Departamento de Engenharia de Eletrónica e Telecomunicações e de Computadores
Licenciatura em Engenharia Informática e de Computadores

Second Practical Project

Artificial Intelligence course

2021/2022 Summer Semester

Version 1.00

Teacher: Nuno Leite

ISEL, 22 April 2022

ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Artificial Intelligence course

Second Practical Project

2021/2022 Summer Semester

Learning objectives

At the end of the **Second practical project**, students should be able to:

- ☐ Approach Problem-Solving as Search, using basic (non-informed) and more advanced algorithms (informed);
- ☐ Understand the concept of *state space*;
- ☐ Apply non-informed and informed search algorithms to solve the Sokoban game;
- ☐ Approach Problem-Solving as search in a solution space;
- ☐ Understand optimisation algorithms that explore a space of solutions, such as *simulated annealing* and *genetic algorithms*;
- ☐ Apply *simulated annealing* and *genetic algorithms* to solve the Sokoban game.

Sokoban game

In this project, your group will program an *automated solver* for the famous Sokoban game. The word Sokoban literally means "depositor warehouse". Figure 1 illustrates three different phases of a given game level.

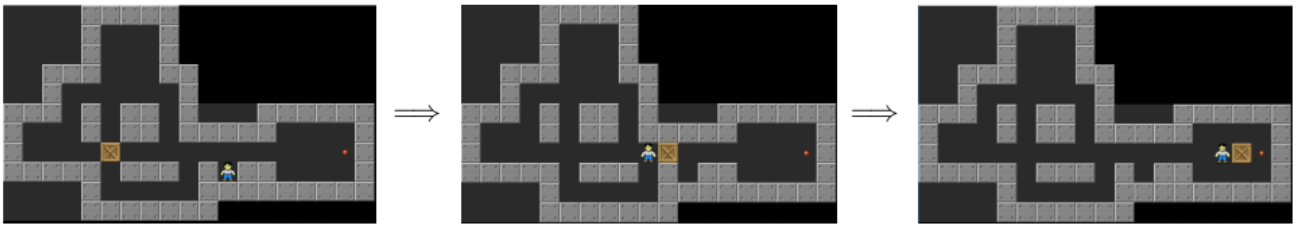

Figure 1: Three phases of the Sokoban Game: (a) Initial state, (b) and (c) Intermediate states.

The objective of the game is to move the warehouse worker (according to the directions North, South, East and West) in order to push the wooden boxes to the final positions (marked with a red dot in the figure). The employee can only push one box (not being possible to pull it). The warehouse is bordered by walls and corridors, represented by gray blocks in Figure 1. More information about the game can be consulted in: `https://en.wikipedia.org/wiki/Sokoban`.

Your assignment is to develop programs (in Prolog and in other language of your choice) that demonstrate the behaviour of different solvers, namely solvers that employ the following algorithms:

- iterative-deepening (non-informed), programmed in Prolog;
- best-first search or A* (informed), programmed in Prolog. You have to devise a proper *admissible heuristic*;
- simulated annealing (optimisation algorithm), programmed in a language of your choice;
- genetic algorithms (optimisation algorithm), programmed in a language of your choice.

These algorithms will be available in the GitHub's Artificial Intelligence course site, and are written in Prolog and Matlab languages.

The input/output is text-based.

You should demonstrate the behaviour of the automated solvers for some example levels, starting with a very small level with just one box. You don't need to implement level loaders (the levels could be hard-coded for simplicity), but if you want to do so, you could use the level format published in: `http://www.sokobano.de/wiki/index.php?title=Level_format`.

Due date: 23 May 2022 until 23:59.

The delivery of the work must present the report, Prolog code (solver using search algorithms), and other developed code (solver using optimization algorithms, *simulated annealing* and *genetic algorithms*), delivered in the Moodle system. The report must be concise and justify all decisions taken. It must indicate the student group composition and the curricular unit info.