

Entrega tres

3)

Modelo MTZ

```
forall ( j in cities )
    flow_in:
        sum ( i in cities : i != j ) x[< i, j >] == 1;

forall ( i in cities )
    flow_out:
        sum ( j in cities : i != j ) x[< i, j >] == 1;
```

Lo primero que se hace es fijarse que haya un solo camino que entre y luego un solo camino que salga, esto hace que se visite una sola vez. Pero puede generar subtours.

```
forall ( i in cities : i > 1, j in cities : j > 1 && j != i )
    subtour:
        u[i] - u[j] + ( n - 1 ) * x[< i, j >] <= n - 2;
```

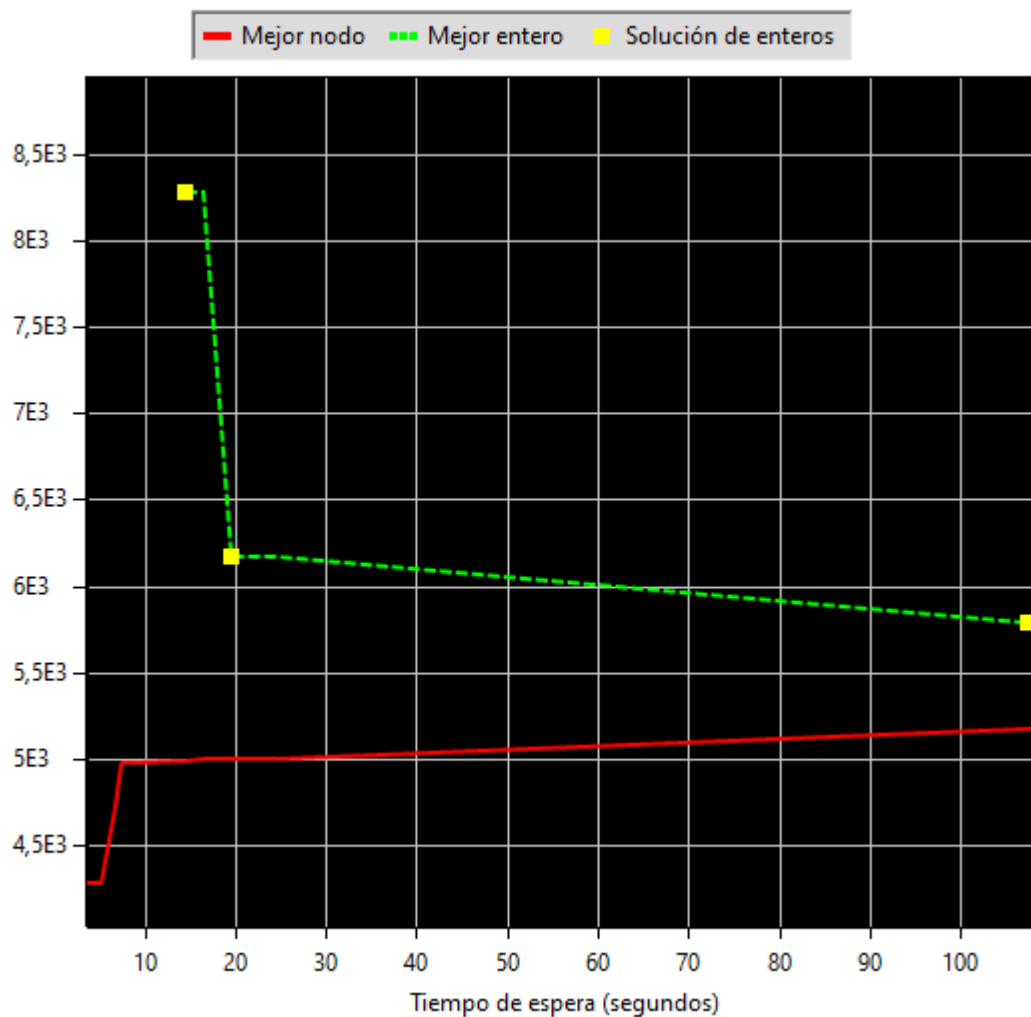
Luego se hace esta restricción que obliga a cada vértice que tenga un valor de cuando se visita. Esto genera que no pueda haber más de un vértice que tenga la misma posición de orden que otro, y a su vez obliga que todos los vértices tengan un vértice contiguo, por lo tanto todos están en un mismo recorrido.

Ventajas:

La solución encontrada es la solución óptima para el problema.

Desventajas:

El modelo tarda un tiempo en resolverse y cuando se agregan algunas ciudades más, el tiempo de ejecución crece enormemente.



Se puede ver en el gráfico como la curva solución es decreciente, mostrando así que este modelo siempre va mejorando su solución hasta llegar a la solución óptima. La curva roja se puede explicar sabiendo que el programa cplex busca la solución del problema con variables continuas y haya una solución óptima que vendría a ser el primer valor que obtiene esta curva, esta solución lo más probable es que sea inalcanzable con variables enteras, pero sirve como una cota inferior de la solución. A medida que se van haciendo cortes de la región, la cota inferior va aumentando hasta llegar al valor igual al de la mejor solución con variables enteras.

Modelo Subtours

```
forall ( j in Cities )
    sum ( < i, j > in Edges ) x[< i, j >] + sum ( < j, k > in Edges ) x[< j,
        k >] == 2;
```

El problema de esta restricción es que permite que haya dos caminos de entrada o dos caminos de salida, si el primer término de la suma suma 2 o si el segundo suma 2. (igual

por el código si ya se visitó ese vértice no se va a volver a visitar a menos que sea el primer elemento).

```
forall ( s in subtours )  
  sum ( i in Cities : s.subtour[i] != 0 ) x[< minl ( i, s.subtour[i] ), maxl  
    ( i, s.subtour[i] ) >] <= s.size - 1;
```

Lo que hace esta restricción es eliminar la posibilidad de que un subtour pueda tener un ciclo. Cuenta la cantidad de aristas y se fija que sea menor o igual a la cantidad de vértices del subtour menos 1.

Desde el principio el modelo está planteado para que solo contemple los caminos los cuales tienen $i < j$, por lo tanto esto acorta las posibilidades que tiene el modelo entonces hace que sea más rápido pero no nos da una certeza de conseguir la solución óptima al problema. En el caso de este problema que solo hay que encontrar un camino que solo cumpla que la distancia sea mínima, con esta restricción de los caminos es asegurado que se encuentra una solución, que es la que va de vértice en vértice en orden ascendente comenzando por el vértice 1. Si se le agregase algún objetivo más, como cumplir que lleve un monto acumulado que no debe superar o se le obliga que visite un vértice antes que otro, en esos casos no es tan seguro que el modelo pueda encontrar una solución al problema.

Ventajas:

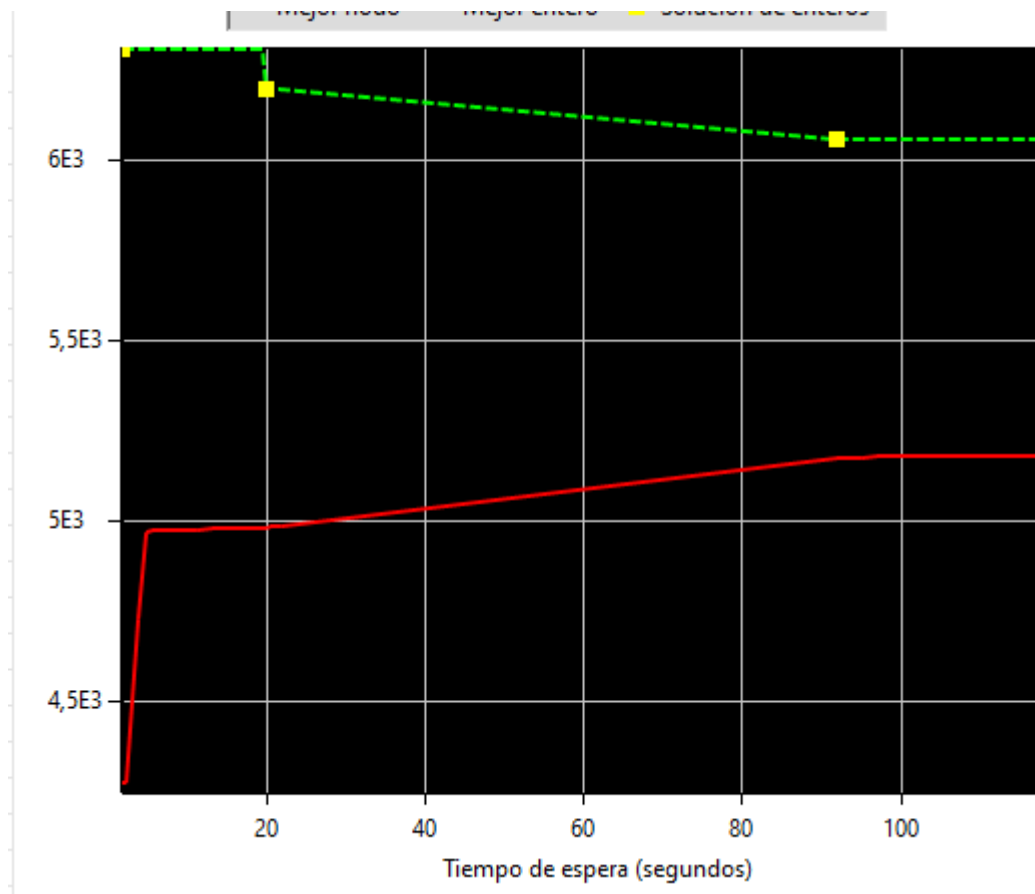
La solución se encuentra más rápido, aunque también el modelo es más chico ya que no permite aristas con $i > j$.

Desventajas:

Como el modelo es más chico, probablemente no se alcance la solución óptima al problema, ya que la solución óptima podría contener a una arista que no está contemplada en este modelo.

5)

Modelo MTZ



Ya se puede visualizar en el gráfico que empieza a buscar la solución mucho más cerca que sin la solución inicial. Al agregar la solución inicial al problema, el software toma esta solución para acortar los posibles caminos. Como ya se sabe que un camino tiene una longitud determinada, cada camino que supere este número o llegue a este número antes de haber llegado al final del camino, se deberá descartar este camino probable, entonces la cantidad de soluciones se acorta.

Modelo Subtours

La solución inicial del problema contempla que todos los caminos entre vértices son posibles, entonces la solución inicial no necesariamente puede ayudar, ya que si al menos 1 camino no existe en el modelo planteado, la solución podría cambiar completamente.