

Práctica 2

Alejandro Luque Villegas, Maria Solórzano Gómez
Comentarios sobre la Práctica 2

May 12, 2025

1 Comentarios sobre la solución con MiniZinc

1.1 Parámetros y variables

Para poder resolver el problema con MiniZinc se usan los siguientes parámetros y variables:

1.1.1 mensualMin

El array mensualMin guarda el número mínimo de aceites diferentes que deben usarse cada mes. Por ejemplo, si $\text{mensualMin}[m] = 2$, significa que al menos 2 aceites deben refinarse en el mes m .

1.1.2 cantidadMin

El array cantidadMin guarda la cantidad mínima de aceite que debe refinarse si se usa un aceite específico. Por ejemplo, si $\text{cantidadMin}[a] = 100$, significa que si el aceite a se refina, debe refinarse al menos 100 unidades.

1.1.3 Incompatibles

El array incompatibles es una matriz que indica si dos aceites son incompatibles. Si $\text{incompatibles}[i, j] = 1$, los aceites i y j no pueden refinarse juntos en el mismo mes.

1.1.4 Requeridos

El array requeridos es una matriz que indica si un aceite requiere otro. Si $\text{requeridos}[i, j] = 1$, significa que si el aceite i se refina, el aceite j también debe refinarse.

1.1.5 aceiteTotal

Las variables aceiteTotal $[m, a]$ representan la cantidad total de aceite a disponible en el mes m (antes de refinar). Esto incluye el aceite inicial, el comprado y el que queda del mes anterior.

1.1.6 aceiteComprado

Las variables `aceiteComprado[m, a]` representan la cantidad de aceite a comprada en el mes `m`.

1.1.7 cuantoRefinar

Las variables `cuantoRefinar[m, a]` representan la cantidad de aceite a refinada en el mes `m`.

1.2 Restricciones

La primera restriccion que tenemos calcula el aceite total disponible en cada mes. En el primer mes, se suma el aceite inicial y el comprado. En los meses posteriores, se calcula en función del aceite que quedó del mes anterior, el comprado y el refinado. Otra restricción asegura que no se compre más aceite del que cabe en el almacenamiento. También hay restricciones que limitan la cantidad máxima de aceite que puede almacenarse (MCAP) y que aseguran que no se desvíe el inventario final de aceite respecto al inicial más allá de un porcentaje (PV). Además, se asegura que no se refine más aceite del disponible y que no se excedan los límites de refinación para aceites vegetales (MAXV) y no vegetales (MAXN). También se verifica que la densidad de los aceites refinados esté dentro de un rango (MinD y MaxD) si se usan aceites no vegetales.

Otras restricciones aseguran que se alcance un beneficio mínimo (MinB), que se usen al menos `mensualMin[m]` aceites diferentes en cada mes, y que si se usa un aceite, se refine al menos `cantidadMin[a]` unidades. También se imponen restricciones sobre aceites incompatibles, asegurando que si dos aceites son incompatibles, no se refinan juntos en el mismo mes. Finalmente, se imponen restricciones sobre aceites requeridos, asegurando que si un aceite requiere otro, ambos deben refinarse juntos en el mismo mes.

Para el problema de optimización se busca maximizar el beneficio total obtenido al refinar y vender aceites durante los seis meses. La suma de estas diferencias para todos los meses y todos los tipos de aceite da como resultado el beneficio total.

2 Comentarios sobre la solución con Python y Z3

En este código, hemos desarrollado un modelo de optimización utilizando Z3 para gestionar la refinación de aceites durante seis meses. Nos enfocamos en cumplir tanto restricciones duras como suaves. Las restricciones duras garantizan que se respeten los límites básicos del problema, como la cantidad máxima de aceite que se puede refinar, las capacidades de almacenamiento, los requisitos de dureza y los beneficios mínimos que debemos alcanzar.

Por otro lado, implementamos restricciones suaves para manejar situaciones más flexibles. Por ejemplo, si un aceite es requerido, otro también debe ser refinado, o si dos aceites son incompatibles, no deberían refinarse al mismo tiempo. Estas restricciones suaves tienen

penalizaciones asociadas, lo que nos permite priorizar su cumplimiento sin hacer que el problema sea infactible si no se cumplen completamente.

Además, diseñamos una función de minimización que busca reducir el número total de aceites utilizados en los seis meses. Esto lo logramos sumando las variables que indican si un aceite es refinado en cada mes y minimizando este valor. El objetivo es encontrar una solución eficiente que cumpla con las restricciones y optimice el uso de recursos. Al final, el modelo evalúa todas las condiciones y nos indica si existe una solución.

3 Restricciones adicionales propuestas

La restricción adicional que hemos decidido implementar en minizinc ha sido añadir un coste de refinamiento por cada aceite, y este coste se tiene en cuenta a la hora de calcular el beneficio final. Adicionalmente, podrían añadirse costes de almacenamiento (pudiendo almacenar mas aceite, pero pagando por cada contenedor), o asegurarse de mantener X beneficios cada mes. Estas restricciones son interesantes de cara a simular los beneficios y costes de una empresa más fielmente.