# Práctica 3: Planificación Clásica (PDDL)

## Curso 2023-24
## Tercer Curso de Ingeniería Informática

**José Antonio Zamora Reyes**

## Técnicas de los Sistemas Inteligentes


UNIVERSIDAD
DE GRANADA

**1. Tabla de resultados . Se pide entregar una tabla de resultados donde se especifique, para cada ejercicio, el número de acciones del plan encontrado y el tiempo invertido por MetricFF para encontrar dicho plan para cada ejercicio. Se deben comentar, valorar y contextualizar los resultados mostrados en la tabla. Por ejemplo, ¿cómo evolucionan el tiempo de resolución y el número de estados evaluados para cada ejercicio? ¿Cuáles han sido las claves fundamentales a nivel de implementación (precondiciones o postcondiciones en acciones, predicados) en cada ejercicio?**

| | EJ 1 | EJ 2 | EJ 3 | EJ 4 | EJ 5 | EJ 6 |
|---|---|---|---|---|---|---|
| **Tiempo** | **0.00s** | **0.00s** | **0.03s** | **0.03s** | **15.04s** | **90 min** |
| **Acciones** | **4** | **11** | **17** | **32** | **52** | **52** |

No consigo optimizar más el problema , lo que me ocasiona tiempos gigantes.
Se puede observar como los tiempos aumentan en proporción a la complejidad del problema, llegando como vemos en el ejercicio 6 a tiempos aproximados a 90 minutos , el número de estado evaluados también aumentan considerablemente.
Como conclusión podemos sacar que la optimización del código es clave para poder abordar cierto tipo de problemas , ya que vemos que se puede dar situaciones en el que el número de estados a valorar sea muy grande e innecesario , lo que nos produce grandes tiempos de ejecución.

Ejercicio 1:
En el ejercicio el objetivo era conseguir que un determinado operador consiguiera ser asignado a un recurso , para ello he hecho uso de diferentes predicados para saber la posición de la unidad , para saber si existe un camino entre localizaciones , para determinar a qué tipo de unidad pertenece una unidad en concreto(ejemplo operador1 es operador) , para saber si una unidad(operador) está extrayendo un recurso…etc , todos estos predicados son necesarios para realizar las acciones de “Navegar” y “Asignar” las cuales requiere el ejercicio.
La acción navegar consiste básicamente en que una unidad pase de una localización a otra, para ello debemos garantizar que existe un camino entre el origen y el destino y que dicha unidad que se quiere mover no está extrayendo ningún recurso , como consecuencia de esta acción la posición de la unidad cambiará.
La acción Asignar consiste básicamente en poner a un determinado operador a extraer un recurso , es decir , asignar un recurso , para ello se deben cumplir las precondiciones de que este operador no esté extrayendo otro recurso y que el operador se encuentre en la localización donde se encuentra el recurso, como consecuencia de esta acción , el operador quedará asignado e inmovil extrayendo el recurso.

Ejercicio 2:
A diferencia del ejercicio 1 , este ejercicio tiene una nueva acción que es construir, para llevar a cabo esta acción debemos incluir algunos predicados para saber si tengo un

determinado recurso , para saber si el edificio se ha construido y para saber qué recurso necesita el edificio para ser construido .
Al usar el predicado que nos dice si tengo un recurso o no ( Tengo ?recurso ) , tengo que modificar la acción asignar del ejercicio 1 , para que cuando asigne un operador a un recurso , nos indique mediante este predicado que ese determinado recurso lo tenemos , este predicado será utili par evitar tener que hacer bucles es otras acciones , lo que nos ocasiona una mejor eficiencia.
En cuanto a la acción construir , tenemos de precondiciones que la unidad que construye sea un operador y que este no esté extrayendo ningún recurso , también debemos de tener el recurso necesario para realizar la construcción y que el operador que construye este en la localización donde queremos construir , como efecto de esta acción , el edificio quedará construido en la localización correspondiente.

Ejercicio 3:
En este ejercicio debemos de tener en cuenta que un edificio puede necesitar más de un recurso para ser construido y que no se pueden construir dos edificios en la misma localización, como consecuencia de esto , la acción construir deberá ser modificada.
Deberemos de añadir a la anterior acción construir algunas modificaciones , como recorrer todos recursos debido a que el edificio puede necesitar más de uno y garantizar que no hay edificios construidos donde deseamos construir el nuevo edificio.

Ejercicio 4:

En este ejercicio se añade una nueva acción “Reclutar” , que como su nombre indica consiste en reclutar a una unidad , para ello debemos hacer uso de nuevos predicado entre ellos uno que nos diga qué en qué edificio recluta un tipo de unidad y otro que nos indique qué recursos necesita la unidad para reclutar .
La nueva acción Reclutar tendrá como precondiciones que el edificio en el que se quiere reclutar , esté construido y posicionado en la localización en la que vamos a reclutar la unidad , debemos garantizar también que la unidad a reclutar no está ya creada , también debemos garantizar que el edificio que pasamos por argumento a la función sea del tipo de edificios del cual la unidad requiere para poder ser reclutada y por último debemos garantizar que tenemos todos los recursos necesarios para reclutar la unidad .
Como efecto de la acción la unidad quedará creada y en la posición determinada.

Ejercicio 5:

En este ejercicio necesitamos una nueva acción que es Investigar , la cual debemos implementar , también existe un nuevo edificio que es Laboratorio que es donde se hacen las investigaciones .
A parte de añadir la acción Investigar , debemos de modificara la acción de reclutar debido a que los soldados no podrán ser creados hasta que se haya realizado la investigación “InvestigarSoldado” , también hay que modificar la acción construir debido a que el edificio Teletransportador solo puede ser construido cuando la investigación del teletransporte se ha realizado .
Para realizar la acción Investigar , necesito añadir un predicado que me indique si se ha realizado una investigación , otro que diga los recursos que necesita un tipo de

investigación y otro para identificar a qué tipo de investigación pertenece una investigación concreta.

La acción investigar , tendrá de precondiciones que el edificio donde se va a investigar , sea un Laboratorio , que este esté construido y que se tengan los recursos necesarios para realizar la investigación , como resultado la investigación se realizará y se marcará a partir del predicado, que tenemos realizada esa investigación en concreto.

NOTA: En el ejercicio 5 acciones como construir y reclutar se ven modificadas con respectos los ejercicios anteriores , debido a que me di cuenta que necesitaba mejorar la eficiencia.

Ejercicio 6:

Este ejercicio es básicamente igual que el cinco solo que en este vamos a garantizar una ruta óptima , para ello añadimos la función coste. En el dominio para cada una de las acciones indicamos que cuando cualquiera de estas se ejecute , el coste incrementa en 1 . En el problema el objetivo será que el coste sea menor que 52 , de esta manera garantizamos el camino óptimo.

Nota : Este ejercicio está realizado , pero el tiempo que tarda es grandísimo. No he conseguido optimizar más el problema.

**2. En las distintas llamadas a MetricFF necesarias para resolver el Ejercicio 6 (o cualquier otro en donde se busque optimizar de modo efectivo el tamaño del plan o algún otro criterio), ¿MetricFF suele tardar aproximadamente el mismo tiempo en todas ellas?**
**¿A qué cree que se debe este fenómeno? Si en un ejercicio comparamos el tiempo de resolución minimizando una métrica y sin minimizarla. ¿Debería tardar lo mismo el planificador en proponer un plan? Razone su respuesta.**

Está claro que no , el ejemplo lo vemos en mi caso entre el ejercicio 5 y 6 , si compramos la resolución de problemas minimizando una métrica y sin minimizar , generalmente no debe tardar lo mismo . Cuando minimizas una métrica añadimos una complejidad más al problema lo que provoca tiempos de ejecución mayores. Sin embargo , la variabilidad de los tiempos de ejecución no solo depende de la complejidad del problema , sino que también depende de la eficiencia de las heurísticas empleadas por Metric.
En resumen los tiempos de ejecución bajos pueden ocurrir o bien porque el problema no sea tan complejo o bien cuando las heurśiticas sean muy eficientes.

**3. Revise la bibliografía recomendada, así como la literatura científica que considere relevante, y profundice un poco en la complejidad computacional de la planificación automática. ¿Cómo considera que escalaría la resolución de los problemas a medida que incorporásemos más acciones, objetos y nodos del mapa? De hecho, a modo de sencillo experimento, pártase del Ejercicio 4 e incorpórense, progresivamente, más operadores a la definición del problema: (unidadEs Operador4 Operador) ¿Cómo evolucionan los tiempos del Ejercicio 4 al realizar este sencillo cambio? ¿Cuánto tarda el planificador en encontrar un plan si declaramos, por ejemplo, 7 posibles operadores?**

Pienso que medida que en un problema incorporamos más acciones , objetos y nodos en el mapa , los tiempos de ejecución incrementarían significativamente , para corroborar esto hacemos uso del ejercicio 4 , añadiremos diferentes operadores y nos damos cuenta que los tiempos de ejecución son bastante mayores , si probamos con 7 operadores el problema 4 tendría un tiempo de ejecución bastante alto.
En el ejercicio 4 por defecto , el tiempo de ejecución es de 0.03 segundos , mientras que cuando añadimos 7 operadores , el tiempo de ejecución se dispara ( Paro la ejecución debido a que ya he visto que el tiempo de ejecución aumenta).
Este incremento del tiempo de ejecución cuando se añaden nuevos operadores , objetos o nodos de mapa , se debe a un incremento exponencial del espacio de búsqueda. Este incremento se debe a la mayor cantidad de combinaciones posibles que el planificador debe explorar para encontrar un plan óptimo.