



Deusto

Facultad de Ingeniería
Ingeniaritza Fakultatea

Grado en Ingeniería Informática

Informatikako Ingeniaritzako gradua

Proyecto fin de grado

Gradu amaierako proiektua

Aplicación de técnicas de reinforcement learning al
problema "flexible job-shop scheduling problem"

Asier Villar Marzo

Director: Iker Pastor Lopez

Bilbao, junio de 2023



Deusto

Facultad de Ingeniería
Ingeniaritza Fakultatea

Grado en Ingeniería Informática

Informatikako Ingeniaritzako gradua

Proyecto fin de grado

Gradu amaierako proiektua

Aplicación de técnicas de reinforcement learning al
problema "flexible job-shop scheduling problem"

Asier Villar Marzo

Director: Iker Pastor Lopez

Bilbao, junio de 2023

Contents

1	Introducción	1
1.1	Motivación	1
1.2	Explicación del problema	1
1.3	Estructura del documento	1
2	Antecedentes y justificación	3
2.1	Estado del arte	3
2.2	Antecedentes	3
3	Objetivos y alcance	4
3.1	Objetivos generales	4
3.2	Alcance	4
3.2.1	Dentro del alcance	4
3.2.2	Fuera del alcance	4
4	Metodología del Proyecto	5
4.1	Metodologías ágiles	5
4.2	Metodología de desarrollo	6
4.2.1	Scrum y MLOps	6
4.2.2	Elementos de Scrum	6
4.2.3	Principios de MLOps	7
4.2.4	Flujo de trabajo	7
4.3	Metodología de investigación	8
5	Conclusiones y trabajo a futuro	10
5.1	Conclusiones	10
5.2	Trabajo a futuro	10

List of Tables

List of Figures

4.1	Flujo de trabajo de una metodología ágil.	5
4.2	Arquitectura MLOps	6
4.3	Metodología MLOps	8

1. INTRODUCCIÓN

1.1. MOTIVACIÓN

1.2. EXPLICACIÓN DEL PROBLEMA

1.3. ESTRUCTURA DEL DOCUMENTO

En esta sección, se presenta la estructura del documento de forma clara y organizada. Se brinda una visión general de cómo se han organizado los diferentes capítulos y secciones para abordar de manera coherente y completa todos los aspectos relevantes del proyecto. Además, se proporciona una breve descripción de cada capítulo, destacando su contenido y su contribución al conjunto de la memoria. Esta sección permite al lector tener una guía clara sobre cómo está estructurado el documento y qué puede esperar encontrar en cada sección.

- **Introducción.** En este capítulo se presenta de forma breve el objetivo principal del proyecto, su impacto deseado y la motivación detrás de su realización. Además, se realiza una breve descripción del problema a resolver y se enumeran de manera ordenada los capítulos que componen el proyecto.
- **Antecedentes y justificación.** Se proporciona un estudio del estado del arte y las últimas tendencias, y se justifican las antecedentes existentes durante el desarrollo del proyecto.
- **Alcance y objetivos.** Se definen de manera detallada tanto el objetivo principal como los objetivos secundarios del proyecto. También se establece el alcance del proyecto, que se describe mediante una lista concisa de elementos que se encuentran dentro y fuera del proyecto.
- **Metodología.** Se describe la metodología de trabajo utilizada durante el desarrollo del proyecto, así como la metodología creada para la resolución del problema.
- **Memoria técnica.** Se explican en detalle todos los aspectos técnicos del mismo. Se incluyen la arquitectura del sistema integral, las herramientas utilizadas para el desarrollo, los requisitos del sistema y las incidencias encontradas entre otros.
- **Proceso de desarrollo.** En este capítulo se presenta el proceso de desarrollo utilizado en el proyecto. Se describe de manera detallada la metodología y las prácticas empleadas durante la resolución del problema. Proporciona una visión general del enfoque adoptado en el desarrollo del proyecto y cómo se aseguró la calidad y eficiencia en la implementación del sistema. También se discuten posibles limitaciones de los métodos y se proponen recomendaciones para investigaciones futuras.
- **Experimentación.** En este apartado se describe el proceso de experimentación llevado a cabo en el proyecto. Se detallan los experimentos realizados, las diferentes representaciones del problema, los datos recopilados y los resultados obtenidos. Además, se analizan e interpretan los resultados para sacar conclusiones relevantes y respaldar las decisiones tomadas en el proyecto.
- **Planificación y presupuesto.** Se detallan las fases y tareas del proyecto, se organizan cronológicamente indicando su duración. También se incluye un esquema de descomposición del trabajo y el plan de recursos humanos. Además, se incluyen los costes totales del proyecto, incluyendo los materiales y los recursos humanos.

- **Conclusiones y trabajo a futuro.** Se presentan las reflexiones realizadas tras la finalización del proyecto, así como las lecciones aprendidas y los conocimientos adquiridos. Además, se presentan ideas o propuestas que podrían ser utilizadas o implementadas en futuras investigaciones.
- **Abreviaturas, acrónimos y definiciones.** Se proporcionan explicaciones sobre el significado de ciertos términos, acrónimos o abreviaturas mencionadas en la memoria y que se consideran relevantes.
- **Bibliografía.** Se incluye una lista de referencias bibliográficas utilizadas durante el desarrollo de la memoria.
- **Anexos.** Se incluyen documentos independientes a la memoria del proyecto, pero considerados lo suficientemente relevantes como para ser adjuntados en documentos separados.
 - **Anexo I, Manual de usuario.** Se proporcionan las instrucciones necesarias para que cualquier usuario, independientemente de su nivel de conocimiento sobre el tema del proyecto, pueda poner en marcha el sistema inteligente y aprovechar todas sus funcionalidades.
 - **Anexo II, Dimensión ética del proyecto.** Se realiza un análisis ético del proyecto para garantizar que en su conjunto sea considerado éticamente aceptable y una contribución positiva para la sociedad.

2. ANTECEDENTES Y JUSTIFICACIÓN

2.1. ESTADO DEL ARTE

2.2. ANTECEDENTES

3. OBJETIVOS Y ALCANCE

En esta sección se introducen los objetivos en los que va consistir el proyecto, habiéndose realizado una división entre el principal y los secundarios. Además de ello, se presentan los elementos que forman el alcance, así como se comentarán otros que no forman parte de este.

3.1. OBJETIVOS GENERALES

3.2. ALCANCE

En esta sección se definen los límites del proyecto, estableciendo lo que está incluido y excluido dentro mismo. Se describirá de manera detallada las actividades que forman parte del desarrollo final, así como aquellos elementos que no están incluidos en el alcance del proyecto. Esta sección es fundamental para tener una comprensión clara y evitar posibles malentendidos o confusiones durante su ejecución.

3.2.1 Dentro del alcance

3.2.2 Fuera del alcance

4. METODOLOGÍA DEL PROYECTO

En esta sección, se describirá la metodología para abordar tanto el desarrollo del software como la propia metodología de investigación. La metodología de desarrollo de software se basa en el uso de metodologías ágiles [1], concretamente en la metodología Scrum [2]. Por otro lado, la metodología de investigación es una metodología propia diseñada para abordar problemas dentro del marco de trabajo de los problemas combinatorios NP-Hard diseñada por el autor de este trabajo. Esta metodología se basa en el uso de técnicas de IL.

4.1. METODOLOGÍAS ÁGILES

Las metodologías ágiles son un conjunto de metodologías de desarrollo de software que se caracterizan por ser iterativas e incrementales. Estas metodologías se basan en el desarrollo de software en ciclos cortos de tiempo, llamados iteraciones, en los que se desarrolla una parte del software. Estas iteraciones son incrementales, es decir, cada iteración añade una funcionalidad al software que se está desarrollando. De esta forma, se consigue que el software se desarrolle de forma incremental, y que se pueda ir probando y testeando a medida que se desarrolla. Esto permite que se puedan detectar errores de forma temprana y que se puedan corregir de forma rápida. Además, permite que el software se pueda adaptar a los cambios que se produzcan en los requisitos del proyecto.

A diferencia de las metodologías tradicionales o pesadas [3] (como por ejemplo el modelo en cascada), las metodologías ágiles no se basan en la planificación inicial del proyecto, sino que se basan en la adaptación constante, cualidad que es muy importante especialmente en proyectos de investigación, en los que la planificación inicial es muy difícil de realizar debido a la incertidumbre que existe en este tipo de proyectos y a los diversos factores externos que pueden afectar al proyecto.

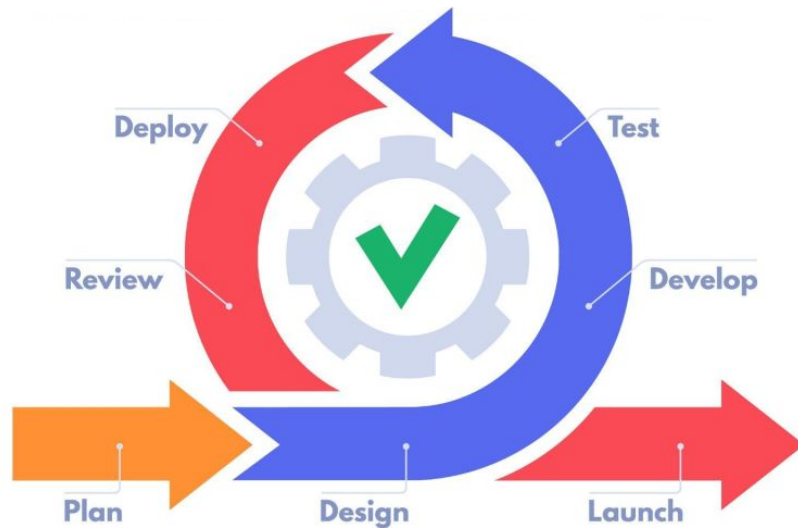


Figure 4.1: Flujo de trabajo de una metodología ágil.

4.2. METODOLOGÍA DE DESARROLLO

4.2.1 Scrum y MLOps

Machine Learning Operations (MLOps) [4] y Scrum [2] son dos enfoques diferentes pero complementarios que se pueden utilizar en proyectos de desarrollo de software que involucran el desarrollo de modelos de aprendizaje automático. Scrum es una metodología de trabajo ágil utilizado para gestionar proyectos de desarrollo de software. Se basa en la idea de dividir el trabajo en iteraciones cortas y enfocadas llamadas "sprints", donde se entregan incrementos de software funcionales al final de cada sprint. Scrum se centra en la flexibilidad, la colaboración y la capacidad de respuesta al cambio. Por otro lado, MLOps se centra en la gestión del ciclo de vida del modelo, desde el desarrollo hasta la implementación y la puesta en producción.

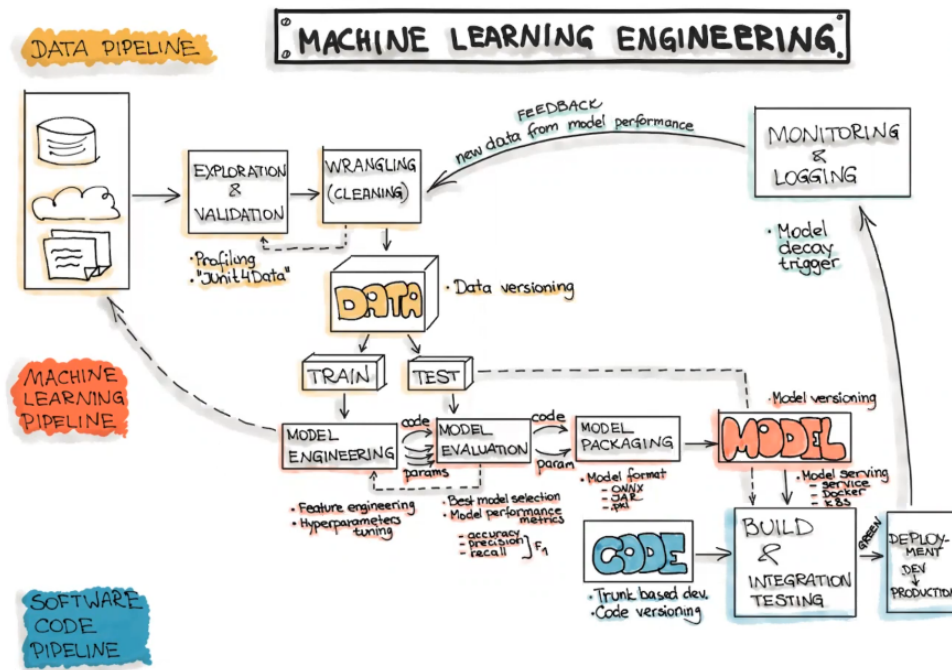


Figure 4.2: Arquitectura MLOps

En este proyecto, vamos a utilizar una combinación de ambas metodologías para gestionar el desarrollo del proyecto. Para ello vamos a recoger los principios fundamentales de Scrum y MLOps y los vamos a adaptar a nuestro proyecto.

4.2.2 Elementos de Scrum

A continuación, vamos a identificar aquellos elementos de Scrum esenciales para el desarrollo de nuestro proyecto.

- **Planificación del sprint:** Durante la planificación del sprint en Scrum, se incluyen las tareas relacionadas con la recopilación y preparación de datos, la selección y entrenamiento de modelos, la evaluación o implementación de la infraestructura necesaria para poner en producción los modelos.

- **Reuniones diarias:** Las reuniones diarias son una parte fundamental de Scrum. En estas reuniones se revisan las tareas realizadas, las tareas pendientes y los posibles impedimentos que puedan surgir.
- **Revisión del sprint:** Al final de cada sprint, se realiza una revisión del trabajo realizado durante el mismo.

Scrum también incluye una serie de roles:

- **Product Owner:** responsable de definir las funcionalidades del producto y priorizarlas.
- **Scrum Master:** responsable de asegurar que el equipo de desarrollo sigue la metodología
- **Equipo de desarrollo:** equipo encargado de desarrollar el producto.

En nuestro caso, el equipo de desarrollo estaría formado por el alumno, mientras que el tutor del proyecto haría las funciones de Product Owner y Scrum Master. Más adelante, en la sección de planificación, se detallará la lista de tareas a realizar durante el proyecto y los tiempos de desarrollo de cada una de ellas.

4.2.3 Principios de MLOps

A continuación, vamos a detallar los principios de MLOps que vamos a seguir en el desarrollo de nuestro proyecto. En esta sección se incluyen los puntos más importantes de la metodología, durante todo el desarrollo intentaremos cumplir con ellos en la medida de lo posible, ya que pueden darse situaciones en las que sea totalmente imposible satisfacerlos al completo.

- **Automatización:** la automatización es la clave para la eficiencia y el escalado. Aquí incluimos tareas como la generación de datos, el despliegue del modelo, la evaluación y la puesta en producción.
- **Testeo:** garantiza que tanto la funcionalidad como el desempeño del modelo están evaluados correctamente. Esencial en Machine Learning para identificar problemas y mejorar la confianza en los resultados.
- **Versionado:** es importante tener un control de versiones de los datos, el código y los modelos. Puede variar el método dependiendo de la herramienta que se utilice, pero existen estándares como Git para la gestión de código y GitHub/GitLab para el almacenamiento de los repositorios.
- **Reproducibilidad:** es necesario poder reproducir los resultados de forma consistente, puede ser complicado en Machine Learning debido a la naturaleza aleatoria de los algoritmos. Igualmente aquí se tratarán las herramientas y medidas necesarias para lograrlo.
- **Monitorización:** es importante tener un control de los modelos en proceso de entrenamiento o de producción. Por ello, recolectar estadísticas en tiempo de procesamiento nos ayudará a generar nuevas hipótesis para las próximas iteraciones.

4.2.4 Flujo de trabajo

Una de las características de los proyectos de Machine Learning es que son tecnologías en constante evolución, el desarrollo no es un proceso sencillo y requiere de un enfoque coordinado por parte de los diferentes equipos que lo conforman para asegurar su sostenibilidad a largo plazo.

Es fundamental poder adaptarse a los cambios de forma ágil, sin que esto suponga un deterioro del nivel de productividad o que ralentice el avance de nuevas funcionalidades.

Podemos identificar tres fases principales: diseño, desarrollo del modelo y operaciones. Cada fase tiene un enfoque específico y requiere una serie de tareas para garantizar el éxito del proyecto. La fase de diseño, se definen los objetivos y requisitos del proyecto, se identifican los datos necesarios para el entrenamiento y se establecen los criterios de éxito. La fase de desarrollo del modelo, se preparan los datos, se entrena el modelo y se evalúa su rendimiento. También se realiza la validación y se toman las decisiones sobre el modelo final a utilizar. Finalmente, en la fase de operaciones, se integra el modelo en la infraestructura existente, se monitoriza su rendimiento y se extraen conclusiones para futuras iteraciones. Al repartir las responsabilidades en diferentes fases, aislamos los errores junto con la carga de trabajo y podemos avanzar sin la preocupación de que un bug o un cambio requisitos afecte al resto del equipo.

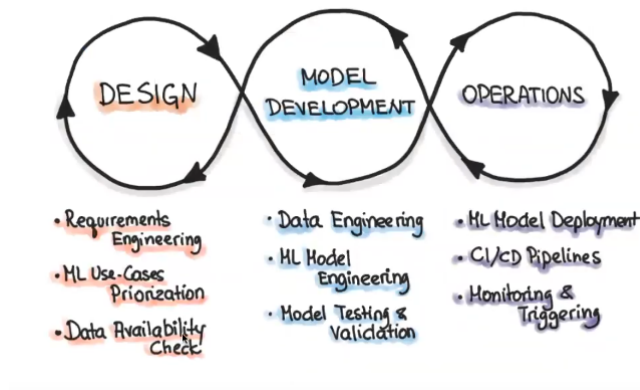


Figure 4.3: Metodología MLOps

4.3. METODOLOGÍA DE INVESTIGACIÓN

En esta sección, se describirá la metodología desarrollada para abordar problemas dentro del marco de trabajo de los problema NP-Hard. La metodología propuesta se basa en el uso de técnicas de IL para abordar problemas computacionalmente complejos, dando resultados cercanos al óptimo en un tiempo de ejecución muy reducido. Se explicará cómo esta metodología se ha diseñado y aplicado para resolver el caso concreto del FJSP, pero como se ha mencionado anteriormente, esta metodología se podría aplicar a otros problema de optimización combinatoria NP-Hard.

Una vez que sea descrita la metodología para abordar el problema utilizando técnicas de IL, se procederá con el caso específico de cómo esta metodología ha sido aplicada para diseñar un modelo del FJSP. Se explicará cómo se ha procedido en cada uno de los pasos propuestos a nivel de razonamiento y ejecución. A continuación se van a listar una serie de puntos que se deben seguir para poder aplicar la metodología propuesta:

1. **Modelado del problema como un environment de RL.** El problema se modeliza como un environment de RL, definiendo el espacio de estados y el espacio de acciones disponibles para el agente. A diferencia de un enfoque típico de RL, no es necesario definir una función

de recompensa ya que el modelo se entrenará mediante aprendizaje supervisado y no por refuerzo. Es importante destacar que el espacio de estados y el espacio de acciones estén enfocados en construir gradualmente la solución del problema y ser capaces de representar una solución válida.

2. **Generación de instancias aleatorias para el entrenamiento del modelo.** Se generan instancias aleatorias del problema con el fin de crear un conjunto de datos para el modelo de aprendizaje automático. Estas instancias aleatorias se utilizarán para entrenar el modelo de deep learning, que servirá para predecir soluciones en instancias del problema no vistas previamente. Una recomendación es que adicionalmente se generen diferentes grupos de instancias, con diferentes tamaños y características, para poder evaluar el rendimiento del modelo en diferentes escenarios.
3. **Utilización de métodos de programación matemática para obtener la solución óptima de las instancias.** Mediante métodos de programación matemática, se busca obtener las soluciones óptimas para las instancias generadas en el paso anterior. Estas soluciones óptimas se utilizarán como los datos del experto, lo que permitirá al modelo desarrollar una estrategia para resolver instancias no vistas.
4. **Entrenamiento del modelo supervisado de deep learning.** Una vez que se tienen las instancias generadas y las soluciones óptimas, se procesarán a través del environment de tal forma que para cada instancia se obtenga una secuencia de acciones que representan la solución óptima. Estas secuencias de acciones serán las que el modelo de deep learning tratará de predecir.
5. **Evaluación de resultados y benchmark.** Se utilizó el modelo entrenado en el paso anterior para predecir soluciones en instancias del problema que forman parte del benchmark o conjunto de datos de prueba. El objetivo es evaluar la capacidad del modelo para generalizar y proporcionar soluciones adecuadas en instancias no vistas previamente.

5. CONCLUSIONES Y TRABAJO A FUTURO

5.1. CONCLUSIONES

5.2. TRABAJO A FUTURO

BIBLIOGRAFÍA

- [1] “WHAT IS AGILE”. en. In: *The Power of the Agile Business Analyst, second edition*. IT Governance Publishing, 2018, pp. 22–25. ISBN: 9781849289955.
- [2] Atlassian. *What is Scrum?* 2018.
- [3] G. José. *Metodologías tradicionales vs metodologías ágiles*. en. 2011. URL: <https://es.linkedin.com/pulse/metodolog%C3%ADas-tradicionales-vs-%C3%A1giles-jos%C3%A9-gabald%C3%B3n>.
- [4] Redacción KeepCoding. “¿Qué es MLOps? [Guía completa]”. In: (2022).