Modelo con formulación MTZ

En el modelo con la formulación MTZ se utilizan las siguientes variables

X_{ij} = Bivalente. Indica 1 si se el tour va desde la ciudad i a la j, 0 sino.
U_i = Entera. Número de secuencia en la cual la ciudad i es visitada.

Luego tenemos las siguientes restricciones

- Exactamente una ciudad debe ser visitada despues de la ciudad i

  $\sum_{j=0;\, i!=j}^{n} X_{ij} = 1$ para todo i = 0,1, …., n

- Exactamente una ciudad debe ser visitada antes de la ciudad j

  $\sum_{i=0;\, i!=j}^{n} X_{ij} = 1$ para todo j = 0,1, …., n

- Formulación MTZ para eliminar subtours

  $U_i - U_j + n X_{ij} \leq (n - 1)$ para i,j = 1,2,....n ; i != j

- Por último tenemos nuestro funcional de minimización

  $Z = \sum_{i,j=0;\, i!=j}^{n} X_{ij}\, d_{ij}$ (MINIMIZAR)

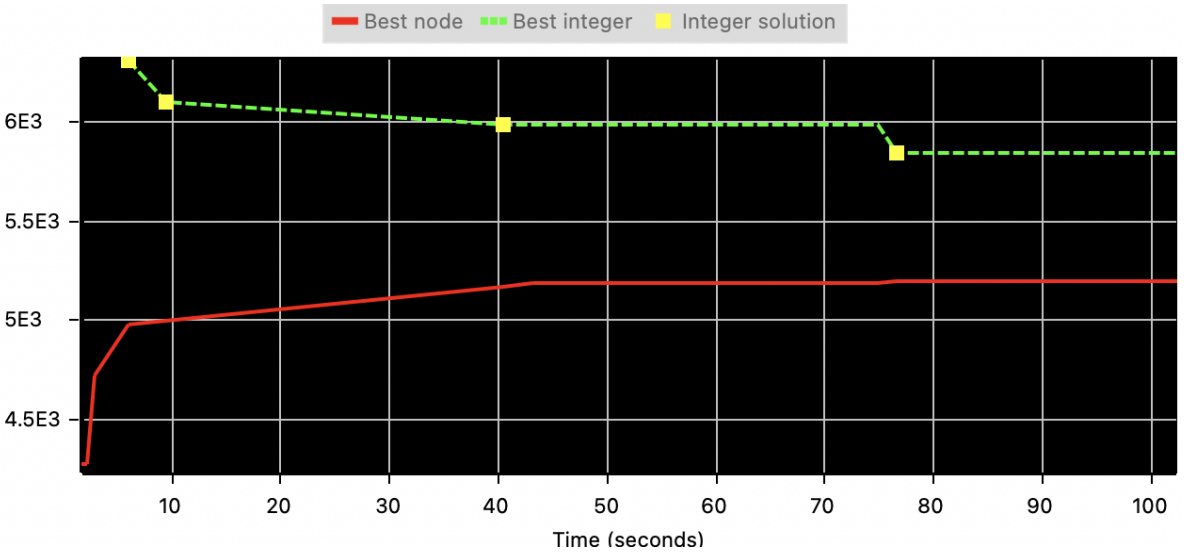
  Donde d_{ij} representa la distancia entre la ciudad i y la ciudad j

Cabe la aclaración de que en el código la ciudad 1 es el inicio y final, por lo tanto sería nuestro 0 en las restricciones del viajante estándar. Por esto mismo la formulación MTZ arranca en 2, ya que en esta restricción no se toma en cuenta a la ciudad de origen ya que el numero de secuencia en la cual es visitada es 0.

La ventaja que obtenemos con la formulación MTZ nos aseguramos que el resultado que se obtiene es el mejor posible, es decir 5249.622889689. Además no se requiere de post procesamiento para calcular subtours de ningún tipo ya que la misma restricción se encarga de esto. La desventaja es que este algoritmo puede tardar un poco más y depende mucho de la cantidad de ciudades o nodos que tengamos para procesarse en un tiempo razonable. El tiempo de corrida con 100 ciudades es de aproximadamente 5 minutos.

En el gráfico de Statistics podemos ver dos líneas graficadas en función del tiempo medido en segundos. La línea roja se refiere al valor del nodo (o rama) que tiene la mejor solución en cuanto al proceso de branch and cut que realiza el optimizador. En verde punteado podemos ver los valores que van tomando dichas soluciones y tambien vemos puntos

amarillos en los cuales se encuentra una solución entera. Como es una solución de minimización vemos que dicha línea verde debería ir decrementando hasta encontrar la mejor solución.



Modelo utilizando heurísticas

En el modelo con heurísticas vemos que se trabaja incrementalmente agregando cada vez más subtours en cada iteración, siendo restringidos por las restricciones mismas en cada una, hasta llegar a un subtour que tenga el mismo valor que la cantidad de ciudades recorridas. Intentaremos pasar las restricciones y funcional a notación matemática.

Hay que tener en cuenta que en este caso cuando armamos todas las combinaciones posibles de nodos (en el código se crea una tupla denominada Edge para esto) las creamos de forma ordered (también se puede pensar en que estas tuplas serán creadas siempre que i < j)

Esto quiere decir que si pensamos en una matriz, solo tendríamos elementos por arriba de la diagonal principal. Por ejemplo nunca tendríamos el elemento <2,1> o <1,1> en nuestro Edge, pero si el <1,2> porque en ese caso el i es menor a j.

En este caso esto es posible, porque estamos pensando las ciudades como nodos de un grafo, y el peso de cada arista es la distancia entre cada una. Por lo que la distancia entre 1 y 2 es idéntica tanto para <2,1> como para <1,1> es decir es un grafo no dirigido porque se conecta en ambos sentidos. Si fuera un grafo dirigido, esta optimización no sería posible. La ventaja de esto es que tenemos una estructura mucho más pequeña que la anterior con la formulación MTZ, por lo que el algoritmo en teoría debería ser mucho más rápido ya que tiene menos restricciones que verificar en cada iteración.

- Como primer punto vemos que el funcional es el mismo que el anterior, solo tenemos en cuenta que i siempre es menor a j

  $$Z = \sum_{i,j=0;\, i<j}^{n} X_{ij}\, d_{ij}$$ (MINIMIZAR)

  Donde d_{ij} representa la distancia entre la ciudad i y la ciudad j

- Luego tenemos una restricción en la cuál no dejamos que ningún nodo quede suelto, es decir todos los nodos están conectados con otros, tanto a derecha como a izquierda.

  $$\sum_{i,j=0;\, i<j}^{n} X_{ij} + \sum_{j,k=0;\, j<k}^{n} X_{jk} = 2$$ para todo j = 0,1,....,n

- Para eliminar subtours se creará en el post procesamiento una variable subtours la cual contiene cada subtour que se va encontrando incrementalmente. Se itera sobre este conjunto en cada iteración por lo que en cada una de las mismas, la cantidad de restricciones irá incrementando. La tupla de subtours tiene dos variables:
  - size: tamaño del subtour
  - subtour[i]: valor del nodo que le sigue al nodo i. Si no forma parte del subtour el valor será 0, sino un numero diferente de 0. Por ejemplo el subtour 1-2-1 el subtour[1] sería 2.

  $$\sum_{i=0;\, subtour[i]\ !=0}^{n} X_{min\,(i,\, subtour[i]),\, max\,(i,\, subtour[i])} \leq size - 1$$ para todo subtour

  Podemos ver que aquí tenemos los índices de la X forzados a cumplir la restricción de que en cada iteración, el valor de i sea menor que j, ya que del lado de i siempre tenemos el menor entre el y el siguiente, y del lado de j sería el mayor. Esto limitamos al tamaño del subtour a size - 1, por ejemplo si es un subtour de 2, lo forzamos que sea menor igual a 1, de esa forma eliminamos una conexión entre ambos nodos.

  En el post procesamiento tendremos un algoritmo que irá iterando buscando el subtour de menor tamaño incrementalmente hasta que hayamos visitado todos los nodos. Se realizaron 36 iteraciones restringiendo 35 subtours. La duración del mismo es de 30 segundos aproximadamente.

  Como vemos la mayor ventaja es la velocidad con respecto a la formulación MTZ, esto reside en la cantidad de restricciones que se van agregando en cada iteración, la cual a lo sumo serán 100 restricciones de conexión de los nodos (uno por cada nodo) y 36 restricciones de subtours agregadas iterativamente. En cambio en la formulación MTZ nosotros tenemos n * (n - 1) restricciones además de las 200 ya realizadas. Por lo que es una gran diferencia entre uno y otro algoritmo.
  La desventaja es que llegar a una buena solución o a la mejor posible, depende del tipo de heurística que realicemos sobre el grafo, además que el algoritmo de post

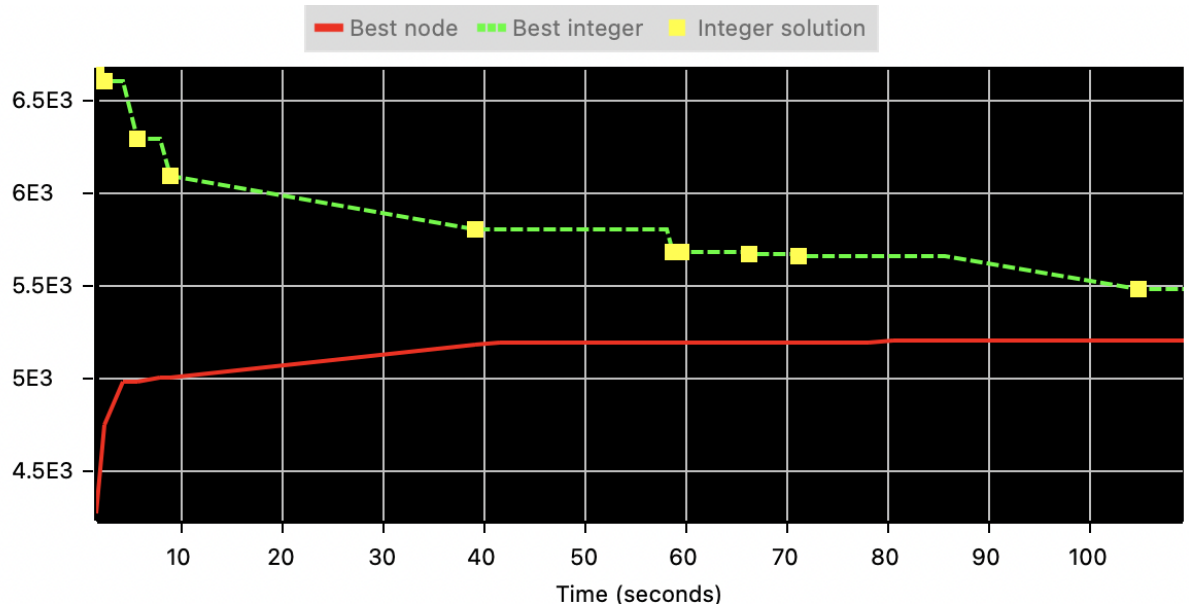
procesamiento tambien cumple un rol muy importante, por lo que agrega un poco de complejidad a la solución. También que la solución puede llegar a ser aproximada y no la mejor. Esto no pasa con la formulación MTZ la cuál funciona de forma segura y exacta, a pesar de que puede tardar más tiempo, y el mismo crece incrementalmente dependiendo de la cantidad de ciudades involucradas.

Modificación segunda entrega

Para esta segunda entrega se modificó el código utilizado en la segunda entrega para eliminar las restricciones de las demandas. Además se corrió el código con 100 ciudades para que sea válido a la hora de pasarlo a CPLEX. Se correrá ambos algoritmos utilizando la salida de nuestro algoritmo como solución inicial en ambos modelos. El resultado que se obtuvo en la distancia es de 6412.687, una diferencia de 1163, lo cual sería un error relativo del casi 20% con la solución óptima, una diferencia bastante notoria.

Formulación MTZ

Para la formulación MTZ cuando agregamos la solución inicial, no se cambió el tiempo de corrida del mismo, sigue siendo de aproximadamente 5 minutos. Tampoco se ha cambiado el resultado del objetivo y el orden de las ciudades recorridas. Esto se debe a que con el método con MTZ siempre se llegará al mismo resultado haya solución inicial o no.



La diferencia que encontramos entre este gráfico y el anterior es que en este caso las líneas del mejor nodo encontrado y de los valores que va tomando la solución se empiezan a acercar mucho antes, por ejemplo en 60 segundos vemos que ambas líneas están mucho mas cercanas que cuando no teníamos la solución inicial. Además vemos que las

soluciones enteras que se encuentran a medida que se optimiza, son el doble del anterior modelo.

Heurística

Para la parte de heuristica tampoco se ha obtenido cambios en cuanto a la velocidad del algoritmo. Tampoco se registraron cambios en cuanto a la solución final ni a la cantidad de subtours en que se corta el proceso.

Se cree que los pocos cambios realizados en cuanto al ingreso de la solución inicial, es que el valor de la solución inicial tenian una gran diferencia en distancia comparado con los óptimos.