



Tarea Computacional 2: El problema del Vendedor Viajero y su solución mediante Python con PuLP

Integrantes

ANSORENA CARRASCO JAIME IGNACIO - 2020401497
COVARRUBIAS CHÁVEZ PATRICIO ANDRÉS - 2020409609

1. Situación Propuesta

En un sombrío futuro, la Tierra, nuestro hermoso planeta azul, ha sido devastada por la imprudencia del ser humano, llevando a la humanidad al borde de la extinción.

La solución ante esta crisis se presenta en la figura de un intrépido astronauta, cuya nave espacial está equipada con tecnología de punta y un sistema de navegación avanzado. Su misión es clara: explorar una serie de planetas en diferentes sistemas solares con el objetivo de determinar cuál de ellos es habitable para la supervivencia de la especie humana.

La lista de planetas es tan variada como desconcertante. Entre ellos se encuentran gigantes gaseosos con tormentas violentas y gravedad abrumadora, así como exuberantes mundos llenos de vida exótica. Sin embargo, las distancias entre estos planetas son totalmente asimétricas debido a la gran cantidad de eventos espaciales que ocurren, lo que obliga a recalcular constantemente las rutas de navegación.

Es en este punto donde entra en juego tu habilidad como ingeniero de la UdeC, quien ha aprobado Optimización I con las más altas calificaciones. Tu tarea es crucial: ayudar a calcular la ruta más eficiente para visitar cada planeta una sola vez en busca de signos de habitabilidad. Una ruta no óptima podría marcar la diferencia entre la supervivencia de la humanidad y su extinción.

La humanidad confía en tu ingenio para resolver este enigma espacial, minimizando la distancia recorrida por el astronauta y, de esta manera, encontrar un nuevo hogar para nuestra especie.



Tu misión



2. Modelo Matemático

Sea $G = (V, A)$ un digrafo completo, donde $V = \{1, \dots, n\}$ es el conjunto de vértices (planetas), $A = \{(i, j) : i, j \in V\}$ el conjunto de aristas (distancias entre los planetas) y $S \subset V$ (subconjunto de V).

Por lo tanto, se define para las diferentes formulaciones la siguiente variable de decisión:

$$x_{ij} = \begin{cases} 1, & \text{si el arco } (i, j) \text{ es parte del camino óptimo} \\ 0, & \text{no es parte del camino óptimo} \end{cases} \quad i, j = 1, \dots, n \quad (1)$$

2.1. Formulación DFJ

Se formula el modelo DFJ como:

$$\text{Minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (2)$$

$$\text{s.a} \quad \sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \quad j \neq i \quad (3)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n \quad i \neq j \quad (4)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad S \subset V : S \neq \emptyset \quad (5)$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, \dots, n \quad (6)$$

donde c_{ij} = distancia de la arista desde el planeta i hacia el planeta j [años luz]
 n = cantidad de planetas contenidos en V [unidad]

2.2. Formulación MTZ

Se formula el modelo MTZ como:

$$\text{Minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (7)$$

$$\text{s.a} \quad \sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \quad j \neq i \quad (8)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n \quad i \neq j \quad (9)$$

$$u_i - u_j + (n - 1)x_{ij} \leq n - 2 \quad i, j = 2, \dots, n \quad (10)$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, \dots, n \quad (11)$$

$$u_i \in \mathbb{Z} \quad i = 1, \dots, n \quad (12)$$



donde c_{ij} = distancia de la arista desde el planeta i hacia el planeta j [años luz]
 n = cantidad de planetas contenidos en V [unidad]
 u_i = número arbitrario que representa el orden del planeta i en el camino óptimo [unidad]

2.3. Formulación GG

Se formula el modelo GG como:

$$\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (13)$$

$$\text{s.a } \sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \quad j \neq i \quad (14)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n \quad i \neq j \quad (15)$$

$$\sum_{j=1}^n g_{ij} - \sum_{j=2}^n g_{ji} = 1 \quad i = 2, \dots, n \quad (16)$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, \dots, n \quad (17)$$

$$0 \leq g_{ij} \leq (n-1)x_{ij} \quad i = 2, \dots, n \quad j = 1, \dots, n \quad (18)$$

donde c_{ij} = distancia de la arista desde el planeta i hacia el planeta j [años luz]
 n = cantidad de planetas contenidos en V [unidad]
 g_{ij} = número de arcos en el camino desde el planeta 1 al arco ij en el camino óptimo [unidad]

3. Descripción de las instancias

Se utilizó un conjunto diverso de 12 instancias en la exploración de planetas habitables, el cual incluyó la generación de 9 escenarios planetarios variados (instancias de tamaño 10x10 a 18x18), considerando complejidades en la distancia asimétrica entre planetas de manera aleatoria.

Además, se incluyeron 3 instancias de TSPLIB (ftv35, ftv38 y ftv44), las cuales fueron incluidas para estudiar el tiempo de ejecución en los algoritmos MTZ y GG, y así denotar con mayor margen sus diferencias temporales. Estos datos proporcionaron una base de comparación relevante para evaluar las soluciones generadas aleatoriamente. Al examinar los resultados, se analizaron las instancias generadas al azar y las de TSPLIB. Esto permite una evaluación integral de la eficacia de los algoritmos en el problema del vendedor viajero asimétrico.

En términos de tiempos de ejecución, se analizaron estadísticas detalladas, incluyendo el tiempo promedio de ejecución para cada algoritmo en cada instancia.

Para la ejecución de los distintos algoritmos en el archivo proyecto2.py se requieren de las siguientes librerías:

```
pip install pulp tsplib95 networkx numpy
```

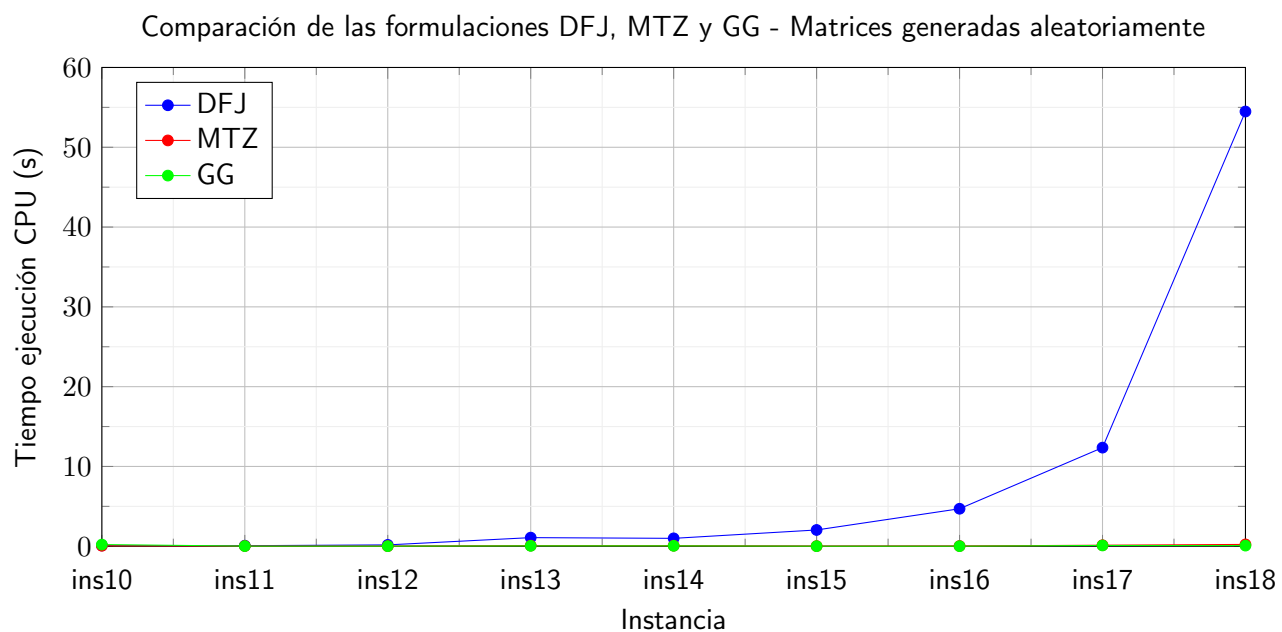


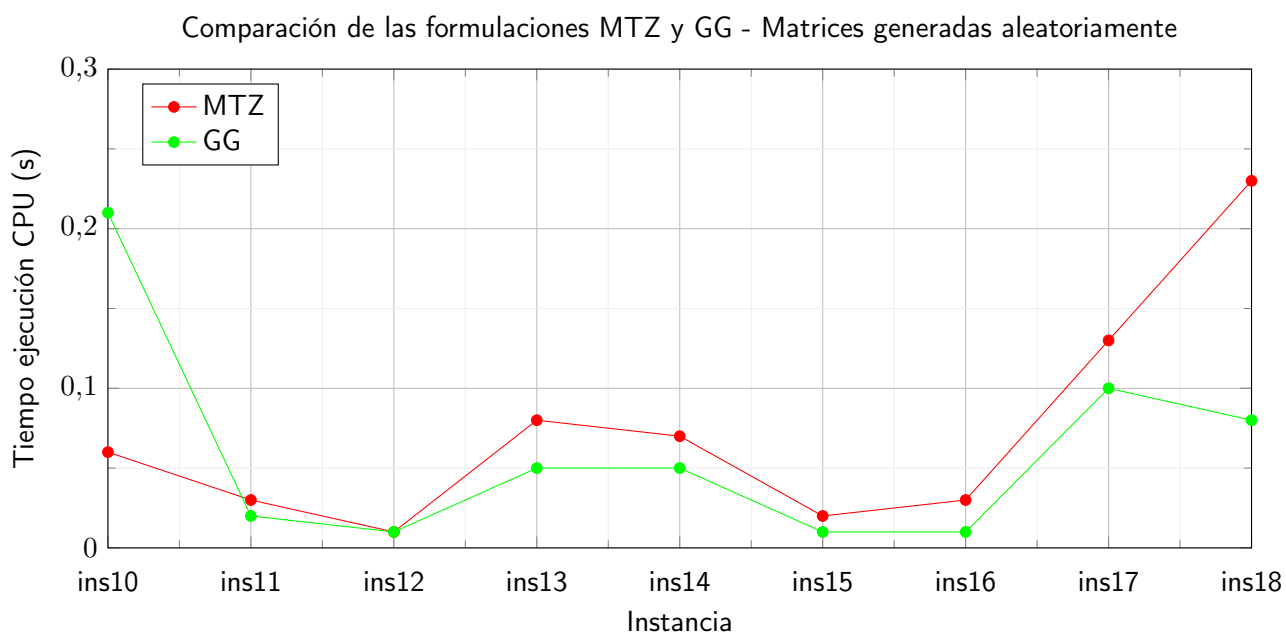
4. Resultados

4.1. Matrices generadas aleatoriamente

Instancia	DFJ (s)	MTZ (s)	GG (s)	Valor óptimo (u)
ins10	0.10	0.06	0.21	195
ins11	0.07	0.03	0.02	127
ins12	0.17	0.01	0.01	152
ins13	1.08	0.08	0.05	148
ins14	0.99	0.07	0.05	160
ins15	2.04	0.02	0.01	76
ins16	4.70	0.03	0.01	161
ins17	12.36	0.13	0.10	127
ins18	54.48	0.23	0.08	133

Cuadro 1: Tiempos de ejecución CPU para las tres formulaciones

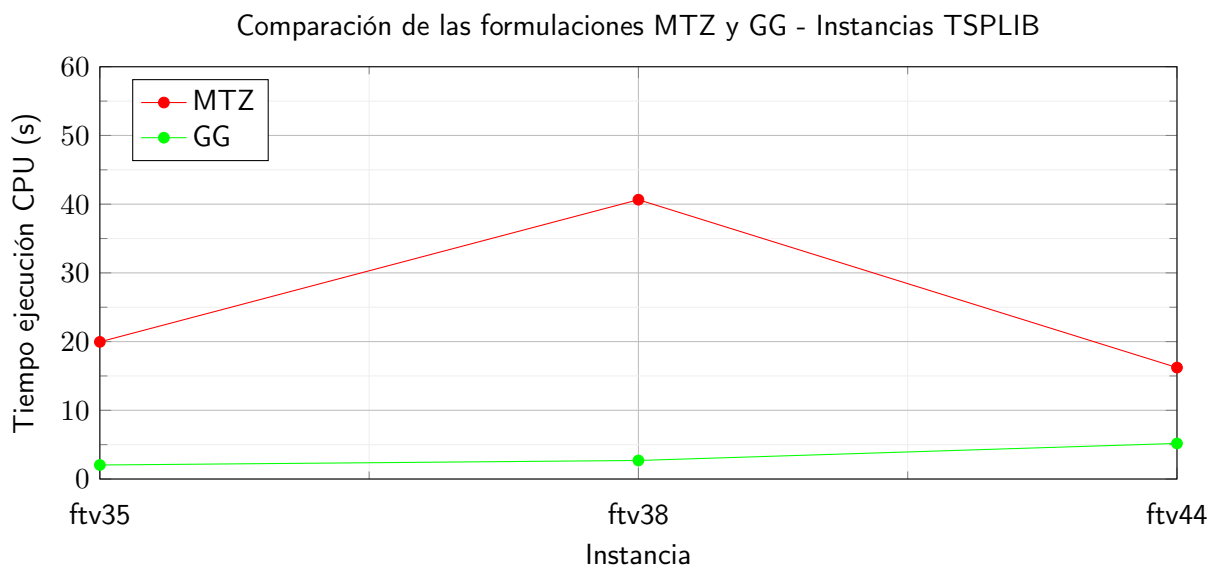




4.2. Instancias de TSPLIB

Instancia	MTZ (s)	GG (s)	Valor óptimo (u)
ftv35	19.96	2.04	1473
ftv38	40.65	2.70	1530
ftv44	16.22	5.18	1613

Cuadro 2: Tiempos de ejecución CPU para las formulaciones MTZ y GG





5. Conclusiones

En este informe se abordó el Problema del Vendedor Viajero Asimétrico (ATSP), usando el argumento de minimizar la ruta posible en la búsqueda de planetas potencialmente habitables para la humanidad mediante la implementación de tres algoritmos: MTZ, GG y DFJ. Cada uno de estos métodos fue aplicado a instancias generadas al azar, y se evaluaron sus desempeños en términos de tiempos de ejecución.

Los resultados indican que los tres algoritmos lograron encontrar soluciones óptimas para las instancias del ATSP consideradas. La ruta en la búsqueda de planetas habitables y la distancia total recorrida fueron determinadas de manera eficiente para cada método.

Se observó una consistencia en las soluciones obtenidas por MTZ, GG y DFJ en las instancias generadas al azar, puesto que no hubo variabilidad en las soluciones óptimas, lo que sugiere una robustez en la capacidad de estos algoritmos para abordar diferentes configuraciones del problema. Con respecto a los tiempos de ejecución, se vieron variaciones significativas entre los algoritmos, esto debido que MTZ demostró ser computacionalmente eficiente, proporcionando soluciones óptimas en un tiempo relativamente corto. GG también mostró un buen desempeño, y en la mayoría de situaciones fue el que menos se demoró, mientras que DFJ, aunque efectivo, requirió de mucho más tiempo para converger en algunas instancias de gran tamaño. Esto se debe a la restricción de eliminar los subcaminos que se pueden formar en el grafo. DFJ aplica la restricción para cada subcamino posible, por lo que la complejidad del algoritmo crece exponencialmente al número de nodos, mientras que MTZ y GG lo hacen en tiempo polinomial.

Finalmente mencionar la coherencia en la identificación de óptimos consistentes entre los algoritmos, a pesar de la generación de caminos distintos que conectan los mismos puntos, destaca la robustez de los métodos empleados en el problema del vendedor viajero asimétrico. Aunque convergen hacia soluciones de calidad igual, la variabilidad en los caminos evidencia la diversidad de rutas eficientes posibles. Esta dualidad entre consistencia en la calidad de la solución y variabilidad en los caminos revela la complejidad intrínseca del problema y subraya la capacidad de los algoritmos para explorar diversas configuraciones de ruta.