**ENTREGA 2 – Modelado, optimización y simulación**

**Definición del problema:**

A la hora de viajar, hay cientos de miles de destinos y un abanico casi interminable de posibles rutas y calendarios por seguir. En la mayoría de los casos es necesario ajustar el tiempo y el presupuesto invertido en los viajes. Por esta razón, dada una ventana de tiempo y un conjunto de ciudades que se quiera visitar, buscamos generar un plan de viajes que maximice los lugares visitados y minimice el presupuesto empleado.

En vista de mejorar la experiencia de viaje, se pueden especificar el mínimo y el máximo de días que se puede estar en cada ciudad, con el fin de ajustarse a diferentes planes de viaje.

Para abordar este problema, usaremos datos de costo de transporte entre las ciudades, el costo de vida por día promedio (para un turista) en cada ciudad, y los puntajes y cantidad de reviews de los sitios turísticos de cada ciudad.

Costos (y tiempos) aproximados entre ciudades: <https://www.rome2rio.com/es/map/Bogot%C3%A1/Santa-Marta>[[1]](#footnote-1)

Costo de vida promedio: <https://www.numbeo.com/cost-of-living/>

Puntajes y cantidad de reviews: <https://www.google.com.co/destination/map/topsights?q=Bogot%C3%A1&sa=X&site=search&output=search&dest_mid=%2Fm%2F01dzyc&dest_mid=%2Fm%2F01dzyc&tcfs> [[2]](#footnote-2)

**Construcción del modelo:**

Parámetros:

* mind; Mínimo número de días.
* maxd: Máximo número de días.
* s: Punto de inicio.
* d: Número de días.
* CT: Matriz de costos para ir de una ciudad a otra. Es el mínimo costo entre los medios de transporte evaluados.
* CV: Arreglo de costo de vida promedio diario por cada ciudad.
* n: Número de ciudades.
* : Prioridad de maximizar el número de días.
* : Prioridad de minimizar el presupuesto.
* S: Matriz de puntajes que existen para una ciudad i con un punto de interés j.
* R: Matriz de cantidad de reviews que existen para una ciudad i con un punto de interés j.

1. Variable de decisión:

Que representa en qué ciudad *j* debe quedarse el viajero en el día *i*.

1. Función objetivo: Se plantea un modelo multi-objetivo usando un método de sumatoria ponderada entre las dos funciones propuestas: cantidad de días y presupuesto gastado (divido en costo de transporte y costo de vida).

Donde:

Esta fórmula asigna un puntaje a un destino l proporcional a su rating agregado (promedio) y al número de personas que opinaron sobre el destino. La fórmula ‘normaliza’ los puntajes dividiéndolos por 5 y busca penalizar a los destinos con mayor cantidad de reseñas al aplicarle la raíz cuadrada. Esto, porque puede haber lugares no tan populares que deberían ser considerados dentro de un plan de viajes.

1. Restricciones:

La estructura de costos para ir de un lugar a otro se representa en números reales. No se contemplan que el costo sea menor a 0 para ninguna de estas.

La estructura de costos de vida promedio por día se representan en números reales. No se contemplan que el costo sea menor a 0 para ninguna de estas.

La estructura de puntajes se representa en números reales. No se contempla que el costo sea menor a 0 para ninguna de estas.

El puntaje de cada atracción (de cada ciudad) está entre 1 y 5

Donde l es el número predefinido de puntos de interés por ciudad.

Por cada ciudad, el vector de puntaje de las atracciones está ordenado descendentemente (los más altos primero):

Las estructuras de puntajes se representan en números reales. No se contemplan que el costo sea menor a 0 para ninguna de estas.

Donde l es el número de puntos de interés predefinidos por ciudad.

La sumatoria de las prioridades asignadas debe ser igual a 1:

Las prioridades son números racionales entre 0 y 1:

La persona no puede sobrepasar la cota mínima y máxima de días que se puso para cada una de las ciudades seleccionadas por el algoritmo durante el viaje:

La persona no puede viajar a dos ciudades el mismo día

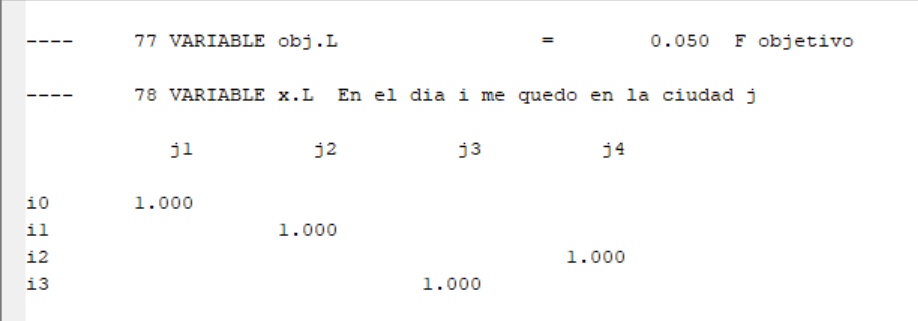
En el primer día (para el ejercicio se asume que k=0) la persona tiene que viajar a algún lado.

**Definición de escenarios:**

1. Escenarios base 1 (Funcionamiento básico)
   1. P2=0
   2. Mind=1 y Maxd=1
   3. d=4
   4. n=4
   5. Resto de parámetros del modelo son aleatorios

RESULTADO ESPERADO: Se escogen las 4 ciudades con 1 día por cada una independiente del orden.

RESULTADO OBTENIDO:



De lo anterior la ruta generada sería:

- Ciudad 1 a Ciudad 2.

- Ciudad 2 a Ciudad 4.

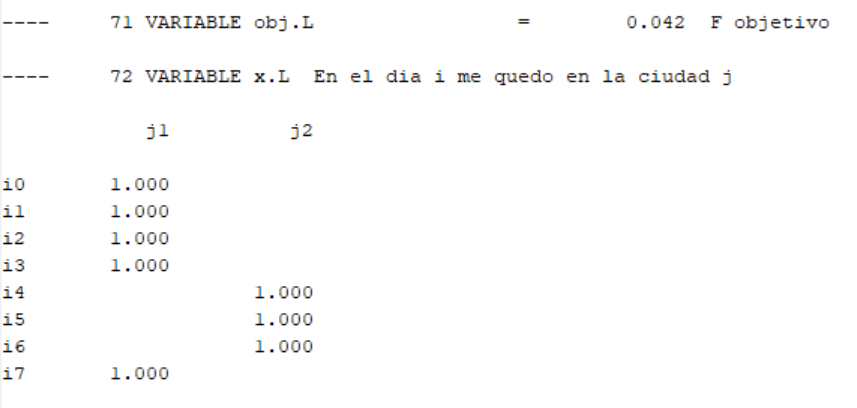
- Ciudad 4 a Ciudad 3.

Lo cual cumple con la estimación que habíamos generado antes.

1. Escenario base 2 (Número de días):
   1. n=2
   2. mind=3
   3. maxd=5
   4. d=8
   5. s=1
   6. Puntaje(1)=10
   7. Puntaje(2)=1
   8. p1=1
   9. p2=0
   10. Los otros parámetros son aleatorios

RESULTADO ESPERADO: Se queda 5 días en la primera ciudad y 3 en la otra.

RESULTADO OBTENIDO:

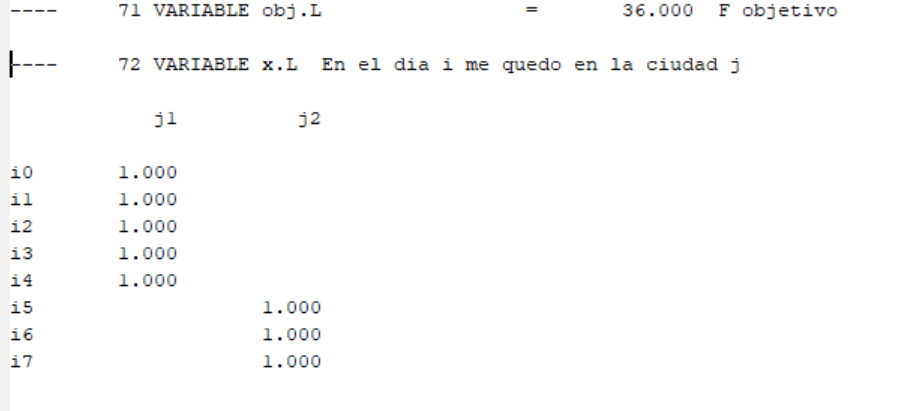


De la anterior imagen se ve que se queda 4 días en la ciudad 1, se desplaza 3 días a la ciudad 2, y se queda el último día en la ciudad 1. Esto cumple con las expectativas que se tienen del caso base.

1. Escenario base 3 (Costo de vida promedio):
   1. n=2
   2. mind=3
   3. maxd=5
   4. d=8
   5. s=1
   6. Los valores de la matriz de costos son iguales
   7. p1=0
   8. p2=1
   9. Los otros parámetros son aleatorios

RESULTADO ESPERADO: Se queda 5 días en la primera ciudad y 3 en la otra.

RESULTADO OBTENIDO:



Se queda 5 días en la primera ciudad y luego tres en la segunda, justo como se esperaba.

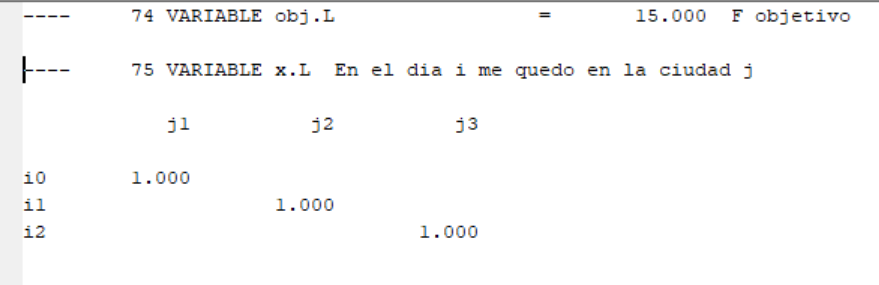
1. Escenario base 4 (Costo de transporte):
   1. n=3
   2. mind=|
   3. maxd=1
   4. d=3
   5. s=1
   6. p2=1
   7. p1=0

|  |  |  |
| --- | --- | --- |
| Infinito | 10 | 10000000 |
| 10 | Infinito | 2 |
| 999999 | 12 | Infinito |

* 1. Costo de vida promedio es igual para todas las ciudades
  2. Los otros parámetros del modelo son aleatorios

RESULTADO ESPERADO: Va de la ciudad 1 a la 2 y luego a la 3.

RESULTADO OBTENIDO:

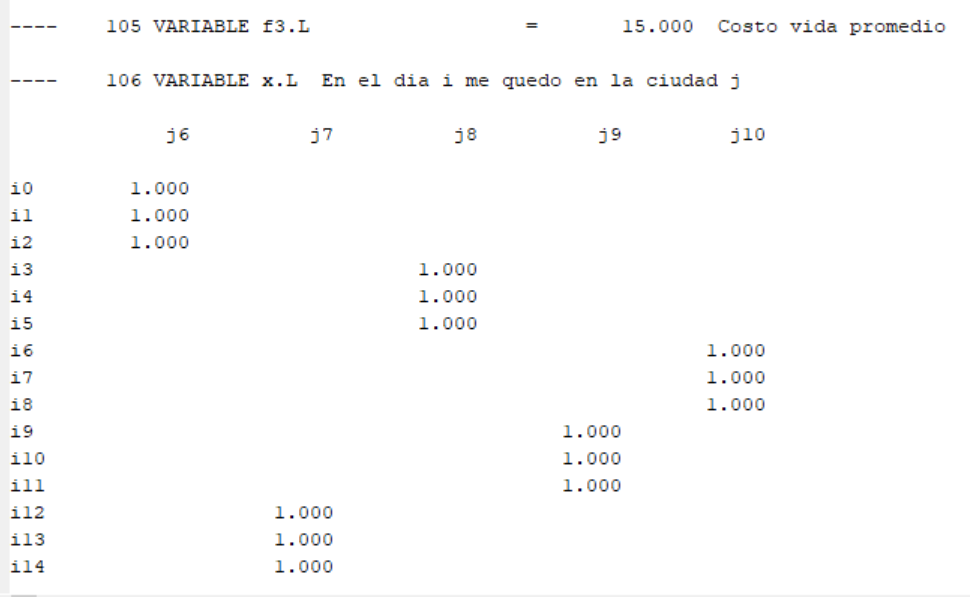


De acuerdo con la anterior imagen, como fue de esperarse se desplaza de la ciudad 1 a la 2, y de la ciudad 2 a la 3.

1. Escenario intermedio:
   1. n=10
   2. 5 ciudades tienen un costo de vida alto y puntajes bajos (Ciudad 1 a 5)
   3. 5 ciudades tienen un costo de vida bajo y puntajes altos (Ciudad 6 a 10).
   4. Costos de transporte iguales.
   5. d=15
   6. maxd=3
   7. mind=1
   8. p1=0.5
   9. p2=0.5
   10. s: Una ciudad de las baratas (6 a 10).
   11. Los otros parámetros son aleatorios

RESULTADO ESPERADO: Irse por las ciudades más baratas.

RESULTADO OBTENIDO:



Como era de esperarse, solo se estuvo en las 5 ciudades más baratas.

DECISIONES DE DISEÑO

* Alelo: <Número de ciudad visitada en el día 1>-<Número de ciudad visitada en el día 2>-….<Número de ciudad visitada en el día n>
* Algoritmo: SPEA
* Distancia euclidiana con las dos funciones de la variable objetivo:
* Ponderación de las funciones: Para el caso de F1 se multiplica por este número para asegurar la reducción de las influencias.
  + F1:
  + F2:
* N es un número poblacional cualquiera que cumpla con las restricciones. Se puede ensayar con 10 y 100. N prima, es el 25% de esa población y también cumple con la condición de ser válido.
* F1 y F2 son para el cálculo de las distancias y para evaluar la dominancia de las soluciones. P1 y P2 no se utilizan para este algoritmo.

1. Costos de transporte: Se mandó la propuesta para solicitar acceso al API de Rome2Río que contiene costos de desplazamiento entre ciudades sobre diferentes medios de transporte. Sin embargo, la solicitud no ha sido respondida por lo que es necesario extraer los datos manualmente . [↑](#footnote-ref-1)
2. Puntajes y cantidad de reviews: Se buscó utilizar el API oficial de google places: <https://developers.google.com/places/web-service/details?hl=es> . Sin embargo, este no ofrece la cantidad total de reviews realizadas a cada lugar, por lo que no era de utilidad los datos del API. Por este motivo, se escogió extraer los datos manualmente. [↑](#footnote-ref-2)