Trabajo Práctico: *Balanceando la felicidad*

Modelos y Técnicas de Programación Lineal Entera.

Universidad Torcuato Di Tella
Master in Management + Analytics.

Basigalup, Paula
Dalla Via, Josefina

11 de diciembre de 2018

Resumen

En el siguiente informe se presenta la forma en que fue abordado y resuelto el problema de optimización combinatoria planteado en el trabajo práctico de la materia.

En primer lugar, se realiza una introducción al problema de decisión en cuestión y se describe la forma en que el mismo fue modelado mediante técnicas de programación lineal entera mixta. Luego se detallan algunas cuestiones relativas a la implementación técnica y se presenta la solución obtenida para la instancia dada. Por último, se plantean alternativas y extensiones al modelo inicial y se analizan los resultados obtenidos de la experimentación con cada una de ellas. Se finaliza con las conclusiones del trabajo reazalido.

Junto con el informe se adjunta el código fuente utilizado para la resolución del trabajo en el archivo: `balanced_assignment.zip`.

1. Introducción al problema

En el presente informe se aborda el problema de asignación que enfrenta una compañía de charters empresariales al tener que decidir cómo asignar sus unidades a los diferentes servicios que se deben cubrir en un día dado.

La empresa le paga a los choferes de sus unidades en función de los kms. recorridos en servicio y, en este sentido, busca una asignación que, además de factible, sea justa en términos de cómo se distribuyen los kms. asignados entre los choferes, de manera de maximizar la felicidad de los mismos.

Además existen condiciones deseables de la asignación pero no necesarias, como por ejemplo, que los choferes no tengan que esperar demasiado tiempo entre sus viajes o que no tengan que recorrer demasiados kilométros fuera de servicio, que son modeladas mediante restricciones *soft*.

El problema es resuelto mediante técnicas de programación lineal entera mixta. Se utiliza el software `Cplex` para la implementación del modelo. Se resuelve la instancia en cuestión mediante la aplicación de los algoritmos: `Branch & Bound` y `Brand & Cut`.

2. Descripción del modelo

En esta sección se formula el modelo matemático que describe el problema de optimización que enfrenta la compañía, de manera generalizada, para después traducirlo a un problema de programación lineal entera mixta.

Formulación matemática

En forma generalizada, la formulación matemática del problema que enfrenta la compañía resulta en:

Dados los parámetros:

- U la cantidad de unidades (choferes) con que cuenta la compañía
- M la cantidad máxima de viajes por unidad
- N la cantidad de servicios a cubrir en un día dado
- V la velocidad promedio de circulación

El objetivo de la compañía es:

$$Max\ W(x_{111}, x_{112}....., x_{nuk})$$

sujeto a:

$$\sum_j \sum_k x_{ijk} = 1 \quad \text{para} \quad i = 1, ...n$$

$$\sum_i x_{ijk} \leq 1 \quad \text{para} \quad j = 1, ...u \quad , \quad k = 1, ...m$$

$$a_{i_1 i_2 jk}[(start_{i_2} - end_{i_1})V - d_{end_{i_1}, start_{i_2}}] \geq 0 \quad \text{para} \quad i_1 \neq i_2 \quad , \quad j = 1, ..., u \quad , \quad k = 1, ..., m-1$$

donde:

- x_{ijk} es una variable binaria que toma valor 1 si el servicio i es asignado a la unidad j en su servicio k, 0 en caso contrario.
- $a_{i_1 i_2 jk} = \begin{cases} 1 & \text{si } x_{i_1 jk} = 1 \cap x_{i_2 jk+1} = 1 \\ 0 & \text{en otro caso} \end{cases}$
- $start_i$ representa el tiempo de salida del servicio i.
- end_i representa el tiempo de llegada del servicio i.
- $d_{end_{i_1}, start_{i_2}}$ mide la distancia entre la zona de llegada del servicio i_1 y la zona de salida del i_2.
- $W(...)$ es una función de bienestar de los choferes que depende de la asignación.

Modelo de Programación Lineal

El problema formalizado en el inciso anterior se puede modelar como un problema de programación lineal entera mixta con las siguientes características:

Variables

- Variables de decisión

$$x_{ijk} \quad \forall \quad i, j, k$$

Variables binarias que indican si el servicio i fue asignado a la unidad j en su k-ésimo viaje cuando toman valor 1.

Cantidad de variables: $N * U * M$

- Variables auxiliares

$$y_j \quad \forall \quad j$$

Variables continuas que sirven para representar la diferencia absoluta de la asignación del chofer j con respecto a la asignación más justa (es decir, aquella que asigna a todos los choferes la misma cantidad de kms).

Cantidad de variables: U

Función objetivo

$$Min \sum_j y_j$$

Esta función viene a representar la suma de las diferencias absolutas de las asignaciones de los choferes con respecto a la media. (Notar que esto depende fuertemente del sentido de la optimización y de las restricciones a establecer sobre las variables y_j).

Se busca minimizar dicha función para encontrar la asignación más justa posible en términos de la equidad de kms asignados a cada chofer.

Restricciones

- Restricción de una unidad asignada por servicio

$$\sum_j \sum_k x_{ijk} = 1 \quad \forall \quad i$$

Esta restricción garantiza que todos los servicios sean asignados exactamente una vez.

Cantidad de restricciones: N

- Restricción de máximo un servicio por viaje

$$\sum_i \sum_k x_{ijk} \leq 1 \quad \forall \quad j, k$$

Esta restricción garantiza que a cada chofer no se le asigne más de un servicio en cada uno de sus viajes. Podría no asignársele ninguno.

Cantidad de restricciones: $U * M$

- Restricción de asignación consecutiva de viajes a un chofer

$$\sum_i x_{ijk} - \sum_i x_{ijk+1} \geq 0 \quad \forall \quad j, k, \ / \ k < M$$

Esta restricción exige que un servicio pueda asignársele a un chofer en un viaje dado si y solo si en el viaje anterior ya se le asignó algún servicio.

Cantidad de restricciones: $U * (M - 1)$

- Restricción de compatibilidad entre viajes asignados a la misma unidad

$$x_{i_1jk} + x_{i_2jk+1} \leq 1 \quad \forall \quad j, k, i_1, i_2 \ / \ k < M, \ holgura(i_1, i_2) < 0$$

donde:
$holgura(i_1, i_2) = (start_{i_2} - end_{i_1}) * V - d_{end_{i_1}, start_{i_2}}$

Esta restricción garantiza que, dados dos servicios i_1 i_2 inviables de ser servidos consecutivamente por el mismo chofer (tomando la velocidad promedio de circulación como dada), los mismos no sean asignados de esta manera.

Cantidad de restricciones: En este caso es variable y dependerá de la instancia en cuestión. Sea I la cantidad de pares de servicios i_1 i_2 con $holgura(i_1, i_2) < 0$ que existan en la instancia, la cantidad de restricciones viene dada por: $U * (M - 1) * I$

En el peor de los casos, en que todos los servicios sean *incompatibles* entre sí, la cantidad de restricciones asciende a: $U * (M - 1) * N * (N - 1)$

- Restricciones auxiliares

  Restricciones de y "por arriba".

$$y_j - \sum_i \sum_k x_{ijk} * r_i \quad \geq \quad \frac{-\sum_i r_i}{U} \quad \forall \quad j$$

  Restricciones de y "por abajo".

$$y_j + \sum_i \sum_k x_{ijk} * r_i \quad \geq \quad \frac{\sum_i r_i}{U} \quad \forall \quad j$$

  donde r_i representa la cantidad de kms de recorrido del servicio i

Estas restricciones junto con el sentido de la optimización (minimización) son las que llevan a cada y_j a tomar el valor absoluto de la diferencia entre el total de kms de la asignación otorgada a el chofer j y la asignación promedio.

Cuando $\sum_i \sum_k x_{ijk} * r_i - \frac{\sum_i r_i}{U} \geq 0$, y_j tiene que ser al menos tan grande como esa diferencia según la restricción "por arriba". Dado que estamos minimizando la suma de las y_j, el algoritmo elegirá que valgan exactamente esa diferencia y no más.

Cuando la diferencia es negativa, la restricción "por abajo" fuerza a que $-y_j$ sea menor o igual que esa diferencia. Dado que estamos minimizando, el algoritmo elegirá que valgan exactamente menos la diferencia.

Cantidad de restricciones: $2 * U$

3. Resolución de la instancia

En esta sección se describe la implementación en `Cplex` del modelo desarrollado en el inciso anterior y la solución obtenida para la instancia input del trabajo práctico.

La instancia en cuestión consta de un total de 59 servicios ($N = 59$). Se eligieron los siguientes valores de parámetros para esta formulación:

- $U = 15$ unidades en servicio.
- $M = 7$ máximo de viajes por unidad.
- $V = 50$ km/h velocidad promedio de circulación.

El modelo construido en `Cplex` cuenta con un total de 188,654 restricciones y 6,210 variables de las cuales 6,195 son variables binarias.

Se resolvió esta instancia utilizando el algoritmo `Branch & Cut` y permitiendo la heurística inicial de `Cplex`, en un tiempo total de cómputo de 6.52 hs. Se aplicaron 380 cortes de distinto tipo y se exploraron 496,901 nodos, siguiendo el criterio de la mejor cota.

El gráfico de barras a continuación ilustra que la solución obtenida logra producir una asignación equitativa en términos de la distancia en servicio asignada a las diferentes unidades, tal como era el objetivo. En particular, se observa que la distancia asignada a cada unidad oscila entre 152 y 156 kms mientras que la asignación "ideal" sería de 152.8 kms por chofer. La suma de los desvíos de las asignaciones con respecto a este promedio asciende a 13.87 kms que es exactamente el valor del funcional en el óptimo obtenido por `Cplex`.

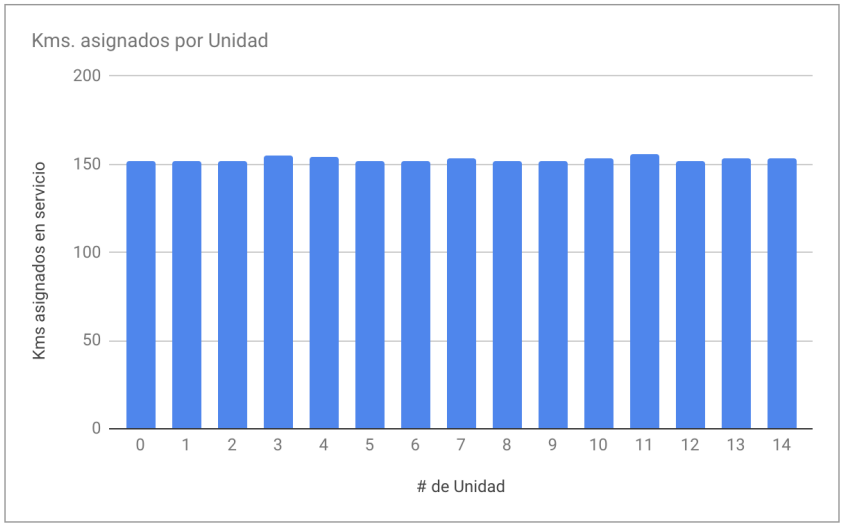

Figura 1: Distribución de los kms en servicio de las diferentes unidades bajo la asignación óptima.

4. Extensiones

En esta sección se desarollan algunas alternativas o extensiones sobre el modelo planteado inicialmente. Por un lado, se plantea una función objetivo alternativa. Por el otro, se formulan algunas restricciones *soft* que podrían ser incluidas en el modelo a fines de controlar por ciertas características deseables de la solución.

Función objetivo alternativa

Como alternativa a la minimización del módulo, el problema de balancear la asignación total de kms entre las unidades puede modelarse mediante la maximización de la mínima asignación de kms a un chofer.

En términos de programación lineal, esto implica crear las siguientes variables auxiliares y restricciones para poder linealizar una función objetivo que es no lineal:

- Variables auxiliares

  $$y$$

  Variable única que tomará el valor correspondiente a la mínima cantidad de kms asignados a las U unidades. Es decir, y representará $Min\left\{\sum_{i\ k} x_{i1k} * r_i, ..., \sum_{i\ k} x_{iuk} * r_i\right\}$

  Cantidad de variables: 1

- Restricciones auxiliares

  $$\sum_i \sum_k x_{ijk} * r_i - y \quad \geq \quad 0 \quad \forall \quad j$$

  Estas restricciones junto con el sentido de la optimización (maximización) son las que llevan a que y tome el valor de la mínima asignación total de kms realizada a un chofer.

  En particular, la restricción fuerza a que y sea, como mucho, tan grande como el total de kms asignados a cada chofer. En la práctica, de las j restricciones creadas, la única que va a estar limitando el problema es la correspondiente a la unidad cuya asignación sea la mínima entre las U unidades.

  En ese caso, dado que estamos maximizando y, el algoritmo buscará asignarle a esta variable un valor tan grande como sea posible. En consecuencia, y valdrá exactamente el mínimo, ya que es lo máximo que esta restricción le permitirá.

  Cantidad de restricciones: U

- Función objetivo

  $$Max\ y$$

  Se busca maximizar el mínimo de kms asignados a un chofer, de modo de alcanzar la asignación más justa posible en términos de equidad de distancia a recorrer.

Con respecto al modelo inicial, esta extensión implica cambiar únicamente las variables auxiliares y las restricciones que ayudan a definirlas (restricciones auxiliares), así como la función objetivo y el sentido de la optimización. Todas las demás restricciones y variables se mantienen.

Restricciones *soft*

Otra extensión al modelo inicial viene dada por la inclusión de restricciones *soft* para modelar características deseables, pero no necesarias, de la asignación óptima. En la práctica no se trata de restricciones propiamente dichas sino de términos de costo que se agregan a la función objetivo.

En particular, se formularon dos alternativas de restricciones *soft* para el modelo de minimización del módulo:

1. **Restrición *soft* sobre la distancia recorrida no remunerada**

   En la descripción del problema se plantea que las unidades cobran por la distancia recorrida **en servicio**. Por lo tanto, además de buscar una asignación que sea equitativa en término de los kms asignados a cada chofer, también es deseable minimizar el costo de traslado.
   Dicho costo, dado por la distancia entre el punto de llegada de uno de sus viajes k del chofer j y el punto de salida del viaje inmediato posterior $k+1$ del mismo chofer, corre a cuentas de la unidad en cuestión y, por lo tanto, atenta contra su "felicidad".

   A efectos prácticos, esto requiere la siguiente implementación:

   - Variables adicionales

     $$a_{i_1 i_2} \quad \forall \quad i_1, i_2 \ / \ holgura(i_1, i_2) \ > \ 0$$

     Variables binarias que indican si el servicio i_1 y el servicio i_2 fueron asignados de manera consecutiva (es decir, en los slots k y $k+1$) para alguna unidad j, cuando toman valor 1.

     Cantidad de variables: Nuevamente la cantidad de variables depende de la cantidad de pares de servicios i_1,i_2 con $holgura(i_1, i_2) \ > 0$. En el peor caso posible se trata de: $N * (N-1)$ variables.

   - Restricciones sobre las variables adicionales

     $$x_{i_1 jk} + x_{i_2 jk+1} - a_{i_1 i_2} \leq 1 \quad \forall \quad j, k, i_1, i_2 \ / \ k < M, \ holgura(i_1, i_2) \ > \ 0$$

     Estas restricciones obligan a $a_{i_1 i_2}$ a valer 1 cuando este par de servicios i_1,i_2 han sido asignados consecutivamente a alguna unidad.

     Notar que no hace falta agregar la restricción que obliga a estas variables adicionales a valer 0 en caso contrario. Puesto que dichas variables van a estar sumando en la función objetivo y el problema es de minimización, en la práctica, esta condición se cumple aunque no se explicite en la formulación.[1]

     Cantidad de restricciones: Nuevamente la cantidad de restricciones depende de la cantidad de pares de servicios i_1,i_2 con $holgura(i_1, i_2) \ > 0$. En el peor caso posible se trata de: $U * M * N * (N-1)$ restricciones.

   - Reformulación de la función objetivo

     $$Min \sum_j y_j + \sum_{i_1 i_2 / holgura > 0} w * a_{i_1 i_2} * d_{end_{i_1}, start_{i_2}}$$

[1] Este resultado fue comprobado de manera empírica.

Esta función objetivo agrega, a la función del modelo inicial, un término que corresponde a la sumatoria de las distancias que deben recorrer (y financiar) los choferes para poder cumplir con la asignación obtenida. Las mismas están ponderadas por un parámetro w que indica el peso que se le quiere dar a esta restricción *soft* en el modelo.

2. **Restrición *soft* sobre el tiempo ocioso entre viajes**

Otra característica deseable de la solución es que las unidades no tengan demasiado tiempo ocioso entre uno de sus viajes y el posterior. Notar que esta restricción es ligeramente distinta a la anterior. Si bien es cierto que, si dos servicios asignados consecutivamente están muy lejos el uno del otro necesariamente deberán tener tiempo ocioso (caso contrario la asignación sería "incompatible"), podría darse el caso en que a los choferes se les asignen, de manera consecutiva, servicios que estén cerca en el espacio físico pero lejanos en el tiempo. Si bien la espera no tiene un costo explícito para el chofer, existe un costo de oportunidad de tener una unidad frenada y por lo tanto vale la pena incluirlo a tráves de una restricción *soft*.

La formulación de esta restricción tiene implicancias idénticas a la anterior en cuánto a las variables adicionales y la restricciones para definir dichas variables. La diferencia viene dada por la función objetivo.

- Reformulación de la función objetivo

$$Min \sum_{j} y_j + \sum_{i_1 i_2 / holgura > 0} w * a_{i_1 i_2} * (end_{i_1} - start_{i_2})$$

Esta función objetivo agrega, a la función del modelo inicial, un término que corresponde a la sumatoria de los tiempos ociosos de los choferes para poder cumplir con la asignación obtenida. Nuevamente, están ponderadas por un parámetro w que indica el peso que se le quiere dar a esta restricción *soft* en el modelo.

5. Experimentación y análisis

En esta sección se presentan los resultados de la experimentación con las diferentes extensiones al modelo inicial planteadas en el inciso anterior. También se realiza un análisis de sensibilidad sobre los parámetros del modelo. Finalmente se experimenta con diferentes algoritmos de resolución.

En primer lugar, se definen algunas métricas relevantes utilizadas para analizar los resultados de la experimentación.

- `Equidad (en km)` - Objetivo
  Interesa entender qué tan equitativa es la solución en términos de los kms asignados a cada chofer. Se utilizó el *MAD* (*mean absolute deviation*) para comparar la calidad de las diferntes soluciones en esta dimensión.
- `Distancia ociosa promedio (en km)` - Característica deseable
  Es una medida de los kms recorridos fuera de servicio que refleja los costos de traslado de los choferes. Se calcula como la distancia promedio que tienen que recorrer las unidades para llegar del punto de llegada de un servicio al punto de salida del próximo. Es deseable que esta distancia sea la menor posible para incrementar la felicidad de los choferes.

- `Tiempo ocioso promedio (en minutos)` - Característica deseable
  Mide el tiempo ocioso promedio que tienen las unidades entre sus viajes. Se considera que este tiempo ocioso tiene un costo de oportunidad para las unidades y, por lo tanto, también se busca que esta métrica sea lo más baja posible.

A continuación se presentan los resultados de las diversas experimentaciones realizadas:

- **Implementación con Max Min**

  Se implementó el modelo con la función objetivo alternativa (Max Min) planteada en la sección anterior y se analizó su performance con respecto al modelo inicial (Min Mod).

Función objetivo	Max Min	Min Mod
Cantidad de variables	6,196	6,210
Cantidad de restricciones	188,639	188,654
Cantidad de nodos	794,064	496,901
Cantitad de cortes	2,825	380
Tiempo total (sec)	11,835.82	23,461.39
Status solución	GAP 1.24 %	Solución óptima
Tiempo ocioso promedio (min)	151.75	133.53
Distancia ociosa promedio (km)	31.52	28.08
MAD (km)	1.86	0.92

MaxMin vs MinMod ($U = 15$, $M = 7$, $V = 50$)

  Se observa que el modelo de maximización del mínimo es ciertamente más rápido en acercarse a la solución óptima que el modelo inicial. En particular, luego de 60 segundos de corrida ya llega a estar a 2.6 % del óptimo mientras que la formulación con Min Mod, en el mismo tiempo, está a cerca del 80 %.

  La diferencia en tiempo de cómputo podría debserse a la menor cantidad de variables y restricciones de MaxMin o al hecho de que el algoritmo de `Branch & Cut` logra aplicar una gran cantidad de cortes, reduciendo el espacio de búsqueda y mejorando la cota rápidamente.

  Por último, se verifica que la solución de Min Mod es superior en términos de equidad tal como era de esperar pues es la solución óptima. En particular, el MAD se reduce en un 49 %. Sin embargo, en términos de tiempo y distancia ociosa las diferencias son menores.

- **Implementación con restricciones *soft*** [2]

  Se implementaron los dos tipos de restricciones soft propuestas en la sección anterior. La tabla a continuación muestra los resultados de cada alternativa:

[2] Para esta experimentación se trabajó con una instancia de menor tamaño por facilidad de cómputo.

Función	Sin restricciones soft	Soft Distancia	Soft Tiempo
Cantidad de variables	392	446	446
Cantidad de restricciones	2,912	3,899	3,899
Cantidad de nodos	0	16,504	49,477
Cantitad de cortes	3	73	116
Tiempo total (sec)	0.17	7.73	24.59
Tiempo ocioso promedio(min)	68	68	41
Distancia ociosa promedio (km)	17	15	12
Equidad (km)	6.73	7.35	20.16

Función MinMod ($U = 7$, $M = 4$, $V = 50$, $W = 0{,}5$)

Las implementaciones de ambas restricciones *soft* tienen un impacto significativo en la complejidad y el tiempo de resolución del algoritmo tal como se ve reflejado en el tiempo de cómputo del modelo.

La restrición *soft* sobre la distancia recorrida no remunerada mejora la métrica de distancia ociosa promedio. Sin embargo, no se modifica el tiempo ocioso con respecto al modelo sin restricciones soft.

La restrición *soft* sobre el tiempo ocioso entre viajes, por su parte, mejora ambas métricas (tanto el tiempo como la distancia ociosa). Este resultado es esperable ya que la restricción *soft* de tiempo es más exigente que la restricción *soft* de distancia.

La equidad de la solución se ve impactada negativamente en ambos casos, reflejando que existe un *trade-off* entre la "justicia" de la asignación y su "eficiencia" (entendida como disminución de tiempos y distancias ociosas).

En términos generales, se puede decir que la introducción de estas restricciones soft incrementa muchísimo la complejidad del modelo y, además, sacrifica en buena parte la equidad de la solución. Para lidiar con esto último, una alternativa posible es disminuir el peso (w) que se le da a dichas restricciones en la función objetivo. Aún así, esto no soluciona la limitación de la poca escalabilidad del modelo.

Alternativamente, podrían incluirse restricciones *hard* a la formulación, que bloqueen la asignación consecutiva de ciertos servicios si estos conllevan una espera de más de cierta cantidad de minutos o un traslado superior a cierta cantidad de kms.

- **Sensibilidad a los parámetros U, K, V** [3]

En esta experimentación se evaluó la variación en la equidad de la asignación obtenida al modificar uno de los parámetros mantiendo todo lo demás constante. A continuación se presentan los resultados:

[3] Para esta experimentación se trabajó con una instancia de menor tamaño por facilidad de cómputo.

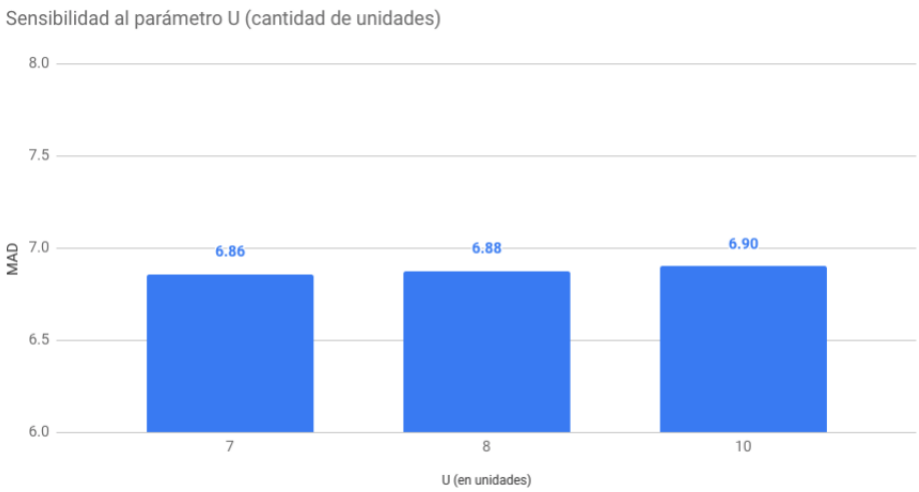

Figura 2: Función Min Mod ($M = 2$, $V = 50$)

La cantidad de unidades en servicio tiene un impacto negativo sobre la equidad de la asignación lograda. Esto es razonable dado que, a medida que aumenta la cantidad de unidades en servicio, si los servicios no se incrementan, ya no se le puede asignar más viajes a quienes les hayan tocado los viajes más cortos para compensar por la variabilidad de los trayectos.

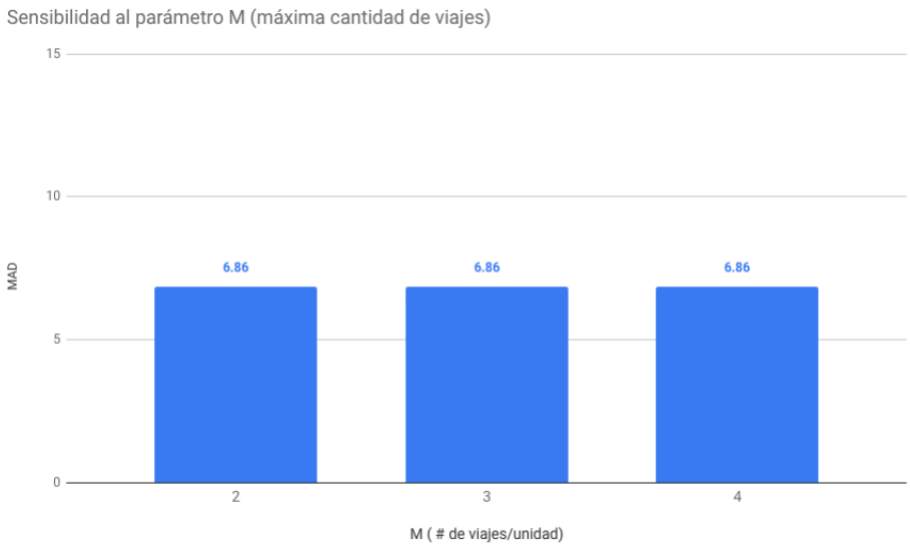

Figura 3: Función Min Mod ($U = 7$, $V = 50$)

La cantidad de viajes es un parámetro que, si bien resulta fundamental para determinar la factibilidad de una formulación, una vez que hay viajes y unidades suficientes como para cubrir todos los servicios pierde protagonismo. Este también es un resultado razonable considerando que el objetivo del modelo es generar la asignación más equitativa posible. Frente al mismo listado de servicios y el mismo número de unidades, permitir que una misma unidad pueda hacer más viajes, difícilmente contribuya a lograr una asignación más justa.

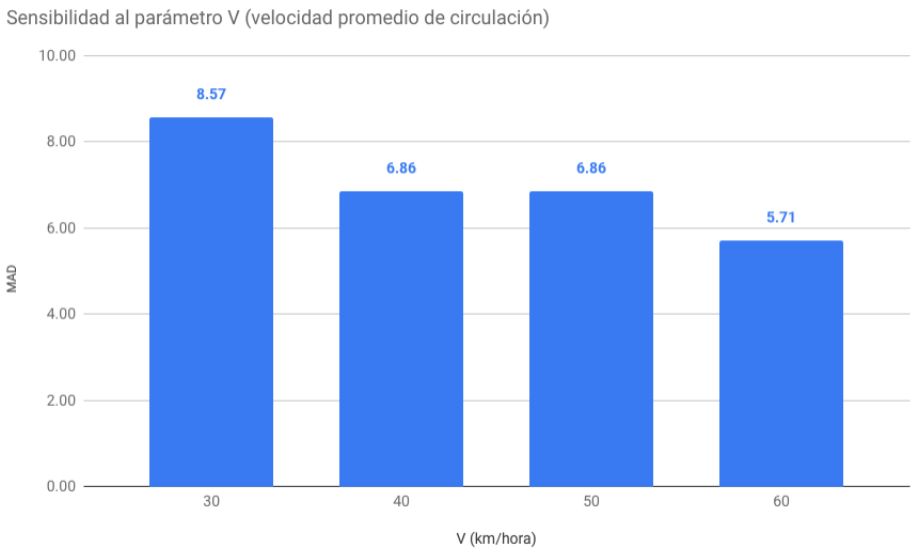

Figura 4: Función Min Mod ($U = 7$, $M = 3$)

La velocidad promedio de circulación resulta ser, de lo tres parámetros analizados, el que más impacta sobre la equidad de la asignación. En particular se observa que, a medida que disminuye la velocidad, la asignación óptima se vuelve cada vez menos equitativa en término del MAD. Lo que sucede es que este parámetro "ajusta" las restricciones de compatibilidad de asignaciones consecutivas. A medida que se necesita más tiempo para recorrer las distancias de traslados, más pares de servicios son los que se vuelven infactibles de ser asginados consecutivamente. Al perder flexibilidad, el modelo se ve obligado a sacrificar equidad.

- **Algoritmos de resolución**[4]

En este caso se resolvió el problema de minimización del módulo de las diferencias variando parámetros de los algoritmos de resolución tales como la aplicación de cortes, la utilización de la heurística y el criterio de selección de nodos. Se presentan los resultados a continuación:

En primer lugar, se observa que el algoritmo `Branch & Bound` con el criterio de búsqueda de la mejor cota es el que mejor performa para esta instancia del problema ya que, antes de los 30 minutos, logra llegar a la solución óptima (Columna 2).

El agregado de cortes, para esta instancia, no parece tener buenos resultados: en el tiempo que `Branch & Cut` insume en la generación de cortes y la resolución de relajaciones lineales de formulaciones más complejas, `Branch & Bound` llega a explorar más 100,000 nodos extra, lo que le permite acercarse más rápido a la solución óptima. Mientras tanto, `Branch & Cut` se queda a un 23.5 % de la mejor asignación (Columna 1).

Por otro lado, la heurísitca resulta un factor clave en la performance del algoritmo de resolución. En la única iteración en la que no fue incluida la solución obtenida queda muy lejos del óptimo (73 %) y, además, solo se llegan a encontrar 7 soluciones (Columna 3).

[4]Para esta experimentación se trabajó con la instancia original, configurando un tiempo máximo de corrida de 30 minutos.

Algoritmo	`Heuristic + Branch & Cut Best bound`	`Heuristic + Branch & Bound Best bound`	`No Heuristic + Branch & Cut Best bound`	`Heuristic + Branch & Cut FIFO`
Cantidad de nodos	190,184	336,117	309,914	195,117
Cantitad de cortes	334	0	270	338
Cantidad soluciones	26	21	7	28
Gap	23.53 %	0.00 %	72.92 %	31.58 %
Tiempo total (sec)	1807.19	1673.41	1800.06	1802.39
Valor del funcional	15	13	48	19

Función MinMod ($U = 15$, $M = 7$, $V = 50$)

Por último, al comparar las estrategias de selección de nodos (manteniendo al algoritmo `Branch & Cut` con heurística) , se observa que el criterio de búsqueda en profundidad (FIFO) llega a explorar algunos nodos más y encuentra más soluciones (Columna 4). Sin embargo, el criterio de la mejor cota, en el mismo tiempo, alcanza una asignación más justa.

6. Conclusiones

A continuación se presentan las conclusiones del presente informe.

En primer lugar, se planteó un modelo de programación lineal entera mixta para resolver el problema de asignación en cuestión que busca maximizar la equidad de los kms asignados a las diferentes unidades en servicio. El modelo toma como input los datos de una instancia, la cantidad de unidades contratadas (U), la cantidad de viajes máxima por unidad (M) y la velocidad promedio de circulación y devuelve una asignación óptima en tiempo razonable.

En segundo lugar, se desarollaron algunas extensiones y alternativas a este primer modelo. Por un lado, se presentó una variante de función objetivo que transforma el problema de minimización del módulo de las diferencias en un modelo de maximización de la mínima asignación. Por otro lado, se modelaron algunas restricciones *soft* sobre condiciones deseables de la solución tales como que las unidades no tengan demasiado tiempo ocioso entre sus viajes o que se minimice también, en cierto grado, la distancia de traslado.

Por último, se realizaron experimentaciones con dos instancias del problema en cuestión y se presentaron sus resultados. Del análisis se desprenden los siguientes hallazgos:

- Se comprobó que la implementación de restricciones *soft* dificulta muchísimo la formulación del problema y lo vuelve poco escalable para instancias más grandes, sugiriendo que quizás vale la pena limitar directamente la asignación consecutiva de ciertos servicios mediante restricciones *hard*.

- Se encontró que la velocidad promedio de circulación es determinante en la equidad de la asginación óptima. Una versión mejorada de este modelo pondría mayor énfasis y grado de dificultad en la incorporación de esta variable que en nuestra propuesta no es más que un parámetro exógeno.

- También se hicieron algunos hallazgos relacionados a cuestiones técnicas de los algoritmos existentes para resolver problemas de programación lineal entera. En particular, se comprobó empíricamente que `Branch & Bound` performa mejor que `Branch & Cut` para la instancia en cuestión, que la heurística

primal es un factor clave en ayudar a encontrar la solución óptima rápidamente y que abrir los nodos del árbol siguiendo el criterio de la mejor cota arroja mejores resultados que yendo en profundidad.

Referencias

[1] Castillo Ron, E.; Conejo Navarro, A.J. y P. Pedregal Tercero, (2002) *Formulación y resolución de modelos de programación matemática en Ingeniería y Ciencia*, La Mancha, Universidad De Castilla.

[2] IBM (2015),"IBM ILOG CPLEX Optimization Studio Documentation" en *IBM Knowledge Center*. [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.1

[3] Méndez Días, I.; Pousa, F. y P. Zabala. "Clase 1: Objetivos Generales. Optimización Combinatoria. Programación lineal. Programación lineal entera mixta. Modelado". *Modelos y Técnicas de Programación Lineal Entera*. Universidad Torcuato Di Tella, noviembre 2018.