

Analytics aplicado a residencias médicas: un enfoque basado en programación matemática

El problema

Entidades gubernamentales se encuentran analizando la posibilidad de modificar el sistema de asignación de residencias médicas actual para hacerlo a nivel nacional. Hoy en día, el procedimiento se basa en un sistema clásico de *orden de méritos*: se genera un ranking de residentes en base a distintas métricas y que luego los mismos eligen el hospital donde realizar su residencia en base a este orden. Sin embargo, el organismo busca cambiar este esquema por uno más balanceado, que incorpore también las preferencias de las instituciones (en este caso, los hospitales).

El objetivo es tomar como base el reconocido procedimiento *The National Resident Matching Program* (NRMP), o *The Match*, que ha recibido gran cantidad de reconocimiento. Sin embargo, sobre el mismo se realizarán algunas modificaciones particulares respecto a la forma de evaluar la calidad de una solución. Para decidir llevar adelante el proyecto, se realiza primero una etapa de pruebas de concepto donde distintas empresas de base tecnológica participarán proponiendo soluciones de gran valor agregado para resolver el problema. Se sospecha que aquellas que obtengan la mejor solución correrá con ventaja a la hora de adjudicar el proyecto en caso de decidir llevarlo adelante. Nosotros participaremos del mismo a través de nuestra consultora, *DM Analytics*.

El problema en cuestión es el siguiente. Se cuenta con n médicos residentes que deben ser asignados a m hospitales. Por simplicidad, en esta etapa se asume que cada hospital cuenta con p puestos disponibles, de forma tal que $n = m \times p$. Como etapa previa al problema, se han realizado rondas de entrevistas para guiar la selección. En base a las mismas, tanto los residentes como las instituciones han generado sus correspondientes listas de preferencias. En este sentido, cada hospital i , $1 \leq i \leq m$, posee una lista de k residentes L_i^h , ordenados de mayor a menor de acuerdo a su preferencia. De forma análoga, cada residente j , $1 \leq j \leq n$, entrega un ranking de l hospitales L_j^r , también ordenados de mayor a menor preferencia. Todas las vacantes de un hospital poseen la misma lista de preferencia y pueden considerarse *idénticas*. Para cuantificar una posible solución, se definen los siguientes índices de satisfacción individuales dada una solución:

- satisfacción del residente j si es asignado al hospital i desde la perspectiva del residente, w_{ji}^r , definido por

$$w_{ji}^r = \begin{cases} 2 \times (|L_j^r| - \text{pos}(i, L_j^r)) & \text{si } i \in L_j^r \\ -1 & \text{si } i \notin L_j^r \end{cases}$$

- satisfacción de una vacante hospital i mediante el residente j desde la perspectiva del hospital, w_{ij}^h , definido por

$$w_{ij}^h = \begin{cases} 2 \times (|L_i^h| - \text{pos}(j, L_i^h)) & \text{si } j \in L_i^h \\ -1 & \text{si } j \notin L_i^h \end{cases}$$

A modo de ejemplo, si un residente tiene la lista de hospitales $[1, 4, 0, 2, 3]$ y le toca el hospital 0, entonces $w_{ji}^r = 2 \times (5 - 2) = 6$. Si el mismo residente es enviado al hospital 3, entonces $w_{ji}^r = 2 \times (5 - 4) = 2$. De forma análoga puede calcularse la satisfacción de una vacante de un hospital.

Notar que tanto residentes como hospitales poseen listas del mismo tamaño, $|L_i^r| = l$ y $|L_j^h| = k$. Con el fin de obtener una métrica de satisfacción global, dada una solución al problema se busca maximizar la *satisfacción global agregada*. Ésta considera la suma total de la satisfacción de cada residente así como también la satisfacción total de cada hospital.

En base a contactos informales con especialistas del área, tenemos cierta información respecto a la estrategia a seguir por nuestro principal competidor en este proyecto. Su estrategia se basa en obtener una solución utilizando técnicas heurísticas, en particular una de las denominadas *greedy (codiciosas)*, buscando soluciones de buena calidad de manera rápida. Nuestro equipo de desarrollo nos provee una implementación de lo que, se estima, sería su enfoque para abordar el problema.

Nuestro objetivo es proveer una solución para el problema que sea competitiva y mejore los resultados de la competencia.

El contexto

Contamos con un desarrollo parcial para nuestro problema, incluyendo una clase `ResidentAssignmentInstance` que se encarga de leer la definición de una instancia del problema de un archivo de entrada. La misma contiene los siguientes atributos:

- `n (int)`: cantidad de residentes.
- `m (int)`: cantidad de hospitales.
- `k (int)`: tamaño lista de preferencia de hospital.
- `l (int)`: tamaño lista de preferencia de residente.
- `p (int)`: cantidad de vacantes por hospital.
- `penalty (double)`: penalidad α a aplicar para el cálculo de w_{ij}^r y w_{ji}^h . En la definición previa, $\alpha = 1.0$.
- `res (dict(int, list(int)))`: diccionario con clave id del residente (numerados de 0 a $n - 1$) y significado lista de l hospitales, ordenados de forma decreciente por preferencia.
- `hos (dict(int, list(int)))`: diccionario con clave id del hospital (numerados de 0 a $m - 1$) y significado lista de k residentes, ordenados de forma decreciente por preferencia.
- `var_idx`: estructura auxiliar que puede ser utilizada para almacenar índices de variables. El tipo queda sujeto a elección del grupo.

La clase provee dos funciones adicionales que, dada la información necesaria, devuelve las preferencias definidas anteriormente:

- `get_resident_pref(j,i)`: retorna la satisfacción de asignar el hospital i para el residente j , desde la perspectiva del residente (o -1 si $i \notin L_j^r$).

- `get_hospital_pref(i,j)`: retorna la satisfacción de asignar el residente j para el hospital i , desde la perspectiva del hospital (o -1 si $j \notin L_i^h$).

Para la evaluación del problema, se cuenta con 3 sets de instancias con distintas características y tamaños. Las mismas deben ser utilizadas para comparar la efectividad de nuestro enfoque y realizar una comparación extensiva con el método de la competencia. Las características de las instancias son:

- **small**: $n = 20, m = 5, l = 5, k = 5, p = 4$.
- **medium**: $n = 500, m = 50, l = 10, k = 10, p = 10$.
- **large**: $n = 1000, m = 100, l = 10, k = 10, p = 10$.

El trabajo

La resolución del trabajo consiste en las siguientes etapas, incluyendo completar el código del archivo `resident_assignment.py` completando y agregando aquellas funciones auxiliares que consideren necesarias. Se pide:

1. **(1.5 puntos) Formulación del modelo.** Proponer un modelo de programación lineal para el problema a abordar. Incluir la definición de las variables de decisión, la función objetivo y las restricciones de factibilidad. Justificar la elección del mismo y por qué lo consideran apropiado para el problema. Calcular la cantidad de variables de decisión y la cantidad de restricciones que tiene el problema para una instancia genérica.
2. **(2 puntos) Implementación.** Realizar una implementación del modelo propuesto utilizando CPLEX.
3. **(1.5 puntos) Experimentación.** Realizar experimentos sobre todas las instancias comparando el modelo propuesto respecto al de la solución de la competencia. En cada caso, medir la mejora porcentual obtenida. Sean z_0 y z_1 una solución (satisfacción global) de nuestro modelo y el de la competencia, respectivamente, definimos la mejora porcentual relativa como

$$\%gap = \frac{z_0 - z_1}{z_1}.$$

Este factor puede ser relevante a la hora de justificar la utilización de un método por sobre el otro. En caso de considerarlo conveniente, pueden agregar otras métricas complementarias (respecto al método o a las soluciones) para el análisis de los resultados.

4. **(2 puntos) Discusión y análisis de resultados.** Analizar los resultados obtenidos en función de la comparación entre ambos métodos. Incluir como parte de la discusión aquello que considere relevante en relación a las características del modelo propuesto para resolver el problema.
5. **(1 punto) Discusión extendida.** La solución obtenida puede ser analizada en función de otras métricas, por ejemplo:

- *Preferencia mutua*: residentes que fueron asignados a un hospital de su lista de preferencia, y que a su vez también pertenecen a la lista de preferencia de residentes del hospital.
- *Preferencia residente*: residentes que fueron asignados a un hospital de su lista de preferencias, pero **no** pertenecen a la lista de preferencia de residentes del hospital.
- *Preferencia hospital*: residentes que fueron asignados a un hospital que **no** está entre sus preferencias, pero si pertenece a la lista de preferencia del hospital.

Agregar métricas complementarias que permitan profundizar el análisis de los resultados.

6. **(1 puntos) Limitaciones y posibles extensiones.** Discutir respecto a limitaciones del modelo, la propuesta y el enfoque. Proponer al menos una variación al proceso con el objetivo de mejorarlo (ejemplo: otra función objetivo, incluir alguna información complementaria, analizar alguna extensión, etc).
7. **(1 puntos) Modelo alternativo.** Proponer un modelo alternativo que resuelva el problema mediante un problema de flujo de costo mínimo con una única fuente y un único destino. Discutir las ventajas y desventajas respecto al modelo original propuesto.

Modalidad de entrega

Se pide presentar el modelo y la experimentación en un informe de máximo 10 páginas que contenga:

- introducción al problema y la decisión,
- descripción del modelo propuesto, según corresponda,
- consideraciones generales respecto a la implementación del modelo, incluyendo dificultades que hayan encontrado,
- resumen de resultados obtenidos en la experimentación,
- conclusiones, posibles mejoras y observaciones adicionales que consideren pertinentes.

Junto con el informe debe entregarse el código con la implementación del modelo. El mismo debe ser entendible, incluyendo comentarios que faciliten su corrección y ejecución.

Fechas de entrega

Formato Electrónico: **domingo 23 de Julio de 2023, 23:59 hs**, enviando el trabajo (informe + código) a las dirección `dm_mim@utdt.edu`. El subject del email debe comenzar con el texto **Trabajo Práctico 2 - Modelos de Decisión** seguido de la lista de apellidos de los integrantes del grupo. Todos los integrantes del grupo deben estar copiados en el email.

Importante: El horario es estricto. Los correos recibidos después de la hora indicada serán considerados re-entrega.