

Modelo TSP_MTZ

Funcionamiento

El algoritmo que utiliza este modelo es el clásico modelo del viajante que vimos en la materia. La eliminación de subtours está reflejada por la restricción $U_i - U_j + nY_{ij} \leq n - 1$ para todo i, j de 1 a n , con i distinto de j . Donde la variable U_i es el orden en que se visita la sucursal bancaria i .

Ventajas y Desventajas

- ✓ Algoritmo más corto y simple.
- ✗ Gran cantidad de variables X_{ij} .
- ✗ Tarda mucho tiempo en encontrar la solución (13 minutos aproximadamente).

Modelo TSP_subtours

Funcionamiento

Para el caso de este modelo, la clave del algoritmo está en la eliminación de los subtours que se producen a partir de las restricciones que se van agregando a lo largo de las ejecuciones. Siempre toma el subtour de menor cantidad de nodos y agrega la restricción. Esto se hace hasta que no haya más subtours. CPLEX utiliza el método de Branch-and-Cut el cual combina Branch-and-Bound con planos de corte para ir eliminando restricciones, esto lo podemos ver en detalle en el engine log del programa.

Ventajas y Desventajas

- ✓ Menor cantidad de variables X_{ij} .
- ✓ Poco tiempo en encontrar la solución (36 segundos aproximadamente).
- ✗ Algoritmo más largo y complejo.

Diferencias entre modelos

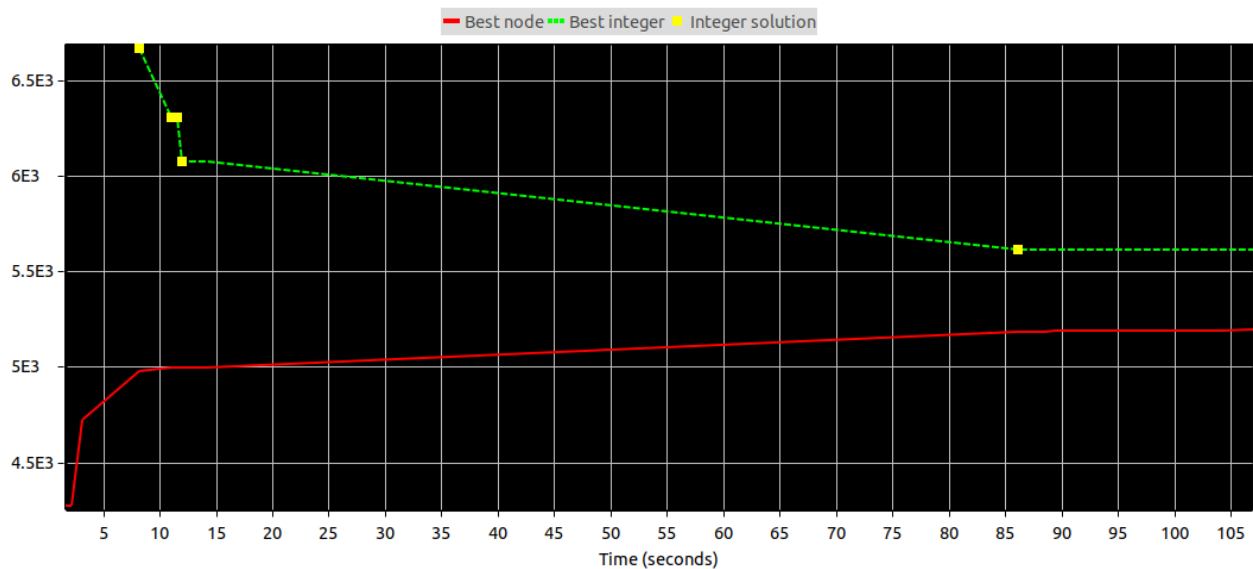
El modelo MTZ tiene todas las variables X_{ij} , mientras que el otro modelo tiene un “ordered” por lo que solo tiene X_{ij} cuando $i < j$. Esto hace que el modelo de subtours tenga muchas menos variables, por lo que las opciones a considerar por el algoritmo son mucho menores y esto hace

que sea más rápido. Además, esto es posible ya que el recorrido que tiene que realizar el camión es hacia adelante y en ningún momento tiene que retroceder.

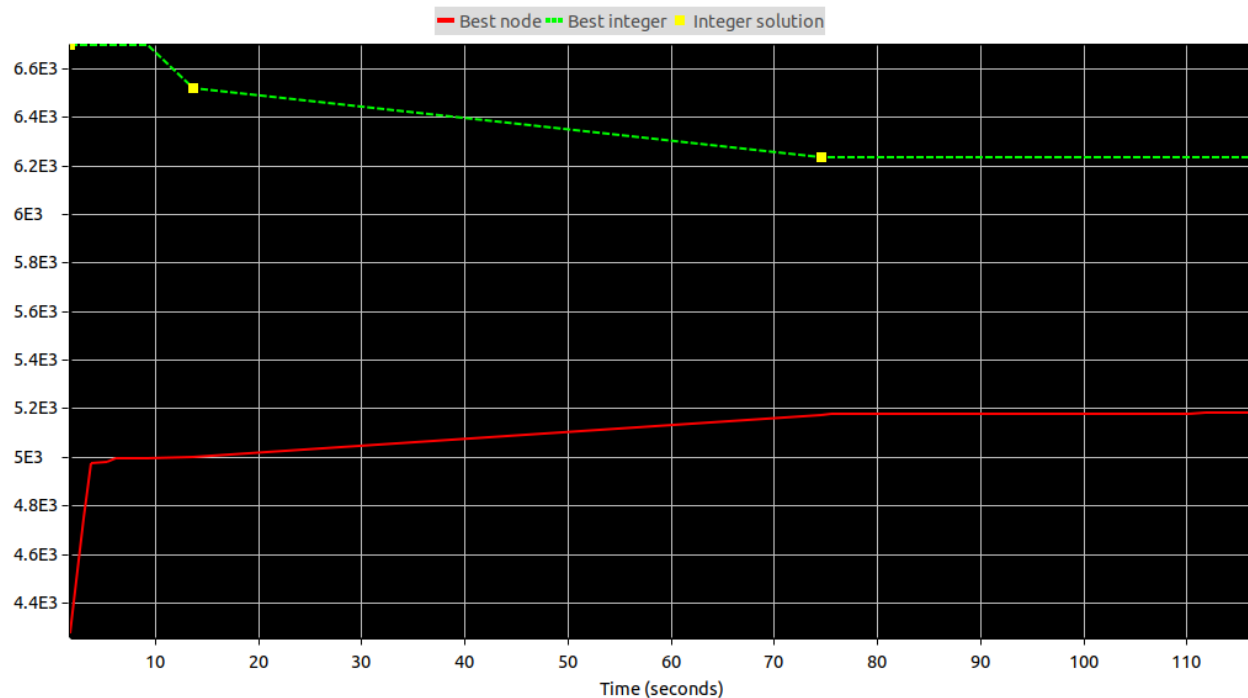
Gráficos (MTZ)

```
int ordenInicial[Cities] = [68, 91, 45, 71, 44, 64, 72, 49, 31, 27, 41, 57, 39, 60, 66, 17, 11, 61, 36, 69, 24, 12, 53, 40, 42, 9, 28, 6, 37, 2, 19, 99, 70, 86, 29, 81, 25, 20, 51, 43, 67, 32, 23, 38, 77, 14, 80, 15, 78, 59, 16, 79, 88, 94, 10, 63, 48, 73, 76, 87, 1, 98, 34, 30, 84, 7, 8, 89, 96, 35, 93, 52, 33, 92, 54, 46, 90, 56, 26, 75, 18, 85, 65, 55, 58, 50, 83, 97, 100, 5, 95, 82, 3, 62, 22, 4, 21, 13, 74, 47];
```

Sin solución inicial



Con solución inicial



Tanto como para el modelo MTZ como para el subtours, se tarda más tiempo en encontrar la solución óptima. Para el primero, el tiempo pasó de 13 minutos a 16 minutos, mientras que para el segundo paso de 36 segundos a casi 38 segundos. En este último caso, el tiempo es casi despreciable.

En este caso, el costo de haber agregado la solución es mayor al beneficio ya que la solución no aporta demasiado. A priori, se podría pensar que se iba a reducir el tiempo y se iba a encontrar el óptimo mucho más rápido, pero como se puede ver en los gráficos, para el caso de **sin** solución en los primeros 100 segundos se encuentran 5 soluciones enteras mientras que en el de **con** solución inicial, se encuentran 2. Además se ve clara la distancia que hay entre las líneas verde y roja.

Por otro lado, como esto me llamó la atención, le compartí mi solución inicial a un compañero para que la pruebe en su PC y llamativamente obtuvo otros resultados. Por empezar, el modelo con la solución inicial es más rápido que el sin solución inicial, y además el tiempo de ejecución del modelo también es menor. Los tiempos fueron de 3 minutos aproximadamente con la solución inicial y 6 minutos sin.