Fiuba
Facultad de Ingeniería
Universidad de Buenos Aires

U.B.A. - FACULTAD DE INGENIERÍA

Departamento de Computación

71.14 Modelos y optimización I

TP 3

Primer cuatrimestre de 2022

Curso Oitana

Alumna:
Valeria Magdalena Rocha Bartaburu

Padrón: 90596

Eliminación de subtours

Modelo TSP_MTZ

La formulación MTZ nos dice que para evitar subtours se deben agregar las variables Ui, es decir, el número de secuencia en la cual la ciudad i es visitada. A partir de esto se formula la siguiente restricción:
Ui - Uj + n * Yij <= n - 1 siendo n la cantidad de ciudades del problema.
Con i,j = 1,..., n para todo i distinto de j
No se incluye a la ciudad 0 (origen). De esta forma al quedar fuera del arco, el mismo no se puede cerrar. Para esto habría que unir a la primera ciudad con la última, evitando que se formen subtours.

En el MTZ se generan todas las aristas, las idas y vueltas.

En el problema planteado las ciudades van de 1 a n, es por esto que la restricción utilizada es:
Ui - Uj + (n - 1) * Yij <= n - 2

Modelo tsp

Xij vale 1 si se visita desde la ciudad i a la ciudad j. Para que se eviten tours se debe cumplir que Xij + Xji <= 1

Para cada iteración:
Mientras haya un nodo adyacente sin visitar, que no sea el de inicio, hay que moverse a ese nodo. De esta forma genera subtours y se va quedando con el de menor tamaño, es decir, el que recorre la menor cantidad de ciudades. Dicho recorrido lo guarda en la variable subtours, la cual va actualizando en cada iteración. Al crear subtour no se pueden agregar ciudades que ya fueron visitadas (es decir que se encuentran en el subtour).

En cada nueva iteración se trata de eliminar el subtour a partir de la colección subtours, utilizando la siguiente restricción:

```
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;
```

El algoritmo terminará cuando el tamaño del subtour sea igual a la cantidad de ciudades n, en nuestro caso 100. Logrando de esta forma que el camino final no tenga ciclos.

Comparación de modelos

El modelo tsp elimina los subtours mientras que el modelo MTZ los evita directamente.

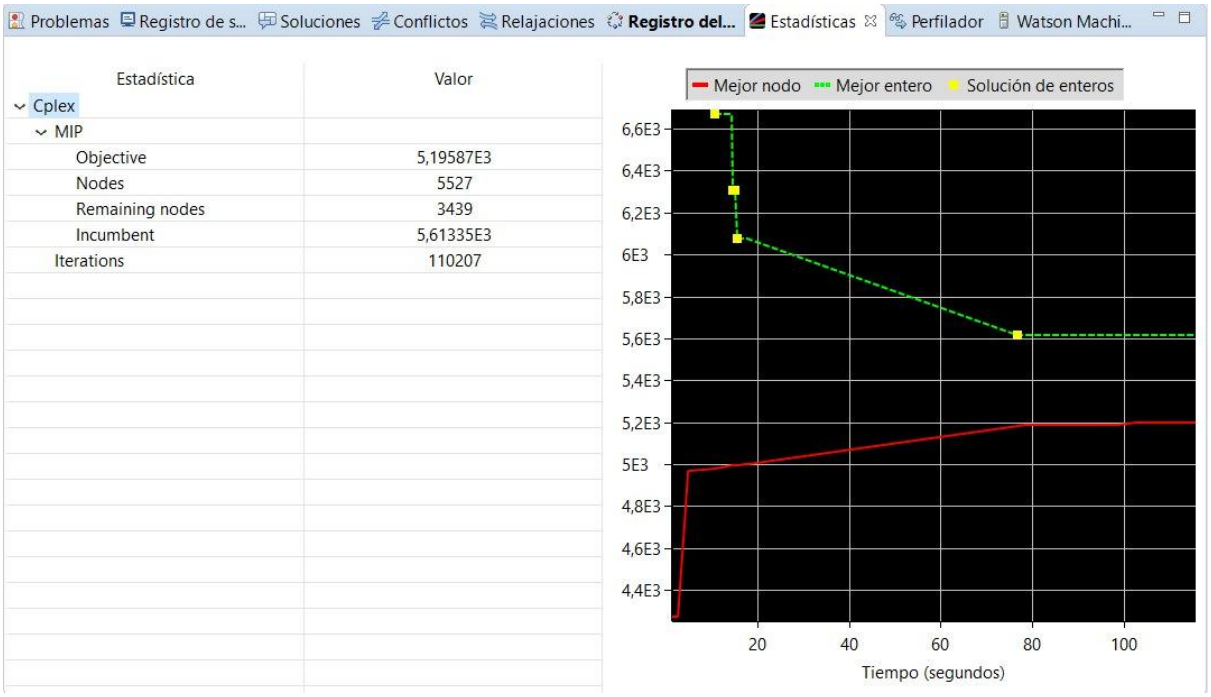
El modelo MTZ tiene todas las variables Xij mientras que el otro tiene un "ordered" por lo tanto solo tiene Xij cuando i < j. El beneficio es que en el modelo de subtours se reduce el problema ya que se quitan aristas que unen ciudades. Esto resulta posible porque se busca cumplir que sea mínimo el camino, las únicas restricciones que se agregan son las del modelo del viajante. En este caso funciona con problemas que se puedan resolver con un grafo dirigido.

Es decir que como ventaja se tiene que en el modelo de subtours se simplifica el problema, en contraposición con la desventaja del modelo MTZ que resulta más trabajoso en cuanto a procesamiento al contar con todas las aristas. Esto puede verse reflejado en la siguiente sección, en los tiempos de ejecución de cada modelo y la cantidad de iteraciones necesarias para resolver el problema. Vemos que ambos (tiempo e iteraciones) resultan mayor en el caso del modelo MTZ.

Tiempos de ejecución

Modelo	Tiempo de ejecución (h:m:s)	Iteraciones
TSP_MTZ	09:56:94	1230028
tsp	00:30:55	1515

Gráfico de la solapa "Statistics" para TSP_MTZ



La parte superior del gráfico muestra la solución con enteros que va encontrando al recortar el polígono, es por esto que la curva verde va descendiendo hasta que encuentra la solución óptima. De esta forma vemos que va tendiendo a la curva roja (punto óptimo). Esto ocurre a los 100 segundos, momento en que ambas curvas se acercan.

Modificación de Heurística

Recordando que al tener una solución inicial estaríamos “achicando el poliedro”: si la solución inicial proporcionada al modelo no resulta mejor que la utilizada por el motor inicialmente, podría aumentar el tiempo de ejecución, ya que aunque se la descarte se estaría sumando el tiempo de evaluar dicha solución.
Si la solución inicial proporcionada es mejor que la utilizada por el motor, va a mejorar el tiempo de ejecución al mejorar el funcional. Estaríamos acotando el tiempo.
Es decir que no siempre podría mejorar el tiempo agregar una solución inicial.

A continuación se detallan los tiempos de ejecución al agregar a los modelos la solución inicial obtenida al modificar la heurística del TP2, sin que tome en cuenta las restricciones de capacidad:

Modelo	Tiempo de ejecución (h:m:s)	Tiempo con heurística modificada (h:m:s)
TSP_MTZ	07:02:30	06:40:50
tsp	00:30:55	00:29:85

Se verifica que para el caso del modelo MTZ, agregar una solución inicial redujo el tiempo de ejecución.

Modelo	Iteraciones	Iteraciones con heurística modificada
TSP_MTZ	1230028	1260810
tsp	1515	1515

Las iteraciones obtenidas fueron al completar la ejecución del modelo.

El gráfico de la solapa "Statistics" obtenido para el modelo TSP_MTZ con la modificación de la heurística es el siguiente:

Cplex utiliza el método de ramificación y corte por lo que al agregar una solución le permite eliminar parte del espacio donde va a buscar la solución. De esta forma se pueden obtener árboles de ramificación y corte más pequeño, lo que se traduce en un menor tiempo de ejecución.