



UNIVERSIDAD DE BUENOS AIRES
Facultad de Ingeniería

Departamento de Gestión

TRABAJO PRÁCTICO FINAL

71.14 - Modelos y Optimización I
2^{do} cuatrimestre 2022

Alumno:

Rinaldi, Lautaro Ezequiel

Padrón:

89.609

mail:

lerinaldi@fi.uba.ar

Docentes teórica:

Ramos, Silvia Ramos
Echevarría, Pablo

Docentes práctica:

OITANA, Diego
ROMANO, Pablo
TERZANO, Iván

Horario Práctica:

Lunes 19 – 22 hs

Índice

<u>Enunciado.....</u>	<u>3</u>
<u>Primer Problema: Solución mediante programación algorítmica con heurísticas.....</u>	<u>4</u>
<u>Segundo Problema: Solución mediante programación algorítmica con heurísticas.....</u>	<u>6</u>
<u>Tercer Problema: Modelo de Programación Lineal Entera.....</u>	<u>9</u>
<u>Cuarto Problema: Solución del modelo mediante un software.....</u>	<u>11</u>
<u>Solución con Heurística propia.....</u>	<u>11</u>
<u>Solución con CPLEX.....</u>	<u>12</u>
Solución con CPLEX (sin modificaciones: sin límite de lavados, sin simetría).....	12
Solución con CPLEX (límite de 15 lavados, sin simetría).....	13
Solución con CPLEX (sin límite de lavados, con simetría).....	14
Solución con CPLEX (con límite de 15 lavados, con simetría).....	17
Solución con CPLEX (con límite de 11 lavados, sin simetría).....	20
<u>Conclusiones.....</u>	<u>24</u>

Enunciado

En esta ocasión, se debe resolver un problema que tiene las siguientes características:

- Una lavandería tiene que lavar prendas, algunas pueden ir juntas y otras no (destiñen).
- El tiempo de cada lavado es el tiempo que lleva lavar la prenda más sucia de ese lavado.

Formato del archivo de entrada del problema

Cada línea comienza con un caracter que indica el contenido de la misma

- "**c**": comentario formato: c Comentario ej: "*c esto es un comentario*"
- "**p**": definición del problema formato: p c n m "c" es un comentario, "n" es la cantidad de prendas y "m" la cantidad de incompatibilidades ej: "*p edges 10 30*"
- "**e**": incompatibilidad formato: e n1 n2 "n1" y "n2" son los números de prenda incompatibles entre ellas ej: "*e 1 2*"
- "**n**": tiempo de lavado formato: n n1 c1 "n1" es el número de prenda y "c1" el tiempo de lavado ej: "*n 5 3*"

Formato del archivo de la solución esperado

- Cada renglón tiene dos valores separados por un espacio, el primero es el número de prenda, el según el número de lavado asignado. ej: "*1 5*" Esto sería lavar la prenda "1" en el lavado "5"

Primer Problema: Solución mediante programación algorítmica con heurísticas

Primera impresión del problema

El problema a resolver no me pareció complicado. Yo lo encontré costoso computacionalmente, porque lo notaba de naturaleza algorítmica e iterativa.

Idea general de cómo resolver el problema planteado

Se me ocurrió tener un conjunto de lavados consolidados, que se van a ir creando de forma dinámica en la medida en la que voy analizando nuevas prendas.

Inicialmente no hay ningún lavado, y se va a crear el primer lavado con la primera prenda de la que se lean restricciones del archivo.

Para la siguiente prenda que se encuentren restricciones, se va a ir comparando con el conjunto de lavados ya consolidados:

- La prenda se asignará al primer lavado contra el cual ninguna de las restricciones presente incompatibilidades;
- en caso de que la prenda presente incompatibilidades con todos los lavados ya creados, se creará un nuevo lavado que contendrá en principio solo a esta prenda.

Una vez que se terminan de analizar todas las prendas con restricciones, se van a asignar al primer lavado todas las prendas que no tienen incompatibilidades con ninguna otra (tomé como supuesto que cada lavado no tiene cantidad máxima de prendas).

Comentarios sobre el código

Me encontré con el caso no detallado en el enunciado ni tampoco entre el conjunto de datos de entrada, de que puede haber prendas que no tengan ninguna incompatibilidad con otras prendas, y deben ser agregadas a algún lavado luego de comparar todas las restricciones.

También asumí que el archivo de entrada viene siempre bien formateado, y que los registros vienen agrupados según el orden que detalla el enunciado (líneas de comentarios "c", única línea con definición del problema "p", todo el conjunto de incompatibilidades entre prendas "e" -ordenados de menor a mayor por el número de la primera prenda-, y al final todo el conjunto de tiempos de lavado "n" - también ordenados de menor a mayor por el número de prenda).

Comentarios finales de la entrega

Creo que la solución implementada llega a un resultado cercano al óptimo (en cuanto a minimizar el tiempo total de lavado), pero no tengo forma de garantizarlo.

Sin embargo, creo también que se puede analizar resolverlo con otro tipo de enfoques algorítmicos para ver si se puede mejorar la eficiencia computacional (por ejemplo, plantenado algún tipo de matriz de incompatibilidades en lugar de tener una naturaleza tan iterativa).

Análisis de la solución

Al correr la heurística propia, la solución se obtiene instantáneamente.

El principal problema que existe al correr una heurística propia, es que desconocemos totalmente qué tan cerca o qué tan lejos del óptimo se encuentra la solución obtenida.

Solución encontrada:

- Tiempo total de lavado: 61 minutos
- Cantidad de lavados requeridos: 8

Salida de la aplicación con heurística propia

Cantidad Prendas: 20

Cantidad Incompatibilidades Informadas: 210

Tiempo total de lavado (prendas individuales): 133 minutos

Lavados agrupados:

Lavado 1, 2 prendas, 8 minutos. Prendas: 4 10

Lavado 2, 2 prendas, 8 minutos. Prendas: 1 11

Lavado 3, 3 prendas, 10 minutos. Prendas: 2 12 17

Lavado 4, 3 prendas, 9 minutos. Prendas: 3 9 20

Lavado 5, 3 prendas, 5 minutos. Prendas: 5 15 18

Lavado 6, 3 prendas, 9 minutos. Prendas: 6 16 19

Lavado 7, 2 prendas, 10 minutos. Prendas: 7 13

Lavado 8, 2 prendas, 2 minutos. Prendas: 8 14

Cantidad de lavados conjuntos: 8

Tiempo total de lavado (prendas agrupadas): 61 minutos

Segundo Problema: Solución mediante programación algorítmica con heurísticas

Primera impresión del problema:

El problema a resolver no me pareció complicado, sino que es casi idéntico al anterior. Sin embarlo, las formas que se me ocurrieron para encararlo, obtienen una solución posible que probablemente no sea la óptima.

Yo lo encontré más costoso computacionalmente, porque tiene mayor cantidad de prendas y muchísimas más incompatibilidades entre ellas.

Le sumó algún grado de complicación adicional que el archivo de entrada no tenga cargadas todas las incompatibilidades y su recíproca, como pasaba en la entrega anterior.

Idea general de cómo resolver el problema planteado

Se me ocurrió tener un conjunto de lavados consolidados, que se van a ir creando de forma dinámica en la medida en la que voy analizando nuevas prendas.

En la búsqueda de minimizar el tiempo total de lavado, entre varios intentos que hice, encontré que lo más importante era procesar las prendas ordenándolas previamente por tiempo de lavado de mayor a menor. Eso hace que se reduzca considerablemente el tiempo.

Como mejora adicional, agregué que si dos prendas tienen el mismo tiempo de lavado, se procesa primero la que tiene mayor cantidad de restricciones (esto permite reducir un 5% adicional el tiempo total de lavado).

Otra mejora adicional, que también mejora un 3 % adicional el tiempo de lavado, es que si hay dos lavados con el mismo tiempo en el que la prenda puede agregarse, se elige poner la prenda en el que tenga menor número de incompatibilidades.

En cuanto al algoritmo:

- Inicialmente no hay ningún lavado, y se va a crear el primer lavado la prenda que tarde más tiempo en lavarse.
- Luego, se continua con el resto de las prendas, empezando por las que tardan más tiempo en lavarse, y se va a ir comparando con el conjunto de lavados ya consolidados:
 - La prenda se asignará al primer lavado contra el cual ninguna de las restricciones presente incompatibilidades;
 - En caso de que la prenda presente incompatibilidades con todos los lavados ya creados, se creará un nuevo lavado que contendrá en principio solo a esta prenda.

Comentarios sobre el código

En esta entrega, en lugar de hacer programación tan estructurada como en la entrega anterior, aposté por crear algunos objetos que me permitieron simplificar enormemente el código, y evitar errores por omisión.

En relación al problema anterior, lo más relevante fue que el archivo no tenía cargadas de forma simétrica las incompatibilidades entre prendas (es decir, si la prenda p1 es incompatible con la p2, por ende también vale la recíproca). Como el archivo no estaba formateado de esa manera, tuve que hacer un preprocesamiento para tener una matriz simétrica con cada incompatibilidad y su recíproco, y luego trabajar en memoria con dichos datos.

Asumí que el archivo de entrada viene siempre bien formateado, y que los registros vienen agrupados según el orden que detalla el enunciado (líneas de comentarios "c", única línea con definición del problema "p", todo el conjunto de incompatibilidades entre prendas "e" -ordenados de menor a mayor por el número de la primer prenda-, y al final todo el conjunto de tiempos de lavado "n" - también ordenados de menor a mayor por el número de prenda).

Comentarios finales de la entrega

En este caso, no se puede ver a ojo si la solución implementada se acerca a la óptima o no (en cuanto a minimizar el tiempo total de lavado).

Lo que si puedo destacar, es que el tiempo computacional de procesamiento es mínimo (casi instantáneo), con respecto a la gran demora que yo esperaba que tuviera antes de empezar a resolverlo.

Creo que se puede analizar resolverlo con otros enfoques algorítmicos para ver si se pueden evaluar más posibilidades, y efectivamente acercarnos más al óptimo (por ejemplo, plantenado algún tipo de matriz sobre la que se pueda operar iterativamente).

Me llamó la atención que los mejores tiempos se obtengan ordenando las prendas por tiempo de lavado decreciente. Antes de codificarlo y probarlo, yo me hubiese inclinado a que el mejor resultado lo iba a obtener ordenando las prendas por cantidad de restricciones de mayor a menor.

Análisis de la solución

Al correr la heurística propia, la solución se obtiene instantáneamente.

El principal problema que existe al correr una heurística propia, es que desconocemos totalmente qué tan cerca o qué tan lejos del óptimo se encuentra la solución obtenida.

Solución encontrada

- Tiempo total de lavado: 463 minutos
- Cantidad de lavados requeridos: 35

Salida de la aplicación con heurística propia

Cantidad Prendas: 385

Cantidad Incompatibilidades Informadas: 19095

Tiempo total de lavado (prendas individuales): 4008 minutos

Lavados agrupados:

Lavado 1, 27 prendas, 20 minutos. Prendas: 17 22 46 54 57 69 86 99 114 116 118 137 156 168 173 197 209 236 238 241 251 270 314 316 328 354 355

Lavado 2, 24 prendas, 20 minutos. Prendas: 3 55 85 104 107 112 138 140 142 174 179 185 187 193 246 274 285 310 312 313 315 333 361 379

Lavado 3, 22 prendas, 20 minutos. Prendas: 13 19 40 68 80 97 136 162 188 189 194 208 235 239 269 284 290 319 348 353 362 381

Lavado 4, 15 prendas, 19 minutos. Prendas: 2 105 176 195 229 309 357 364 365 366 367 368 369 377 378

Lavado 5, 14 prendas, 19 minutos. Prendas: 37 78 83 115 132 146 169 186 217 220 300 325 344 373

Lavado 6, 14 prendas, 19 minutos. Prendas: 34 43 64 92 119 133 144 165 201 218 232 306 318 383

Lavado 7, 14 prendas, 19 minutos. Prendas: 28 44 56 95 147 160 161 196 211 244 261 327 347 358

Lavado 8, 15 prendas, 19 minutos. Prendas: 11 42 58 81 102 139 141 163 181 205 257 299 337 341 382

Lavado 9, 14 prendas, 18 minutos. Prendas: 8 38 53 62 117 127 151 154 171 230 276 282 329 350

Lavado 10, 16 prendas, 17 minutos. Prendas: 14 21 63 100 129 148 155 167 192 237 243 253 265 283 363 375

Lavado 11, 11 prendas, 17 minutos. Prendas: 7 113 130 149 159 175 224 263 297 346 380

Lavado 12, 19 prendas, 16 minutos. Prendas: 6 66 74 91 96 111 124 152 183 221 234 250 280 305 326 351 360 384 385

Lavado 13, 13 prendas, 16 minutos. Prendas: 9 27 47 106 135 150 177 198 200 219 266 286 339

Lavado 14, 11 prendas, 16 minutos. Prendas: 41 45 49 67 254 267 271 289 320 336 352

Lavado 15, 9 prendas, 15 minutos. Prendas: 35 50 121 157 256 281 323 331 359

Lavado 16, 14 prendas, 15 minutos. Prendas: 16 20 59 79 98 108 122 170 216 252 264 292 338 345

Lavado 17, 9 prendas, 14 minutos. Prendas: 26 52 70 76 87 172 262 273 279

Lavado 18, 11 prendas, 14 minutos. Prendas: 23 82 109 166 214 227 258 268 288 343 376

Lavado 19, 9 prendas, 14 minutos. Prendas: 12 51 61 126 225 248 275 277 308

Lavado 20, 10 prendas, 14 minutos. Prendas: 5 33 110 123 158 184 191 226 296 302

Lavado 21, 8 prendas, 14 minutos. Prendas: 32 84 101 190 202 203 294 324

Lavado 22, 8 prendas, 13 minutos. Prendas: 25 60 77 164 213 233 295 307

Lavado 23, 8 prendas, 12 minutos. Prendas: 75 182 204 228 231 245 303 342

Lavado 24, 9 prendas, 11 minutos. Prendas: 1 24 128 143 210 272 298 335 370

Lavado 25, 12 prendas, 10 minutos. Prendas: 65 73 90 125 145 207 215 255 260 322 334 372

Lavado 26, 7 prendas, 9 minutos. Prendas: 48 94 178 212 242 317 349

Lavado 27, 7 prendas, 9 minutos. Prendas: 4 30 180 206 249 311 321

Lavado 28, 8 prendas, 8 minutos. Prendas: 10 36 134 222 247 287 330 356

Lavado 29, 4 prendas, 8 minutos. Prendas: 88 120 223 301

Lavado 30, 6 prendas, 8 minutos. Prendas: 29 71 93 131 278 340

Lavado 31, 4 prendas, 7 minutos. Prendas: 18 72 259 291

Lavado 32, 2 prendas, 4 minutos. Prendas: 371 374

Lavado 33, 3 prendas, 4 minutos. Prendas: 89 153 293

Lavado 34, 5 prendas, 4 minutos. Prendas: 15 31 39 199 240

Lavado 35, 3 prendas, 1 minutos. Prendas: 103 304 332

Cantidad de lavados conjuntos: 35

Tiempo total de lavado (prendas agrupadas): 463 minutos

Tercer Problema: Modelo de Programación Lineal Entera

Situación Problemática

Una lavandería tiene que lavar una cantidad N de prendas, y sabe que algunas de ellas las puede lavar en conjunto y otras las tiene que lavar en lavados diferentes.

Se conocen el tiempo que llevaría lavar cada prenda individualmente. Además, cuando se lavan prendas en conjunto, el tiempo del lavado es el tiempo que le lleva lavar la prenda más sucia de ese lavado (es decir, la que lleva más tiempo).

Tipo de Problema

La forma en la que decidí encarar el problema es tomando como base un problema de *coloreo de grafos*.

Hipótesis y Supuestos

- La cantidad total de prendas a lavar y los tiempos de lavado de cada prenda individuales son conocidos de antemano.
- No hay límite en la cantidad máxima de prendas que puede tener un lavado.
- Todas las prendas deben ser lavadas al menos una vez.
- Como mínimo, en cada grupo de prendas a lavar, hay una prenda.
- En el peor caso, se considera que se hace un lavado para cada prenda.

Objetivo

Determinar cuántos lavados deben realizarse, y qué prendas deben ponerse en cada uno de dichos lavados, para minimizar el tiempo total de lavado en el periodo P , respetando las restricciones de incompatibilidades entre prendas.

Constantes conocidas

TP_i : Tiempo de lavado de la prenda i

N : Número total de prendas a lavar

Variables de Decisión

$$YP_{ij} : \begin{cases} 1 & \text{si la prenda } i \text{ se lava en el lavado } j \\ 0 & \text{en otro caso} \end{cases}$$

$$YL_j : \begin{cases} 1 & \text{si el lavado } j \text{ va a ser necesario} \\ 0 & \text{en otro caso} \end{cases}$$

TL_j : Tiempo del lavado j

Restricciones

- Si la prenda **i** y la prenda **k** son incompatibles, tienen que asignarse a distintos lavados (distintos **j**)

$$Y P_{ij} + Y P_{kj} \leq Y L_j \quad \forall 1 \leq j \leq N \quad \forall i, k \text{ que presenten incompatibilidad}$$

- Toda prenda debe ser asignada a un único lavado (único **j**)

$$\sum_{j=1}^N Y P_{ij} = 1 \quad \forall 1 \leq i \leq N$$

- El tiempo del lavado **j** es igual al tiempo de lavado que requiere la prenda de mayor tiempo

$$T P_i \cdot Y P_{ij} \leq T L_j \quad \forall 1 \leq i \leq N \quad \forall 1 \leq j \leq N$$

Funcional

$$Z_{min} = \sum_{j=1}^N T L_j$$

Cuarto Problema: Solución del modelo mediante un software

En esta cuarta entrega se pide buscar el óptimo utilizando el software IBM ILOG CPLEX.

Se recomienda utilizar el código provisto en https://modelosuno.okapii.com/content/modelos_wvcp.zip, y efectivamente es lo que se hizo.

Solución con Heurística propia

Al correr la heurística propia (es la misma que se utilizó para el segundo problema), la solución se obtiene instantáneamente.

El principal problema que existe al correr una heurística propia, es que desconocemos totalmente qué tan cerca o qué tan lejos del óptimo se encuentra la solución obtenida.

Solución encontrada:

- Tiempo total de lavado: 123 minutos
- Cantidad de lavados requeridos: 11

Salida de la aplicación:

Cantidad Prendas: 138

Cantidad Incompatibilidades Informadas: 986

Tiempo total de lavado (prendas individuales): 1481 minutos

Lavados agrupados:

Lavado 1, 75 prendas, 20 minutos. Prendas: 1 2 3 4 5 6 9 11 14 15 16 17 19 22 23 24 25 26 30 31 34 37 38 39 41 42 43 44 46 47 48 50 51 55 56 59 60 62 63 65 66 67 71 73 75 79 80 82 85 86 87 88 90 91 93 98 102 103 104 105 106 107 110 111 113 114 119 120 121 125 128 129 131 132 137

Lavado 2, 30 prendas, 19 minutos. Prendas: 8 10 12 13 27 28 33 35 40 49 52 58 68 69 76 77 84 92 94 96 97 109 112 117 118 122 124 127 134 136

Lavado 3, 7 prendas, 14 minutos. Prendas: 18 54 61 101 115 126 133

Lavado 4, 10 prendas, 13 minutos. Prendas: 7 21 29 53 64 70 89 100 123 130

Lavado 5, 2 prendas, 12 minutos. Prendas: 32 138

Lavado 6, 6 prendas, 11 minutos. Prendas: 20 45 57 83 99 108

Lavado 7, 3 prendas, 9 minutos. Prendas: 78 81 95

Lavado 8, 2 prendas, 9 minutos. Prendas: 72 135

Lavado 9, 1 prendas, 8 minutos. Prendas: 36

Lavado 10, 1 prendas, 5 minutos. Prendas: 116

Lavado 11, 1 prendas, 3 minutos. Prendas: 74

Cantidad de lavados conjuntos: 11

Tiempo total de lavado (prendas agrupadas): 123 minutos

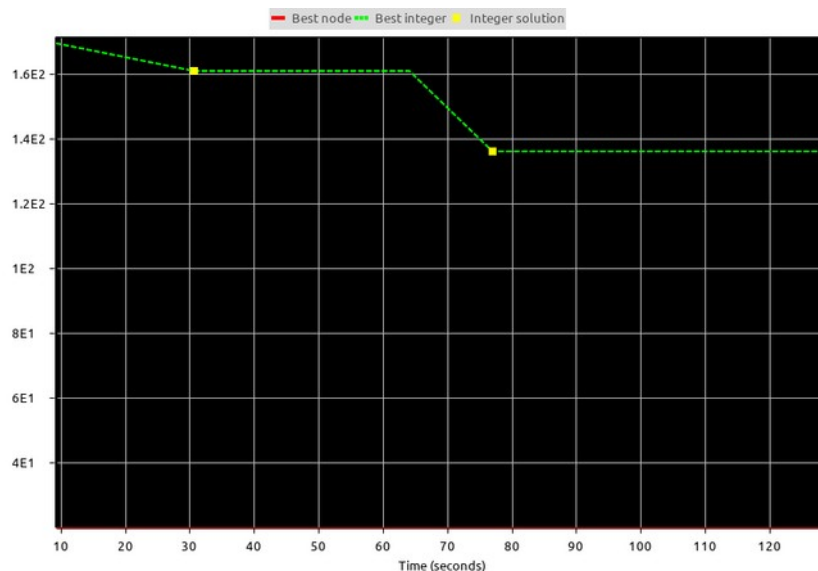
Solución con CPLEX

Solución con CPLEX (sin modificaciones: sin límite de lavados, sin simetría)

En esta primera prueba, se ejecutó el código tal cual como fue descargado, sin realizarle ningún tipo de modificaciones.

A pesar de haber pasado 10 minutos, el software no pudo terminar de procesar.

Solapa Statics:



De la captura anterior, lo más destacable es, que en los primeros dos minutos, el software pudo encontrar y mejorar muy poco las soluciones enteras. Se ve que entre que encuentra una solución y luego encuentra otra solución que mejora la anterior, le lleva bastante tiempo.

Solapa Engine: (solo el final)

```
Elapsed time = 455,45 sec. (97166,15 ticks, tree = 5,20 MB, solutions = 19)
596 399 44,1979 1025 126,0000 20,0000 949565 84,13%
606 403 46,2868 961 126,0000 20,0000 970239 84,13%
619 405 46,2868 1297 126,0000 20,0000 975446 84,13%
639 456 52,0000 1061 126,0000 20,0000 1098217 84,13%
654 448 73,1439 694 126,0000 20,0000 1064938 84,13%
677 452 46,0000 832 126,0000 20,0000 1096523 84,13%
702 495 59,9752 729 126,0000 20,0000 1205431 84,13%
735 496 65,0000 831 126,0000 20,0000 1209704 84,13%
760 546 118,4789 152 126,0000 20,0000 1305124 84,13%
807 570 78,5610 495 126,0000 20,0000 1374877 84,13%
Elapsed time = 582,08 sec. (107052,45 ticks, tree = 6,60 MB, solutions = 19)
854 572 124,0000 94 126,0000 20,0000 1347747 84,13%
* 916 609 integral 0 125,0000 20,0000 1414712 84,00%
924 615 79,0087 708 125,0000 20,0000 1460561 84,00%
* 925+ 626 124,0000 20,0000 83,87%
* 939 620 integral 0 123,0000 20,0000 1476979 83,74%
955 619 95,8291 517 123,0000 20,0000 1470987 83,74%
985 616 50,0000 834 123,0000 20,0000 1504752 83,74%
1008 647 106,6556 289 123,0000 20,0000 1555542 83,74%
1034 656 117,0000 153 123,0000 20,0000 1560151 83,74%
* 1042 657 integral 0 122,0000 20,0000 1561512 83,61%
* 1044 700 integral 0 121,0000 20,0000 1668705 83,47%
1063 688 81,0000 751 121,0000 20,0000 1696393 83,47%
1072 696 56,0000 858 121,0000 37,0000 1754374 69,42%
1086 695 46,2868 945 121,0000 37,0000 1670060 69,42%
1100 714 47,5064 1198 121,0000 37,0000 1808835 69,42%
Elapsed time = 722,83 sec. (117208,89 ticks, tree = 9,48 MB, solutions = 24)
```

Lo más destacable que se ve en la captura anterior, es que aunque hayan pasado más de 12 minutos de procesamiento, el software recién está encontrando soluciones con una garantía de calidad muy baja, ya que están a una distancia de 69,42% del óptimo.

También se puede destacar que aunque la mejor solución que obtuvo hasta el momento es de 121 minutos (es decir, muy cerca del óptimo que se mostrará más adelante), el software no tuvo tiempo suficiente de seguir probando soluciones y acotando inferiormente y superiormente para garantizar una mejor garantía de calidad de la solución obtenida parcialmente.

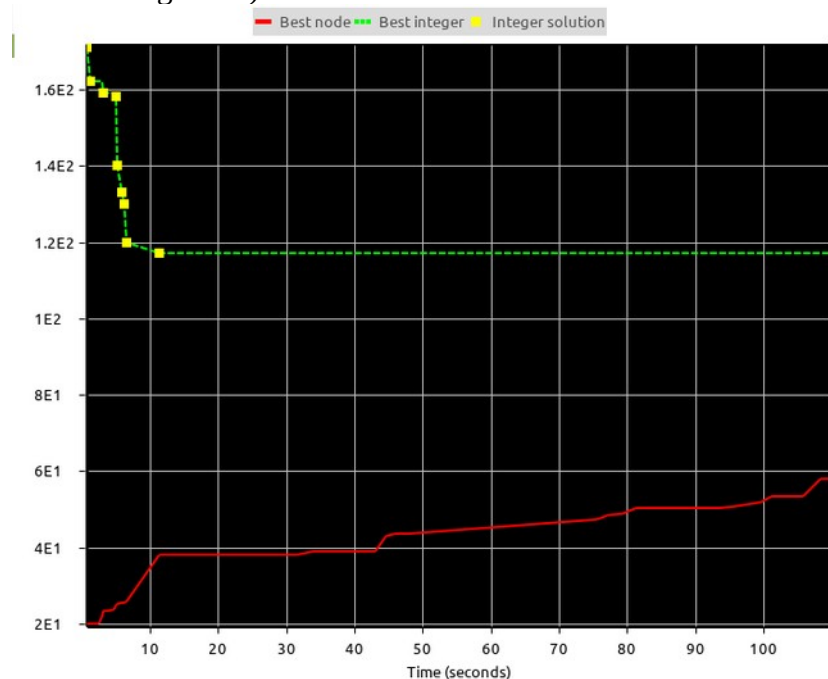
Solución con CPLEX (límite de 15 lavados, sin simetría)

En esta prueba, vamos a restringir dentro del código la cantidad de lavados (O colores en el modelo del grafo), a solamente 15 (dato obtenido mediante alguna heurística).

En principio, como se eliminan un montón de posibilidades que el software ya no va a tener que probar, esperamos una convergencia más rápida, o al menos, que en un menor tiempo consiga acercarse lo más posible con un mayor grado de garantía de calidad.

Con esta prueba, se va a poder ver si esta limitación es determinante o no para la convergencia.

Solapa Statics: (primeros 90 segundos)



De la captura anterior, lo más destacable es que en los primeros 10 segundos, el software pudo encontrar una solución entera y enseguida consiguió encontrar varias soluciones enteras mejores.

Solapa Engine: (solo el final)

```
Elapsed time = 437,38 sec. (188870,21 ticks, tree = 316,44 MB, solutions = 14)
85794 14111      116,0000    370      117,0000      105,1623    8844635    10,12%
87754 15326      116,0000    248      117,0000      105,2117    8993968     10,08%
89690 17167      109,8700    362      117,0000      105,3740    9194196     9,94%
91670 19075      cutoff          117,0000      105,5410    9397957     9,79%
93378 20242      111,3066    352      117,0000      105,6510    9555903     9,70%
95506 22168      112,0000    314      117,0000      105,7681    9770771     9,60%
97369 23278      111,0638    323      117,0000      105,7978    9908468     9,57%
99277 24915      112,5887    306      117,0000      105,8407    10087197    9,54%
101271 26448      112,3158    347      117,0000      105,8811    10275616    9,50%
102549 27823      106,0000    475      117,0000      105,8811    10436092    9,50%
Elapsed time = 530,89 sec. (227062,55 ticks, tree = 817,66 MB, solutions = 14)
104232 28679      108,8552    350      117,0000      105,9129    10555828    9,48%
105493 30371      116,0000    235      117,0000      106,0000    10768302    9,40%
107111 31519      112,0000    226      117,0000      106,0000    10915386    9,40%
108714 32984      110,5167    369      117,0000      106,0000    11081014    9,40%
110015 34135      114,8500    283      117,0000      106,0000    11227313    9,40%
111576 34903      112,0000    242      117,0000      106,0000    11334392    9,40%
112979 36533      114,0000    279      117,0000      106,0000    11526726    9,40%
114109 37556      114,0000    306      117,0000      106,0000    11687715    9,40%
115424 38313      113,2857    276      117,0000      106,0000    11820207    9,40%
116724 39830      112,0000    255      117,0000      106,0000    12032540    9,40%
Elapsed time = 622,63 sec. (265254,65 ticks, tree = 1251,87 MB, solutions = 14)
```

En este caso, aunque pasaron 10 minutos, el solver no terminó. Sin embargo, puede notarse que en los mismos 10 minutos está encontrando soluciones que tienen una garantía de calidad de estar a tan solo un 9,40% del óptimo.

También se puede destacar que aunque el software aun no pudo confirmarlo, la solución de 117 minutos que obtuvo hasta el momento es la que más adelante terminará determinando que es el óptimo.

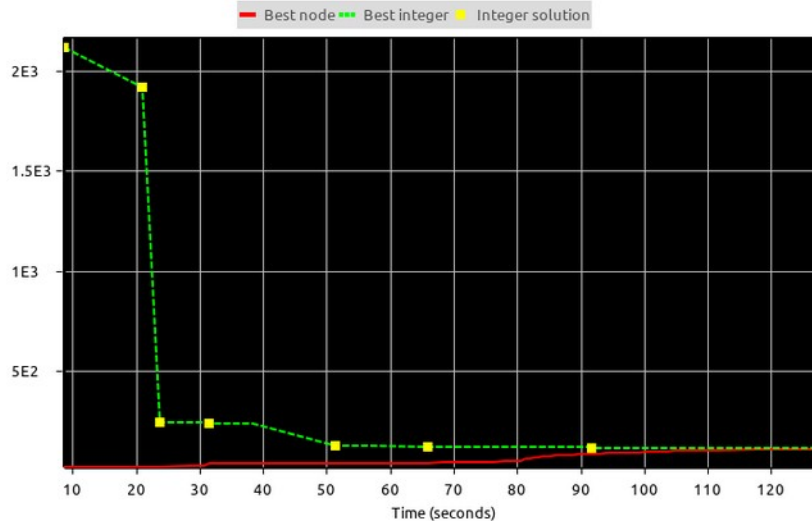
Solución con CPLEX (sin límite de lavados, con simetría)

Ahora se habilita la restricción de simetría dentro del código, de esta manera le estamos diciendo al solver que lo importante es cómo agrupe la prendas, pero no el número de lavado en el que va ese grupo de prendas.

Esto elimina una enorme cantidad de casos que el solver ya no necesita probar, y esperamos que ahora sí el software pueda converger en el tiempo estipulado.

Efectivamente, por primera vez, el solver consiguió terminar la ejecución y la solución óptima la halló en un tiempo de alrededor de 2 minutos.

Solapa Statics:



En este caso, lo más destacable del gráfico, es la rápida convergencia tanto de la solución continua como también de la solución entera, que es lo que termina garantizando que el solver termine.

También se ve que con pocas mejoras sobre la primera solución obtenida, llega al óptimo (esto se debe a que eliminamos del conjunto solución todas las soluciones alternativas, ya que nos alcanza con una sola porque el resto salen por simetría).

Solución encontrada:

- Tiempo total de lavado: 117 minutos
- Cantidad de lavados requeridos: 11

Solapa Scripting Log:

solution: 117 /size: 138 /time: 1669175488.980947971

Nodo 1: 1	Nodo 47: 1	Nodo 93: 1
Nodo 2: 1	Nodo 48: 1	Nodo 94: 1
Nodo 3: 1	Nodo 49: 4	Nodo 95: 4
Nodo 4: 1	Nodo 50: 1	Nodo 96: 2
Nodo 5: 1	Nodo 51: 1	Nodo 97: 8
Nodo 6: 5	Nodo 52: 4	Nodo 98: 1
Nodo 7: 4	Nodo 53: 7	Nodo 99: 7
Nodo 8: 2	Nodo 54: 8	Nodo 100: 5
Nodo 9: 1	Nodo 55: 1	Nodo 101: 3
Nodo 10: 2	Nodo 56: 1	Nodo 102: 1
Nodo 11: 1	Nodo 57: 6	Nodo 103: 1
Nodo 12: 2	Nodo 58: 6	Nodo 104: 1
Nodo 13: 2	Nodo 59: 1	Nodo 105: 1
Nodo 14: 1	Nodo 60: 4	Nodo 106: 1
Nodo 15: 1	Nodo 61: 7	Nodo 107: 1
Nodo 16: 1	Nodo 62: 2	Nodo 108: 6
Nodo 17: 2	Nodo 63: 1	Nodo 109: 3
Nodo 18: 3	Nodo 64: 1	Nodo 110: 1
Nodo 19: 1	Nodo 65: 1	Nodo 111: 1
Nodo 20: 6	Nodo 66: 1	Nodo 112: 1
Nodo 21: 4	Nodo 67: 2	Nodo 113: 5
Nodo 22: 2	Nodo 68: 2	Nodo 114: 1
Nodo 23: 1	Nodo 69: 2	Nodo 115: 6
Nodo 24: 1	Nodo 70: 5	Nodo 116: 6
Nodo 25: 2	Nodo 71: 1	Nodo 117: 1
Nodo 26: 5	Nodo 72: 7	Nodo 118: 2
Nodo 27: 2	Nodo 73: 2	Nodo 119: 1
Nodo 28: 4	Nodo 74: 11	Nodo 120: 3
Nodo 29: 7	Nodo 75: 1	Nodo 121: 1
Nodo 30: 1	Nodo 76: 2	Nodo 122: 1
Nodo 31: 1	Nodo 77: 1	Nodo 123: 10
Nodo 32: 4	Nodo 78: 8	Nodo 124: 2
Nodo 33: 2	Nodo 79: 1	Nodo 125: 1
Nodo 34: 3	Nodo 80: 1	Nodo 126: 3
Nodo 35: 4	Nodo 81: 10	Nodo 127: 2
Nodo 36: 8	Nodo 82: 2	Nodo 128: 1
Nodo 37: 9	Nodo 83: 6	Nodo 129: 1
Nodo 38: 1	Nodo 84: 2	Nodo 130: 4
Nodo 39: 1	Nodo 85: 1	Nodo 131: 1
Nodo 40: 4	Nodo 86: 1	Nodo 132: 1
Nodo 41: 1	Nodo 87: 1	Nodo 133: 5
Nodo 42: 1	Nodo 88: 1	Nodo 134: 4
Nodo 43: 4	Nodo 89: 4	Nodo 135: 9
Nodo 44: 7	Nodo 90: 2	Nodo 136: 2
Nodo 45: 10	Nodo 91: 1	Nodo 137: 1
Nodo 46: 1	Nodo 92: 2	Nodo 138: 5

Solapa Engine: (solo el final)

```
Performing restart 1

Repeating presolve.
Tried aggregator 4 times.
MIP Presolve eliminated 19683 rows and 10577 columns.
MIP Presolve modified 195 coefficients.
Aggregator did 15 substitutions.
Reduced MIP has 4214 rows, 2556 columns, and 13179 nonzeros.
Reduced MIP has 2440 binaries, 116 generals, 0 SOSs, and 0 indicators.
Presolve time = 0,03 sec. (34,26 ticks)
Tried aggregator 1 time.
Reduced MIP has 4214 rows, 2556 columns, and 13179 nonzeros.
Reduced MIP has 2441 binaries, 115 generals, 0 SOSs, and 0 indicators.
Presolve time = 0,01 sec. (12,31 ticks)
Represolve time = 0,20 sec. (343,43 ticks)
 4908   0   48,0377   476   117,0000   Cuts: 485   255198   8,88%
 4908   0   49,8985   500   117,0000   Cuts: 495   257214   8,88%
 4908   0   50,0702   511   117,0000   Cuts: 475   258071   8,88%
 4908   0   50,1639   446   117,0000   Cuts: 440   258666   8,88%
 4908   0   50,3685   500   117,0000   Cuts: 429   260241   8,88%
 4908   0   50,9331   491   117,0000   Cuts: 207   262348   8,88%
 4908   0   51,0704   498   117,0000   Cuts: 514   263065   8,88%
 4908   2   51,0704   433   117,0000   106,6080   263065   8,88%
 4927   4   101,2820   248   117,0000   106,6080   267162   8,88%
 5222   69   112,2162   149   117,0000   106,6080   306078   8,88%
 6051  467   92,8716   289   117,0000   106,6080   353187   8,88%
 6572  814   111,0000   127   117,0000   106,6080   417941   8,88%
Elapsed time = 122,57 sec. (59714,31 ticks, tree = 1,60 MB, solutions = 25)
 7404  823   99,8906   212   117,0000   106,6080   487667   8,88%
 8718  689   cutoff           117,0000   107,2105   559601   8,37%

Clique cuts applied: 9
Implied bound cuts applied: 51
Flow cuts applied: 53
Mixed integer rounding cuts applied: 83
Zero-half cuts applied: 31
Gomory fractional cuts applied: 22

Root node processing (before b&c):
  Real time = 73,71 sec. (43623,61 ticks)
Parallel b&c, 8 threads:
  Real time = 54,73 sec. (18298,01 ticks)
  Sync time (average) = 8,85 sec.
  Wait time (average) = 0,01 sec.
-----
Total (root+branch&cut) = 128,44 sec. (61921,62 ticks)
```

En este caso, es relevante notar que pudo resolver el problema en un tiempo razonable (alrededor de 2 minutos), y además da la garantía de que la solución hallada está a 8,37% del óptimo en el peor caso.

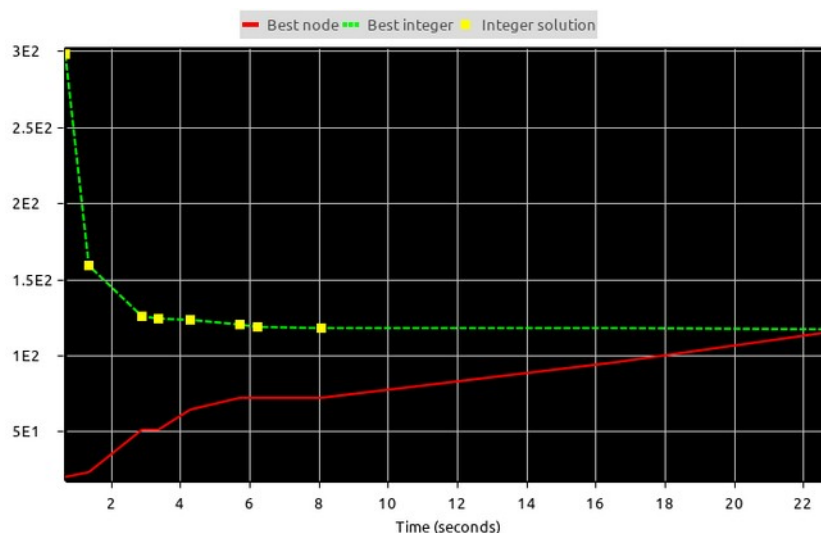
Lo más destacable, es que el hecho de aprovechar la simetría, es lo que nos pudo garantizar que la solución converja.

Solución con CPLEX (con límite de 15 lavados, con simetría)

En este caso, vamos a aprovechar para mejorar y acelerar lo más posible solución, todos los datos de los que disponemos: Que el problema tiene simetría, y que la cantidad de lavados necesarios para lavar todas las prendas está acotada por 15. Esperamos obtener una convergencia mucho más rápida.

Efectivamente, hallar la solución tardó alrededor de 22 segundos (1/6 de lo que tardó en el punto anterior, sin tener en cuenta el límite de lavados pero sí la simetría).

Solapa Statics:



En este caso, lo más destacable del gráfico, es la rápida convergencia tanto de la solución continua como también de la solución entera, que es lo que termina garantizando que el solver termine.

Solución encontrada:

- Tiempo total de lavado: 117 minutos
- Cantidad de lavados requeridos: 11

Solapa Scripting Log:

solution: 117 /size: 138 /time: 1669176463.879160881

Nodo 1: 1	Nodo 67: 2	Nodo 133: 4
Nodo 2: 1	Nodo 68: 2	Nodo 134: 2
Nodo 3: 1	Nodo 69: 9	Nodo 135: 7
Nodo 4: 1	Nodo 70: 5	Nodo 136: 2
Nodo 5: 1	Nodo 71: 1	Nodo 137: 1
Nodo 6: 1	Nodo 72: 7	Nodo 138: 5
Nodo 7: 4	Nodo 73: 2	
Nodo 8: 1	Nodo 74: 11	
Nodo 9: 1	Nodo 75: 1	
Nodo 10: 2	Nodo 76: 2	
Nodo 11: 1	Nodo 77: 1	
Nodo 12: 2	Nodo 78: 8	
Nodo 13: 2	Nodo 79: 2	
Nodo 14: 1	Nodo 80: 1	
Nodo 15: 1	Nodo 81: 10	
Nodo 16: 2	Nodo 82: 2	
Nodo 17: 1	Nodo 83: 6	
Nodo 18: 3	Nodo 84: 2	
Nodo 19: 1	Nodo 85: 1	
Nodo 20: 9	Nodo 86: 1	
Nodo 21: 4	Nodo 87: 1	
Nodo 22: 2	Nodo 88: 1	
Nodo 23: 1	Nodo 89: 4	
Nodo 24: 1	Nodo 90: 1	
Nodo 25: 1	Nodo 91: 1	
Nodo 26: 5	Nodo 92: 2	
Nodo 27: 2	Nodo 93: 1	
Nodo 28: 4	Nodo 94: 1	
Nodo 29: 6	Nodo 95: 4	
Nodo 30: 1	Nodo 96: 2	
Nodo 31: 1	Nodo 97: 2	
Nodo 32: 8	Nodo 98: 1	
Nodo 33: 1	Nodo 99: 6	
Nodo 34: 3	Nodo 100: 5	
Nodo 35: 2	Nodo 101: 3	
Nodo 36: 8	Nodo 102: 1	
Nodo 37: 3	Nodo 103: 1	
Nodo 38: 2	Nodo 104: 1	
Nodo 39: 1	Nodo 105: 1	
Nodo 40: 2	Nodo 106: 2	
Nodo 41: 2	Nodo 107: 1	
Nodo 42: 1	Nodo 108: 6	
Nodo 43: 1	Nodo 109: 2	
Nodo 44: 2	Nodo 110: 2	
Nodo 45: 6	Nodo 111: 2	
Nodo 46: 1	Nodo 112: 2	
Nodo 47: 1	Nodo 113: 1	
Nodo 48: 1	Nodo 114: 1	
Nodo 49: 2	Nodo 115: 9	
Nodo 50: 1	Nodo 116: 9	
Nodo 51: 1	Nodo 117: 2	
Nodo 52: 1	Nodo 118: 2	
Nodo 53: 4	Nodo 119: 1	
Nodo 54: 6	Nodo 120: 3	
Nodo 55: 1	Nodo 121: 1	
Nodo 56: 1	Nodo 122: 6	
Nodo 57: 10	Nodo 123: 4	
Nodo 58: 2	Nodo 124: 2	
Nodo 59: 4	Nodo 125: 1	
Nodo 60: 1	Nodo 126: 6	
Nodo 61: 8	Nodo 127: 1	
Nodo 62: 1	Nodo 128: 1	
Nodo 63: 1	Nodo 129: 1	
Nodo 64: 9	Nodo 130: 4	
Nodo 65: 2	Nodo 131: 3	
Nodo 66: 1	Nodo 132: 2	

Solapa Engine: (solo el final)

```
Presolve time = 0,01 sec. (9,19 ticks)
Tried aggregator 1 time.
MIP Presolve eliminated 127 rows and 0 columns.
MIP Presolve modified 14 coefficients.
Reduced MIP has 2657 rows, 1450 columns, and 8973 nonzeros.
Reduced MIP has 1435 binaries, 15 generals, 0 SOSs, and 0 indicators.
Presolve time = 0,01 sec. (8,38 ticks)
Represolve time = 0,06 sec. (61,18 ticks)
 3445  0      88,9862  359    118,0000    Cuts: 367    236594    24,59%
 3445  0      89,8636  355    118,0000    Cuts: 266    237860    23,84%
 3445  0      90,0897  368    118,0000    Cuts: 266    238533    23,57%
 3445  0      90,1633  360    118,0000    Cuts: 292    238790    23,57%
 3445  0      90,3472  354    118,0000    Cuts: 297    239236    23,36%
 3445  0      90,3945  362    118,0000    Cuts: 264    239593    23,23%
 3445  0      90,4372  362    118,0000    Cuts: 229    239881    23,20%
 3445  0      90,5967  358    118,0000    Cuts: 235    240363    23,18%
 3445  0      90,7315  364    118,0000    Cuts: 265    240624    23,11%
 3445  0      90,8098  360    118,0000    Cuts: 209    240945    23,04%
 3445  0      90,8731  374    118,0000    Cuts: 172    241191    22,99%
 3445  0      90,9395  386    118,0000    Cuts: 210    241465    22,93%
 3445  0      91,0301  368    118,0000    Cuts: 168    241649    22,86%
 3445  0      91,0846  359    118,0000    Cuts: 191    242089    22,81%
 3445  0      91,1345  353    118,0000    Cuts: 191    242461    22,77%
 3445  0      91,1345  356    118,0000    Cuts: 165    242596    21,56%
 3445  0      91,1864  322    118,0000    Cuts:  77    242862    21,56%
 3445  0      91,2794  327    118,0000    Cuts: 191    243295    21,56%
 3445  0      91,3452  352    118,0000    Cuts:  98    243546    19,07%
 3445  0      91,3706  310    118,0000    Cuts: 129    243758    19,07%
 3445  0      91,4634  318    118,0000    Cuts: 192    243951    19,07%
 3445  2      91,4634  297    118,0000    95,4975    243953    19,07%
 3469  7      cutoff    118,0000    95,4975    249769    19,07%
Elapsed time = 17,40 sec. (9652,52 ticks, tree = 0,02 MB, solutions = 13)
 3753  62     112,0000  183    118,0000    95,4975    264961    19,07%
 5048 322     112,4348  211    118,0000    96,4398    305126    18,27%
* 7017 416     integral    0     117,0000    114,3163    358270    2,29%

Clique cuts applied:  7
Implied bound cuts applied:  6
Flow cuts applied:  35
Mixed integer rounding cuts applied: 174
Zero-half cuts applied: 19
Gomory fractional cuts applied: 16

Root node processing (before b&c):
  Real time      =  5,35 sec. (3178,43 ticks)
Parallel b&c, 8 threads:
  Real time      = 17,14 sec. (9116,47 ticks)
  Sync time (average) =  2,44 sec.
  Wait time (average) =  0,01 sec.
-----
Total (root+branch&cut) = 22,48 sec. (12294,90 ticks)
```

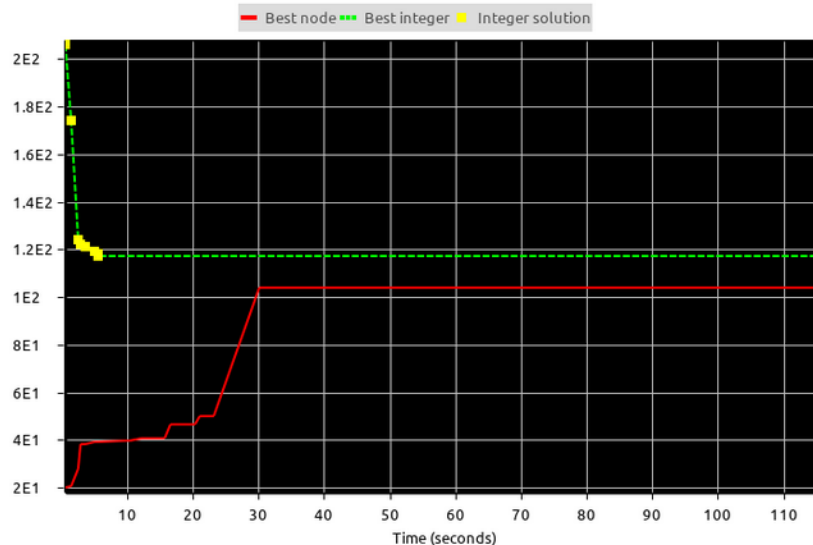
En este caso, es relevante notar que el hecho de limitar la cantidad máxima de lavados, hizo que la convergencia se acelere de forma impresionante.

Pudo resolver el problema en un tiempo de alrededor de 20 segundos, y además da la garantía de que la solución hallada está a 2,29% del óptimo en el peor caso. Es decir, esta forma de procesamiento tiene dos ventajas notorias: tiempo mínimo, y buena garantía de calidad.

Solución con CPLEX (con límite de 11 lavados, sin simetría)

En esta prueba, suponemos que por alguna otra heurística se ha conseguido estimar aún mejor la cantidad máxima de lavados necesarios. Sin embargo, no vamos a aprovechar la condición de simetría, para ver cuál de los dos factores tiene más peso.

Solapa Statics: (primeros 90 segundos)



En este caso, aunque al comienzo se ve una tendencia muy rápida a la convergencia tanto de la solución continua como también de la solución entera, luego se estanca y le cuesta muchísimo seguir acotando la solución, y por eso el solver no terminó de procesar en los 10 minutos de tiempo que le dimos antes de cortarlo.

Como a pesar de tener menos lavados como limitantes, el solver sigue sin converger, nos permite concluir que lo que realmente ayuda a la convergencia es la condición de simetría.

Solapa Engine: (solo el final)

```
Performing restart 2
Repeating presolve.
Tried aggregator 1 time.
MIP Presolve eliminated 94 rows and 0 columns.
Reduced MIP has 2734 rows, 1485 columns, and 17361 nonzeros.
Reduced MIP has 1474 binaries, 11 generals, 0 SOSs, and 0 indicators.
Presolve time = 0,03 sec. (15,19 ticks)
Tried aggregator 1 time.
Reduced MIP has 2734 rows, 1485 columns, and 17361 nonzeros.
Reduced MIP has 1474 binaries, 11 generals, 0 SOSs, and 0 indicators.
Presolve time = 0,02 sec. (15,66 ticks)
Represolve time = 0,36 sec. (94,68 ticks)
130708 0 107,0000 196 117,0000 Cuts: 52 12798434 8,55%
130708 0 107,0000 325 117,0000 Cuts: 336 12798984 8,55%
130708 0 107,0000 272 117,0000 Cuts: 99 12799370 8,55%
130708 0 107,0000 319 117,0000 Cuts: 361 12799809 8,55%
130708 0 107,0000 253 117,0000 Cuts: 59 12800017 8,55%
130708 0 107,0000 328 117,0000 Cuts: 350 12800356 8,55%
130708 0 107,0000 226 117,0000 Cuts: 108 12800611 8,55%
130708 0 107,0000 296 117,0000 Cuts: 269 12801009 8,55%
130708 0 107,0000 206 117,0000 Cuts: 98 12801250 8,55%
130708 0 107,0000 307 117,0000 Cuts: 361 12801618 8,55%
130708 0 107,0000 207 117,0000 Cuts: 91 12801745 8,55%
130708 0 107,0000 281 117,0000 Cuts: 333 12802088 8,55%
130708 2 107,0000 174 117,0000 107,0000 12802088 8,55%
130786 53 112,6667 157 117,0000 107,0000 12821026 8,55%
Elapsed time = 545,28 sec. (253999,53 ticks, tree = 0,21 MB, solutions = 9)
132402 826 108,8462 311 117,0000 107,0000 12887069 8,55%
136332 4212 114,0000 260 117,0000 107,0000 13117044 8,55%
139716 7124 108,0000 269 117,0000 107,0000 13355342 8,55%
142571 9470 116,0000 137 117,0000 107,0000 13577333 8,55%
144690 10973 110,2222 309 117,0000 107,0000 13777463 8,55%
146769 12634 112,0000 200 117,0000 107,0000 13987011 8,55%
148799 14259 110,6211 333 117,0000 107,0000 14195950 8,55%
150961 15939 110,5556 326 117,0000 107,0000 14412816 8,55%
153052 17636 110,3500 360 117,0000 107,0000 14622570 8,55%
154836 19094 112,6667 210 117,0000 107,0000 14826519 8,55%
Elapsed time = 624,03 sec. (292161,29 ticks, tree = 341,71 MB, solutions = 9)
```

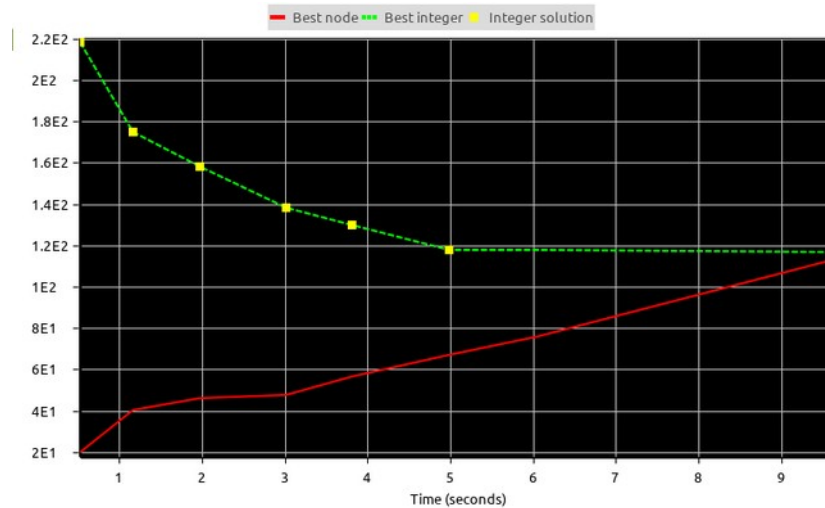
En este caso, aunque pasaron 10 minutos, el solver no terminó. Sin embargo, puede notarse que en los mismos 10 minutos está encontrando soluciones que están a tan solo un 8,55% del óptimo en el peor caso.

Solución con CPLEX (con límite de 11 lavados, con simetría)

En principio, ya podemos intuir que la convergencia va a ser rápida, por estar habilitada la condición de simetría. Ahora se va a poder cuantificar cuánto acelera la convergencia limitar lo más posible la cantidad de lavados.

En este caso, la solución tardó solamente alrededor de 10 segundos (la mitad de lo que tardó anteriormente restringiendo a 15 lavados).

Solapa Statics:



En este caso, lo más destacable del gráfico, es la rápida convergencia tanto de la solución continua como también de la solución entera, que es lo que termina garantizando que el solver termine.

También se puede ver que hallando menos soluciones óptimas enteras, encuentra más rápido el resultado (porque descarta todas las soluciones simétricas).

Solución encontrada:

- Tiempo total de lavado: 117 minutos
- Cantidad de lavados requeridos: 11

Solapa Scripting Log:

solution: 117 /size: 138 /time: 1669309903.071106911

Nodo 1: 1	Nodo 66: 1	Nodo 131: 1
Nodo 2: 1	Nodo 67: 2	Nodo 132: 1
Nodo 3: 1	Nodo 68: 2	Nodo 133: 3
Nodo 4: 1	Nodo 69: 2	Nodo 134: 3
Nodo 5: 9	Nodo 70: 6	Nodo 135: 7
Nodo 6: 5	Nodo 71: 1	Nodo 136: 2
Nodo 7: 4	Nodo 72: 7	Nodo 137: 1
Nodo 8: 2	Nodo 73: 2	Nodo 138: 5
Nodo 9: 1	Nodo 74: 11	
Nodo 10: 2	Nodo 75: 1	
Nodo 11: 1	Nodo 76: 2	
Nodo 12: 2	Nodo 77: 1	
Nodo 13: 2	Nodo 78: 10	
Nodo 14: 1	Nodo 79: 1	
Nodo 15: 6	Nodo 80: 1	
Nodo 16: 2	Nodo 81: 10	
Nodo 17: 3	Nodo 82: 1	
Nodo 18: 3	Nodo 83: 6	
Nodo 19: 1	Nodo 84: 2	
Nodo 20: 6	Nodo 85: 1	
Nodo 21: 1	Nodo 86: 1	
Nodo 22: 5	Nodo 87: 1	
Nodo 23: 1	Nodo 88: 1	
Nodo 24: 4	Nodo 89: 4	
Nodo 25: 1	Nodo 90: 2	
Nodo 26: 5	Nodo 91: 1	
Nodo 27: 2	Nodo 92: 7	
Nodo 28: 2	Nodo 93: 1	
Nodo 29: 2	Nodo 94: 1	
Nodo 30: 1	Nodo 95: 4	
Nodo 31: 6	Nodo 96: 1	
Nodo 32: 4	Nodo 97: 3	
Nodo 33: 1	Nodo 98: 1	
Nodo 34: 2	Nodo 99: 6	
Nodo 35: 6	Nodo 100: 5	
Nodo 36: 8	Nodo 101: 3	
Nodo 37: 2	Nodo 102: 1	
Nodo 38: 1	Nodo 103: 1	
Nodo 39: 1	Nodo 104: 1	
Nodo 40: 1	Nodo 105: 1	
Nodo 41: 2	Nodo 106: 2	
Nodo 42: 1	Nodo 107: 1	
Nodo 43: 1	Nodo 108: 6	
Nodo 44: 2	Nodo 109: 2	
Nodo 45: 7	Nodo 110: 2	
Nodo 46: 1	Nodo 111: 2	
Nodo 47: 1	Nodo 112: 3	
Nodo 48: 1	Nodo 113: 1	
Nodo 49: 2	Nodo 114: 5	
Nodo 50: 2	Nodo 115: 9	
Nodo 51: 1	Nodo 116: 9	
Nodo 52: 1	Nodo 117: 1	
Nodo 53: 6	Nodo 118: 2	
Nodo 54: 3	Nodo 119: 1	
Nodo 55: 1	Nodo 120: 2	
Nodo 56: 1	Nodo 121: 1	
Nodo 57: 11	Nodo 122: 3	
Nodo 58: 8	Nodo 123: 4	
Nodo 59: 1	Nodo 124: 2	
Nodo 60: 5	Nodo 125: 1	
Nodo 61: 6	Nodo 126: 6	
Nodo 62: 2	Nodo 127: 1	
Nodo 63: 1	Nodo 128: 1	
Nodo 64: 9	Nodo 129: 3	
Nodo 65: 5	Nodo 130: 10	

Solapa Engine: (solo el final)

```
MIP search method: dynamic search.
Parallel mode: deterministic, using up to 8 threads.
Root relaxation solution time = 0,13 sec. (103,70 ticks)

      Nodes
      Node Left      Objective  IInf  Best Integer      Cuts/
                                     Best Bound      ItCnt      Gap
*      0+   0                220,0000      0,0000      100,00%
*      0+   0                218,0000      0,0000      100,00%
      0    0                20,0000      915      218,0000      20,0000      1435      90,83%
*      0+   0                175,0000      20,0000      88,57%
      0    0                26,3014      804      175,0000      Cuts: 605      3775      76,80%
*      0+   0                158,0000      40,5963      74,31%
      0    0                37,0000      688      158,0000      Cuts: 699      5564      70,66%
*      0+   0                138,0000      46,3560      66,41%
      0    0                41,5533      660      138,0000      Cuts: 833      9515      65,45%
*      0+   0                130,0000      47,6766      63,33%
      0    0                47,4095      597      130,0000      Cuts: 831      12422      56,49%
*      0+   0                118,0000      56,5623      52,07%
      0    0                62,0000      528      118,0000      Cuts: 833      16772      43,22%
      0    0                62,0000      519      118,0000      Cuts: 507      17782      43,22%
      0    0                62,2672      461      118,0000      Cuts: 497      20102      35,59%
      0    0                62,3360      592      118,0000      Cuts: 536      21353      31,47%
      0    0                62,3750      471      118,0000      Cuts: 266      21655      31,47%
      0    0                62,3750      532      118,0000      Cuts: 504      22522      31,47%
      0    2                64,5392      394      118,0000      80,8627      23421      31,47%
Elapsed time = 6,87 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                102,5994      67457      13,05%
      1910 713                110,6800      99      118,0000      108,1333      83465      8,36%
      2511 743                cutoff                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%

Clique cuts applied:  3
Implied bound cuts applied:  782
Mixed integer rounding cuts applied:  125
Zero-half cuts applied:  41
Multi commodity flow cuts applied:  9

Root node processing (before b&c):
  Real time      =  6,81 sec. (3718,51 ticks)
Parallel b&c, 8 threads:
  Real time      =  2,65 sec. (1518,77 ticks)
  Sync time (average) =  0,51 sec.
  Wait time (average) =  0,00 sec.
-----
Total (root+branch&cut) =  9,46 sec. (5237,28 ticks)
```

En este caso, no solo es relevante notar que pudo resolver el problema en un tiempo muy bajísimo (10 segundos), sino que también da la garantía de que la solución hallada está a tan solo 2,30% del óptimo en el peor caso.

Nuevamente, tenemos que esta forma de procesamiento tiene dos ventajas notorias: tiempo mínimo, y excelente garantía de calidad.

Conclusiones

Al resolver este trabajo, se pudo comparar la diferencia entre resolver un problema mediante la programación de una heurística propia, contra plantear un modelo de programación lineal y luego utilizar un solver comercial (en este caso, IBM ILOG CPLEX).

Para la *heurística*, se puede plantear como principal ventaja que no se requieren conocimientos matemáticos específicos, ni de ningún software comercial específico.

Como principal desventaja, se puede destacar que si la heurística y la dimensión del problema son lo suficientemente grandes como para que no se puedan probar todos los casos posibles, uno no sabe que tan cerca o qué tan lejos está la solución hallada de la solución óptima.

Además, es de destacar que la misma heurística puede dar un resultado bueno para un conjunto de datos, y un resultado malo para otro conjunto de datos.

Para la *programación lineal entera*, podemos destacar como principal ventaja que nos da herramientas mejor estudiadas y más probadas para resolver los problemas, y que no requieren del ingenio del desarrollador y de la suerte con el conjunto de datos. Sobre todo, es muy valorable el hecho de que el solver nos pueda cuantizar el grado de garantía de calidad de la solución hallada, y así conocer la distancia de la solución hallada contra el peor caso.

Como principal desventaja, podemos destacar que se requieren conocimientos matemáticos específicos, y que para resolver problemas medianos o grandes, requerimos de un software específico que tiene una licencia con un costo muy alto.

En cuanto al problema específico de los lavados, puedo destacar:

- Para el solver es muchísimo más relevante tener la condición de simetría, a tener información sobre la limitación en cuanto a la cantidad de lavados.
- Limitar solo la cantidad de lavados, no garantiza la convergencia en un tiempo razonable.
- Agregar solo la condición de simetría, sí garantiza la convergencia en un tiempo razonable.
- Agregarle la limitante de cantidad de lavados a la solución con simetría, permite acelerar muchísimo la convergencia. Luego, no es tan relevante a qué cantidad de lavados se limite (siempre dentro de cierto rango, como fue en este caso entre 11 y 15); Lo relevante es la acotación.

Dada la naturaleza de los datos, se dio la casualidad de que la heurística que yo desarrollé no estuvo tan lejos de la solución que dio el solver. Ambos coincidieron en que se requieren 11 lavados, y los tiempos totales de lavados fueron levemente mejores con el solver (123 minutos contra 117 minutos, es decir, una diferencia de menos del 5 %).

Una última cosa a destacar al utilizar el solver, es la consistencia en las soluciones. Como era esperable, en todos los casos que llegó a la solución, siempre fue la misma en cuanto a distribución de prendas en los diferentes lavados, siempre se requirieron 11 lavados, y siempre el tiempo total de lavado de las prendas fue de 117 minutos.