

# Métodos escalables de resolución para el Dial-a-Ride Problem (DARP)

Sergio García Prado

24 de septiembre de 2020

# Introducción

“The Dial-a-Ride Problem (DARP) consists of designing vehicle routes and schedules for  $n$  users who specify pickup and delivery requests between origins and destinations. The aim is to plan a set of  $m$  minimum cost vehicle routes capable of accommodating as many users as possible, under a set of constraints. The most common example arises in door-to-door transportation for elderly or disabled people.”

— Jean-François Cordeau & Gilbert Laporte

# Formulación

Minimizar

$$\sum_{k \in K} \sum_{j \in V} \sum_{i \in V} c_{ij}^k x_{ij}^k$$

sujeto a

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1,$$

$$\forall i \in P$$

$$\sum_{i \in V} x_{0i}^k = 1,$$

$$\forall k \in K$$

$$\sum_{i \in V} x_{i,2n+1}^k = 1,$$

$$\forall k \in K$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{n+i,j}^k = 0,$$

$$\forall i \in P, \forall k \in K$$

$$\sum_{j \in V} x_{ji}^k - \sum_{j \in V} x_{ij}^k = 0,$$

$$\forall i \in P \cup D, \forall k \in K$$

$$u_i^k + d_i + t_{ij} - M_{ij}^k (1 - x_{ij}^k) \leq u_j^k,$$

$$\forall i, j \in V, \forall k \in K$$

$$w_i^k + q_j - W_{ij}^k (1 - x_{ij}^k) \leq w_j^k,$$

$$\forall i, j \in V, \forall k \in K$$

$$u_{2n+1}^k - u_0^k \leq T_k,$$

$$\forall k \in K$$

$$e_i \leq u_i^k,$$

$$\forall i \in V, \forall k \in K$$

$$u_i^k \leq l_i,$$

$$\forall i \in V, \forall k \in K$$

$$t_{i,n+i}^k \leq u_{n+i}^k - (u_i^k + d_i),$$

$$\forall i \in P, \forall k \in K$$

$$u_{n+i}^k - (u_i^k + d_i) \leq L_k,$$

$$\forall i \in P, \forall k \in K$$

$$\max\{0, q_i\} \leq w_i^k,$$

$$\forall i \in V, \forall k \in K$$

$$w_i^k \leq Q_k + \min\{0, q_i\},$$

$$\forall i \in V, \forall k \in K$$

$$x_{ij}^k \in \{0, 1\},$$

$$\forall i, j \in V, \forall k \in K$$

# Métodos de Resolución

## ► Exactos:

- *Branch and Bound*: Se basa en la resolución una versión relajada del problema para después añadir restricciones adicionales (valores binarios, etc.) a partir de un proceso de ramificación.

## ► Heurísticos:

- *Fuerza Bruta*: Mediante un proceso iterativo probar todas las posibles combinaciones de soluciones hasta llegar a la mejor solución.
- *Selección Aleatoria*: Generación iterativa de soluciones incrementales tomando decisiones aleatorias en cada paso.
- *Greedy*: Generación iterativa de soluciones incrementales tomando la mejor decisión en cada paso del algoritmo con la esperanza de que la solución final sea lo suficientemente buena.
- *Clarke and Wright*: Algoritmo clásico de generación de rutas, partiendo de una por viaje para después realizar combinaciones entre estas mientras sea posible.
- *Búsqueda Local*: Estrategia de mejora, dada una solución inicial, se trata de aplicar transformaciones sobre esta de tal manera que mejore respecto de una determinada métrica.
- *Redes Neuronales*: Apoyadas en la relación de cercanía entre nodos del grafo generado, en los últimos años se han propuesto métodos de resolución para problemas básicos mediante *SOM*.

# Métodos de Resolución

## ► Meta-Heurísticos:

- *GRASP*: Estrategia combinada basada en generación de soluciones *Greedy* para su posterior mejora mediante *Local Search*.
- *Simulated Annealing*: Para tratar de reducir la problemática de caer en mínimos locales, en las primeras etapas de ejecución del algoritmo se aceptan ciertas soluciones "contraproducentes" para después hacer más estricto dicho criterio conforme aumentan las iteraciones.
- *Tabu Search*: Se permite alcanzar soluciones peores que la actual, siempre y cuando estas no se hayan visitado previamente (para ello se almacena la secuencia de últimas soluciones generadas).
- *Variable Neighborhood Search*: En cada iteración de la estrategia se utiliza una heurística diferente con la esperanza de que la combinación de ambas permita alcanzar resultados mejores.
- *Large Neighborhood Search*: La generación de soluciones iniciales se basa en la recombinación y/o fusión de las soluciones obtenidas en anteriores iteraciones del algoritmo.

# Implementación y Resultados (I)

```
import jinete as jit

solver = jit.Solver(
    loader_kwargs={"file_path": file_path},
    algorithm=jit.GraspAlgorithm,
    algorithm_kwargs={
        "first_solution_kwargs": {
            "episodes": 1,
            "randomized_size": 2,
        },
        "episodes": 5,
    },
    storer=jit.StorerSet,
    storer_kwargs={
        "storer_cls_set": {
            jit.PromptStorer,
            partialjit.GraphPlotStorer, file_path=file_path.parent / f"{file_path.name}.png"),
            partial(jit.FileStorer, file_path=file_path.parent / f"{file_path.name}.output"),
        }
    },
)

result = solver.solve()
```

## Implementación y Resultados (II)

	<b>R</b>	<b>A</b>	<b>B</b>
Finalización	20.0 %	38.09 %	42.85 %

**Tabla:** Indicadores relativos al *porcentaje de finalización* antes de las dos horas obtenido durante el experimento.

	<b>R</b>	<b>A</b>	<b>B</b>
Promedio	50.18 %	14.55 %	10.34 %
Desviación Estándar	29.08	10.68	10.02

**Tabla:** Indicadores relativos a la *diferencia de costes* respecto del mejor valor conocido obtenido durante el experimento.

	<b>R</b>	<b>A</b>	<b>B</b>
Promedio	94.85 %	99.24 %	98.82 %
Desviación Estándar	6.43	1.62	1.79

**Tabla:** Indicadores relativos al *nivel de servicio* obtenido durante el experimento.

# Conclusiones y Próximos Pasos

## ► *Conclusiones:*

- El estudio del problema *Dial-a-Ride* representa un reto interesante, tanto por su complejidad desde el punto de vista teórico como del punto de vista de la implementación de métodos de resolución.
- La definición de una buena metodología de trabajo en proyectos científicos es de vital importancia para el correcto desarrollo de los mismos.

## ► *Próximos Pasos:*

- Profundización en el análisis y posible implementación de los métodos de resolución que representan el actual *estado del arte* en la materia.
- Mejora de la implementación llevada a cabo, aumentando su escalabilidad de tal manera que sea utilizable en instancias de tamaño más realista.



¿Preguntas?