lavados. Indicar en el informe todo lo que notan de esta corrida
- 6) Comparar el paso 3 y el 5, repetir la prueba sabiendo que existe una solución de 11 lavados
- 7) Comparar en el informe la heurística (paso 1) con la solución mediante programación lineal entera

Desarrollo

Primer paso

Se corrió la heurística sobre la instancia nueva (se puede observar en el archivo desarrollo.ipynb) y se obtuvo una solución donde las 138 prendas se agrupan en 11 lavados tardando 123 de tiempo. Esta solución corrió en 0.1s.

Dado que el enunciado en el paso 6 comenta que existe una solución que solo tiene 11 lavados, puedo sacar 2 conclusiones:

1. O tenemos la solución óptima.
2. O la heurística de construcción nos da un buen acercamiento y utilizando una heurística de mejoramiento podemos mejorar esta solución.

Segundo paso

Para este paso, ejecute el modelo sin ningún tipo de cambio y lo corte a los 10 minutos:

Implied bound cuts applied: 6657

Zero-half cuts applied: 300

Gomory fractional cuts applied: 4

Root node processing (before b&c):

Real time = 68,08 sec. (85900,46 ticks)

Parallel b&c, 12 threads:

Real time = 538,91 sec. (117024,43 ticks)

Sync time (average) = 129,75 sec.

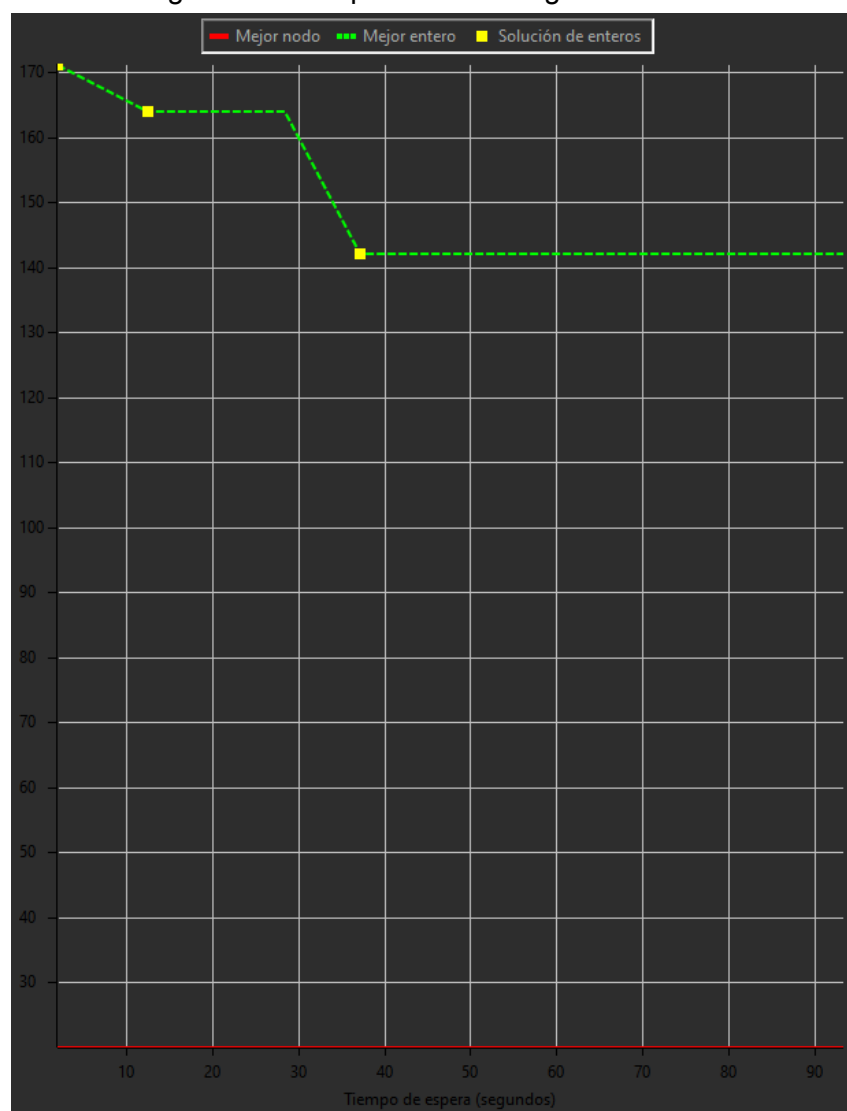
Wait time (average) = 0,00 sec.

Total (root+branch&cut) = 606,99 sec. (202924,89 ticks)

Tanto la solapa de soluciones como la de registro de scripts no muestran salida, por lo tanto no se encontró la solución óptima dentro de los 10 minutos.

Revisando el registro del motor, el mejor resultado que se encuentra es de 119 pero no logra llegar a una solución entera. Como nuestra heurística daba 123, ya sabemos que este no es el óptimo.

Adjunto a continuación el gráfico de los primeros 90 segundos de la sección de estadísticas.

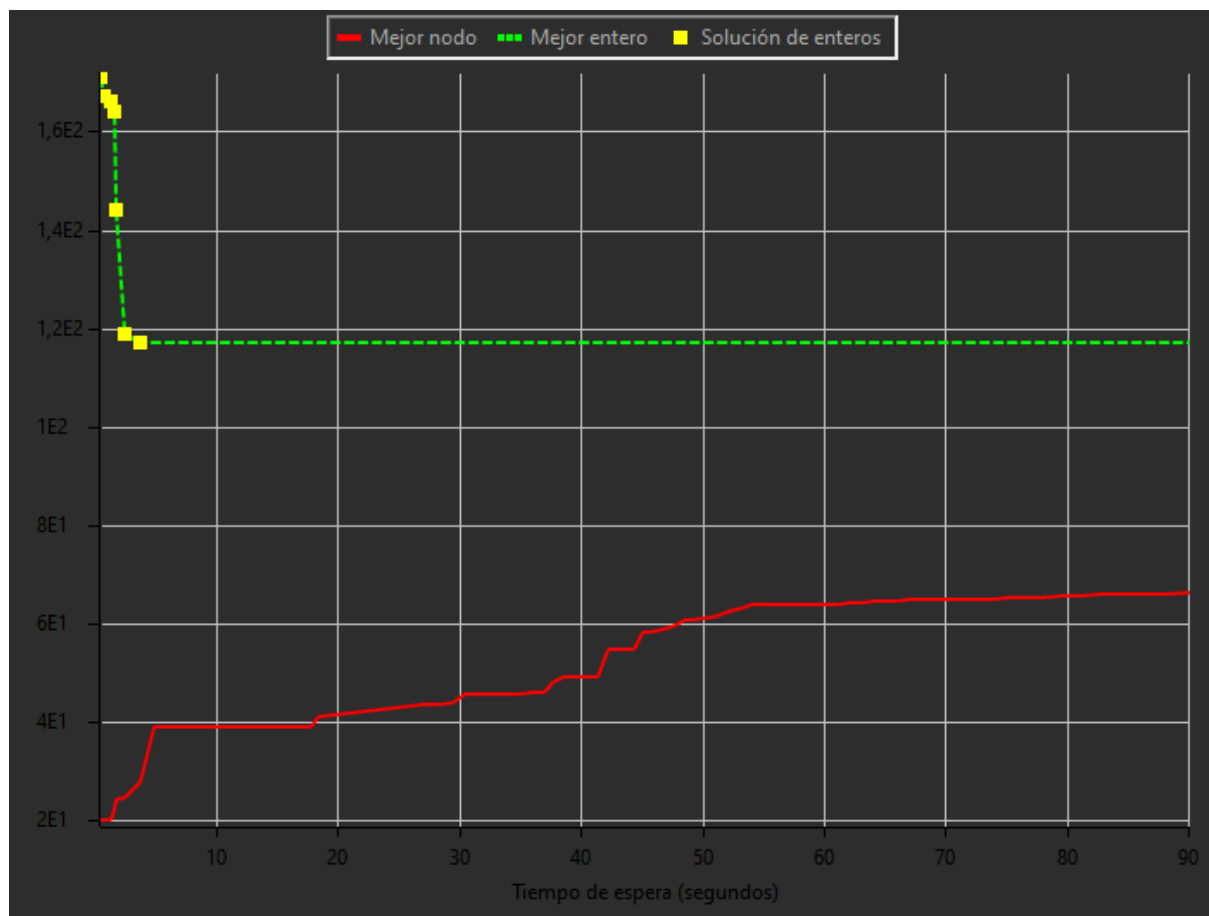


Tercer paso

Para realizar este paso, lo primero que hago es darle valor 15 (en lugar de n) a la variable "limiteColores" para reducir el modelo dado.

Nuevamente, tanto la solapa de soluciones como la de registro de scripts no muestran salida, por lo tanto no se encontró la solución óptima dentro de los 10 minutos.

Viendo el gráfico de estadísticas, podemos observar que se llega a una solución mejor mucho más rápido que en el caso anterior. Esto se debe a la reducción que realizamos sobre el modelo.



Revisando el registro del motor, si bien no se llega a una solución entera, comenta que el tiempo de lavado está en 117 que si de encontrarse una solución entera es mejor que la solución encontrada en la primera heurística.

Agrego parte de lo que informa el registro del motor:

Elapsed time = 413,84 sec. (222594,03 ticks, tree = 1715,82 MB, solutions = 10)

| | | | | | | | |
|--------|-------|----------|-----|----------|----------|----------|-------|
| 153810 | 61203 | 114,7832 | 267 | 117,0000 | 106,0000 | 14780078 | 9,40% |
| 155582 | 63046 | 112,0000 | 373 | 117,0000 | 106,0000 | 15007751 | 9,40% |
| 157605 | 65082 | 114,0000 | 318 | 117,0000 | 106,0000 | 15280142 | 9,40% |
| 159637 | 66565 | 112,2238 | 311 | 117,0000 | 106,0000 | 15475176 | 9,40% |
| 161509 | 67862 | 112,6500 | 375 | 117,0000 | 106,0000 | 15651626 | 9,40% |

| | | | | | | |
|--------------|----------|-----|----------|----------|----------|-------|
| 162880 69663 | 113,0000 | 404 | 117,0000 | 106,0000 | 15931457 | 9,40% |
| 164830 71452 | 112,0000 | 270 | 117,0000 | 106,0000 | 16143933 | 9,40% |
| 166177 72283 | 112,0000 | 339 | 117,0000 | 106,0000 | 16303408 | 9,40% |
| 167432 73430 | 112,0000 | 313 | 117,0000 | 106,0000 | 16509693 | 9,40% |
| 168974 74711 | 114,9625 | 376 | 117,0000 | 106,0000 | 16717036 | 9,40% |

Elapsed time = 543,28 sec. (260766,59 ticks, tree = 2307,12 MB, solutions = 10)

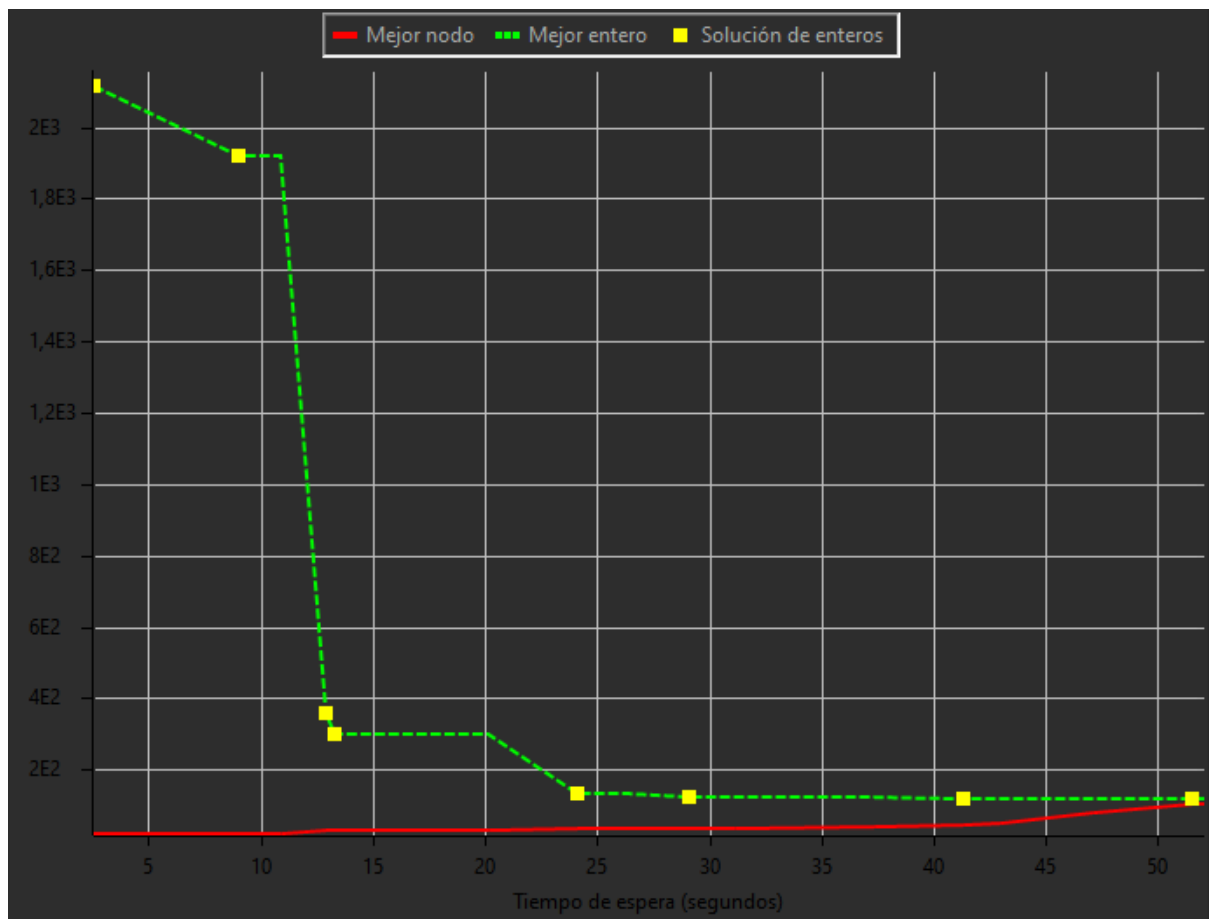
Nodefile size = 253,90 MB (227,36 MB after compression)

| | | | | | | |
|--------------|----------|-----|----------|----------|----------|-------|
| 170576 76222 | 114,0000 | 377 | 117,0000 | 106,0000 | 16956124 | 9,40% |
| 172686 77017 | 106,6626 | 432 | 117,0000 | 106,0009 | 17059406 | 9,40% |
| 174664 78912 | 108,2885 | 407 | 117,0000 | 106,0294 | 17250687 | 9,38% |

Cuarto paso

Para este paso, volvemos la restricción "limiteColores" a n y descomento la restricción de simetría.

A diferencia de los dos pasos anteriores, acá si llegamos a una solución y se puede ver en el gráfico de estadísticas que es muy rápido (finaliza en 56s a diferencia de los dos anteriores que a los 10 minutos no tenían solución)



Si vemos la pestaña de registro de scripts tenemos la solución que es la siguiente:

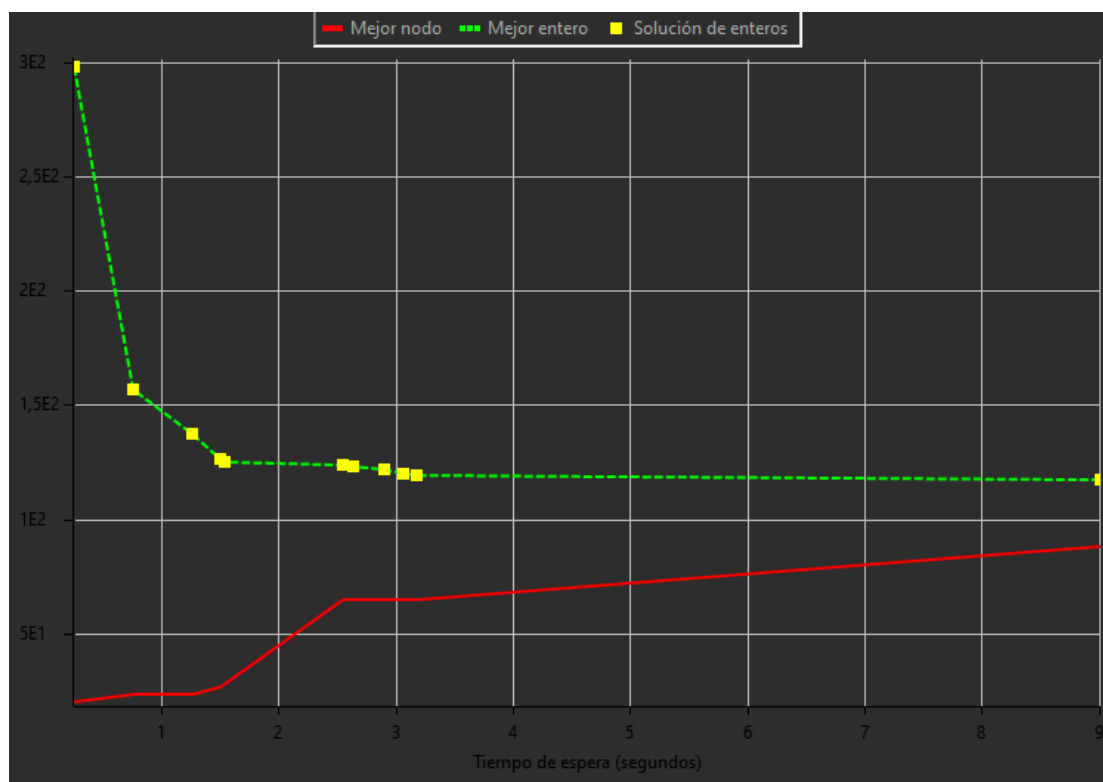
solution: 117 /size: 138 /time: 281729.906

Nodo 1: 1
Nodo 2: 1
Nodo 3: 2
Nodo 4: 1
Nodo 5: 4
Nodo 6: 1
Nodo 7: 3
Nodo 8: 2
Nodo 9: 1
...

Leyendo la solución, podemos observar que utiliza 11 lavados (que leyendo el paso 6 es correcto). Dado que encontramos una solución óptima, esta es mejor que la encontrada por mi heurística por una diferencia de 6 unidades (123 vs 117). Esta diferencia es de aproximadamente un 5%.

Quinto paso

Para realizar este paso, vuelvo a darle valor 15 (en lugar de n) a la variable “limiteColores” para reducir el modelo dado.



Si vemos la pestaña de registro de scripts tenemos la solución que es la siguiente:

solution: 117 /size: 138 /time: 282359.578

Nodo 1: 3
Nodo 2: 1
Nodo 3: 1
Nodo 4: 4
Nodo 5: 1
Nodo 6: 5
Nodo 7: 4
Nodo 8: 1
Nodo 9: 1

...

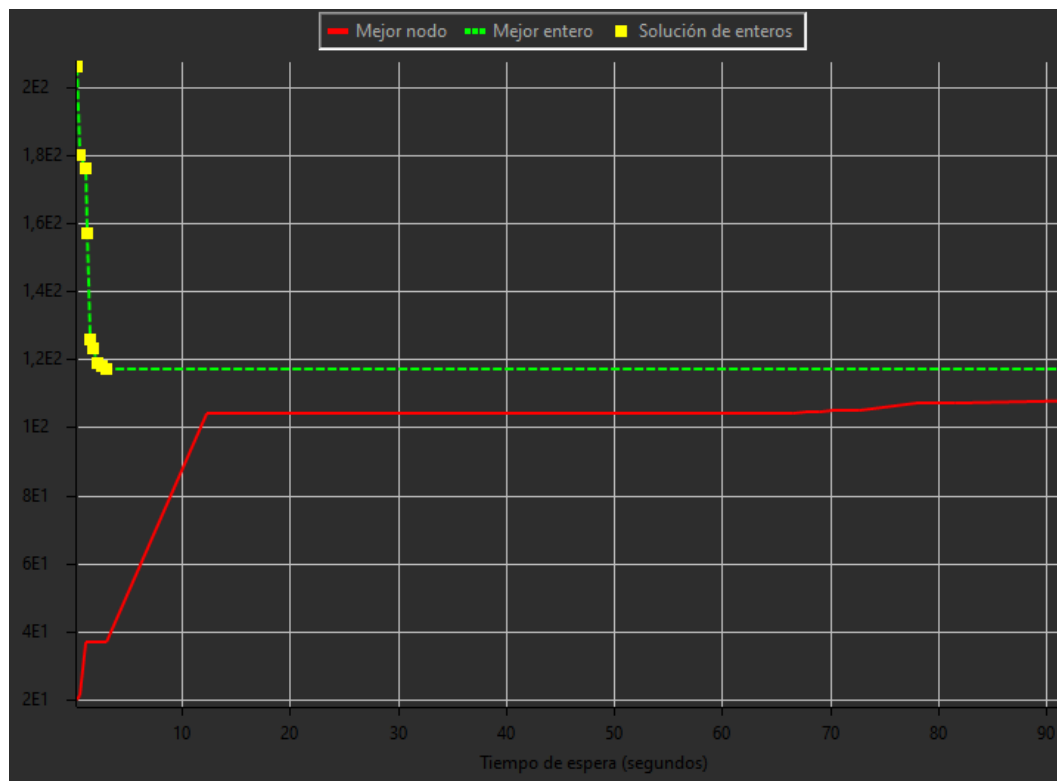
Leyendo la solución, podemos observar nuevamente que utiliza 11 lavados y tarda los mismos 117.

La principal diferencia que encontré es que corre muchísimo más rápido, tardando un poco menos de 12 segundos. Comparando el paso 4 y el paso 5, el mismo salto abrupto se genera con diferencia de que en el paso 4 se genera a los 12 segundos y en el paso 5 se genera antes del segundo.

Sexto paso

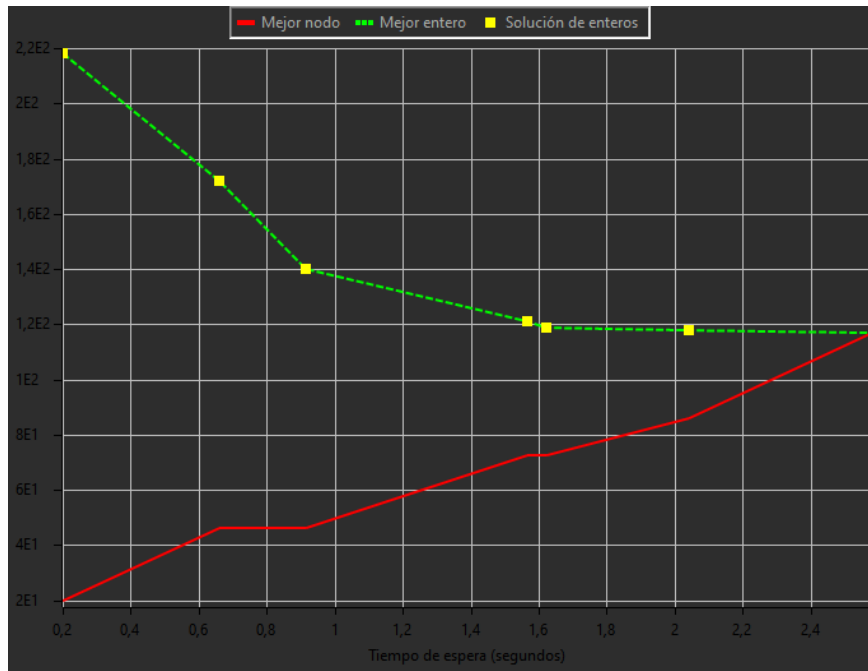
Para este paso, corro el modelo con la restricción "limiteColores" en 11. Primero lo corro con la simetría desactivada y luego con la simetría activada.

Con simetría desactivada:



Debido a tener la simetría desactivada, queda calculando las diferentes soluciones que tienen 117 como valor de funcional y en los 10 minutos no llega a una solución entera. Por lo tanto, esta corrida, no genera un resultado óptimo.

Con simetría activada:



Se puede observar que se llega a la solución óptima en menos de 3 segundos
Si vemos la pestaña de registro de scripts tenemos la solución que es la siguiente:

solution: 117 /size: 138 /time: 283101.281

Nodo 1: 2

Nodo 2: 1

Nodo 3: 4

Nodo 4: 1

Nodo 5: 1

Nodo 6: 4

Nodo 7: 4

Nodo 8: 5

Nodo 9: 1

...

Por lo tanto, se llega al óptimo y en un tiempo muy rápido (pero no más rápido que la heurística creada)

Séptimo paso

La solución encontrada con la heurística genera 11 lavados que tardan 123 y tarda 0.1 segundos en correr. En cambio, la solución óptima más rápida (con simetría activada y máximo de colores en 11) genera 11 lavados que tardan 117 y tarda un poco menos de 3 segundos.

Comparando las diferentes corridas que se realizaron, se llega a la conclusión de que la simetría es un factor clave para que el modelo pueda correrse de forma rápida. Esto se debe a que, con la simetría activada, el modelo puede reducir significativamente el espacio de búsqueda al evitar soluciones redundantes y equivalentes. Sin el uso de simetría, y dada la cantidad de elementos presentes, el número de soluciones posibles aumenta considerablemente, lo que dificulta y ralentiza el proceso de cálculo.

Dicho esto, de tener que elegir un algoritmo, elegiría la heurística debido a su velocidad y simplicidad para encontrar una solución que, en este caso puntual, solo difiere en un 5% de la solución óptima. Otra opción, es utilizar esta heurística para reducir nuestro modelo de PLE y así conseguir un resultado más rápido.

Informe final

En este trabajo práctico, a lo largo de sus distintas entregas, aprendí acerca de problemas de programación lineal entera. En las entregas 1 y 2 realice una heurística que, en la entrega 3, pude probar que da un resultado muy aceptable ya que solo difiere un 5% del algoritmo óptimo.

En la entrega 3, trabajamos con modelos de programación lineal entera y pudimos ver cómo mejorar un modelo inicial para poder hacer un modelo más reducido que nos ayude a que pueda terminar de correrse el algoritmo. Para realizar esto, pude visualizar cómo combinar heurísticas con modelos de PLE ya que en el paso 3 se nos comenta que el número máximo de lavados es 15 y fue obtenido por una heurística. Es decir, podemos utilizar una heurística, que por lo general son reglas sencillas, para poder reducir nuestro modelo de PLE y de esta forma hacer que sea resuelto mucho más rápido.