

FIUBA

Modelos y Optimización I

Cuarta Entrega

Apellido y Nombre: Su, Agustina Doly

Padron: 105708

Fecha de Entrega: 21 de Noviembre de 2022

Introduccion:

En este informe se desarrollará la resolución de la cuarta entrega del trabajo práctico correspondiente a la materia Modelos y Optimización I. El problema consiste en encontrar la combinación de prendas para que requiera la menor cantidad de tiempo posible del lavarropas, dado que el lavarropas puede lavar varias prendas a la vez y algunas prendas no se pueden lavar juntas. El objetivo de esta entrega es analizar las distintas soluciones a partir de un mismo modelo con pequeñas variaciones, y obtener conclusiones a partir de los resultados obtenidos.

Enunciado:

En esta cuarta entrega se pide que busquen el OPTIMO y lo suban a modelosuno.okapii.com, es MUY recomendable usar el código provisto CPLEX https://modelosuno.okapii.com/content/modelos_wvcp.zip

Armar un informe con cada uno de los pasos, incluir gráficos (solapas "Statistics", "Engine log", "Scripting log", etc.) y todo lo que consideren pertinente / interesante. El gráfico de "Statistics" tomarlo en los primeros 90 segundos.

Pasos:

- 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

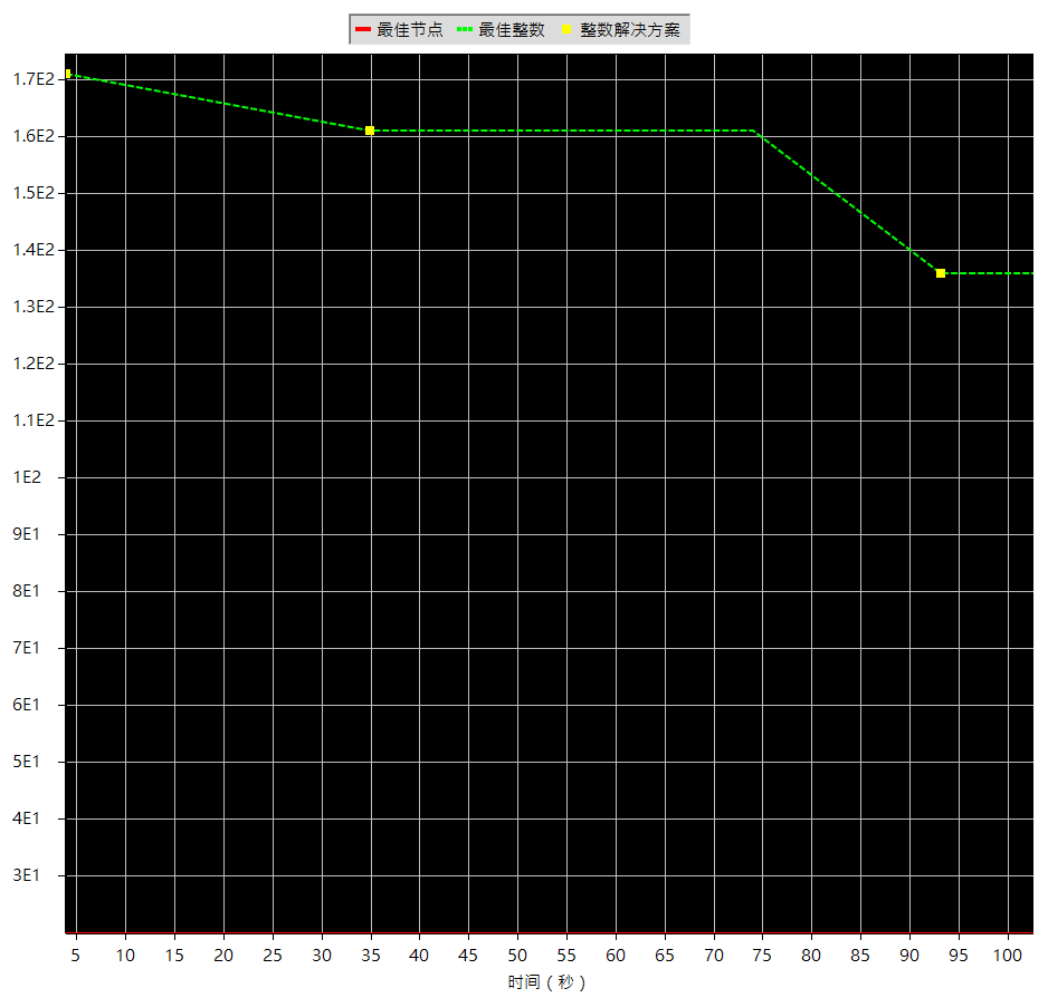
Resolución:

Paso 1:

En este paso se pide correr el algoritmo que propuse en la segunda entrega, que se encuentra en el archivo *entrega2.py* en la carpeta correspondiente. Se usó una técnica similar al coloreo de grafos basado en los tiempos individuales y la cantidad de incompatibilidades que tiene cada prenda. La solución se obtuvo muy rápido, con un tiempo óptimo de 123 segundos y utilizando 11 lavados (la solución se encuentra en el archivo *solucion4.txt*).

Paso 2:

Gráfico:



Engine Log:

1495	1019	82.0000	696	119.0000	37.0000	2707439	68.91%
1509	1000	56.0000	923	119.0000	37.0000	2653119	68.91%
1528	1044	88.0000	568	119.0000	37.0000	2753997	68.91%
1550	1042	109.0000	253	119.0000	37.0000	2752476	68.91%
1586	1051	117.0000	112	119.0000	37.0000	2757717	68.91%
1635	1068	105.9761	463	119.0000	37.0000	2819438	68.91%
1663	1108	114.0000	185	119.0000	37.0000	2925775	68.91%
1677	1081	47.7547	736	119.0000	37.0000	2882906	68.91%
1688	1066	65.0000	882	119.0000	37.0000	2853572	68.91%
1699	1157	58.4794	815	119.0000	37.0000	3035829	68.91%
Elapsed time = 697.03 sec. (147384.63 ticks, tree = 13.95 MB, solutions = 26)							

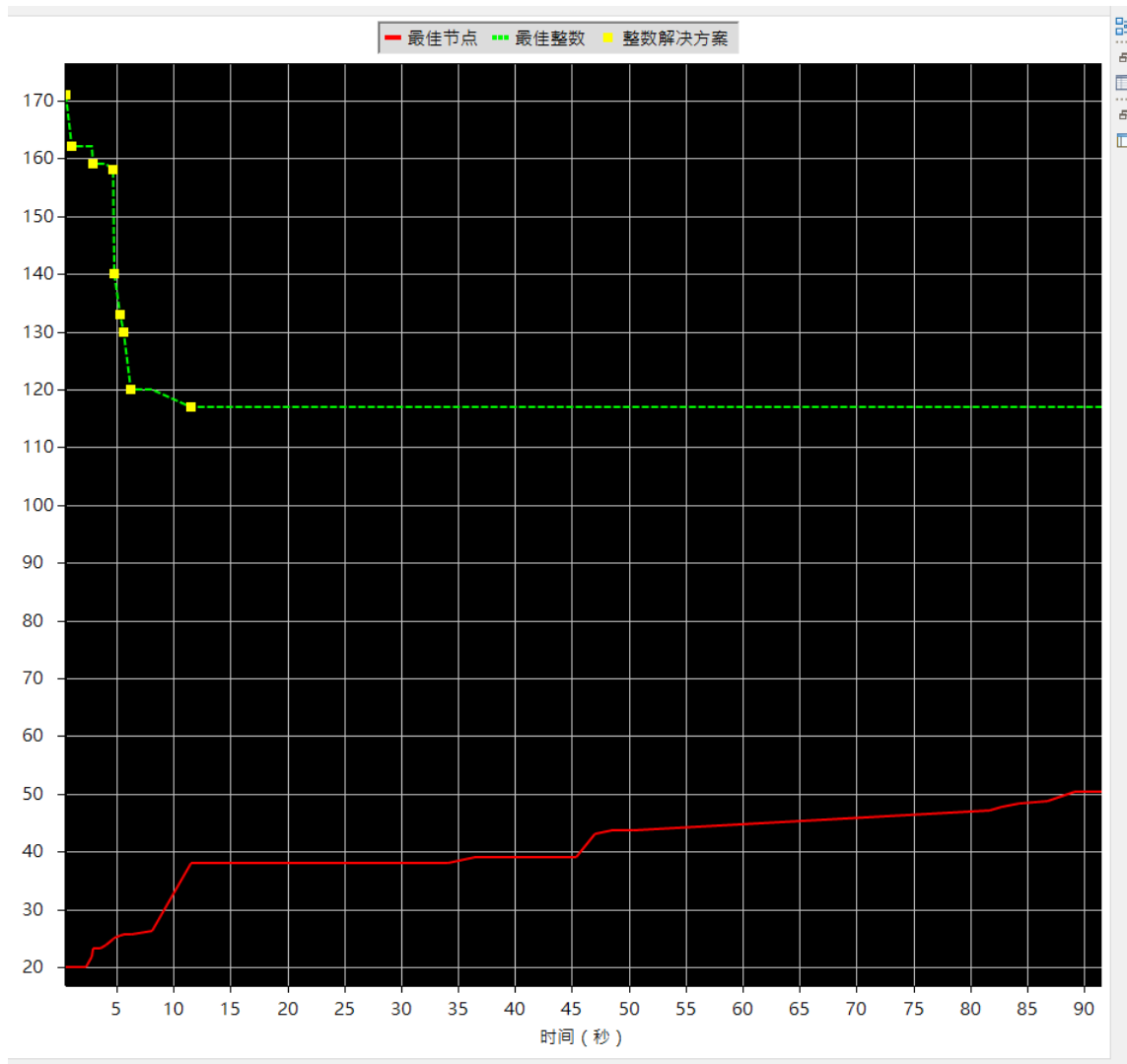
Script Log:

solution: 119 /size: 138 /time: 7828.609

Mirando el gráfico de Statistics se puede ver que la mejor solución desciende lentamente (de 171 a 136 en 90 segundos), encontrando únicamente dos soluciones. La cantidad de iteraciones realizadas es 29019. Observando el engine log, a los 10 minutos todavía no se logró obtener la solución óptima, siendo la mejor solución hasta el momento 119. El gap se mantuvo en 68.91%, y el best bound en 37. Como no se encontró la solución óptima, en el script log figura la mejor encontrada hasta el momento, que es 119.

Paso 3:

Gráfico:



Engine Log:

69136	12	106.0000	376	117.0000	104.0000	7427525	11.11%
69231	75	106.0000	359	117.0000	104.0000	7437018	11.11%
69374	193	112.0000	158	117.0000	104.0000	7463015	11.11%
70699	711	114.0000	143	117.0000	105.0000	7522245	10.26%
73790	3852	113.6667	331	117.0000	105.0000	7751827	10.26%
76412	5988	109.0000	359	117.0000	105.0000	7929199	10.26%
78221	7476	106.1140	417	117.0000	105.0000	8096240	10.26%
80069	9232	111.0000	387	117.0000	105.0000	8289648	10.26%
81783	10520	114.5455	297	117.0000	105.0957	8456645	10.17%
83743	12335	111.5839	324	117.0000	105.1623	8660634	10.12%

Elapsed time = 608.53 sec. (188870.21 ticks, tree = 316.44 MB, solutions = 14)

Script Log:

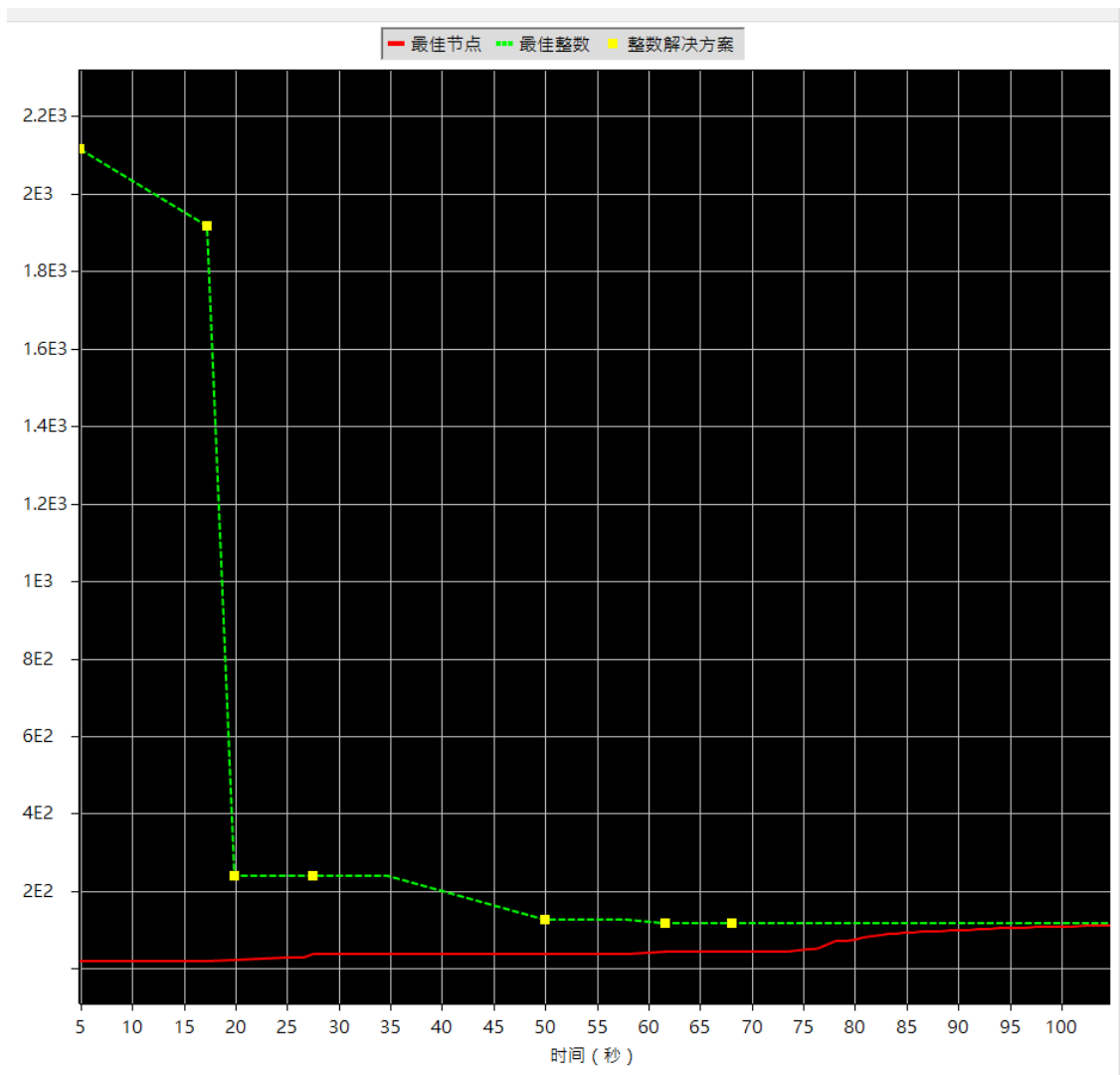
solution: 117 /size: 138 /time: 9171.625
--

En este caso se modifica el código para que el límite de colores (cantidad de lavados) pase a ser 15, en lugar de la cantidad de nodos. Con esto se espera que el tiempo para llegar a la mejor solución sea menor al caso anterior, o de incompatible en caso que no se pueda solucionar el problema en 15 lavados.

A partir del gráfico se puede ver que el tiempo total disminuye rápidamente, llegando al 117 a los 12s y luego permanece igual. La línea roja sube hasta el 50.275, y la cantidad de iteraciones realizadas es mayor a la corrida anterior. Esto se puede deber a que, al tener menos variables, el modelo tarda menos en computar cada iteración. Mirando el engine log, se puede notar que luego de los 11 segundos la mejor solución permanece en 117, teniendo que hacer un restart dos veces, reduciendo la cantidad de filas, columnas y binarios cada vez que se ejecuta. Al cabo de 10 minutos, el programa no encontró la solución óptima, quedando con un gap de 11.11%. Si bien no se logró encontrar la mejor solución, esta reducción de variables ayudó a mejorar mucho la performance.

Paso 4:

Gráfico:



Engine Log:

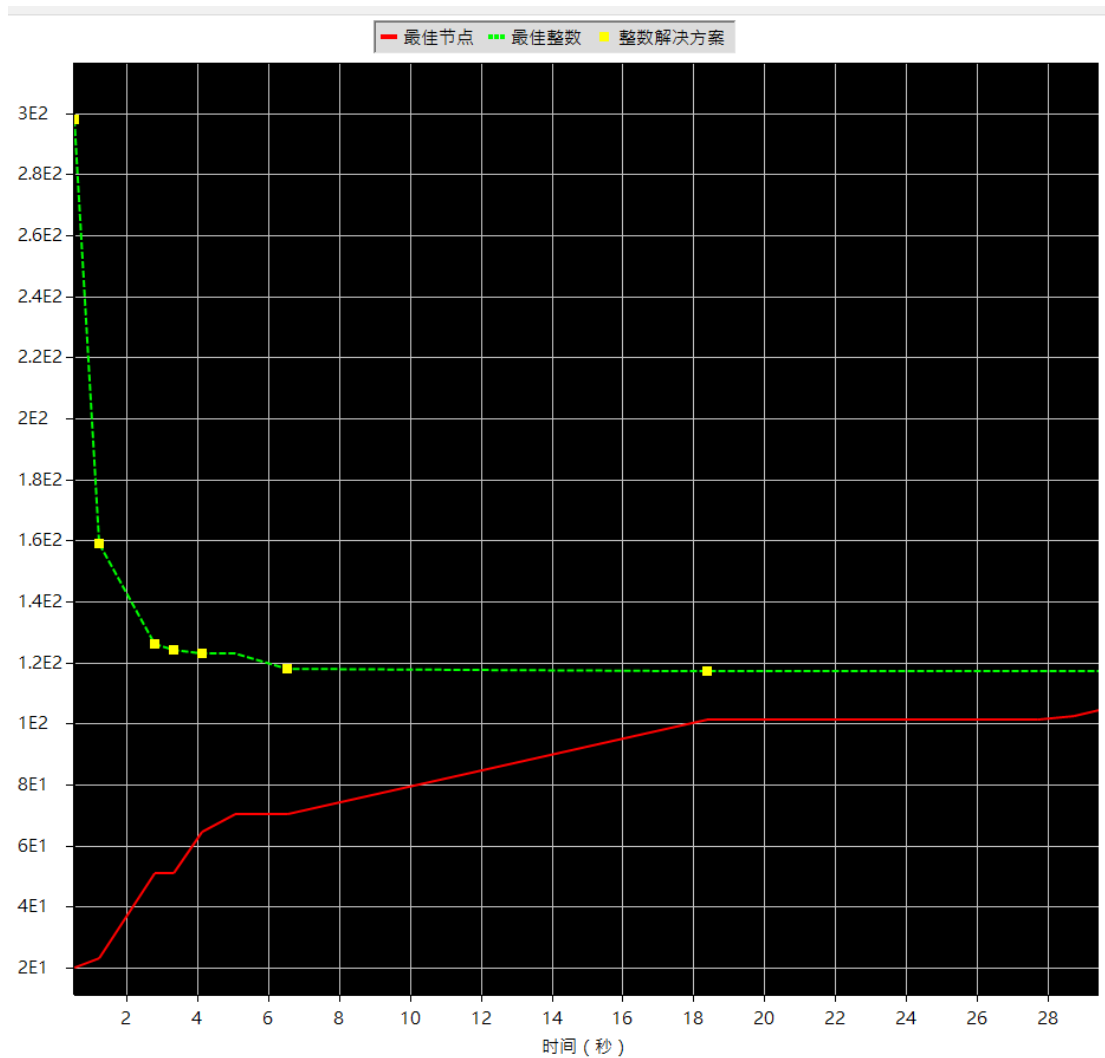
```
Elapsed time = 81.88 sec. (44809.58 ticks, tree = 76.65 MB, solutions = 24)
1981 1026 109.2778 150 117.0000 91.7011 138584 21.62%
2609 1259 112.0921 231 117.0000 97.4807 164931 16.68%
3258 1486 111.0000 136 117.0000 101.3444 192079 13.38%
4065 1697 cutoff 117.0000 105.7273 217807 9.63%
4819 1551 113.0000 154 117.0000 108.1250 240382 7.59%
5707 1385 infeasible 117.0000 110.5000 267637 5.56%
6989 376 cutoff 117.0000 114.7222 292041 1.95%
```

En este paso se pide descomentar la parte del código correspondiente a la simetría, manteniendo el modelo original (sin el límite de 15 lavados).

Mirando el gráfico se puede ver que, si bien tarda más en llegar al 117 (el mejor resultado hasta ahora), las líneas verde y roja convergen más rápido. Observando el engine log, se puede ver que el gap disminuye rápidamente después de detectar las simetrías, llegando a la mejor solución a los 107 segundos. Como esta vez se pudo obtener el resultado, en el script log figura el color que debería tener cada nodo (es decir, el lavado al que corresponde cada prenda), así también la solución óptima, que es 117.

Paso 5:

Gráfico:



Engine Log:

Elapsed time = 19.56 sec. (10141.01 ticks, tree = 0.02 MB, solutions = 8)							
3536	15	101.5000	279	117.0000	101.4461	265760	13.29%
3714	83	105.4286	274	117.0000	101.4461	281747	13.29%
4341	351	106.5106	256	117.0000	101.4461	333315	13.29%
5699	659	cutoff		117.0000	106.0240	427378	9.38%

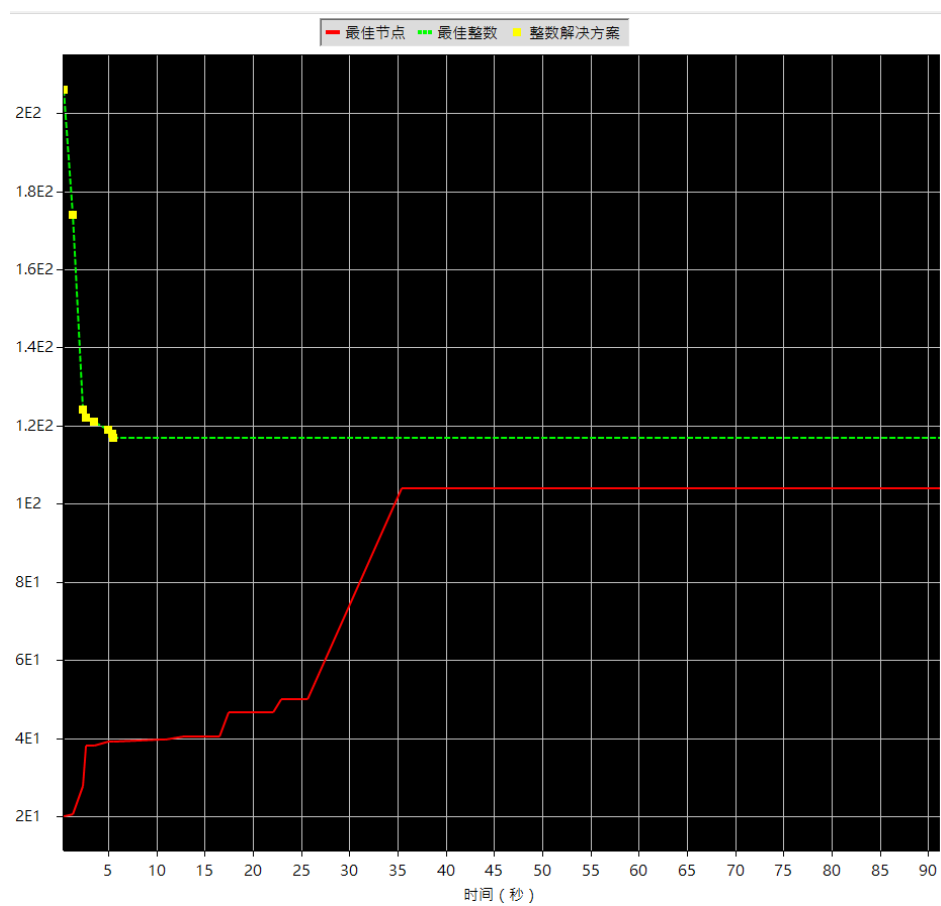
Acá se pide que se ejecute el modelo con el límite de 15 lavados, manteniendo la simetría del paso anterior. Basado en las experiencias, esta vez debería tardar menos en llegar a la solución.

A partir del gráfico se puede ver que el resultado desciende casi de manera exponencial y llega al resultado a los 32 segundos. En el engine log, la cantidad de soluciones es mucho menor a los pasos anteriores. Mirando el script log, se puede ver que los nodos están agrupados de manera distinta al paso anterior, pero el tiempo requerido es el mismo y la cantidad de lavados necesarios también.

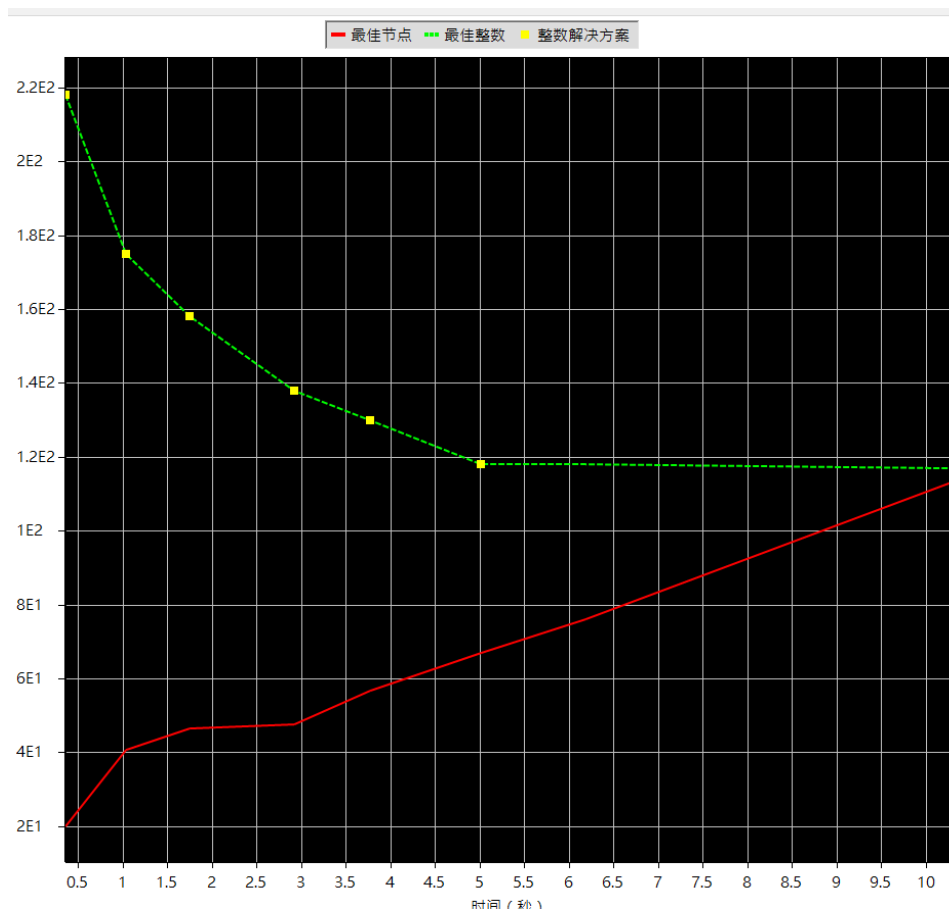
Paso 6:

Gráfico:

Asimetría:



Simetria:



Engine Log:

Asimetria:

87121	67854	114.0094	143	117.0000	104.2244	8819566	10.92%
88869	69557	109.0000	209	117.0000	104.3000	9048159	10.85%
90546	70431	111.7330	336	117.0000	104.3730	9172139	10.79%
91865	71848	108.1071	366	117.0000	104.4444	9398349	10.73%
93314	72740	115.1250	225	117.0000	104.4592	9568233	10.72%
94963	74046	107.7791	357	117.0000	104.5251	9784704	10.66%
97156	75913	112.0992	280	117.0000	104.6047	9999988	10.59%
99258	77451	cutoff		117.0000	104.6955	10195190	10.52%
101752	78920	107.2364	257	117.0000	104.7933	10365504	10.43%
104156	80766	112.2818	241	117.0000	104.8571	10574792	10.38%

Elapsed time = 577.97 sec. (212321.63 ticks, tree = 2270.37 MB, solutions = 9)

Simetria:

Elapsed time = 7.22 sec. (3794.95 ticks, tree = 0.02 MB, solutions = 7)							
105	28	110.4781	179	118.0000	80.8627	34088	31.47%
728	326	115.5385	75	118.0000	95.7751	54124	18.83%
1333	629	cutoff		118.0000	102.5994	67457	13.05%
1910	713	110.6800	99	118.0000	108.1333	83465	8.36%
2511	743	cutoff		118.0000	110.4000	106722	6.44%
* 2846+	539			117.0000	113.0000		3.42%
* 2889	500	integral	0	117.0000	113.0000	120674	3.42%
3224	47	cutoff		117.0000	114.3053	127394	2.30%

Como en los pasos anteriores se obtuvo que el mejor resultado usaba 11 lavados, por lo tanto se puede reducir la cantidad de colores máxima sin tener que preocuparse de que la solución sea incompatible.

Como es esperado, el tiempo que tardó en llegar a la solución óptima es mucho menor a los tiempos tardados en los pasos 3 y 5: se pasó de tardar más de 10 minutos a un minuto y medio, y de 30 segundos a 10. En ambos casos la cantidad de lavados es 11 y el tiempo óptimo es 117.

Como es esperado, los resultados fueron mejores a los obtenidos en los pasos 3 y 5. Para el caso en que no se aplicó la condición de simetría, si bien no se logró encontrar la solución óptima dentro de los 10 minutos, en el gráfico se puede observar que llegó al 117 mucho más rápido que en el paso 3. En el caso en que se aplicó la condición, comparando con el paso 5 el tiempo tardado pasó de 30 segundos a 10, en ambos casos la cantidad de lavados es 11 y el tiempo óptimo es 117.

Paso 7:

Si bien el tiempo que tarda en correrse la heurística es menor al que tarda en correr el modelo lineal, el resultado obtenido es peor debido a que no contempla las varias combinaciones que puede haber, y en parte depende en el orden en que se recibe el dataset. En cambio, si bien parece tardar mucho utilizando un modelo lineal, se puede disminuir el tiempo aplicando técnicas de simetría y estudiando los resultados anteriores.

Conclusiones sobre esta parte:

El modelo lineal propuesto para esta entrega utiliza un método similar al coloreo de grafos, con la diferencia de que en lugar de buscar que la cantidad de colores sea la menor posible (menor cantidad de lavados = mejor solución), acá se busca que el tiempo tardado sea el mínimo. Esto se vuelve difícil de solucionar según la manera tradicional, ya que agrega restricciones que es difícil de aplicar en un grafo.

Debido a esto, aplicar un modelo lineal es la mejor solución para este tipo de problemas. Sin embargo, tiene una desventaja: los modelos lineales con variables discretas tardan mucho en computarse, y esto se evidencia en los pasos 2 y 3 donde tardó más de 10 minutos y no logró obtener la solución óptima. Otro factor que afectó fue la existencia de múltiples soluciones posibles, ya que al modelo le da lo mismo el orden en que se realizan los lavados siempre que el tiempo total sea el mismo.

A medida que se fueron siguiendo los pasos del TP, se pudo observar como el modelo mejoraba de a poco: entre el paso 2 y 3, se llegó a una mejor solución reduciendo la máxima cantidad de lavados (de 138 colores a 15); en los pasos 4 y 5 se pudo obtener una solución óptima mediante la reducción de soluciones posibles; y en el paso 6 se pudo obtener la solución en una cantidad menor de tiempo conociendo la cantidad de lavados necesarios.

A partir de los resultados obtenidos se puede concluir en que la cantidad de variables discretas afecta mucho al tiempo de ejecución. En los pasos 4, 5 y 6 se puede ver cómo la disminución de la cantidad de colores hace que el tiempo de ejecución pase de 107 segundos a 30, y luego a 10. Pese a eso, el factor que más afectó fue la condición de simetría: si no se descomentaba esa parte del código, en ningún caso logró llegar a la solución óptima dentro de los 10 minutos.

Conclusion final:

Este trabajo se desarrolló en 4 partes: la primera fue el planteo inicial con pocas prendas para encontrar la mejor solución a partir de los conocimientos previos que teníamos; la segunda fue tratar de mejorar la solución sobre un dataset con más prendas, utilizando también los conocimientos previos; la tercera fue plantear el problema mediante un modelo lineal con los temas aprendidos en la materia, y finalmente esta última.

La primera entrega, en mi caso, fue más una etapa de descubrimiento y conocimiento inicial del problema, y la solución propuesta dependía fuertemente del orden en que se daban los datos. La “mejor” solución también era relativa, era solamente la mejor entre mis algoritmos para esos datos, y no tenía manera de saber si era la óptima.

En la segunda entrega traté de aplicar una técnica similar al coloreo de grafos según el tiempo de lavado de cada prenda y la cantidad de incompatibilidades que tiene, agrupando así las prendas que más tardan con las que más tardan, y las que menos tardan con las que menos tardan. Si bien el algoritmo era mejor al planteado en la primera entrega, seguía teniendo el mismo problema de no saber si era la mejor solución.

Con las primeras dos entregas pude conseguir un conocimiento bastante amplio del problema, lo cual me ayudó a identificarlo rápidamente cuando se dio el tema en la clase. Gracias a esto, pude plantear el modelo en la tercera entrega, con casi total seguridad de que el resultado que saldrá será el óptimo (si las restricciones eran correctas y no me había olvidado ninguna, claramente). Sin embargo, me encontré con otro problema: debido a la cantidad de variables discretas que tenía, tardaba mucho tiempo en ejecutarse. En mi caso tardé más de ocho horas en correrlo con GLPK y no pude llegar a la solución óptima.

En esta cuarta entrega, corrí el modelo usando CPLEX, un programa que corre más rápido que GLPK y ofrece distintos análisis. No solo pude llegar a una solución, sino también aprendí muchas maneras de resolverlo, como se mencionó más arriba. De esta manera se da por concluido el trabajo práctico, y durante el proceso no solo aprendí a usar distintas herramientas nuevas, sino también a pensar de una manera distinta a lo habitual y conseguir soluciones a problemas que son difíciles de resolver de la manera tradicional.