

Proyecto EPTP 2025-2

Marcel Silva
msilvan@usm.cl

1 Introducción

El Enhanced Profitable Tour Problem (EPTP) es un problema de optimización combinatoria, el cual es una generalización de Traveling Salesman Problem (TSP). A diferencia del TSP clásico, en el EPTP no es obligatorio visitar todos los nodos, sino únicamente aquellos que aporten beneficio neto al recorrido del usuario. Este problema surge en aplicaciones como planificación de rutas turísticas, logística de entregas, y plataformas de movilidad (e.g. Uber), donde no todos los puntos deben necesariamente ser visitados, sino aquellos que aumentan la rentabilidad global de la ruta.

2 Descripción del problema

Dado un grafo dirigido $G = (V, A)$ donde V es un conjunto de n nodos, y A es un conjunto de m arcos, el objetivo consiste en hallar un subconjunto de nodos de tal modo que maximice la **valorización neta** del recorrido. Cada nodo y arco tienen asociada una **valorización** y un **tiempo de uso** (tiempo de servicio en nodos, tiempo de viaje en arcos) que dependen del usuario que desea realizar el tour.

Además, para esta versión del problema, a cada nodo le corresponde una ventana de tiempo $[e_i, l_i]$, que indica el intervalo en el cual el nodo está disponible para ser visitado. Esto hace que nuestro problema sea más realista: por ejemplo, un turista debe visitar los atractivos en sus horarios de apertura, o un servicio de transporte como Uber debe realizar entregas en los horarios solicitados por el cliente.

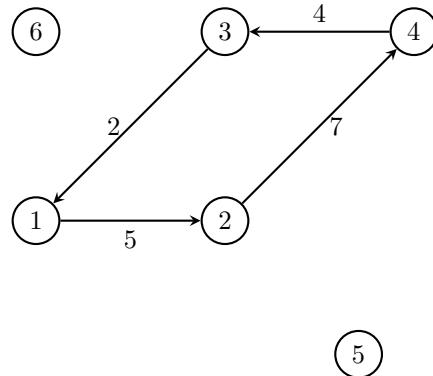


Figure 1: Representación gráfica simplificada del recorrido como un grafo dirigido. Notar que no necesariamente todos los nodos están conectados.

3 Parámetros y constantes

- $V = \{1, 2, \dots, n\}$: conjunto de nodos.
- $A \subseteq V \times V$: conjunto de arcos dirigidos.
- $U = \{1, 2, \dots, m\}$: conjunto de usuarios.
- t_i : tiempo de servicio en el nodo i , $i \in V$
- d_{ij} : tiempo de viaje asociado al arco $(i, j) \in A$.
- T_k : tiempo total para el usuario k , $k \in U$.
- c_{ijk} : valorización del arco (i, j) que le da el usuario k , $i, j \in V$, $k \in U$.
- s_{ik} : valorización del nodo i que le da el usuario k
- $[e_i, l_i]$: ventana de tiempo en la que el nodo i está disponible.

4 Variables

- $x_{ij} \in \{0, 1\}$: variable que indica si el arco (i, j) es parte del tour.
- $y_i \in \{0, 1\}$: variable que indica si el nodo i es visitado.

5 Restricciones

- El tour comienza y termina en el nodo 1
- El tour consiste en un subconjunto de nodos

- Cada nodo del tour es visitado a lo más una vez
- Todos los usuarios comienzan en un tiempo inicial $t = 0$.
- La duración total del recorrido no debe exceder el tiempo total de cada usuario
- Cada nodo i sólo puede visitarse dentro de su ventana de tiempo $[e_i, l_i]$. Si se arriba antes de e_i , se espera; si se arriba después de l_i , el nodo no puede visitarse.

6 Instancias

Las instancias requieren de dos archivos. Los archivos tipo 1 contienen información acerca de todos los puntos del tour. Estos incluyen la cantidad de puntos, el tiempo en que debe estar el usuario en cada nodo, ventanas de tiempo y una matriz acerca del tiempo entre nodos. Por otro lado, los archivos tipo 2 detallan información sobre los usuarios que participan en el tour. Muestran qué valor le dan los usuarios a aquellos nodos y arcos.

6.1 Archivo tipo 1

- **Nombre del archivo:** `n_instancia.txt`, donde n es el número de nodos (atractivos/puntos de interés) de la instancia. Ejemplo: `5_instancia.txt` corresponde a una instancia con 5 nodos.
- **Formato:**
 - Línea 1: número de nodos n .
 - Línea 2: tiempos de permanencia de cada nodo t_i
 - Línea 3: ventana de tiempo de disponibilidad del nodo i ($[e_i, l_i]$), separados por espacios.
 - Matriz de distancias/tiempos de viaje d_{ij} , de tamaño $n \times n$. Cada entrada d_{ij} indica el costo o tiempo de viaje desde el nodo i hasta el nodo j .

```

5
2 3 1 4 2
0 10 2 8 5 12 3 15 7 20
0 9 7 3 6
5 0 4 8 7
1 4 0 3 5
9 8 4 0 5
3 7 2 5 0

```

Figure 2: Ejemplo de formato archivo 1

6.2 Archivo tipo 2

Este archivo es referente a los perfiles de usuarios, por lo que tiene los datos de tiempo disponible y valorizaciones de los atractivos.

- **Nombre del archivo:** `nus_NúmeroDeLaInstancia_instancia.txt`, donde n es el número de usuarios definidos para la instancia, y `NúmeroDeLaInstancia` es el archivo tipo 1 correspondiente a los perfiles de usuario.

A cada archivo de tipo 1 le corresponde un archivo de tipo 2. Ejemplo `2us_5_instancia.txt` corresponde a los datos de 2 usuarios para una instancia con 5 nodos.

- **Formato:**

- Línea 1: número de usuarios m .
- Para cada usuario $k \in \{1, 2, \dots, m\}$:
 - * Línea con el tiempo total disponible T_k .
 - * Línea con los beneficios de los nodos: s_{ik} , $i \in V$
 - * Una matriz de la valorización de los arcos: c_{ijk} , $i, j \in V$

```
2
30
8 2 0 5 6
0 2 4 0 5
8 0 1 1 3
5 2 0 7 9
1 4 7 0 2
8 6 3 6 0
40
5 3 10 1 4
0 5 3 6 3
4 0 1 2 5
2 4 0 7 2
9 3 2 0 8
5 4 5 1 0
```

Figure 3: Ejemplo de formato de archivo 2 usando 2 usuarios

7 Salida

Se debe generar un archivo txt llamado **sol_NumeroDeLaInstancia_instancia.txt**, donde **NumeroDeLaInstancia** corresponde al número de nodos del tour. Para cada uno de los casos de prueba (usuarios):

- Primera linea: Valor del Tour (suma de valoraciones de nodos y arcos pertenecientes al tour generado).
- Segunda linea : Tiempo total disponible y tiempo empleado en el tour
- Tercera linea: Nodos pertenecientes al tour

```

142
30 20
1 4 2 5 3
168
40 25
5 2 3

```

Figure 4: Ejemplo de salida usando 2 usuarios. El ejemplo de salida es sólo para fines de mostrar el formato, no es en ningún caso una solución real

8 Consideraciones

- En este proyecto se consideran múltiples usuarios, cada uno con un perfil de valorizaciones distinto. Sin embargo, el algoritmo no debe ejecutar todos los usuarios en paralelo, sino uno a la vez por usuario. Es decir, el algoritmo se ejecuta para cada perfil de usuario existente en archivo 2 en combinación con el correspondiente archivo 1.
- El tour siempre comienza y termina en el nodo 1.
- El tiempo empleado en el tour incluye tiempos de viaje, tiempos de servicio y tiempos de espera por ventanas de tiempo.
- Si un nodo no puede visitarse dentro de su ventana, debe descartarse.
- El servicio de un nodo debe iniciarse y completarse dentro de su ventana de tiempo $[e_i, l_i]$. Si el servicio no cabe entero en el intervalo, el nodo no puede visitarse. Por ejemplo, con una ventana [3:00, 5:00] y un servicio de 2 horas, el nodo solo es factible si el servicio comienza a las 3:00 y termina a las 5:00.
- Todos los usuarios empiezan con un tiempo inicial $t = 0$, y a medida que ocupan los servicios de los nodos, tiempo entre arcos, y tiempos de espera, este valor aumenta.
- Los tiempos de espera por ventanas cuentan en la duración total del tour y consumen el tiempo total disponible del usuario.

References

- [1] Kasi Viswanath Dasari, Venkatesh Pandiri, and Alok Singh. Multi-start heuristics for the profitable tour problem. *Swarm and Evolutionary Computation*, 64:100897, 2021.
- [2] Pengfei He, Wenchong Chen, Qinghua Wu, and Fengjun Xiao. Capacitated profitable tour problem with cross-docking. *Computers & Operations Research*, 181:107077, 2025.
- [3] Matthias Joest and Wolfgang Stille. A User-Aware Tour Proposal Framework using a Hybrid Optimization Approach. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*, McLean, Virginia, USA, 2002. ACM.
- [4] Gonzalo Lera-Romero and Juan José Miranda-Bront. A branch and cut algorithm for the time-dependent profitable tour problem with resource constraints. *European Journal of Operational Research*, 289(3):879–896, 2021.