

# Sistemas Inteligentes - Appunti

Francesco Lorenzoni

Febrero 2025



# Contents

<b>I</b>	<b>Introduction to SIN</b>	<b>5</b>
<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Informaciones generales . . . . .	9
1.2	Inteligencia Artificial . . . . .	10
1.2.1	Aplicaciones de la IA . . . . .	10
<b>2</b>	<b>Planificación Inteligente</b>	<b>11</b>
2.0.0.1	Modelo “planificación clásica” . . . . .	12
2.1	Planificación Jerarquica . . . . .	12
<b>3</b>	<b>Pyhop</b>	<b>13</b>
<b>4</b>	<b>Optimización Inteligente</b>	<b>15</b>
4.1	Scheduling . . . . .	15
4.1.1	JSP . . . . .	15
4.1.2	CSP - Constraint Satisfaction Problem . . . . .	15



## Part I

# Introduction to SIN



---

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Informaciones generales . . . . .	9
1.2	Inteligencia Artificial . . . . .	10
1.2.1	Aplicaciones de la IA . . . . .	10
<b>2</b>	<b>Planificación Inteligente</b>	<b>11</b>
2.1	Planificación Jerarquica . . . . .	12
<b>3</b>	<b>Pyhop</b>	<b>13</b>
<b>4</b>	<b>Optimización Inteligente</b>	<b>15</b>
4.1	Scheduling . . . . .	15
4.1.1	JSP . . . . .	15
4.1.2	CSP - Constraint Satisfaction Problem . . . . .	15

---





# Chapter 1

## Introduction

### 1.1 Informaciones generales

En los Viernes las clases son online y sincronas.

La asignatura se divide en does partes:

1. Parte 1

T1 Introducción

T2 Planificación inteligente

Algunos ejemplos son la planificación de procesos/planes de producción, o la planificación de rutas de reparto y logística; en general, aplicaciones incluyen sistemas de ayuda a la toma de decisiones y recomendación.

T3 Optimización inteligente Optimización de la logistica empresarial (horarios, recursos, personal, turnos de trabajo,...)

T4 Agentes inteligentes Sistemas multiagentes que tienen que negociar, coordinarse y cooperar para un objetivo común. Pero algo videojuegos donde existe una competición o cooperación entre agentes y se requiere una búsqueda de alternativas.

2. Parte 2

T5 Modelado de lenguaje y autómatas finitos

T6 Aprendizaje profundo

T7 Aprendizaje profundo para clasificación de texto

T8 Aprendizaje profundo para clasificación de imágenes

Profesor	T1	T2	T3	T4	T5	T6	T7	T8
Jaume Jordán								
Carlos Carrascosa								
Carlos Martínez (resp.)								

Figure 1.1: Profesores

La evaluación se efectuará mediante:

- ◇ Dos tests (**20 de junio a las 15:30**) sobre contenidos teóricos y prácticos de la parte 1 (15%) y parte 2 (15%), totalizando 30%.
- ◇ Trabajos académicos (70%):
  - Parte 1 (35%): tema 2 (15%) y tema 4 (20%).
  - Parte 2 (35%): tema 5 (10%), tema 7 (10%) y tema 8 (15%).

## 1.2 Inteligencia Artificial

IA no es solo IA generativa. IA es un campo muy amplio que incluye otras cosas. Inicialmente se asociaba con el Machine Learning, pero ahora se ha ampliado a otras cosas.

La Inteligencia Artificial es un campo interdisciplinario que implica máquinas capaces de imitar determinadas funcionalidades de la inteligencia humana, incluidas características como la percepción, el aprendizaje, el razonamiento, la resolución de problemas, la interacción lingüística e incluso la producción de trabajos creativos.

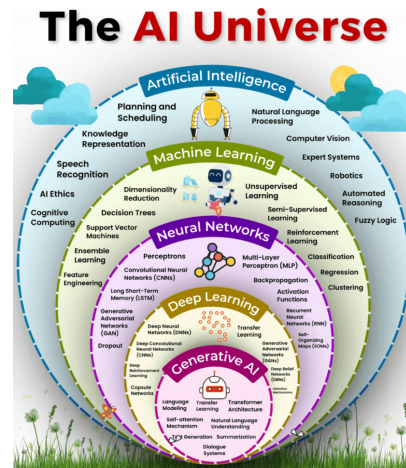


Figure 1.2: AI esquema. Según el profesor esto esquema no es completo, pero es cilustrativo.

## Modelos de IA:

- ◇ Aprendizaje supervisado
- ◇ Aprendizaje no supervisado
- ◇ Aprendizaje por refuerzo

### 1.2.1 Aplicaciones de la IA

La Inteligencia artificial puede ser util para estas cosas: Áreas y aplicaciones

- ◇ Recuperación inteligente de la información
  - Recuperación de información no explícitamente representada
  - Bases de datos deductivas. Procesos inferenciales. Sentido común. Interfaz natural
- ◇ Ingeniería del conocimiento (SBC y Sistemas Expertos)
  - Problemas NP: restricción de dominios, heurísticas y meta-heurísticas
  - Planificación. Scheduling. Optimización. Sistemas de Ayuda a la Toma de Decisiones
- ◇ Problemas (de búsqueda) combinatorios y de planificación
  - Este es el problema más importante de la IA.
  - Problemas NP: restricción de dominios, heurísticas y meta-heurísticas
  - Planificación. Scheduling. Optimización. Sistemas de Ayuda a la Toma de Decisiones

Hay también otras aplicaciones

- ◇ Sistemas Multiagente y distribuidos que interactúan (cooperación)
- ◇ Robótica
- ◇ Percepción
- ◇ Procesamiento de lenguaje natural

## Estado inicial

Información estática	Información dinámica
distance[home][airport_city1]=15 distance[home][center_city1]=5 distance[home][connection_c1_c2]=2 ..... vehicles(taxi)={taxi1} vehicles(plane)={plane1} ... location(airport)={airport_city1, airport_city2} location(to_stay)={home, hotel_city2, ...} .... city(home)=city1 city(airport_city1)=city1 city(airport_city2)=city2 .....	at(me)=home at(taxi1)=center_city1 at(plane1)=airport_city1 at(car1)=home cash(me)=20 ;; dinero que tengo owe(me)=0 ;; dinero que debo  <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <b>Objetivos</b>  at(me)=hotel_city2 </div>

Figure 2.2: Suponen que tengo que viajar desde una ciudad a otra. Esta puede ser una formalización de la planificación.

## Chapter 2

# Planificación Inteligente

Podemos ver la Planificación Inteligente desde varios puntos de vista:

- ◊ Deducción  
A partir de algunos hechos ciertos, conocer qué otros hechos también son ciertos (propagación)
- ◊ Aprendizaje  
“Mejorar la conudeta a partir de la experiencia”
- ◊ Razonamiento sobre acciones A partir de lagunos hechos ciertos (estado inicial), decidir qué acciones se deben ejecutar para alcanzar ciertos objetivos (es decir, que otros hechos sean también ciertos) en base a relaciones de causa-efecto.

El ser humano es inteligente porque es capaz de razonar acerca de las consecuencias de sus actos. Al fin y al cabo estamos planificando constantemente aunque no nos demos cuenta.

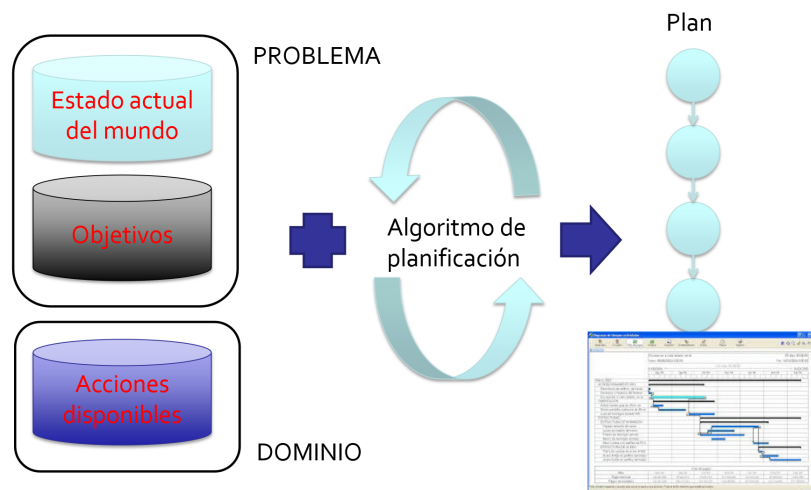


Figure 2.1: Planificación esquematizada

En Fig. 2.1 podemos ver un esquema de planificación. En el estado actual del mundo corresponde al “*dominio*”, y el objetivo es llegar a un estado deseado nel mundo.

**Definition 2.1 (Plan)** Un plan  $P$  es una secuencia de acciones que transforma el estado inicial ( $I$ ) en un estado que satisface los objetivos ( $G$ )

Un plan puede estar parcial o totalmente ordenado. Un enlace causal  $(a_i, p, a_j)$  determina que la acción  $a_i$  resuelve la condición  $p$  de la acción  $a_j$ .

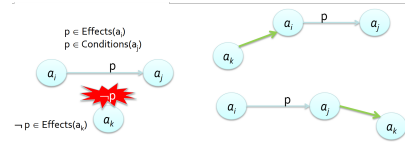


Figure 2.4: Plan

### 2.0.0.1 Modelo “planificación clásica”

- ◊ Determinista
- ◊ Estático (solo cambia por las acciones)
- ◊ Completamente observable
- ◊ Razonamiento temporal y numérico no contemplado

Es decir, el resultado de una acción aplicada a un estado completamente conocido se puede predecir correctamente, ya que no hay influencias externas que afecten al entorno. Para resolver este problema se utilizan algoritmos de búsqueda.

Este modelo pero es muy limitado, y sus asunciones necesitan ser relajadas para que sea más realista y aplicable a problemas reales.

En primer lugar, las acciones tienen distintas **duraciones**, usan recursos y tienen **coste**; además, El mundo *no* es totalmente conocido (existe incertidumbre) y es **dinámico** (cambia debido a eventos exógenos).

## 2.1 Planificación Jerárquica

La idea principal de **HTN** Hierarchical Task Network planning es proceder de una forma más parecida a como los humanos resolvemos los problemas complejos: Primero los aspectos generales (tareas) y después los detalles.

Se definen métodos abstractos que se deben descomponer para resolver el problema. Métodos son los triángulos invertidos en la figura.

Cada método incluye estos componentes:

- ◊ Nombre
- ◊ Parámetros
- ◊ Precondiciones
- ◊ Red de tareas

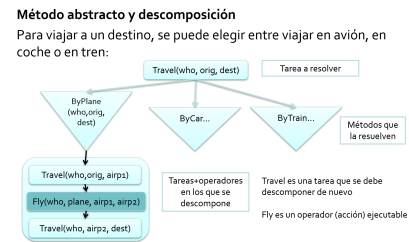


Figure 2.5: Método abstracto y descomposición

Métodos son tareas a descomponer, y los operadores (como **Fly**) representan las acciones que se pueden llevar a cabo dentro del mundo para poder alcanzar los distintos objetivos (acciones ejecutables que producen cambios en el estado del mundo).

- ◊ Nombre
- ◊ Parámetros
- ◊ Condiciones
- ◊ Efectos

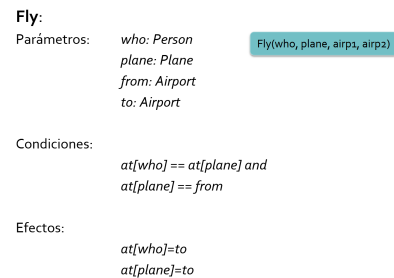


Figure 2.6: Acción Fly

## Chapter 3

# Pyhop

Para resolver un problema tenemos tres fases

1. Modelado
2. Planificación
3. Ejecución

El problema se define como una declaración de un nuevo estado del mundo (inicial), y una declaración de un objetivo. El **dominio**, son las acciones disponibles, que se estructuran en dos niveles: tareas/operadores y métodos.

- ◇ Información estática: no cambia durante la ejecución del problema y que sirve de soporte. Por ejemplo: distancias entre ciudades.
- ◇ Información dinámica: representa el estado actual del mundo. Por ejemplo: cantidad de dinero disponible por el agente en un momento determinado

Representan las **acciones ejecutables** en el dominio

```
def nombre_operador(state, otros_parametros):  
    <extracción valores estado>  
    if condicion1 and condicion2 and ... :  
        state.variable = nuevo_valor  
        ...  
        state.variable = nuevo_valor  
        return state  
    else: return False
```

Condiciones

Efectos

**Se devuelve el nuevo estado tras aplicarse el operador  
o False si el operador no es aplicable**

```
pyhop.declare_operators(nombre_todos_operadores)
```

Figure 3.1: Como definir un operador en Pyhop



# Chapter 4

# Optimización Inteligente

Planificación y Scheduling representa un área de gran relevancia en Inteligencia Artificial. Muchos problemas reales se modelan como problemas de P&S.

La planificación es un proceso de deliberación que escoge y organiza acciones anticipando sus resultados o consecuencias.

Lo scheduling es un proceso de asignación de recursos a tareas sobre el tiempo para optimizar uno o más objetivos.

PLANNING	SCHEDULING
La tarea de planificación determina <u>QUÉ</u> acciones son necesarias llevar a cabo para alcanzar un estado objetivo	La tarea de scheduling se centra en <u>CUÁNDO/CÓMO</u> llevar a cabo las acciones para optimizar el problema

Figure 4.1: P&S

## 4.1 Scheduling

Un problema de scheduling consiste en organizar en el tiempo un conjunto de actividades que compiten por el uso limitado de recursos

### 4.1.1 JSP

JSP stands for Job-Shop Scheduling, y es un paradigma de los problemas de scheduling es muy simple de enunciar y muy difícil de resolver.

- ◊  $n$  trabajos cada uno compuesto por un conjunto ordenado de tareas
- ◊  $m$  maquinas donde se procesan las tareas
- ◊ Cada tarea debe ser procesada en una única máquina durante un tiempo determinado y en un orden pre-fijado;
- ◊ El objetivo es minimizar *makespan*: instante de finalización de la última tarea

- ◊ **Datos** -  $p_{ij}$  es el tiempo de proceso de la tarea  $i$  en el trabajo  $j$ ;
- ◊ **Variables** -  $st_{ij}$  es el tiempo de inicio de tarea  $i$  en el trabajo  $j$ ;
- ◊ **Restricciones** -
  - Secuencial -  $st_{ij} + p_{ij} \leq st_{i(j+1)}$
  - Capacidad -  $st_{ij} + p_{ij} \leq st_{kl} \vee st_{kl} + p_{kl} \leq st_{ij}$
  - Sin interrupción -  $C_{ij} = st_{ij} + p_{ij}$
- ◊ **Objetivo** - Construir una ordenación de las tareas en el tiempo de manera que se satisfagan todas las restricciones sobre cada máquina.
  - Minimize the makespan:  $C_{max} = \max C_{ij}$
  - Minimize total flow time:  $C_{sum} = \sum C_i$
  - Minimize makespan + energy

### 4.1.2 CSP - Constraint Satisfaction Problem

Muchos problemas pueden ser expresados mediante:

- ◊ Un conjunto de variables.

- ◊ Un dominio de interpretación (valores) para las variables.
- ◊ Un conjunto de restricciones entre las variables.
- ◊ La solución al problema es una asignación válida de valores a las variables.

Formalization omitted here

En general, los problemas de optimización son particularmente difíciles de resolver. En general, basta con una solución razonablemente buena (factible, optimizada, pero no la óptima) en un tiempo razonable, y si puede obtener con métodos aproximados (como alternativa a los métodos exactos), cómo la **Búsqueda Heurística** / Metaheurística: La solución es obtenida (generada / mejorada), mediante un proceso de búsqueda en un amplio espacio de estados/-soluciones.

- ◊ Búsqueda Local (Local Search) - Se parte de una solución inicial y se busca una solución mejor en el vecindario de la solución actual.
- ◊ Búsqueda Sistemática (Systematic Search) - Se parte de una solución inicial y se busca una solución mejor en todo el espacio de soluciones.
- ◊ Métodos Constructivos - Construyen una solución paso a paso, añadiendo o eliminando elementos de la solución.
- ◊ Métodos de Mejora - Parten de una solución inicial y la mejoran iterativamente mediante movimientos locales.
- ◊ Métodos Evolutivos - Combinan soluciones previas

Los **algoritmos genéticos** son un tipo de algoritmo evolutivo que se utiliza para encontrar soluciones aproximadas a problemas de optimización y búsqueda. Estos algoritmos reflejan el proceso de selección natural en la evolución de las especies, y se basan en la idea de que las soluciones a un problema pueden ser representadas como individuos en una población, y que la evolución de la población puede conducir a la generación de soluciones cada vez mejores.

La operación fundamental es la **cruce**, que toma  $n$  padres (típicamente 2) y genera  $m$  hijos (típicamente 2); consiste en heredar material genético de los padres. La asunción es que si los padres tienen buen material genético, la mezcla del material genético de ambos también será buena.

Entonces, el efecto de la cruce es *explorar* el espacio de búsqueda con **material prometedor**.

La operación de **mutación** es una operación que se aplica a un individuo y cambia aleatoriamente uno o más genes de ese individuo. Corisponde a una búsqueda local, porque sólo se cambia un gen a la vez, y entonces el hijo será cerca de su padre.

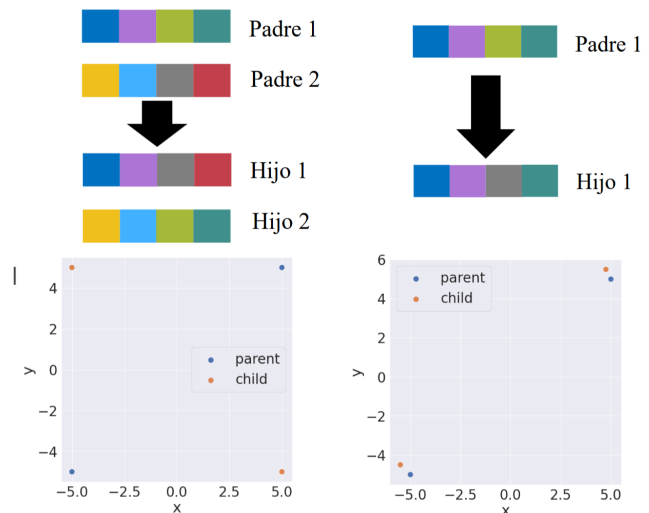


Figure 4.2: Cruce y mutación

TODO selección des padres TODO criterio de parada