Una aplicación del método MTZ a la solución del problema del agente viajero

An Application of the MTZ method to the solution of the Travelman Salesman Problem.

Jairo Alberto Villegas Flórez, Carlos Julio Zapata Grisales¸ Gustavo Gatica
Ingeniería Industrial, Universidad Tecnológica de Pereira, Pereira, Colombia.
Ingeniería Industrial, Universidad Andrés Bello, Chile
javi@utp.edu.co
cjzapata@utp.edu.co
gustavo.gatica@usach.cl

***Resumen*— El Problema del Agente Viajero o TSP por sus siglas en inglés (Travelman Salesman Problem), es un problema de optimización que consiste en hallar la ruta mínima de un recorrido de n ciudades, saliendo desde la ciudad de origen o nodo cero, recorriendo las n ciudades y retornando al punto o ciudad de origen nuevamente, en el siguiente artículo se expone una solución al problema del agente viajero utilizando una heurística exacta llamada "método MTZ", la cual se va a solucionar en el software AMPL utilizando algunas de sus librerías de optimización.**

***Palabras clave*— Agente Viajero, Heurística, TSP.**

***Abstract* — The Traveling Salesman Problem is an optimization problem that consists in finding the minimum path of a tour of n cities, departing from the city of origin or zero node, running the n cities and returning to the point of origin or zero node in the next article a solution for the traveling salesman problem is exposed using an exact heuristic called "method MTZ" wich will be solved using some of its optimization libraries.**

***Key Word* — Heuristic, TSP**

I. INTRODUCCIÓN

El problema del Agente Viajero o TSP[1] por sus siglas en ingles es un problema de optimización combinatorial muy conocido en la literatura de investigación de operaciones aplicados a la logística. Consiste en recorrer un tour de n ciudades partiendo de un punto origen recorriendo las n ciudades solamente y entrando a cada una de ellas un sola vez y retornar al punto de partida. Este recorrido puede estar expresado en distancia o en tiempo, es decir recorrer la menor distancia posible o realizar el recorrido de todas las ciudades en el menor tiempo. Este problema se puede modelar mediante un grafo dirigido en donde los vértices son las ciudades y los arcos son los caminos, dicho arco debe tener un peso que representa la distancia entre dos ciudades o el tiempo en ir de una ciudad a otra.

Para la solución de este problema existen diferente formas de resolverlos, puede ser usando programación lineal, técnicas exacta como las heurísticas o técnicas que resuelven problemas combinatoriales como las metaheurísticas.

En este artículo se aplica una heurística exacta llamada método MTZ para la solución de un agente que tiene que recorrer n ciudades, donde el n puede variar en 1,2,3,4,…..

II. PLANTEAMIENTO MATEMÁTICO DEL TSP

El problema del agente viajero tiene un conjunto de n ciudades que se llaman nodos y un conjunto de distancias entre cada ciudad llamados arcos, cabe aclarar la siguiente situación: si el agente tiene que ir de la ciudad 1 a la ciudad 2, esta distancia la llamaremos d_{12} y no tiene que ser necesariamente igual a d_{21}, por eso se dice que existe un TSP simétrico y un TSP asimétrico que es conocido como ATSP, el objetivo principal es encontrar la ruta más corta posible que visite exactamente una vez cada ciudad y luego retorne al lugar de origen[1].
El TSP se puede formular matemáticamente de una forma sencilla, su complejidad crece a medida que se aumenta el número de ciudades, para formular el TSP como un problema de PLE, haremos uso de una variable X_{ij} la cual tomará el valor de 1 si el arco de i a j es usado y 0 en caso contrario.
Su formulación queda de la siguiente manera:

[1] TSP Travelling Salesman Problem

$$X_{ij} = \begin{cases} 1, si\ se\ visita\ la\ ciudad\ j\ después\ de\ la\ ciudad\ i \\ 0,\ \ si\ no\ se\ visita\ la\ ciudad\ j\ después\ de\ la\ ciudad\ i \end{cases}$$

$C_{ij} = es\ el\ costo\ asociado\ a\ la\ visita\ de\ la\ ciudad\ j, después\ de\ visitar\ la\ ciudad\ i$

$$Min\ Z = \sum_{i=0}^{n} \sum_{j=0}^{n} C_{ij} X_{ij} \quad (1)$$

La ecuación (1) se refiere a la función objetivo, que busca minimizar la distancia recorrida por el agente viajero, adicional a esta función se deben plantear las restricciones de entrada y salida de cada ciudad [2].

$$\sum_{i=0, i\neq j}^{n} X_{ij} = 1 \quad (2)$$

La ecuación (2) garantiza que se llega a una ciudad cada vez.

$$\sum_{j=0, j\neq i}^{n} X_{ij} = 1 \quad (3)$$

La ecuación (3) garantiza que se sale de una ciudad cada vez.

Aunque no basta con las restricciones (2) y (3) para optimizar el problema, ya que se pueden crear subtoures entre conjuntos de ciudades y para ello se tiene la siguiente restricción:

$$\sum_{(i,j)\,:\, i\in L, j\in N \setminus L} X_{ij} \geq 1$$

III. MÉTODOS DE SOLUCIÓN

Los problemas de programación lineal como el agente viajero (TSP), son considerados de difícil solución por tener una explosión combinatorial que hace difícil mediante técnicas exactas encontrar el óptimo global del problema. Dentro de los problemas combinatoriales hay una división que los clasifica en dos tipos problemas tipo P o polinomiales para los cuales es fácil encontrar una solución óptima y los problemas tipo NP o no polinomiales, para los cuales encontrar el óptimo global no es tan sencillo[1], [4] y [5]. El problema del agente viajero cae en la categoría de los no polinomiales, por lo cual encontrar el óptimo global no es tarea fácil y más si se incrementa el número de ciudades n.

El problema del agente viajero se puede resolver utilizando el modelo MTZ[2][2], siempre y cuando sea con un número pequeño de ciudades, el planteamiento del modelo es de la siguiente manera:

Las variables

$$X_{ij} = \begin{cases} 1, si\ se\ va\ de\ la\ ciudad\ i\ a\ la\ ciudad\ j \\ 0, de\ lo\ contrario \end{cases}$$

$$\mu_i = número\ de\ ciudades\ visitadas\ hasta\ la\ ciudad\ i$$

[2] MTZ es un modelo heurístico que se usa para romper los subtours en el problema del agente viajero.

Función Objetivo

$$minZ = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$$

Las restricciones

$$\sum_{i} X_{ij} = 1\ \ entra\ una\ vez\ a\ cada\ ciudad$$

$$\sum_{j} X_{ij} = 1\ \ deja\ una\ vez\ cada\ ciudad$$

$$\mu_i + 1 \leq \mu_j + n * (1 - X_{ij}), rompe\ subtours$$

$$para\ i = 2, \ldots., n\ i \neq j, j = 2, \ldots\ldots., n$$

$$X_{ij} \in \{0,1\}, para\ todo\ i\ y\ j$$

$$\mu_i \geq 1, para\ todo\ i$$

IV. DATOS

Para la realización del presente artículo se planteó un modelo con 5 nodos y así observar como el modelo MTZ funciona cuando n es pequeño y también colocar especial cuidado a las restricciones de subtour.

Los datos de las distancias entre nodos se muestra en la siguiente tabla:

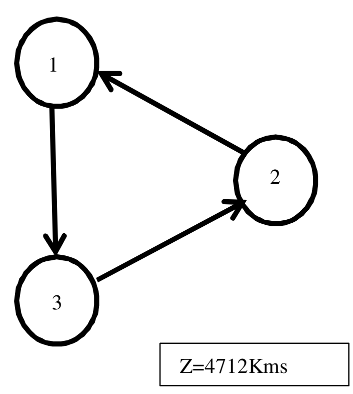
TABLA I

	1	2	3	4	5
1	10000	702	424	842	2396
2	702	10000	324	1093	2136
3	454	324	10000	1137	2180
4	842	1093	1137	10000	1616
5	2396	2136	2180	1616	10000

Tabla 1 matriz de datos para 5 ciudades

V. RESULTADOS

Al correr en el software AMPL el modelo MTZ para un total de 5 ciudades, en primera instancia sin activar las restricciones de subtour el agente deberá realizar el siguiente recorrido, el cual mostraremos a continuación gráficamente:



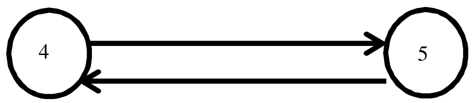


Figura 1. Caso 1 agente viajero sin activar las restricciones de subtour.

Como se puede, ver si en el problema del agente viajero no se tienen en cuenta las restricciones de subtour, los resultados pueden llegan a ser infactibles, tal como se observa en este caso.

Si se activan las restricciones de subtour puede haber cambios significativos en cuanto a la solución como se verá a continuación:

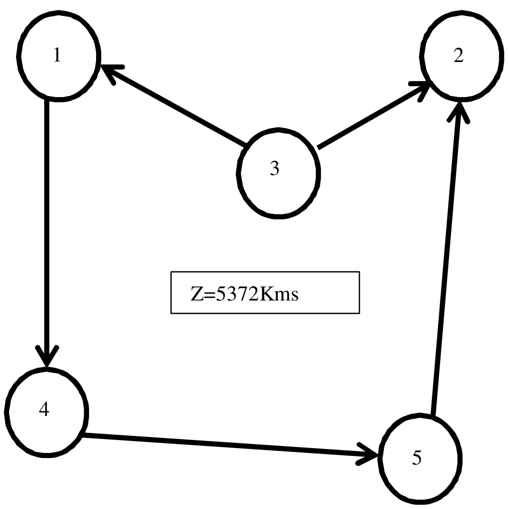
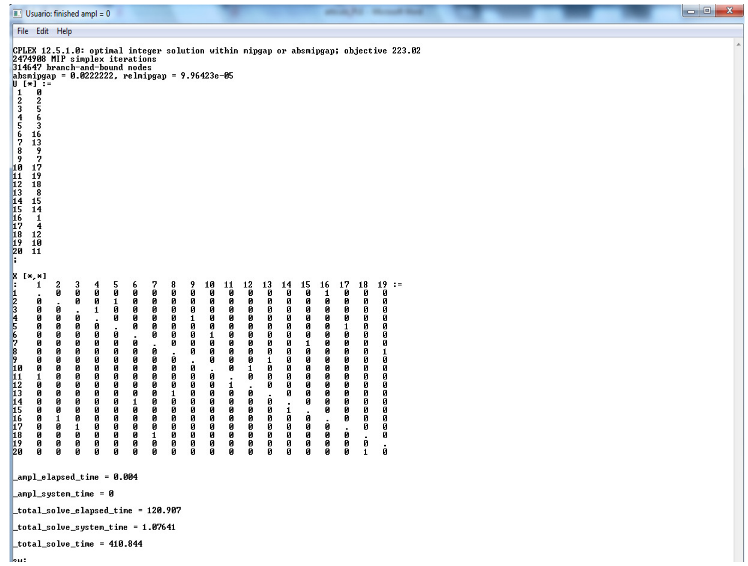


Figura 2. Caso 2 agente viajero activando las restricciones de subtour.

Al activar las restricciones de subtour se observa cómo se cerró el circuito entre las 5 ciudades, con un valor para la función objetivo de 5372 Kms, que aunque mayor comparado con el valor para el caso 1 de 4712Kms, la última solución es mejor ya que es factible.

Con otro ejemplo más complejo se puede ver como el modelo MTZ es de bastante utilidad a la hora de romper los subtours y lograr que la solución sea factible, en el siguiente ejemplo se va a trabajar una matriz de 20 ciudades y se programará con AMPL el modelo MTZ, los resultados obtenidos son los siguientes:



```
Usuario: finished ampl = 0
File  Edit  Help

CPLEX 12.5.1.0: optimal integer solution within mipgap or absmipgap; objective 223.02
2474908 MIP simplex iterations
314647 branch-and-bound nodes
absmipgap = 0.0222222, relmipgap = 9.96423e-05
U [*] :=
 1   0
 2   2
 3   5
 4   6
 5   3
 6  16
 7  13
 8   9
 9   7
10  17
11  19
12  18
13   8
14  15
15  14
16   1
17   4
18  12
19  10
20  11
;

X [*,*]
:    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19 :=
1    .   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0
2    0   .   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
3    0   0   .   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
4    0   0   0   .   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
5    0   0   0   0   .   0   0   0   0   0   0   0   0   0   0   0   1   0   0
6    0   0   0   0   0   .   0   0   0   1   0   0   0   0   0   0   0   0   0
7    0   0   0   0   0   0   .   0   0   0   0   0   0   0   1   0   0   0   0
8    0   0   0   0   0   0   0   .   0   0   0   0   0   0   0   0   0   0   1
9    0   0   0   0   0   0   0   0   .   0   0   0   1   0   0   0   0   0   0
10   0   0   0   0   0   0   0   0   0   .   0   1   0   0   0   0   0   0   0
11   1   0   0   0   0   0   0   0   0   0   .   0   0   0   0   0   0   0   0
12   0   0   0   0   0   0   0   0   0   0   1   .   0   0   0   0   0   0   0
13   0   0   0   0   0   0   0   1   0   0   0   0   .   0   0   0   0   0   0
14   0   0   0   0   0   1   0   0   0   0   0   0   0   .   0   0   0   0   0
15   0   0   0   0   0   0   0   0   0   0   0   0   0   1   .   0   0   0   0
16   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   .   0   0   0
17   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   .   0   0
18   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   .   0
19   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   .
20   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0

_ampl_elapsed_time = 0.004

_ampl_system_time = 0

_total_solve_elapsed_time = 120.907

_total_solve_system_time = 1.07641

_total_solve_time = 410.844
```

Figura 3. Resultado del software AMPL para 20 ciudades

Para observar un poco más a fondo el comportamiento del modelo MTZ vamos empezar con un modelo de 3 ciudades, que luego se irá aumentando paulatinamente a más ciudades para así ver los tiempos totales de solución dependiendo del número de nodos.

La solución óptima para la función objetivo del modelo con 20 ciudades es 223,02

MODELO MTZ	
NRO. CIUDADES	**TOTAL SOLVE TIME [segs]**
3	0.0156
4	0.0468
5	0.0156
6	0.0468
7	0.1248
8	0.1248
9	0.1872
10	0.3432
11	0.2808
12	0.4836
13	1.06081
14	1.29481
15	3.55682
16	7.08245
17	13.6189
18	19.6717
19	132.398
20	417.744

Tabla 2 resultados tiempo total de solución variando el número de ciudades desde 3 hasta 20

Complejidad Algorítmica Modelo MTZ

La complejidad algorítmica del modelo MTZ es de forma cuadrática y se expresa de la siguiente forma:

La función objetivo del modelo MTZ consiste en minimizar el costo de recorrido de toda la ruta y viene dada por la siguiente función:

$\sum_i \sum_j C_{ij} * X_{ij}$ Es complejidad $O(n^2)$ cada sumatoria equivale a un for desde i y j hasta n

Las restricciones son las siguientes:

$$\sum_i X_{ij} = 1 \text{ es complejidad } O(n)$$

$$\sum_j X_{ij} = 1 \text{ es complejidad } O(n)$$

$$\mu_i + 1 \leq \mu_j + n * (1 - X_{ij}) \text{ es complejidad constante } O(c)$$

De ahí se concluye que la complejidad del modelo viene dada por:

$$O(n^2) + O(n) + O(c)$$

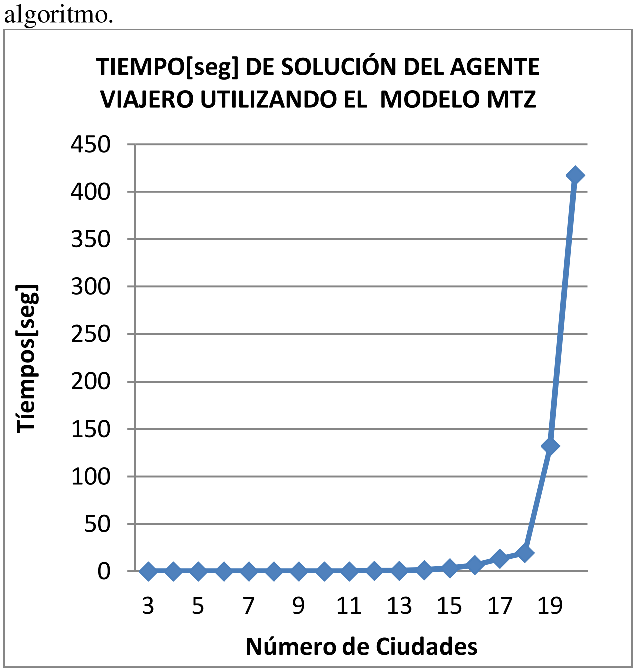
De donde:

$$O(n) \gg O(c)$$

Y

$$O(n^2) \gg O(n)$$

Por lo tanto se puede concluir que el modelo MTZ tiene complejidad $O(n^2)$. También se puede observar de acuerdo a la gráfica 1, como el tiempo de solución va aumentando de forma cuadrática a medida que el número de ciudades se incrementa, lo que corrobora una vez más la complejidad del algoritmo.



Gráfica 1 tiempo de solución vs número de ciudades para el modelo MTZ

VI. CONCLUSIONES

Se pudo observar que el modelo MTZ aunque es una heurística, su comportamiento al momento de eliminar los subtours es bastante aceptable.

Se observa como a medida que el número de ciudades va en aumento el tiempo de solución crece mucho más rápido, lo que hace prever que esta heurística es muy buena cuando el número de ciudades es pequeño pero no así cuando el número de nodos es grande

Por lo tanto no es recomendable el uso modelo MTZ cuando el número de nodos es muy grande pues podría perder eficacia en el tiempo de solución

RECOMENDACIONES

Los resultados obtenidos para el presente artículo se obtuvieron programando el modelo MTZ en el software AMPL, se recomienda programar el modelo en otro software diferente, para ver si existen algunas diferencias en los resultados y en la programación del modelo.

REFERENCIAS

[1]. Hincapie. Isaza. Ricardo, Rios. Porras. Carlos, y Gallego. R. Ramón *Técnicas Heurísticas Aplicadas al Cartero Viajante (TSP),* Revista Scientia Et Technica N° 24. Marzo de 2004. http://www.utp.edu.co/~planeamiento/prod_aca/articulos/Tecnicas_heuristicas_TSP4.pdf

[2]. Velásquez. Escobar. John Willmer, *Sistemas de Transporte: Modelado y Optimización,* Septiembre de 2013

[3]. http://agentefviajerosimpleymultiple.blogspot.com/

[4]. Velásquez. Escobar. John Willmer, Linfati. Rodrigo Adarme. Jaimes. Wilson. *Problema de Localización y Ruteo con Restricciones de Capacidad: Revisión de la Literatura,* Mayo de 2015.

[5]. Rocha, L.; González, C. y Orjuela, J. (2011). Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería, Vol. 16, No. 2, pág. 35 - 55 .