

MEMORIA

Para este trabajo se ha planteado un problema de transportes, siguiendo la propuesta del profesor.

El problema consta de una empresa cuyo objetivo es colocar un almacén de forma que minimice los costes, para ello tenemos como datos: 5 clientes, los costes (coste de transporte e impuestos) y una demanda. El problema consiste en hallar la localización óptima para esta empresa.

El trabajo sigue la línea desarrollada en las tutorías, de forma que el primer problema es el planteamiento inicial y luego se van añadiendo nuevos ingredientes.

DATOS

CLIENTES

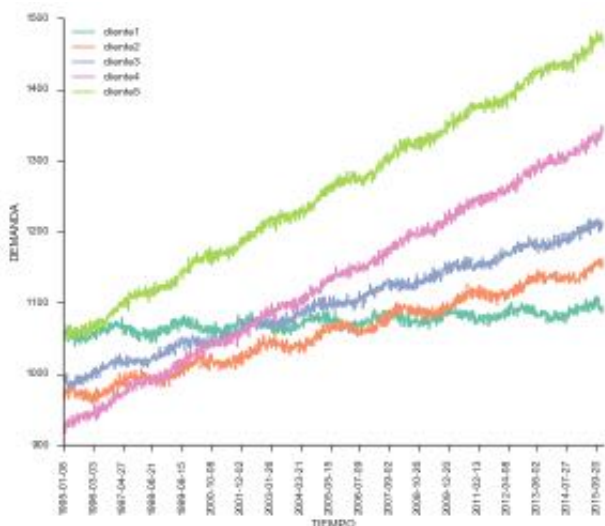
Los datos han sido generados aleatoriamente siguiendo un esquema de serie temporal.

$$y_t = \delta_0 + \delta_1 \cdot t + u_t$$

Donde:

- y_t : Ventas para cada período t .
- δ_0 : Ventas independientes o autónomas. (Stock de Seguridad pe.)
- δ_1 : Pendiente la recta.
- u_t : Ruido $\sim N(0, 5)$

En este caso para darle más realismo, $\delta_0 = \delta_a + \delta_b \cdot \sin(x)$, de esta forma introducimos un posible componente estacional, que depende de cuán grande sea δ_b .



Se ha procurado que cada cliente tenga una pendiente y un valor inicial distinto. La forma de la serie temporal sigue una función seno o coseno aleatoriamente.

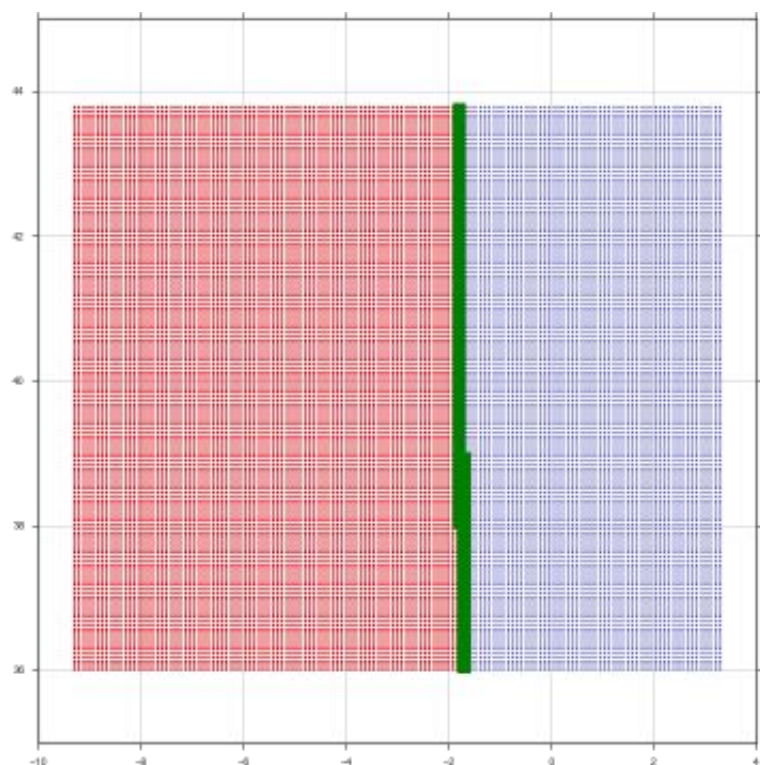
COSTES TRANSPORTE

Los costes de transporte por kilómetro han sido obtenidos del informe proporcionado por el *Observatorio de Costes del Transporte de Mercancías por Carretera (Enero 2015)*, para *VEHÍCULO ARTICULADO DE CARGA GENERAL*^[1]

MAPA

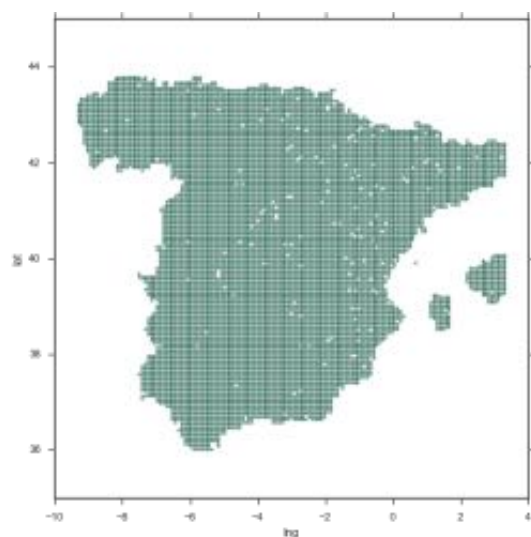
Para poder trabajar con mapas, hemos tenido que hacer un muestreo del mapa de España y obtener las coordenadas aproximadas de los límites de las comunidades autónomas.

Este muestreo consiste en veintemil puntos, dentro del mínimo rectángulo que pueda dibujarse sobre el mapa de España. Para esto se coje el valor con latitud mayor y menor, luego con longitud mayor y menor, y a continuación se generan valores idénticamente distribuidos entre dichos valores. De forma que el resultado es una “rejilla” de este tipo:



Luego con la API de google maps se obtiene información país, comunidad autónoma, provincia de cada punto, y se guarda para su posterior uso. Al final, el nivel de región usado para el trabajo ha sido por Comunidades Autónomas, ya que la API de googlemaps no proporciona información completa para las provincias.

El resultado ha sido el siguiente:



Con este muestreo, que no es más que una discretización del territorio, se soluciona el problema de incertidumbre en el cálculo de

la optimización, al poder asignar a cada punto del mapa de España una única comunidad autónoma a la que poder asignarle características además de crear un “puente” entre las comunidades (cualitativa) y las coordenadas (cuantitativa, variable de decisión) y poder tratar matemáticamente. Hay una ventaja añadida, y es que permite proporcionar un resultado instantáneo sin necesidad de acudir a googlemaps para conocer la situación geográfica del resultado.

También fue necesario realizar una homogeneización de los datos, ya que los nombres de las comunidades autónomas proporcionados por la API de googlemaps estaban en distintos idiomas.

Es importante notar que hay puntos en los que la API no ha devuelto ningún resultado, por eso existen puntos vacíos, sin embargo esto no es un problema ya que el número es muy reducido y el error que pueden provocar es mínimo.

IMPUESTOS

Los impuestos han sido generados aleatoriamente siguiendo una distribución uniforme(0,1).

OPTIMIZACIÓN

En el primer problema, se plantea una optimización lineal de forma que la función objetivo sea minimizar las distancias. En esta optimización se usa el algoritmo del simplex ya que la función objetivo es continua y cóncava.

En el segundo problema se añade la variable impuesto que depende de la comunidad autónoma en la que nos encontremos. Es importante añadir un impuesto infinito para que la solución se encuentre en España, esta restricción se ha incluido añadiendo un condicional, dado que la distancia media entre los puntos es de 0.01 (aproximadamente 7 km) si se supera dicha distancia el impuesto que se asocia es infinito. En esta optimización se hizo necesario utilizar un algoritmo de búsqueda global, se usó el algoritmo de fuerza bruta para hallar el óptimo, ya que el simplex daba una solución local y no global. Esto se debe a que la función objetivo no es continua sino que tiene escalones provocados por los impuestos.

Para la optimización multicriterio, se plantea un problema de Programación por metas MINIMAX o de Tchebychev. Una meta es minimizar el pago de impuestos y la otra meta tiene por objetivo minimizar los costes de transporte. A ambas metas se le asigna un valor superior al óptimo global, esperando que tener en las variables por defecto valores positivos, como al final sucede.

SOFTWARE

Para casi la totalidad del trabajo, se usa el lenguaje de programación Python, como interfaz de usuario Jupyter, para el cálculo matemático la librería SciPy. Se han implementado las ideas dadas en clase sobre Solver en Python.

1. <http://www.fomento.gob.es/NR/rdonlyres/75019EB9-D1D4-48DD-B58C-91FBD81E8E3B/129718/ObservatorioCostesMercanciasenero2015.pdf> ➔