Minería de Procesos - 2024-2025

Universidad de Granada

Análisis Bibliográfico

Miguel García López

Índice

1. Introducción

En el presente trabajo se realiza un análisis detallado del artículo titulado *Learning to select goals in Automated Planning with Deep-Q Learning*, publicado en la revista *Expert Systems with Applications* (2022).

Para ello, se responderán una serie de preguntas orientadas a identificar el problema abordado en el trabajo, su importancia dentro del contexto académico y práctico, las técnicas utilizadas para resolver dicho problema, otras posibles aproximaciones existentes, y las ventajas comparativas de la propuesta de los autores. Además, se establecerá una relación entre los contenidos técnicos descritos en el artículo y los temas tratados en la asignatura correspondiente.

2. Preguntas a responder

2.1. Realizar un resumen basado en respuestas a las siguientes preguntas

2.1.1. ¿Qué problema se aborda en el documento?

En el documento se explica un nuevo método para el campo de *Automated Planning* (**AP**) o planificación automática. Este es un sub-campo de la inteligencia artificial dedicado a obtener una planificación orientada a un objetivo concreto con limitaciones dentro de un dominio. Normalmente se considera un *input* de un dominio, un estado inicial y un objetivo. El agente en cuestión debe devolver una serie de acciones que resuelvan el problema dado ese tipo de *input*.

Concretamente, este nuevo método intenta mejorar los tiempos de ejecución, consecuencia limitante de modelos obtenidos por algoritmos clásicos de **AP**. Por ello se propone un método que combine parte de conocimiento del campo del *Reinforcement Learning* (**RL**) y de **AP**. Los algoritmos de **RL** suelen ser mucho más rápidos, ya que se entrenan una vez y luego actúan sin necesidad de planificar previamente, aunque suelen generalizar peor en nuevos dominios y requieren muchos datos.

En el documento, por tanto, se propone un método que combina las bondades de ambos sub-campos para intentar aplicar agentes planificadores en soluciones en tiempo real y que sean más generalizables entre problemas de distintos dominios.

2.1.2. ¿Por qué es relevante este problema?

El problema abordado en el documento es altamente relevante debido a las limitaciones prácticas de los enfoques tradicionales de planificación automática (**AP**) y a la necesidad de soluciones eficientes y generalizables en escenarios de tiempo real, como los videojuegos o sistemas robóticos.

Por un lado, los algoritmos clásicos de **AP**, aunque efectivos en cuanto a calidad de planes, resultan ser ineficientes en escenarios con restricciones temporales debido al crecimiento exponencial del espacio de búsqueda a medida que aumenta la complejidad del problema. Esto impide su integración directa en arquitecturas de ejecución en línea, donde el agente debe planificar y actuar en tiempo real.

Por otro lado, los algoritmos de **RL**, si bien actúan de forma muy rápida una vez entrenados y no requieren conocimientos explícitos del dominio, suelen necesitar enormes cantidades de datos para aprender políticas efectivas y tienden a generalizar mal ante nuevos problemas, incluso dentro del mismo dominio.

Dado que ambos enfoques presentan ventajas y desventajas complementarias, se hace evidente la necesidad de una solución que combine lo mejor de ambos mundos. La propuesta del documento, que integra AP con *Deep Q-Learning* para la selección de subobjetivos, representa un avance significativo al permitir una planificación más eficiente, generalizable y viable en tiempo real.

2.1.3. ¿Qué técnicas se emplean para solucionar este problema?

Se formula la toma de decisiones como un **MDP** especial, concretamente un *Deterministic Markov Decision Proces* (**DMDP**), que permite la selección de sub-objetivos. Esto permite tomar acciones de forma secuencial como una selección dentro de sub-metas, lo que mejora la generalización entre problemas y reduce drásticamente el espacio de acción.

Se define el proceso de toma de decisiones como $M_g = (S_g, G_s, r_g, t_g)$, donde:

- $S_g \subset S$: espacio de estados reducido a aquellos donde se toman decisiones de subobjetivos.
- G_s: conjunto de subobjetivos elegibles en el estado s, incluyendo siempre el objetivo final g_f.
- $r_g(s, g)$: función de recompensa que devuelve la longitud del plan si g es alcanzable, o una penalización λ si no lo es.
- $t_g(s, g)$: función de transición determinista que devuelve el estado final tras ejecutar el plan hacia g desde s.

Se utiliza además una red neuronal para el módulo de selección de metas, que estima los **Q-valores** $Q(s, g)$ que representan la suma de longitud del plan hacia el sub-objetivo g desde el estado actual s y la longitud del plan desde el sub-objetivo hasta el objetivo final.

Esta red neuronal es una *CNN*, una red convolucional, donde los tensores de entrada codifican información sobre el estado del entorno y la posición del subobjetivo.

Para entrenar esta red se compone una función de pérdida adaptada para **DMDP** y se utilizan algoritmos especiales como *Double Q-Learning*, para reducir sobre-estimaciones, *Fixed Q-targets*, como algoritmo de estabilización, y *Prioritized Experience Replay*, para dar más peso a muestras que sean más informativas.

Todo ello (de la parte de **RL**) se integra con un planificador clásico *Fast-Forward*. La red neuronal no es la que genera directamente los planes, sino que selecciona sub-metas en relación a lo que considere más prometedor. Este *output* se codifica como *PDDL* y se inyecta en el planificador **FF** para su posterior resolución usando una búsqueda *best-first search*.

2.1.4. ¿Hay otras aproximaciones al mismo problema con otras técnicas? ¿En qué consisten?

Existen otras tantas técnicas que intentan solucionar el problema de la planificación automática

La planificación **HTN** (*Hierarchical Task Network*) [1] modela la solución de problemas como la descomposición de tareas complejas en tareas más simples (sub-tareas). Esto imita cómo los humanos abordan problemas grandes: desglosándolos en pasos manejables. La mejora sustancial que incorpora es la reducción del espacio de búsqueda junto a una mejor coordinación entre agentes.

También existe la planificación basada en *landmarks*. Un *landmark* es un hecho (o condición) que debe ocurrir en todo plan válido. Identificarlos permite guiar la búsqueda del planificador hacia metas intermedias críticas. En el paper en [2] proponen **HIGL**, que aprende a identificar *landmarks* como sub-objetivos útiles durante la exploración. Su principal ventaja es que mejora la exploración en entornos con retroalimentación escasa (recompensas difíciles de obtener).

2.1.5. ¿Qué aportan/qué ventajas tienen las técnicas descritas en el trabajo frente a las otras aproximaciones?

La técnica propuesta en el artículo aporta una combinación novedosa de planificación clásica (**AP**) con aprendizaje por refuerzo profundo (**RL**), aprovechando las fortalezas

de ambos enfoques mientras mitiga sus debilidades.

Frente a métodos como la planificación jerárquica (**HTN**), que requieren una estructura rígida y conocimiento explícito del dominio para descomponer tareas, la propuesta permite una mayor flexibilidad al aprender sub-objetivos automáticamente a partir de la experiencia. Además, mientras **HTN** puede presentar dificultades para adaptarse a dominios nuevos sin redefinir sus jerarquías, el enfoque con **Deep Q-Learning** facilita la transferencia entre dominios similares mediante aprendizaje.

En relación con los métodos basados en *landmarks*, aunque estos guían la búsqueda mediante hechos clave, su identificación puede ser costosa computacionalmente y, en muchos casos, requiere pre-procesamiento o heurísticas específicas. En contraste, la propuesta de este artículo integra la selección de sub-metas directamente en la dinámica de aprendizaje, lo que le permite actuar en tiempo real y ajustar las estrategias según el entorno sin necesidad de reconfiguración manual.

La ventaja más significativa de la técnica descrita reside en su capacidad de combinar rapidez de decisión (propia del **RL**) con una alta calidad de planes (proporcionada por un planificador clásico). Además, al estructurar la decisión como un **DMDP** y utilizar un estimador de valores $Q(s, g)$ que incorpora la longitud del plan y su factibilidad, se logra una exploración más informada del espacio de soluciones. Esto no solo mejora la eficiencia, sino también la generalización a nuevos problemas, algo en lo que los enfoques puramente basados en aprendizaje suelen fallar.

2.2. ¿Qué relación tienen las técnicas descritas en el trabajo con los contenidos de la asignatura?

Desde la perspectiva de la Minería de Procesos, la técnica presentada en el artículo se vincula directamente con el objetivo de ofrecer recomendaciones prescriptivas (determinar y proyectar el uso de las alternativas de acción y recursos disponibles) en entornos dinámicos. Al aprender a seleccionar sub-metas mediante aprendizaje por refuerzo, el sistema puede generar decisiones adaptativas en tiempo real, lo que complementa la lógica de los sistemas de soporte a la decisión basados en *logs* de eventos.

Así, esta propuesta representa una evolución hacia agentes que no solo analizan o predicen el comportamiento del proceso, sino que también aprenden a intervenir activamente para optimizar su ejecución, alineándose con los principios de recomendación y mejora continua propios de la minería de procesos.

2.3. Describe de forma resumida la principal aportación técnica del trabajo. Esta descripción debe estar apoyada por una figura (o varias si es necesario) en la que se utilice pseudocódigo (si procede y fuera necesario), o un diagrama funcional explicativo (realizado por tí y no pegado del documento)

La principal innovación de este trabajo es la arquitectura *Deep Q-Planning* (**DQP**), que fusiona de manera sinérgica el aprendizaje por refuerzo con la planificación automática clásica para resolver el problema de selección óptima de submetas en entornos complejos.

En lugar de usar **DQL** para aprender directamente políticas de acción, **DQP** emplea una **CNN** entrenada con **DQL** para predecir la longitud de planes asociados a cada submeta candidata. Esta predicción guía la selección de la submeta más prometedora, que luego es ejecutada por un planificador clásico (*Fast-Forward*) para obtener acciones concretas.

La **CNN** predice planes en milisegundos, lo que hace que la principal aportación del artículo sea la resolución del problema de la explosión combinatoria de sub-metas gracias a este módulo añadido.

En la figura 1 puede observarse como funciona el algoritmo híbrido. Se generan primero las sub-metas con el módulo de **DQL**, el cual selecciona una sub-meta (mucho más sencillo y rápido de resolver por el planificador que un plan completo) y después pasa a la generación tradicional de acciones. Es, en esencia, una forma *greedy* de solucionar el problema que añade una mejora cuantitativa al tiempo de ejecución.

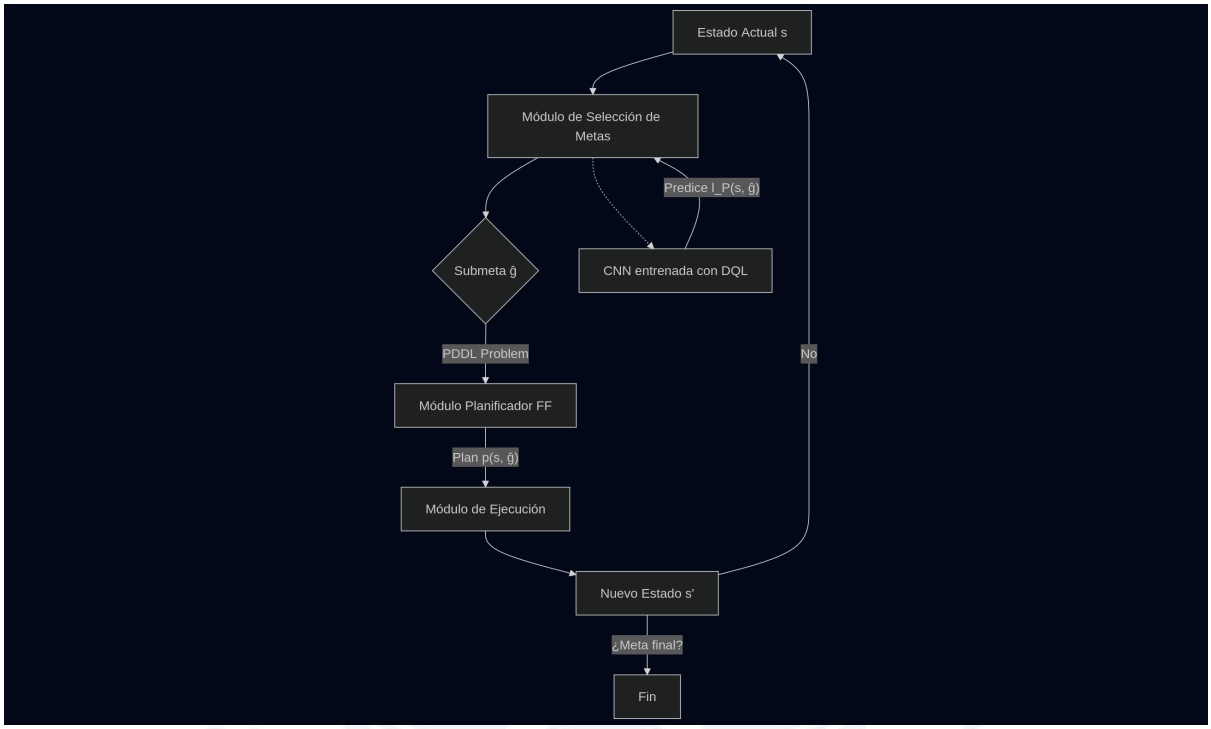

Figura 1: Diagrama Funcional de la Arquitectura **DQP**

2.4. Partiendo de un análisis de los experimentos mostrados en el trabajo (o de los ejemplos/casos de uso mostrados) y de la comprensión de los mismos que se muestran en el trabajo, así como de las conclusiones del trabajo. ¿Cómo de significativa es la solución aportada por los autores al problema?

La experimentación llevada a cabo en el artículo es exhaustiva y está diseñada para validar de manera rigurosa la eficacia del modelo propuesto **DQP**, desde múltiples perspectivas.

Se entrenaron versiones del modelo con $10, 25, 50, 100 y 200$ niveles de entrenamiento, lo que permite estudiar su escalabilidad y eficiencia en el uso de muestras. Los resultados muestran una clara mejora en la calidad de los planes (medida con el coeficiente de acción) conforme aumenta la cantidad de datos, con una reducción progresiva de la desviación estándar, lo que sugiere una generalización estable. Con 100 niveles ya se alcanza un rendimiento cercano al óptimo, usando solo 50000 muestras.

Además, el modelo híbrido se compara con múltiples versiones de **FF** y **DPQ** las supera a todas mejorando el tiempo de varios minutos o horas a solo 2 segundos. Hay que tener en cuenta también, que los planes son entre un 9 % a un 15 % más largos. Al

escoger sub-metas es más que posible que esté escogiendo caminos sub-óptimos en cuanto a número de acciones necesarias.

También se comparó con **DQL**, y a pesar de que el modelo **DQL** fue entrenado con diez veces más datos (un millón de muestras), solo logró resolver un nivel de los 11, con una tasa de éxito del 10 %. En contraste, **DQP** resolvió todos los niveles, mostrando que la combinación de planificación automática con selección de submetas basada en aprendizaje profundo mejora radicalmente la eficiencia y la generalización.

En conjunto, la experimentación demuestra de forma concluyente que la solución propuesta es significativamente superior en eficiencia computacional sin apenas contrapartes significativas. Por ello, considero que la solución aportada es más que significativa, ya que se ha aportado un nuevo método capaz de ser usado en tiempo real.

3. Bibliografía

[1] X. Mu, H. H. Zhuo, C. Chen, K. Zhang, C. Yu y J. Hao, *Hierarchical Task Network Planning for Facilitating Cooperative Multi-Agent Reinforcement Learning*, 2023. arXiv: 2306.08359 [cs.AI]. URL: https://arxiv.org/abs/2306.08359.

[2] J. Kim, Y. Seo y J. Shin, *Landmark-Guided Subgoal Generation in Hierarchical Reinforcement Learning*, 2021. arXiv: 2110.13625 [cs.LG]. URL: https://arxiv.org/abs/2110.13625.