

Desafío Producción Óptima

Un productor de cerveza tiene portfolio de productos $n = 7$ diferentes tipos de cervezas a sus clientes. La utilidad por litro de cada una de estas cervezas es \$2, \$2, \$6, \$10, \$10, \$2 y \$8.

Para lograr estas nuevas variedades ya no solo usa levadura y cebada como vimos en clases, sino también malta, trigo y lúpulo. Debido a la restricciones de capacidad en sus bodegas no puede tener más de 7 toneladas de levadura, 5 de cebada, 5 de malta, 7 de trigo y 2 de lúpulo.

La siguiente tabla muestra los consumos (en kilogramos) de cada materia prima para producir un litro de cada tipo de cerveza.

1. Formule el problema como un problema de optimización lineal.
2. Implemente y resuelva el problema en Python/Gurobi usando el algoritmo Simplex. Escriba la solución y el valor óptimo (se selecciona usando `modelo.params.method = 0`).
3. Resuelva ahora usando los otro de los algoritmos de Gurobi. (los valores son: 1=dual simplex, 2=barrier, 3=concurrent, 4=deterministic concurrent, 5=deterministic concurrent simplex). Escriba la solución y el valor óptimo.
4. Compare los resultados de las partes (2) y (3). ¿Qué puede decir respecto del problema que estamos tratando de resolver?

Desarrollo

1. Como la realización de la pregunta 2 es bastante similar al trabajo realizado en clases de la cerveza, con la única diferencia es en el aumento de ingredientes y tipos de cerveza a realizar. Dado que las restricciones de capacidad en las bodegas es en toneladas, entonces es necesario transformarlas en kilogramos para que se encuentre a par con la tabla de datos.

x_i = cantidad a producir de la cerveza i , con $i = 1, \dots, 7$

$$\max 2x_a + 2x_b + 6x_c + 10x_d + 10x_e + 2x_f + 8x_g$$

$$\text{s.a: } 2x_a + 7x_b + 1x_c + 3x_d + 9x_e + 1x_g \leq 7000$$

$$1x_a + 1x_b + 3x_c + 5x_d + 5x_e + 1x_f + 4x_g \leq 5000$$

$$3x_b + 1x_c + 2x_d + 4x_g \leq 5000$$

$$7x_a + 1x_b + 3x_c + 9x_d + 1x_f + 6x_g \leq 7000$$

$$2x_a + 2x_b + 4x_d + 9x_e + 1x_f + 2x_g \leq 2000$$

$$x_i \geq 0$$

2. El desarrollo del problema se puede ver en el código de python. La utilidad es de \$10.000, con 1296,3 litros de la cerveza C y 222,22 litros de la cerveza E.

3. **Dual simplex (método = 1):** La utilidad es de \$10.000, con 833,33 litros de la cerveza C y 500 litros de la cerveza D.

Barrier (método = 2): La utilidad es de \$10.000, con 1666,67 litros de la cerveza

Concurrent (método = 3): La utilidad es de \$10.000, con 833,33 litros de la cerveza C y 500 litros de la cerveza D.

Deterministic concurrent (método = 4): La utilidad es de \$10.000, con 833,33 litros de la cerveza C y 500 litros de la cerveza D.

Deterministic concurrent simplex (método = 5): La utilidad es de \$10.000, con 833,33 litros de la cerveza C y 500 litros de la cerveza D.

4. Lo preliminarmente destacable de todos los resultados entre algoritmos es en que todos ellos presentan la misma utilidad obtenida, y además de que cuatro de ellos entregan el mismo resultado, conforme a la cantidad de cerveza a producir. Ninguna presenta una ventaja respecto a la otra, pero demuestra que la cerveza C es la más propensa a ser seleccionada

como la indicada a producir.

El método “barrera” entrega el resultado con mayor cantidad de cervezas a producir, en comparación a los otros algoritmos. Conforme a los métodos, los optimizadores “concurrent” corren múltiples métodos simultáneamente, donde entrega el primer resultado realizado. Las opciones deterministas (método = 4 y método = 5) dan siempre el mismo resultado, mientras que el método = 3 es a menudo más rápido pero puede producir diferentes bases óptimas cuando se ejecuta varias veces. Estas reglas pueden verse reflejadas a través de los resultados obtenidos, donde los resultados de los tres métodos “concurrent” son el mismo.

funte: <https://www.gurobi.com/documentation/9.0/refman/method.html>