

# Resolución del TSP mediante PDDL

Marco Quintana García<sup>[UO287872]</sup>, Martín Cancio Barrera<sup>[UO287561]</sup>, and  
Miguel Méndez Murias<sup>[UO287687]</sup>

Sistemas Inteligentes. Grado en Ingeniería Informática. EII. Universidad de Oviedo.  
Campus de los Catalanes. Oviedo

## 1 Introducción

En este trabajo, se aborda la resolución del TSP utilizando el lenguaje de definición de dominios de planificación (*Planning Domain Definition Language*, PDDL) en su versión 2.1. Se ha diseñado un dominio específico y dos instancias de problemas, cuya resolución se llevó a cabo mediante el planificador LPG<sup>1</sup>.

El dominio incluye tres predicados: uno para identificar cuál es la ciudad inicial del recorrido, otro que determina cuál es la ciudad actual en la que se encuentra el viajero en cada momento, y un tercer predicado que permite registrar si una ciudad ya ha sido visitada o no. Además, se definen dos fuentes: uno que representa el coste asociado al trayecto entre dos ciudades específicas y otro que calcula el coste total acumulado del recorrido.

Para construir un camino circular que recorra todas las ciudades, se definen dos acciones: la primera permite desplazarse entre dos ciudades, marcando la ciudad de destino como visitada y sumando el coste del trayecto al total, mientras que la segunda permite completar el recorrido circular al conectar la última ciudad visitada con la ciudad inicial, por lo que esta acción solo se ejecuta cuando todas las ciudades ya han sido visitadas.

En la definición de cada uno de los problemas, se indican las ciudades y los costes de los trayectos entre cada par de ellas, siendo el objetivo final que todas las ciudades estén visitadas y que la ciudad actual cuando termina la ejecución sea la inicial. Además, se indica que la métrica de optimización consiste en minimizar el valor de la función *total-travel-cost-moved*, que representa el coste acumulado de los desplazamientos realizados a lo largo del recorrido.

## 2 Capturas de ejecución

En la captura de la Figura 1 se muestra el resultado de la ejecución con 17 ciudades. Se puede observar que se visitan todas las ciudades una sola vez, iniciando en *c0*, que también es la última ciudad que se visita cuando se ejecuta la acción *visit-and-end*. Además, también se puede ver que la ciudad destino de un paso siempre coincide con la ciudad inicial del siguiente, y hay exactamente el mismo número de pasos que de ciudades (16 para visitar todas las ciudades y uno extra para volver a la inicial). En cuanto al valor de la métrica, este es de

---

<sup>1</sup> <https://lpg.unibs.it/lpg/>

3992 y representa el coste de la solución, el cuál es bastante superior al coste del camino óptimo (2085), pero el tiempo en el que se obtuvo una solución fue muy bajo (0.07 segundos).

```

; Version LPG-td-1.4
; Seed 99811560
; Command line: ./LPG-td-1.4/lpg-td -o TSP/domain.pddl -f TSP/p17.pddl -n 1
; Problem TSP
; Actions having STRIPS duration
; Time 0.07
; Search time 0.07
; Parsing time 0.00
; Mutex time 0.00
; MetricValue 3992.00

0: (VISIT C0 C2) [1]
1: (VISIT C2 C12) [1]
2: (VISIT C13 C12) [1]
3: (VISIT C12 C6) [1]
4: (VISIT C6 C10) [1]
5: (VISIT C10 C11) [1]
6: (VISIT C11 C15) [1]
7: (VISIT C15 C3) [1]
8: (VISIT C3 C16) [1]
9: (VISIT C16 C5) [1]
10: (VISIT C5 C4) [1]
11: (VISIT C4 C14) [1]
12: (VISIT C14 C7) [1]
13: (VISIT C7 C8) [1]
14: (VISIT C8 C1) [1]
15: (VISIT C1 C9) [1]
16: (VISIT-AND-END C9 C0) [1]

```

**Fig. 1.** Ejecución con 17 ciudades

```

; Version LPG-td-1.4
; Seed 121912799
; Command line: ./LPG-td-1.4/lpg-td -o TSP/domain.pddl -f TSP/p21.pddl -n 1
; Problem TSP
; Actions having STRIPS duration
; Time 0.03
; Search time 0.03
; Parsing time 0.00
; Mutex time 0.00
; MetricValue 6226.00

0: (VISIT C0 C1) [1]
1: (VISIT C1 C20) [1]
2: (VISIT C20 C12) [1]
3: (VISIT C12 C9) [1]
4: (VISIT C9 C11) [1]
5: (VISIT C11 C3) [1]
6: (VISIT C3 C10) [1]
7: (VISIT C10 C16) [1]
8: (VISIT C16 C6) [1]
9: (VISIT C6 C19) [1]
10: (VISIT C19 C5) [1]
11: (VISIT C5 C7) [1]
12: (VISIT C7 C15) [1]
13: (VISIT C15 C4) [1]
14: (VISIT C4 C14) [1]
15: (VISIT C14 C17) [1]
16: (VISIT C17 C18) [1]
17: (VISIT C18 C13) [1]
18: (VISIT C13 C8) [1]
19: (VISIT C8 C2) [1]
20: (VISIT-AND-END C2 C0) [1]

```

**Fig. 2.** Ejecución con 21 ciudades

En la captura de la Figura 2 se muestra el resultado de la ejecución con 21 ciudades. En este caso, y al igual que en el anterior, se cumple que hay el mismo número de pasos que de ciudades y también se ve que se visitan todas ellas, siendo la ciudad final de un paso la inicial del paso siguiente. Respecto al valor de la métrica (6226), este también es bastante superior al coste del camino

óptimo (2707), aunque el tiempo en el que encuentra una solución sigue siendo extremadamente bajo (0.03 segundos) a pesar del tamaño del problema.

### 3 Conclusiones

La resolución del TSP utilizando PDDL 2.1 ha demostrado que este lenguaje es adecuado para modelar problemas de planificación que requieren una representación precisa de acciones, predicados y métricas numéricas. En este trabajo, la estructura del dominio y las instancias se definieron de manera clara y consistente, lo que permitió al planificador LPG generar planes tratando de minimizar el coste de la solución.

El uso de LPG evidenció que los planificadores basados en PDDL pueden ser herramientas útiles para problemas como el TSP, logrando resultados razonables en un tiempo bajo. Sin embargo, al comparar este enfoque con algoritmos como el A\* y los algoritmos genéticos, se observó que estas alternativas ofrecen un desempeño significativamente superior. Tomando como referencia el problema con 17 ciudades, el algoritmo A\* con el heurístico BI (Best Insertion) logró llegar a un coste de 2090 en 0.068 segundos, siendo 2085 el coste óptimo. Por su parte, el programa PDDL halló un coste de 3992 en 0.07 segundos. Para el problema de 21 ciudades, el coste usando A\* es de 2943 en 0.177 segundos frente a 6226 en 0.03 segundos usando PDDL, siendo 2707 el coste óptimo.

En general, PDDL es una herramienta valiosa en el ámbito de la planificación automatizada, pero su aplicabilidad depende del contexto. En el caso concreto del TSP, hemos podido comprobar que no es la mejor herramienta para resolver problemas de este tipo, ni en términos de coste ni de eficiencia.

### 4 Opinión personal

Consideramos que PDDL es una herramienta muy útil en el ámbito de la inteligencia artificial, especialmente para problemas de planificación bien definidos y con restricciones claras. La capacidad de expresar acciones, predicados y objetivos de manera explícita proporciona un marco sólido para resolver problemas de planificación y scheduling. Sin embargo, en el caso concreto del TSP, optaríamos por utilizar A\* con un buen heurístico, o bien un algoritmo genético, ya que ofrecen un desempeño claramente superior.

En general, la experiencia de trabajar con PDDL ha sido positiva y nos ha permitido comprender mejor su aplicabilidad y limitaciones. Aunque no lo consideraríamos nuestra primera opción para resolver problemas como el TSP, valoramos su utilidad como herramienta académica y su relevancia en multitud de problemas de optimización y scheduling.