

# Entrega 4 Modelos I

Noviembre 2022

## 1 Pasos de Resolución

### 1.1 Resolución propia

Usando el modelo que hice en las entregas anteriores, la solución obtenida es la siguiente:

```
Tiempo total consumido: 123. Lavados realizados: 11
```

Como sé que mi implementación no era de las mejores, probablemente no sea una solución muy buena, pero correr el programa toma menos de un segundo.

### 1.2 Correr el modelo de CPLEX sin cambios

En la consigna menciona que si no termina en 10 minutos puedo frenar la ejecución, por lo que a priori ya sé que en términos de tiempo, este método es muchísimo más costoso, así que hay que ver qué tanto mejora el resultado

Tardó 5 minutos en llegar a una solución mejor que la mía (121 vs. 123). Al cabo de 15 minutos la mejor solución obtenida fue de 119

```
solution: 119 /size: 138 /time: 299483.14
```

Con la siguiente tabla, se puede observar como no tarda tanto (aunque sigue siendo mucho más que mi modelo hecho en python) en llegar a un resultado no tan alejado.

	Nodes					Cuts/		
	Node	Left	Objective	IInf	Best Integer	Best Bound	ItCnt	Gap
*	0+	0			2760,0000	0,0000		100,00%
*	0+	0			1467,0000	0,0000		100,00%
*	0+	0			171,0000	0,0000		100,00%
	0	0	20,0000	4286	171,0000	20,0000	11	88,30%
*	0+	0			161,0000	20,0000		87,58%
	0	0	20,0000	1567	161,0000	Cuts: 181	6970	87,58%
	0	0	20,0000	2225	161,0000	Cuts: 1498	17262	87,58%
	0	0	20,0000	1187	161,0000	Cuts: 58	19066	87,58%
*	0+	0			143,0000	20,0000		86,01%
*	0+	0			136,0000	20,0000		85,29%
	0	0	-1,00000e+75	0	136,0000	20,0000	19066	85,29%
	0	0	20,0000	1803	136,0000	Cuts: 1091	29019	85,29%
	0	2	20,0000	944	136,0000	20,0000	29019	85,29%

Elapsed time = 90,50 sec. (69300,89 ticks, tree = 0,02 MB, solutions = 6)

Se puede ver lo mismo en el siguiente gráfico (excepto la parte del principio, donde no se llega a ver cuando el valor es muy alto)

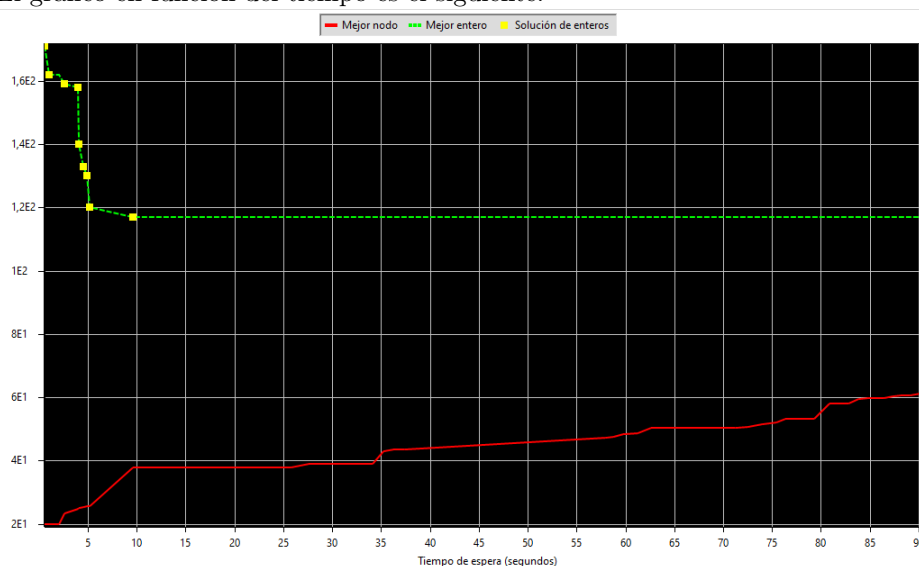


### 1.3 Limitar la cantidad de lavados a 15

Cuando se limita la cantidad de lavados a 15, ya sin correrlo sé que va a ser mucho mas rápido, porque se ahorra probar combinaciones que no van a ser óptimas como por ejemplo que cada prenda se lave sola, sin armar grupos (esta seria la solucion óptima unicamente si todas las prendas fueran incompatibles con todas, pero en ese caso tener un modelo ni siquiera tendría sentido). Al cambiar la restricción en CPLEXy correr el modelo, se ve rápidamente como llega a una solución buena, considerablemente mejor que en el paso anterior.

**solution: 117 /size: 138 /time: 377394.609**

El gráfico en función del tiempo es el siguiente:



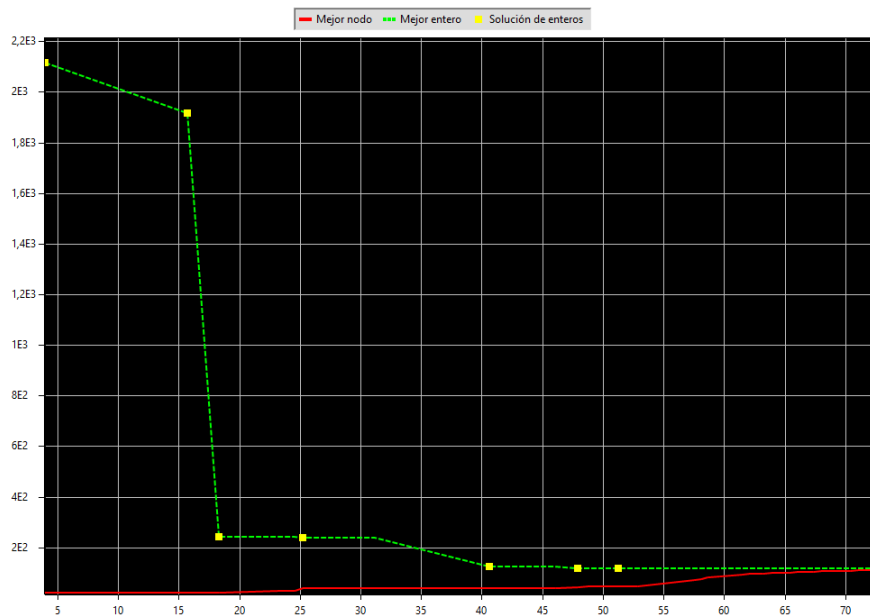
En el gráfico se ve cómo en menos de 10 segundos ya se alcanza su mejor valor, aunque corriéndolo por 12 minutos no llega a terminar la ejecución.

### 1.4 Descomentar la restricción de simetría

Al descomentar la restricción de simetría, lo más llamativo es que el tiempo de ejecución se reduce muchísimo, mientras que antes en 15 minutos todavía no terminaba, ahora en poco más de un minuto sí termina, y además obtiene un resultado un poco mejor que en el punto 1.2 e igual al del 1.3

**solution: 117 /size: 138 /time: 306954.406**

El gráfico en función del tiempo es el siguiente:



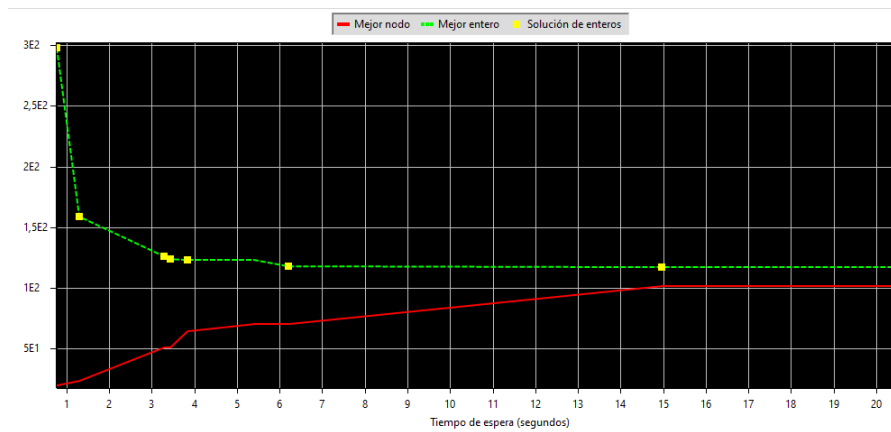
Este método es muchísimo mas rápido porque la restricción de simetría hace que los distintos lavados estén ordenados de mayor a menor, y esto evita que puedan existir soluciones que son idénticas excepto por el orden de sus lavados. Por ejemplo, sin esta restricción existiría una solución donde el primer lavado está compuesto por x, y y z y el segundo por i, j y k y otra en la que el orden sea el opuesto. No aplicar esta restricción implica perder muchísimo tiempo al evaluar soluciones equivalentes a otras ya obtenidas, en las que solo varía el orden de lavados, cosa que no importa en este problema.

### 1.5 Limite de 15 lavados + simetría

Juntando los cambios de los dos puntos anteriores, lo que noto es justamente que el resultado parecería una mezcla de ambos: tarda muy poco en llegar a 117 (como en el 1.3, pero unos pocos segundos mas lento), y además la corrida termina, y lo hace incluso más rápido que en el 1.4, cosa esperable al limitar todavía mas la cantidad de casos posibles

**solution: 117 /size: 138 /time: 378234.812**

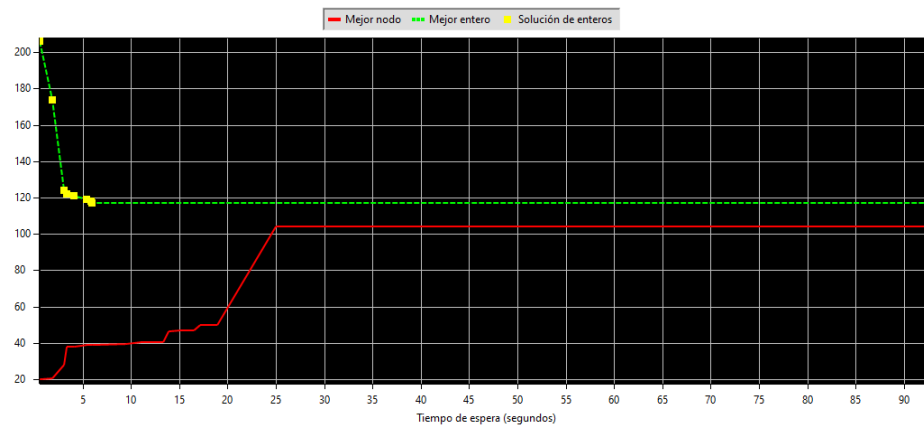
El gráfico es el siguiente:



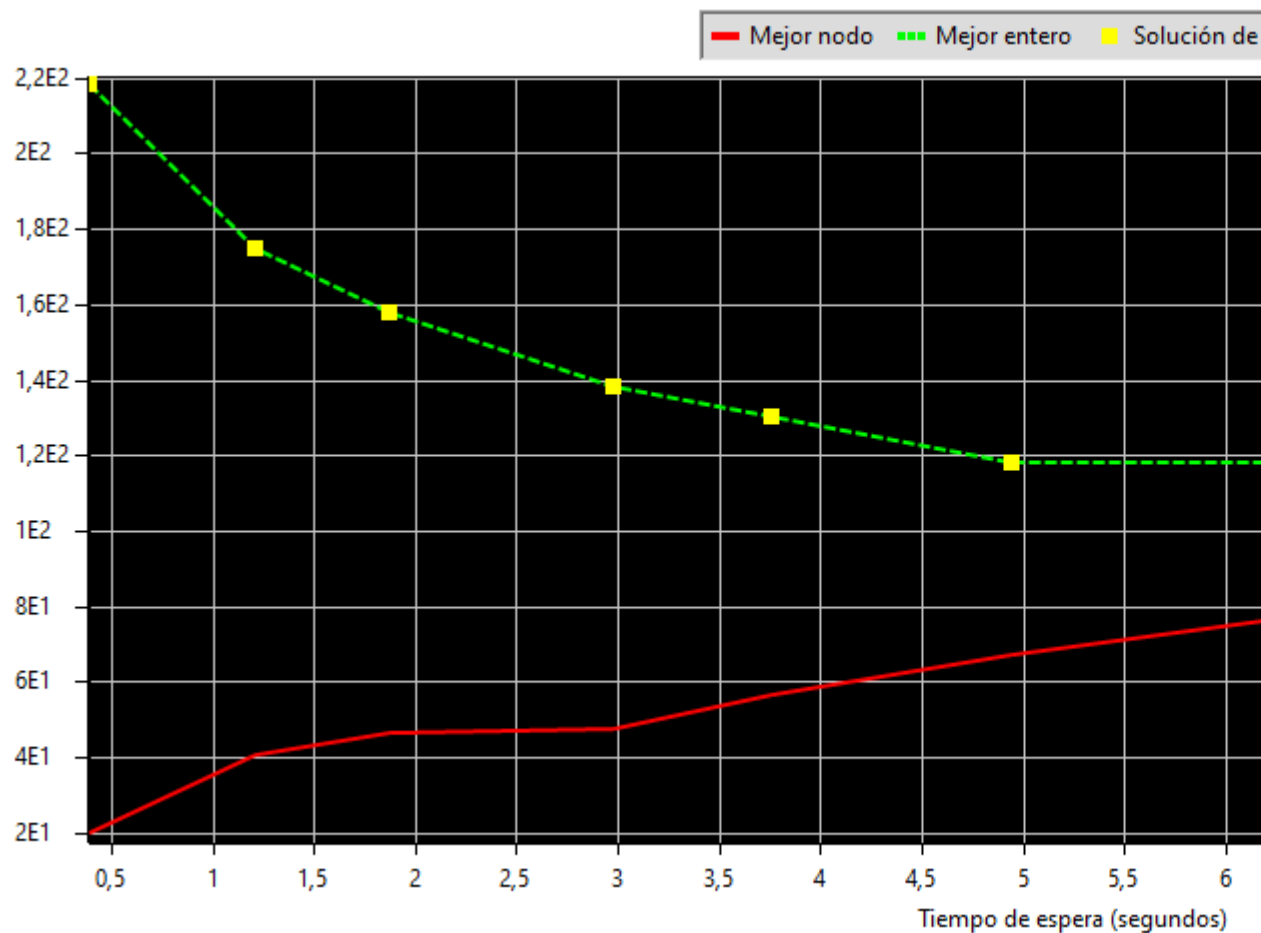
## 1.6 Limitar la cantidad de lavados a 11

Lo que espero es lo mismo que en el 1.3, por las mismas razones. Efectivamente llega a 117 aún mas rápido, en este caso, en alrededor de 6 segundos. Y al igual que en el punto anterior mencionado, al cabo de 12 minutos aún no termina.

**solution: 117 /size: 138 /time: 380306.156**



Agregando la restricción de simetría, se llega al mejor resultado muy rápido y en ese momento la resolución termina:



Lo que entiendo al ver esto, es que una vez que llega a este resultado ya no quedan casos por probar, porque la restricción de simetría hizo que el orden de los grupos de prendas que logran el resultado de 117 sea único, mientras que sin esa restricción se quedaba probando combinaciones de los mismos grupos pero en diferentes ordenes, es decir, 11 factorial formas de ordenar algo en lo cual el orden no es relevante, y eso para cada configuración de grupos.

### 1.7 Comparación entre la heurística y el modelo de programación lineal

A lo largo de esta entrega, se fue mejorando un modelo de programación lineal que inicialmente no me resultó convincente, porque tardaba mucho y conseguía un resultado apenas mejor al de mi heurística. Pero en cada punto, se iban agregando mejoras al modelo, que finalmente hicieron que los resultados obtenidos sean mucho mas convincentes. La restricción de simetría fue muy importante para reducir los tiempos, y muestra cómo hay restricciones que no cambian el

mejor resultado posible, pero que son prácticamente necesarias si se quiere llegar al mismo en un tiempo viable. Por otro lado, las partes del límite de lavados fueron posibles por el uso de una heurística. En este caso mi heurística llegó a esa cantidad de lavados que menciona el enunciado. Yo de esto saco que las heurísticas resultan muy útiles para conseguir una aproximación que sirva para agregar restricciones al modelo de programación lineal. Estas restricciones extra dependerían siempre de los datos con los cuales se busca resolver el problema, por lo que es un trabajo que se debería hacer en cada caso, sin embargo, claramente vale la pena, teniendo en cuenta que mi heurística no tarda ni 1 segundo en correr.