

# Agent Development – Class Scheduling (P01)

Artificial Intelligence, Professor Joaquim Silva, 2022-2023

Nuno Veloso (10411), Augusto Pereira (21136), Duarte Melo (21149) – Grupo 04

# Introdução

Serve o presente documento para explicar e documentar, de uma forma sucinta e superficial, o desenvolvimento de um agente de inteligência artificial que visa encontrar a melhor solução para um problema de *class scheduling* – agendamento de aulas/criação de um horário, com base em determinados *constraints* ou restrições que serão tomadas em conta para a realização do mesmo.

Neste documento serão definidos o objetivo do agente, a explicação do desenvolvimento do mesmo e a estrutura do mesmo (PEAS e *task environment*).

As variáveis, domínio e restrições do agente, assim como o funcionamento e implementação do mesmo, estão explicados no *notebook* anexo a este relatório.

# Agente

## Objetivo

O objetivo deste agente é, após receber como *input* um conjunto de dados relativos às aulas, turmas e disciplinas, deSvolver o melhor horário possível, que vá de acordo com as restrições (*constraints*) do problema.

## Desenvolvimento

Inicialmente, o grupo realizou um pequeno *brainstorm* sobre as restrições que poderia acrescentar ao problema de forma a torná-lo mais realista – por exemplo, atualmente um tema bastante pertinente é o aumento do custo de vida: sendo assim, o grupo criou algumas restrições que visam diminuir, no resultado final, o número de dias por semana que as turmas têm aulas presenciais, reduzindo o número de deslocações à universidade.

Posteriormente, iniciou-se o desenvolvimento do agente começando pelas *constraints* mais básicas, começando a obter alguns resultados satisfatórios, até que se chegou às restrições mais avançadas, obtendo resultados bastante satisfatórios e realistas, que iam de acordo com todas as restrições colocadas ao problema.

Para o desenvolvimento do agente, o grupo utilizou, como auxílio, algumas funções e classes já implementadas no repositório do AIMA: <https://github.com/aimacode/aima-python>

Este repositório tem já implementados alguns algoritmos de definição e resolução de CSPs, como o que foi pelo grupo utilizado (*arc-consistency*).

É possível visualizar toda a evolução da implementação do agente no repositório GitHub do mesmo: <https://github.com/duartemelo/IA_Class_Scheduling>

## Estrutura do agente

### PEAS

O PEAS é um sistema de representação que visa medir a **P**erformance do agente no que diz respeito ao seu ***E****nvironment,* aos **A**tuadores e **S**ensores.

No caso deste agente, o PEAS caracteriza-se da seguinte forma:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo de agente | Performance Measure | Environment | Atuadores | Sensores |
| **Class Scheduling Agent** | Correta atribuição de salas e disciplinas, minimiza o número de deslocações, evita furos, etc. | Conjunto de disciplinas, salas, turmas e “vagas” horárias | Python via script desenvolvido com *display* do horário final consoante as restrições | *Input* recebido via dicionários com dados respetivos ao *environment* |

### *Task Environment*

O *environment*, ambiente em que o agente desenvolvido trabalha, pode ser descrito da seguinte forma:

* *Fully-observable*, completamente observável
* *Single-agent*, um só agente
* *Deterministic*, determinista – o próximo estado é determinável pela ação executada pelo agente
* *Sequential,* sequencial – a decisão atual pode afetar as futuras (atribuir uma sala e uma vaga a uma turma pode afetar na atribuição de uma sala a outra turma)
* *Static,* estático – o ambiente não muda enquanto o agente toma uma decisão
* *Discrete,* discreto – há um número finito de estados, um número discreto de aulas, turmas, disciplinas
* *Known,* conhecido – os *outcomes* das ações são conhecidos

# Conclusão

# Repositório GitHub

<https://github.com/duartemelo/IA_Class_Scheduling>