

Informe Final

Grupo: 6

Nombres: DIEGO VILLEGAS, TOMAS DONOSO, MARTIN MORAGA, NELSON SEPÚLVEDA

Descripción del problema

Un fabricante de chips tiene que planificar la producción para los próximos M meses. Se conoce el precio de venta de los chips así como la demanda mensual de estos (la demanda varia en cada mes y va en aumento ($\text{demandaMes1} < \text{demandaMes2} < \dots < \text{demandaMesM}$). Se prevee una subida mensual de los costos de producción ($\text{costoMes1} < \text{costoMes2} < \dots < \text{costoMesM}$). La fabrica puede generar una producción mensual fija en horario normal y además, en horario extra la producción puede ser incrementada en cierta cantidad de unidades a un costo mensual (diferente para cada mes). Se cuenta con un almacén de capacidad limitada para almacenar el exceso de producción incurriendo en un costo unitario mensual (igual para cada mes). Al final de los M meses la existencia de productos en almacén tiene que ser nula y las demanda de chips siempre debe ser satisfecha.

Además, se ha definido que:

- En el mes 1 la cantidad de chips fabricados tiene que ser a lo mas la quinta parte de la producción total de chips para todo el periodo de planificación (sumatoria de M meses).
- Si decide producir en horario extra entonces, la cantidad de unidades producidas puede ser a lo menos la mitad de la producción mensual en horario normal para el mes en curso.

Función Objetivo: Minimizar el costo total

Modelo Matemático

Parámetros

- d_i : Demanda del mes i .
- c_i : Costo de producción normal en el mes i .
- e_i : Costo de producción en horario extra en el mes i .
- a : Costo de almacenamiento por unidad en el mes.
- CN_{max} : Cantidad máxima de producción en horario normal.
- CE_{max} : Cantidad máxima de producción en horario extra.
- CA_{max} : Cantidad máxima de almacenamiento.

Variables de decisión

- x_i : $\begin{cases} 1, & \text{Se produce en horario extra} \\ 0, & \text{Caso contrario.} \end{cases}$
- y_i : la cantidad de unidades almacenadas al final del mes i .
- h_i : la cantidad de unidades producidas en horario extra en el mes i .
- p_i : Producción mensual final en horario normal en el mes i .

Función Objetivo

$$\min(Z) = \sum_{i=1}^m (c_i * p_i + e_i * h_i + y_i * a)$$

Restricciones

- Restricción de producción máxima en el mes 1: $p_1 + h_1 \leq (\sum_{i=1}^m p_i + h_i) * 1/5, \forall i = \{1...m\}$,
- Restricción de producción en horario extra en relación a la producción normal: $h_i \geq 0,5 * p_i - M(1 - x_i), \forall i = \{1...m\}$, Donde M es una constante muy grande.
- Restricción de balance de inventario: $y_i = y_{i-1} + p_i + h_i - d_i$. Para $i > 1$, $y_1 = p_1 + h_1 - d_1, \forall i = \{1...m\}$
- Restricción de no almacenamiento al final de los m meses: $y_m = 0$
- Restricción de satisfacción de la demanda: $p_i + h_i + y_i \geq d_i, \forall i = \{1...m\}$
- Restricción de la demanda del primer mes sin almacén: $p_1 + h_1 \geq d_1$
- Restricción de producción máxima en horario normal: $CN_{max} \geq p_i, \forall i = \{1...m\}$
- Restricción de producción máxima en horario extra: $CE_{max} * x_i \geq h_i, \forall i = \{1...m\}$
- Restricción de almacenamiento máximo: $CA_{max} \geq y_i, \forall i = \{1...m\}$
- Restricción de no negatividad: $y_i, h_i, p_i \geq 0, \forall i = \{1...m\}$

Generación de instancias y Factibilidad

Para la generación de instancias utilizaremos el lenguaje de programación Python. Comenzamos definiendo la lista de la cantidad de meses a ocupar para cada instancia cumpliendo con los requerimientos del enunciado, además para este caso se ocupará la librería “random” que nos permitirá obtener números aleatorios para un cierto rango de valores y por último en el proceso se utilizará una constante M que será equivalente a un numero muy mayor, es decir, al momento de usarla ningún valor puede ser mayor a esta constante. Realizando todos estos pasos se establecerán los rangos de valores entre los cuales oscilarán los parámetros a entregar, con esto dado se establecerán los valores de los parámetros para cada instancia que posee una cantidad de meses definida al inicio. Con esto realizado se realizarán distintos procedimientos para establecer los valores de cada parámetro, definidos a continuación:

Máxima producción normal CN_{max}

Utilizando los rangos establecidos en un inicio y utilizando la función randint de la librería random para establecer el valor de este parámetro, este valor se mantendrá constante para cada instancia, ya que corresponde al máximo de producción que posee la fábrica.

Máxima producción extra CE_{max}

Este parámetro corresponde al máximo de producción extra, este posee ciertas restricciones y es por esto mismo que para asegurar la factibilidad de la instancia al utilizar randint, se establecerá como rango inferior la mitad del valor de CN_{max} entregado anteriormente y como rango superior el mismo rango superior utilizado para el caso del parámetro CN_{max} , esto ya que como indica el enunciado para nuestro grupo la producción extra tiene que ser a lo menos la mitad de la producción mensual en horario normal, es decir, si utilizamos el caso en que CE_{max} fuera menor que la mitad de CN_{max} , este podría tener valores para los cuales la instancia sería insatisfactoria. Este valor es constante para todos los meses de la instancia.

Máximo almacenamiento CAm_{ax}

Para este valor se utilizarán los rangos establecidos en un inicio y se usara la función randint con estos rangos, de esta forma estableciendo el máximo de almacenaje posible, el cual es constante para todos los meses de la instancia.

Costo de almacenaje a

Este valor corresponde al costo de almacenaje por unidad en el almacén de la fábrica, este valor es constante por cada instancia, para establecer este valor se utilizó la función randint con los valores de los rangos dados en un inicio.

Costo de producción normal c

Este valor corresponde al coste de producción de cada unidad, este valor es distinto para cada mes, además se debe cumplir que este valor aumente cada mes, para realizar esto se estableció un rango inicial entre los que oscilaran estos valores y se partió definiendo el valor del primer mes que tiene como límite inferior el mismo que el rango inicial y como límite superior el del rango inicial menos la cantidad de meses restantes de la instancia, esto para asegurar de que en caso de que al utilizar la función randint el valor podría seguir aumentando dentro de los rangos iniciales, después para los siguientes meses se definirán los valores siguiendo un orden ascendente y se utilizara la misma definición de límite superior con la diferencia de que el límite inferior va a ser equivalente al valor del costo en el mes anterior mas uno, asegurando de esta manera que el valor sea mayor que el mes anterior.

Costo de producción extra e

Este valor corresponde al costo de producción en horario extra, este parámetro es distinto para cada mes de la instancia, es por esto por lo que utilizando los rangos dados en un inicio y se utilizara la función randint para definir los valores para cada mes, en el generador nos aseguramos de que sean distintos y es por esto por lo que si un valor es equivalente a alguno definido para un mes anterior se volverá a generar hasta que no este en esta lista de costos de producción en horario extra.

Demanda d

Este valor corresponderá a la demanda de cada mes en la instancia, este valor debe cumplir con la restricción de aumentar el valor para cada mes de la instancia, para definir cada mes se partirá definiendo el valor del último mes (el mes con mayor demanda), el valor de este tendrá como límite inferior será la suma del límite inferior del rango inicial dado mas la cantidad de meses asegurando que puede disminuir por cada mes y como límite superior se tendrá la suma del máximo de producción en horario normal con el máximo de producción en horario extra y con el máximo de almacenaje, esto ya que este es el máximo valor de producción de la fábrica. Si este valor es mayor que la suma de los máximos de producción en horario normal y extra, se definirán variables que establecerán cuanto almacenaje se debe tener para que el valor de demanda del mes sea factible, los valores de estas variables corresponden al valor dado menos los máximos (variable **almacenado**) y el segundo corresponde al valor del máximo de almacenaje menos la variable **almacenado** (variable dif).

Partiendo de este caso nos pueden dar distintos casos que deben cumplir con algunas restricciones. El primer caso es que el siguiente mes si da una demanda mayor a los máximos de producción, se tendrán que sumar los valores de **almacenado** para este mes cumpliendo con la restricción que este valor no sea mayor que la capacidad máxima de almacenaje además de que el valor de demanda de este mes sea menor que el definido anteriormente. El segundo caso es que el valor de demanda sea menor que las sumas de los máximos de producción y por lo tanto la generación de demandas pase a una siguiente fase en la que se establece el valor de dif a 0 para la diferenciación de las fases, que debe cumplir otras restricciones. Esta fase debe cumplir

que el valor de **almacenado** sea 0 para cuando se defina la última demanda (el menor valor), para realizar esto se diferencian dos casos, el primero es que se este definiendo la demanda de un mes distinto del último (que corresponde a la demanda del mes 1), en este caso se establece el mismo limite inferior que para todos los casos y como limite superior la demanda definida anteriormente menos uno, con este valor se establece una diferencia sumando los máximos y restándoles este valor, el valor de esta diferencia se le restara al valor de **almacenado** cumpliendo, es decir, el valor de **almacenado** es una compensación que deben cumplir los meses con demandas por debajo de las producciones máximas debido a las demandas por encima de estas producciones, el segundo caso que puede darse es que se tenga que definir el valor de la última demanda y por lo tanto toda la compensación se deba realiza en este mes, es por esto que en este caso el límite superior cambia a la suma de los máximos menos el valor de **almacenado**. Como se menciono en un inicio puede darse el caso que nunca suceda lo mencionado anteriormente por lo que se define un último caso que solo cumple con que la demanda disminuya durante la generación de cada mes. Todos estos casos mantienen la factibilidad de la instancia.

Conclusión de generación

Utilizando estos procedimientos se realizarán la creación de todas las instancias y la información que estás posean serán trasladadas al solver Minizinc.

Aleatoriedad

Como se definió con anterioridad al inicio de la generación se establecían ciertos rangos de valores para la creación de los parámetros, esto muestra que la aleatoriedad va a estar restringida desde un inicio para un cierto rango de valores. Utilizando estos rangos y la función randint se definieron los valores de los parámetros de forma aleatoria entre estos rangos para la mayoría de los casos. Para algunos casos como es el de CEmax se tenían los valores de CNmax para su creación, a pesar de esto al usar valores aleatorios en CNmax los valores de CEmax también son aleatorios. Para el caso de la generación de costos de producción en horario normal y de demandas al tener la restricción de aumentar al paso de los meses estos valores tenderán a tener los mismos valores al ir aumentando el valor de los meses en las instancias, es decir, el valor de los costos de producción en horario normal al comenzar por la generación del primer mes hasta llegar al último mes (mes con mayor costo), sus últimos valores van a tender al límite superior del rango inicial cuando el valor de la instancia es alto, por otro lado las demandas al empezar a generarlas desde el ultimo mes hasta el primer (mayor demanda a menor de manda), sus primeros valores van a tender al limite inferior del rango inicial al tener valores de instancia altos, es por esto que a pesar de utilizar valores “aleatorios” al restringir los rangos de valores, al aumentar el numero de instancias, el generador tiende a entregar valores similares. En conclusión, a pesar de utilizar herramientas que entregan valores aleatorios, debido a las restricciones que se deben cumplir para mantener la factibilidad de un problema se tienen que definir rangos entre los que oscilan los valores.

Por ultimo para la creación de nuestras instancias utilizamos los siguientes rangos de valores:

- Rango de costo de producción normal: 5000-10000
- Rango de costo de producción extra: 5000-10000
- Rango de costo de almacenamiento: 10-1000
- Rango de capacidad de almacenamiento: 1-5000
- Rango de producción normal: 10000-20000

Explicación herramienta de Búsqueda

Minizinc es un lenguaje modelado, que trabaja con restricciones y problemas de optimización, de manera que permite especificar restricciones y objetivos con el fin de buscar una o unas soluciones óptimas para el problema planteado. Funciona de tal manera que se deben definir las variables y restricciones del problema, siguiendo de la definición del objetivo, luego se debe ejecutar el modelo y por último se analiza y utiliza la solución entregada para el problema planteado.

CoinBC el solucionador que nosotros utilizamos, ya que resuelve problemas tipo CSOP (Constraint Satisfaction and Optimization), el cual esta estrechamente relacionado con el proyecto. Cuando ejecutamos el modelo en Minizinc con este solucionador, Minizinc traduce el modelo a un formato que CoinBc pueda entender, esto implica transformar todo el modelo, con variables y restricciones incluidas. Una vez que CoinBc puede leer el problema este busca la solución óptima, o las soluciones óptimas. El Solver usa algoritmos que exploran de manera lineal las posibles soluciones buscando la más satisfactoria. Ya con el resultado, Minizinc lo muestra de manera legible.

El algoritmo utilizado por CoinBC es el Branch and Cut (Ramificación y Corte). La “ramificación” busca las diferentes soluciones que pueden ser factibles para el problema, mientras que el “corte” elimina las que no pueden ser posibles, de manera que se van filtrando las soluciones hasta llegar a la más óptima o las más óptimas.

Pulp es una MILP (Mixed Integer Linear Programming) escrita en Python, la cual es compatible con el solucionador CoinBC. Es una interfaz sencilla de utilizar, que se usa para modelar y resolver problemas de optimización lineal y entera.

Una de las ventajas de utilizar Pulp es: permite definir de manera clara y concisa el objetivo y las restricciones de un problema de optimización lineal o entera. Puedes declarar variables, establecer relaciones de igualdad o desigualdad, y agregar restricciones lineales.

Resultados de las instancias

Antes de comenzar cabe recalcar que la calidad de la solución se obtuvo en base a un análisis de sensibilidad menor en los parámetros de la función objetivo, para así observar que tan robusta era la solución, estos valores fueron calculados desde python con pulp

1. Para la instancia 1 de 7 meses, se obtuvieron los siguientes valores:

- Costo total: 1128699535
- Iteraciones: 1
- Tiempo: 149ms
- Calidad de la solución: Buena debido a que esta es robusta frente a variaciones menores, ya que en un 6% de los análisis de sensibilidad cambia el punto óptimo.
- cantidad de variables: 28

2. Para la instancia 2 de 15 meses, se obtuvieron los siguientes valores:

- Costo total: 1507372243
- Iteraciones: 1
- Tiempo: 143ms
- Calidad de la solución: es mala debido a que esta posee 46 nuevas soluciones óptimas de 62 análisis realizados lo que equivale a un 74%.
- cantidad de variables: 60

3. Para la instancia 3 de 25 meses, se obtuvieron los siguientes valores:

-
- Costo total: 1498149534
 - Iteraciones: 1
 - Tiempo: 144ms
 - Calidad de la solución: es mala debido a que esta posee 83 nuevas soluciones óptimas de 102 análisis realizados lo que equivale a un 81 %.
 - cantidad de variables: 100
4. Para la instancia 4 de 35 meses, se obtuvieron los siguientes valores:
- Costo total: 2232656939
 - Iteraciones: 1
 - Tiempo: 135ms
 - Calidad de la solución: es mala debido a que esta posee 137 nuevas soluciones óptimas de 142 análisis realizados lo que equivale a un 96 %.
 - cantidad de variables: 140
5. Para la instancia 5 de 45 meses, se obtuvieron los siguientes valores:
- Costo total: 2514842305
 - Iteraciones: 2
 - Tiempo: 166ms
 - Calidad de la solución: es mala debido a que esta posee 151 nuevas soluciones óptimas de 182 análisis realizados lo que equivale a un 82 %.
 - cantidad de variables: 180
6. Para la instancia 6 de 115 meses, se obtuvieron los siguientes valores:
- Costo total: 5145979366
 - Iteraciones: 1
 - Tiempo: 195ms
 - Calidad de la solución: es mala debido a que esta posee 435 nuevas soluciones óptimas de 462 análisis realizados lo que equivale a un 94 %.
 - cantidad de variables: 460
7. Para la instancia 7 de 145 meses, se obtuvieron los siguientes valores:
- Costo total: 6437200431
 - Iteraciones: 1
 - Tiempo: 206ms
 - Calidad de la solución: es mala debido a que esta posee 574 nuevas soluciones óptimas de 582 análisis realizados lo que equivale a un 98 %.
 - cantidad de variables: 580
8. Para la instancia 8 de 175 meses, se obtuvieron los siguientes valores:
- Costo total: 7156594843
 - Iteraciones: 1
 - Tiempo: 248ms

- Calidad de la solución: es mala debido a que esta posee 633 nuevas soluciones óptimas de 702 análisis realizados lo que equivale a un 90 %.
- cantidad de variables: 700

9. Para la instancia 9 de 185 meses, se obtuvieron los siguientes valores:

- Costo total: 7266433270
- Iteraciones: 1
- Tiempo: 287ms
- Calidad de la solución: es mala debido a que esta posee 724 nuevas soluciones óptimas de 742 análisis realizados lo que equivale a un 97 %.
- cantidad de variables: 740

10. Para la instancia 10 de 195 meses, se obtuvieron los siguientes valores:

- Costo total: 7565493948
- Iteraciones: 1
- Tiempo: 319ms
- Calidad de la solución: es mala debido a que esta posee 767 nuevas soluciones óptimas de 782 análisis realizados lo que equivale a un 98 %.
- cantidad de variables: 780

11. Para la instancia 11 de 500 meses, se obtuvieron los siguientes valores:

- Costo total: 19319536068
- Iteraciones: 1
- Tiempo: 718ms
- Calidad de la solución: es mala debido a que esta posee 1999 nuevas soluciones óptimas de 2002 análisis realizados lo que equivale a un 99 %.
- cantidad de variables: 2000

12. Para la instancia 12 de 650 meses, se obtuvieron los siguientes valores:

- Costo total: 25710357598
- Iteraciones: 1
- Tiempo: 928ms
- Calidad de la solución: es mala debido a que esta posee 2594 nuevas soluciones óptimas de 2602 análisis realizados lo que equivale a un 99 %.
- cantidad de variables: 2600

13. Para la instancia 13 de 700 meses, se obtuvieron los siguientes valores:

- Costo total: 27941800070
- Iteraciones: 1
- Tiempo: 1310ms
- Calidad de la solución: es mala debido a que esta posee 2793 nuevas soluciones óptimas de 2802 análisis realizados lo que equivale a un 99 %.
- cantidad de variables: 2800

14. Para la instancia 14 de 800 meses, se obtuvieron los siguientes valores:

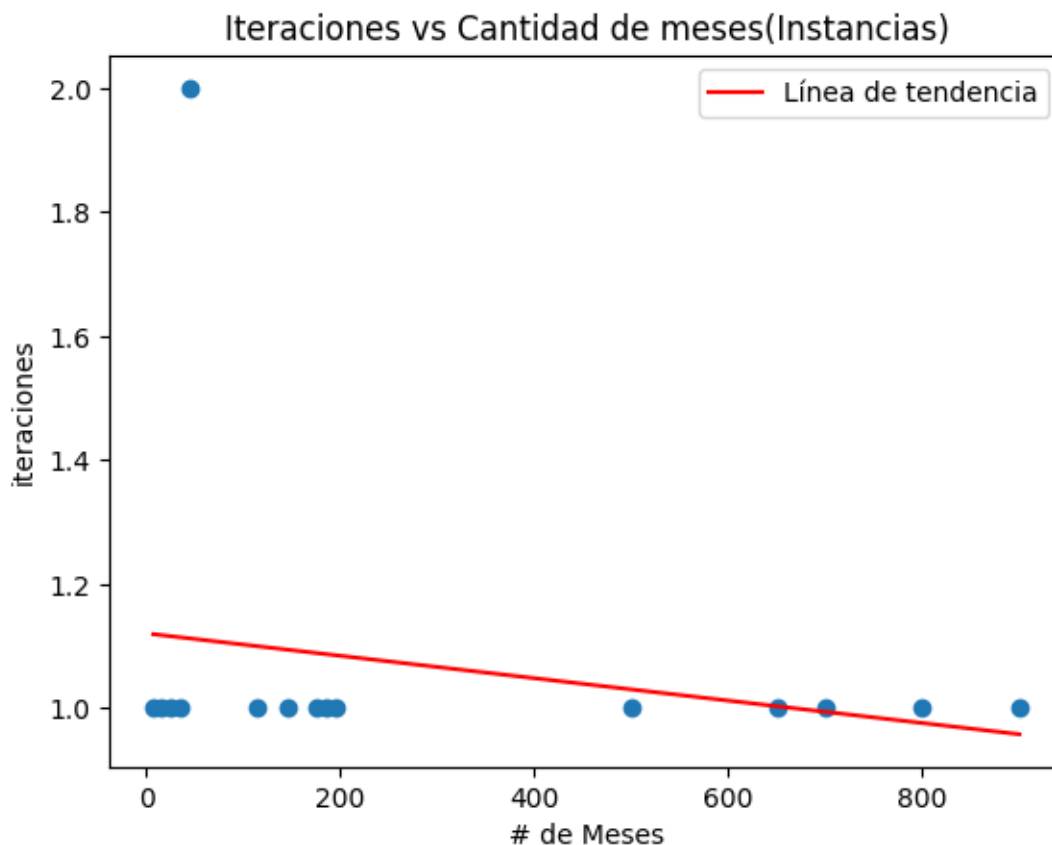
- Costo total: 31661618367
- Iteraciones: 1
- Tiempo: 900ms
- Calidad de la solución: es mala debido a que esta posee 3194 nuevas soluciones óptimas de 3202 análisis realizados lo que equivale a un 99%.
- cantidad de variables: 3200

15. Para la instancia 15 de 900 meses, se obtuvieron los siguientes valores:

- Costo total: 36487417204
- Iteraciones: 1
- Tiempo: 1305ms
- Calidad de la solución: es mala debido a que esta posee 3583 nuevas soluciones óptimas de 3602 análisis realizados lo que equivale a un 99%.
- cantidad de variables: 3600

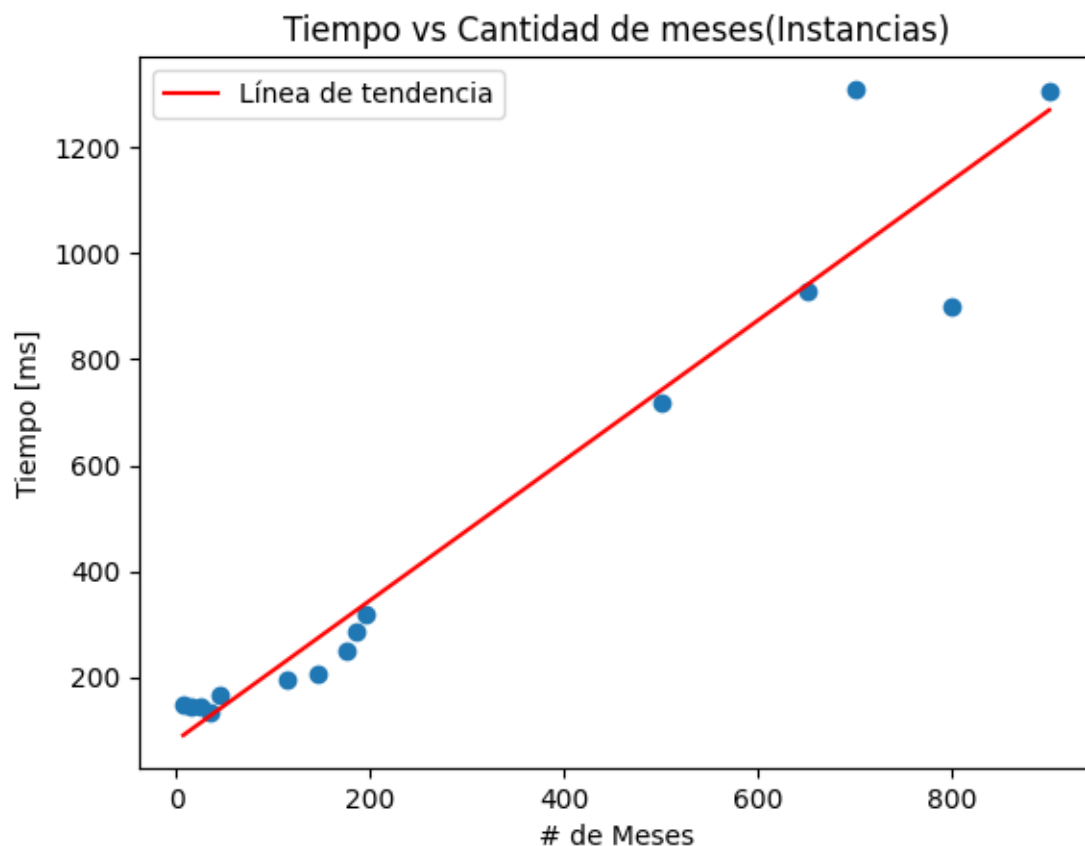
Gráficos de las instancias

En el siguiente gráfico se mostrara la cantidad de iteraciones de soluciones óptimas que realizo el solver CoinBC de Minizinc:



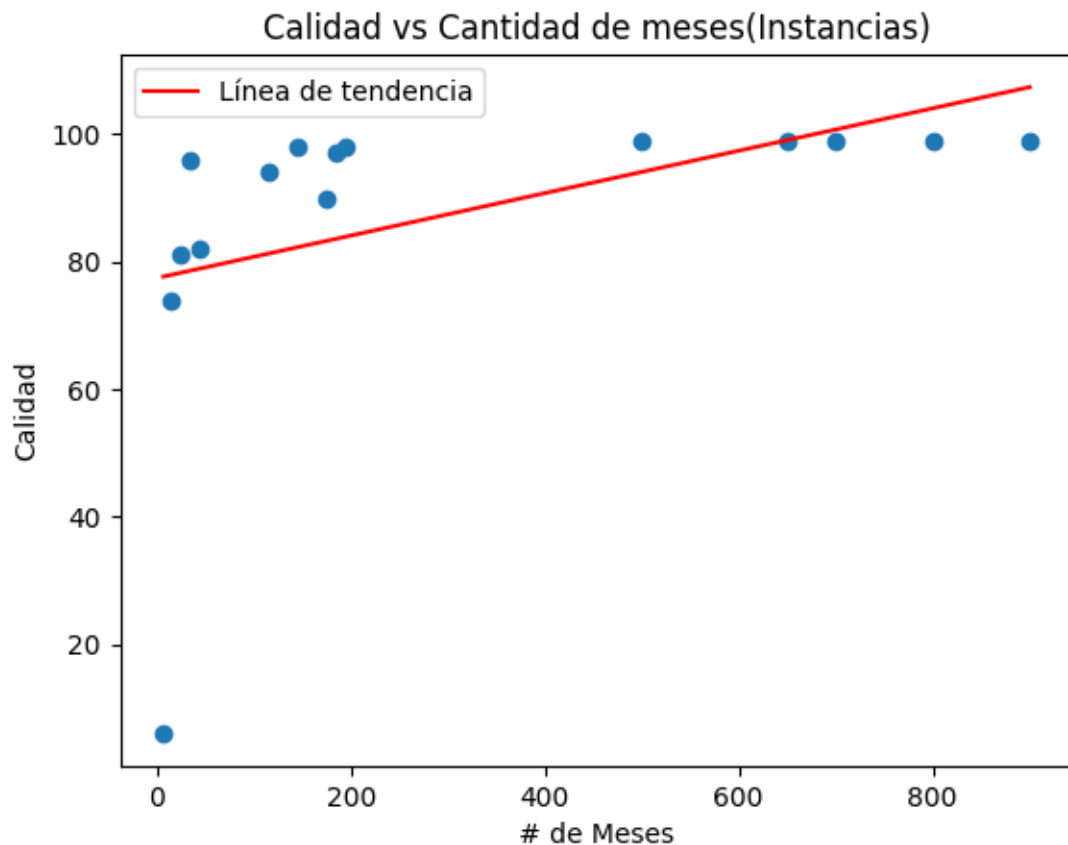
Como se puede observar en este gráfico los valores son prácticamente constantes con respecto a las instancias estudiadas, esto ya que para la mayoría de los casos necesita una sola iteración para encontrar la solución óptima.

En el segundo gráfica se pueden apreciar el tiempo de ejecución de cada instancia utilizando el solver CoinBC de Minizinc:



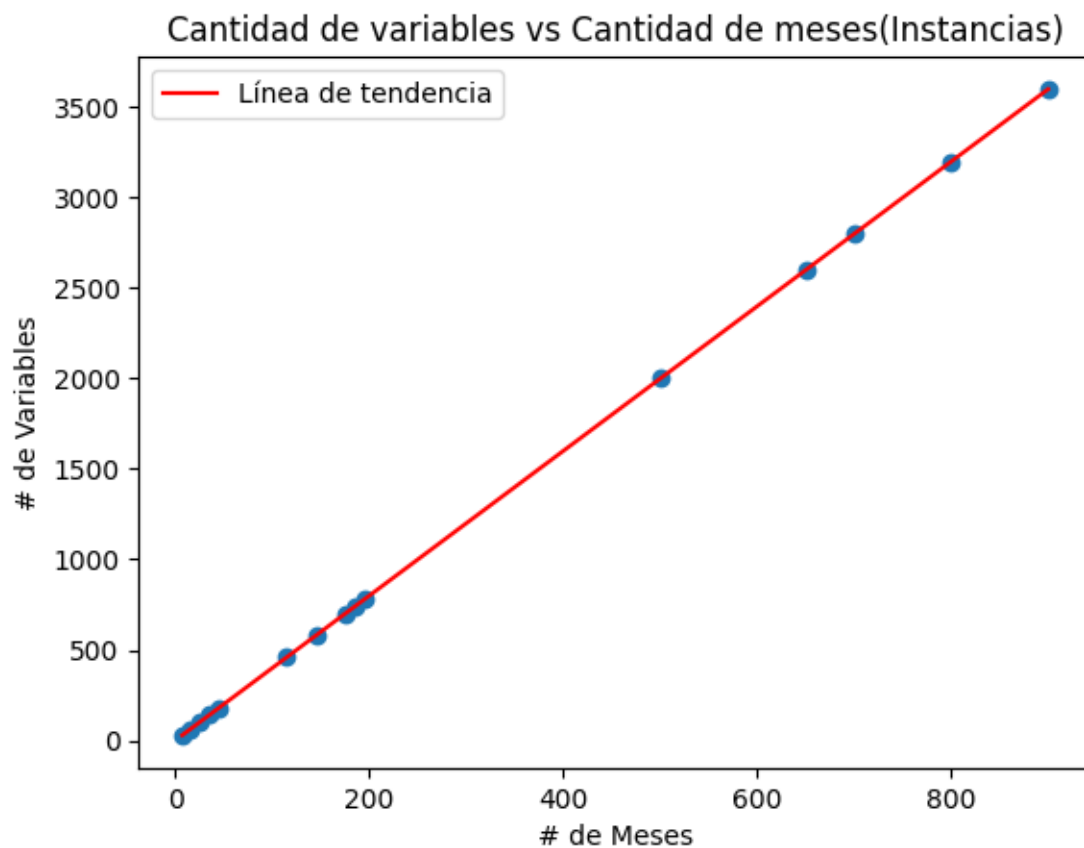
Analizando este gráfico se puede apreciar que el tiempo crece de manera lineal con respecto a la cantidad de meses de cada instancia.

En el tercer gráfico muestra la calidad de las variables de cada instancia, esta calidad se define como un porcentaje de cuanto cambian las soluciones óptimas al cambiar(en este caso un 5%) algún valor de una variable:



Como se puede apreciar el porcentaje aumenta de forma exponencial mientras aumenta en los primeros meses, pero nunca llegando a ser el 100%, esto quiere decir que para un numero mayor de meses la calidad sera menos robusta, es decir, tenderá al cambio si alguna de sus variables es modificada.

Como ultimo gráfico están las cantidad de variables por cada instancia:



Se puede apreciar que la cantidad de variables aumenta de manera lineal al aumentar los valores de las instancias.

Link a GoogleDrive

Click Aquí: [Código del Proyecto](#).